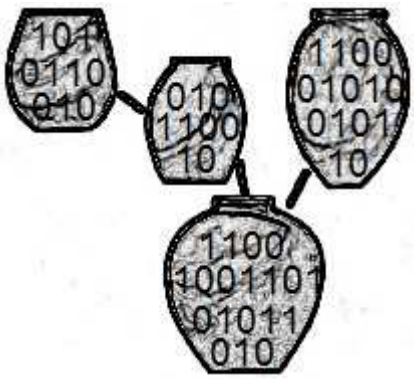
Classification of pottery assemblages in archaeology: a machine learning approach

D'Andrea Curra, Guilherme

Universiteit Leiden
The Netherlands

Classification of pottery assemblages in archaeology: a machine learning approach



Guilherme D'Andrea Curra

Front page figure: design by author after Petrie (1899, Fig. 3).

Classification of pottery assemblages in archaeology: a machine learning approach

Guilherme D'Andrea Curra | s2732467

Master Thesis Archaeological Science | 1084VTSY

Supervisor: Dr. K. Lambers

Leiden University, Faculty of Archaeology

18 March 2022. Final version

ACKNOWLEDGEMENTS

Many people inside and outside Leiden University contributed in one way or another to the development of this thesis and it is not possible to name them all. The first person to be acknowledged, and the one who most influenced this research, is my supervisor Dr. Karsten Lambers. He suggested using machine learning as an approach to my interests in artefact classification and typology, and his comments and support through this long process of thesis writing were invaluable to me.

Before being admitted to the Master's programme I had the opportunity to take the SPOC Academic Skills for Archaeologists course, taught by Dr. Maaike de Waal, which was of great value in preparing for the Master's course. Prof. Richard Thomas from University of Leicester and Dr. Marcos Casa from University of Caxias do Sul provided me the references for the admission process to the MSc Archaeological Science programme. The lessons from Dr. Tuna Kalayici on quantitative methods provided insights and ideas on some of the thesis subjects. Dr. Alex Brandsen recommended the toolkit used to create the ML model and provided me the initial template with classification algorithms. The team responsible for the ARCANE Project created the database of invaluable pottery information that was used in this research, and Dr. Diederik Meijer provided me with additional information about shape classes not available on the project website.

I would especially like to thank my daughter Isabela for her patience and encouragement as I devoted much of my time over the past year to writing this thesis.

For any readers, I hope you enjoy reading this work as much as I enjoyed writing it.

CONTENTS

LIST OF ILLUSTRATIONS

Figures

Tables

Appendices

1 INTRODUCTION

1.1 Overview

Artefact classification has been an important practice since the beginnings of archaeology, with the definition of the Three-age system by Thomsen in the early 19th century (Gräslund, 2009, p. 17-30) and many other achievements such as the application of typology for the definition of prehistoric chronologies in Egypt (Figure 1.1) and the American Southwest in the early 20th century (O'Brien & Lyman, 1999, p. 32-56, p. 84-137). Classification is a constant in archaeology, from a simple separation of artefacts in an excavation according to the raw material to the basis for formulating complex research questions involving human social and cultural systems (Read, 2007, p. 19-20).

Since the introduction of statistical analysis and computers in archaeology in the 1950s, the study of artefact classification evolved considerably (Wilcock, 1999), culminating with recent applications of machine learning concepts and methods.

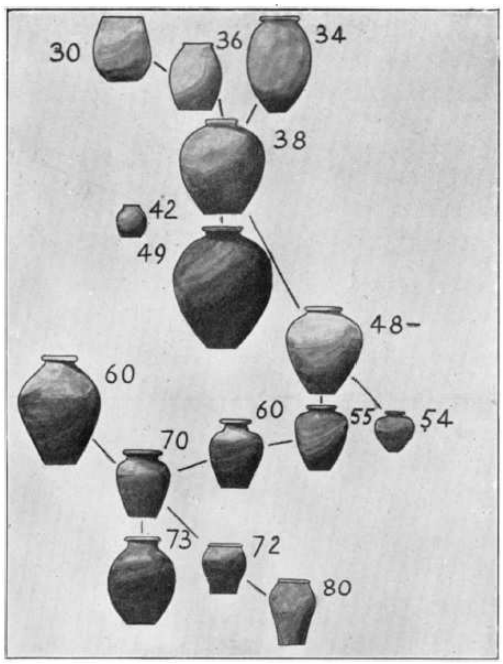


Figure 1.1 – Genealogies of some forms of pottery from Predynastic Egypt (Petrie, 1899, Fig. 3).

Machine Learning (henceforth: ML) uses specific methods to create models that can identify potential patterns in data, initially fitting the model to the observed data and then using the built model to predict values from new sets of data (VanderPlas, 2017). ML concepts were introduced in the late 1950s and early 1960s in areas such as neural network modelling and game programming, since then the discipline of machine learning has been associated with different paradigms and concepts along its history, among the most relevant is pattern recognition (Carbonell et al., 1983, p. 14-6), which is one of the bases of classification. ML uses data and standard algorithms as its basic mechanism in contrast to earlier computational approaches to reproduce human knowledge, such as the extensively programmed expert systems from the 1980s (Alpaydin, 2016, p. 50-2). Figure 1.2 shows a basic comparison between the traditional system development and the ML approaches. A ML model has adjustable parameters that receive different values (data), and makes use of algorithms that can optimise a performance criterion defined for the data through a repetitive and incremental process (Alpaydin, 2016, p. 24-5). ML is now present in a number of applications such as spam filters, detection of diseases based on image analysis, and prediction of customer behaviour (Alpaydin, 2016, p. 16-7, p. 23-4; Géron, 2019, p. 301-40).

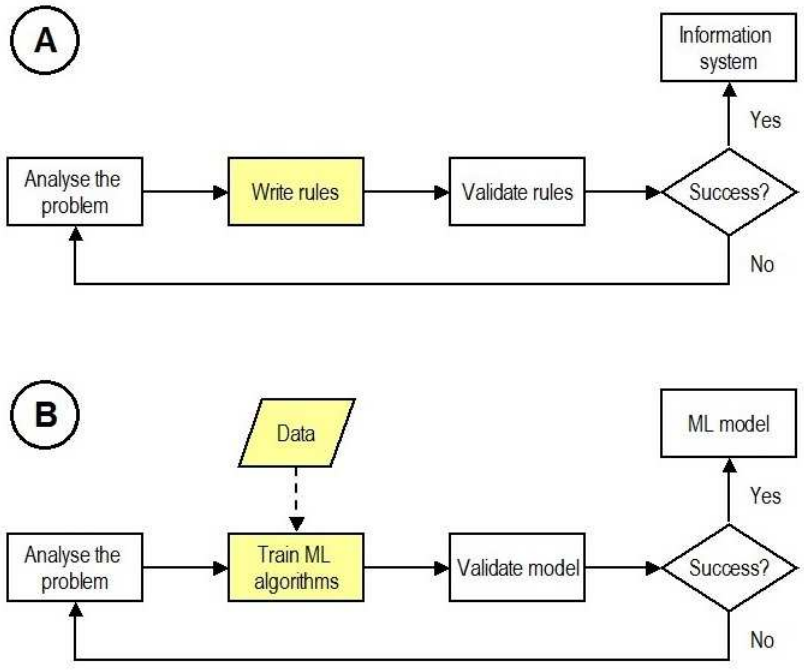


Figure 1.2 – Comparison of simplified diagrams for a traditional systems development approach (A) and a Machine Learning approach (B). After Géron (2019, Figure 1-1, Figure 1-2).

Applications of ML in archaeology started to be published in the 1990s (Barceló, 1995), consolidated in the 2000s (Barceló et al., 2000; van der Maaten et al., 2007), and have increased significantly in the past decade (Davis, 2020). Publications now include a variety of themes such as detection of archaeological features in the landscape (Lambers et al., 2019; Orengo et al., 2020), information retrieval from archaeological reports (Brandsen et al., 2019) and identification of ceramics through photos of shards (Anichini et al., 2021). In the specific case of artefact classification, works such as Hörr et al. (2014) and MacLeod (2018) demonstrate the potential and viability of ML methods.

The remaining of this introduction chapter presents some fundamental concepts for the entire thesis such as the research aim, motives and research questions and briefly presents topics that will be further developed in subsequent chapters: the dataset and methods used in the ML classification model.

1.2 Research aim and questions

The aim of this research is to develop a ML model to classify archaeological pottery assemblages. There are three main motives and objectives associated with this aim.

The **first motive** is to provide a tool to assist pottery experts in their decisions while performing the task of vessel classification. The efficacy of a ML classification model depends on many factors such as the data quality and quantity, and the adaptation of the model to certain characteristics of the input data (Géron, 2019, p. 607-732). An automated classification model will not replace an expert but can give suggestions on how to proceed, and help to identify some mistakes that may occur during data input such as digitising errors. For instance, if a vessel of large dimensions and high volumetric capacity is classified as a beaker due to some code input error, the classification suggested by the model might alert the expert of such occurrence.

The human brain is well equipped for pattern recognition (Alpaydin, 2016, p. 20-4) and can perform some tasks and identify details that cannot be easily matched by computer systems, even more so when the accumulated experience of experts in classification of pottery or other types of artefacts is added. On the other hand,

a ML model may identify patterns that would be left unnoticed by traditional analysis. The efforts provided by traditional analysis/classification and a ML proposed classification would be complementary and the sum of results an optimal alternative (Verschoof-van der Vaart & Lambers, 2021).

The **second motive** is the possibility to provide standard methods of classification, which have clearly defined criteria suggested by the classification model. If there was no concordance between two or more experts regarding a certain classification within an assemblage, the alternative suggested by the model might help to achieve a final decision. When mentioning standard methods it is important to make clear that there is no general standard that might be applied to all sorts of assemblages. We refer here to standard methods that are possible to be applied to assemblages that share common characteristics, such as specific archaeological sites or cultures. A model that was created for one specific assemblage would probably have to be adapted to properly work for a culturally distinct assemblage, depending on the artefact characteristics, and dataset attributes and organisation.

The **third motive** is that, beyond the analysis and classification of new artefact assemblages, it is also possible to perform a new analysis on previously classified collections in accordance with new perspectives and approaches, to investigate new research questions or after new data becomes available on the collection. The remaining parts of this section define the main research question and sub-questions, which specify the research aim and objectives.

Which are the benefits and limitations of a machine learning classification model for pottery assemblages?

To answer the main research question in a more structured way, a set of sub-questions was defined:

1) Which are the minimum features required to provide a basic classification, and which are additional features that could improve it?

2) To what extent can this model replicate classifications made by experts?

3) Which other kinds or levels of classification (e.g., subclasses, groups) that might be archaeologically relevant can the model suggest?

The fist two sub-questions will be addressed through supervised ML methods, the third sub-question will be addressed through unsupervised ML methods as briefly explained in the next section. The term *feature*, referred to in the first research sub-question, has different meanings in archaeology and ML. Section 2.1.2 presents these and other definitions of terms used in this research, for clarification purposes the meaning in the context of ML is the one used through this research.

A relevant note on the scope of this research: the proposed ML model will be prepared to process information on pottery vessels, not necessarily complete vessels (which may be uncommon in most archaeological assemblages), but vessels that have a minimum number of parts or measurements that are significant for shape identification. Associated with the first research sub-question (minimum required features for classification) is the issue of amount and quality of information available considering all features and values for one specific vessel, and how this can affect the model results. The model in this research therefore does not include the processing of smaller vessel parts such as pottery shards.

1.3 Methodology

This section provides a summary of the data and methods used to develop the ML classification model, which are described in more detail in Chapter 3.

1.3.1 Dataset

The source database for the dataset used in this research is the Project ARCANE - Associated Regional Chronologies for the Ancient Near East and the Eastern Mediterranean in the Third Millennium BC (Arcane, 2016), which records more than 8200 pottery objects from 168 archaeological sites.

Samples from four sites from the Arcane database will be used to train the ML model and to test/validate it. The research dataset is composed of 496 vessels from the Tell Brak, Tell Beydar, Tell Leilan and Tell Barri archaeological sites located in northeast Syria; more information on the sites is presented in Section 3.1.1. The pottery assemblage ranges from c. 3000 to 1950 BC, the Bronze Age period in the Near East. The assemblage is composed mostly from vessels types

associated with domestic and storage contexts such as jars, pots, bowls and beakers. A small proportion of vessels types is associated with funerary or ritual contexts.

Figure 1.3 – Selection of pottery vessels from Tell Beydar (JZ002) and Tell Barri (JZ007): (1) Bowl - JZ002_P092; (2) Shallow bowl - JZ002_P620; (3) Open pot - JZ002_P605; (4) Jar (wide neck) - JZ002_P081; (5) Closed pot (high) - JZ002_P805, Closed pot (squat) - JZ002_P806 and Cup/Beaker - JZ002_P807; (6) Juglet - JZ007_P028; (7) Jar (restricted neck) - JZ002_P084; (8) Flask - JZ002_P618. Images at different scales. After Arcane (2016).

Figure 1.3 shows a selection of some pottery from these sites that belong to the dataset used in this research, more information and illustrations of the vessel shapes are presented in Section 3.1.2.

The Arcane pottery database was selected because of data requirements and also because it is an important reference for the archaeology of the region, presenting a classification method that is shared by all sites and objects contained in the database. ML models need both quantity and quality of data (representative

samples, relevant features) in order to identify possible patterns (Géron, 2019, p. 22-6), and for this specific research it was necessary to use a dataset with distinctive types of features (described in detail in Chapter 3.1), belonging from both the categorical (vessel qualitative characteristics) and the continuous (vessel measurements) types.

1.3.2 Machine learning methods and algorithms

One criterion to distinguish the ML methods is according to the amount and type of the model supervision during training (Géron, 2019, p. 340-456):

- Supervised: the training dataset used by the algorithms includes the expected solutions (or labels). In the thesis research objectives, this means to assign new vessels to pre-defined classes.
- Unsupervised: the training dataset does not include the expected results (unlabeled data). In the thesis research objectives, this means the model seeks to identify potential classes based on vessel features (automatic grouping of similar objects).

There are approaches that refer also to semi-supervised methods (Section 2.2.3). In addition to the basic methods of learning, it is necessary also to select methods for specific tasks such as splitting of the dataset, feature encoding and imputing of missing values, which are detailed in Chapter 3.3. Information about the software used in this research is provided in Chapter 3.2.

Algorithms from the scikit-learn library (Scikit, 2021a) were used to build the model. A brief explanation about each algorithm is provided in Chapters 3.3 and 3.4. The algorithms are divided in supervised and unsupervised learning. The algorithms belonging to the supervised learning group are divided according to their main function: classification and regression. In this research only classification algorithms are used. Six algorithms were selected for classification, and two algorithms for clustering, either to compare the results among them or to complement each other.

1.4 Thesis structure

Following this Introduction chapter, this section briefly describes the thesis structure through its chapters. Chapter 2 – 'Background and Context' presents relevant concepts, methods and previous works in the areas of artefact classification, including quantitative methods, and ML. Chapter 3 – 'Data and Methods' details the dataset that was briefly presented in this introduction, including information about vessel shapes and each feature used in the ML model, and details the methods and algorithms applied to the dataset. Chapter 4 – 'Results' presents the results of applying the methods and algorithms to the dataset, in both supervised and unsupervised learning approaches. The classification resulting from the ML model is compared to the classification made by the experts and the relevance of each vessel feature in the ML model is evaluated. Chapter 5 – 'Discussion' interprets the results obtained from the application of the ML model in the dataset and discusses the model benefits, issues and limitations. Chapter 6 – 'Conclusion' presents the answer to the research questions and possibilities of further research in the area of artefact classification through machine learning.

2 BACKGROUND AND CONTEXT

This chapter presents relevant concepts, methods and previous works in the areas of artefact classification and ML, which form the theoretical framework of the research.

2.1 Artefact classification

There are many approaches to archaeological artefacts classification (Dunnell, 1971; Orton et al., 1993, p. 152-65; Read, 2007; Rice, 1987, p. 274-88; Santacreu et al., 2017). The main focus of this chapter is on vessel shape and form since the research questions are related to this aspect of pottery vessels, nevertheless there are other relevant aspects and concepts that are related to vessel shape and classification that are presented. Some methods and techniques used for pottery classification can also be applied to other classes of artefacts that have a certain level of symmetry like some lithic tools (Read, 2007).

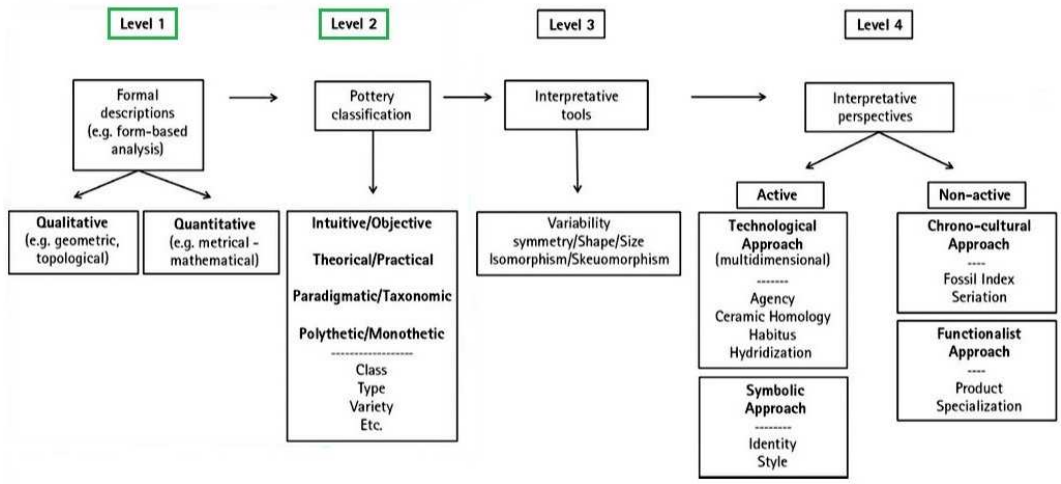


Figure 2.1 – Diagram of the main analytical levels to approach pottery form and classification according to Santacreu et al. (2017). Levels 1 and 2 are within the objectives of this research. The original diagram includes also Level 5, related to Fractal patterns and Homology (extrinsic analysis between technology and remaining social spheres). After Santacreu et al. (2017, Figure 12.1).

Figure 2.1, a diagram showing the main analytical levels to approach this subject, is a convenient summary and starting point for the remaining of the chapter. The most relevant levels for this research are Levels 1 and 2 (approached in Sections 2.1.1 to 2.1.4), however Levels 3 and 4 are equally relevant since they are related to interpretation and practical applications of classification, which is the final goal of this systematic method of arrangement (Dunnel, 1971, p. 43).

2.1.1 Concepts of vessel form, shape and function

According to Read (2007, p. 97-103) there are four conceptually independent operations or stages involved in the production of pottery objects, which include the definition of: 1) the material properties from the which the object is made, for instance type of clay, tempering material, firing techniques; 2) the object form and techniques of production such as coiling or wheel-thrown and inclusion of additional elements such as handles or spouts; 3) the surface treatment (e.g. smoothing, polishing); and 4) the decoration (e.g. incising, painting), if any. The second stage, related to the object form, is the main focus of this research and the basic criteria used for the pottery assemblages' classification. A similar approach is presented by Rouse (1960), where he includes the definition of potential types based on the artefacts resulting from the operations or stages of production (Figure 2.2).

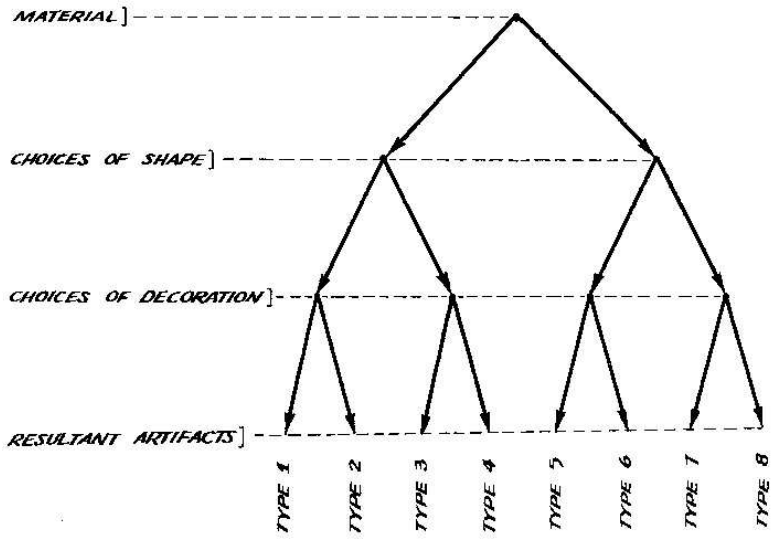


Figure 2.2 – Analytical procedure for making artefacts and definition of potential types (Rouse, 1960, Fig. 1).

Formally, the concepts of form and shape are distinct, however it seems the terms are used as synonyms across many publications in archaeology. According to the Getty Art & Architecture Thesaurus (Getty, 2004a; Getty, 2004b), shape is an attribute or component of form, more like an outline or contour and, with additional characteristics included, a shape can become a form. An example where the terms are aligned with these definitions is Nelson et al. (2017), who define 'form' as a combination of size and shape. For the purposes of this research there will be no formal differentiation between these terms, 'shape' will be used more frequently since it is the term used by the Arcane project to define the classes of vessels (the shape classes).

Rice (1987, p. 215-7) approaches the description and classification of vessel forms based on two basic systems, one is use-oriented and the other one is based on solid geometry. The use-oriented systems are based on inferred use of the vessel according to diverse criteria such as vessel size, ratios of measurements (e.g., height vs. diameter), presence of functional attachments (handles, spouts), complemented by information provided by ethnographic studies and historical documents (Rice, 1987, p. 215). The systems based on solid geometry use a combination of solid shapes, surfaces and sections of these shapes to describe the vessel, some use numerical codes to identify the shapes (Rice, 1987, p. 219-21).

Albeit rigorous and useful in some specific circumstances, these geometry-based systems are not practical enough to replace the more empirical use-oriented systems, even if these may present some inconsistencies and incompatibilities when used among different archaeological assemblages. The vessel shape terminology based on these systems is not standardised and the diversity of terms in different languages, and even within the same language (when different terms are used to define the same shape, or when one term is used to define different shapes), make the definitions of shape class imprecise (Rice, 1987, p. 215).

Use-oriented classifications in archaeology are frequently based on ratios of height to maximum diameter and kind or size of orifice (Figure 2.3), but may use also the presence of functional attachments like handles or spouts (Rice, 1987, p. 215-6). The Arcane project adopted a similar system to define the vessel shapes used in this research (Chapter 3.1).

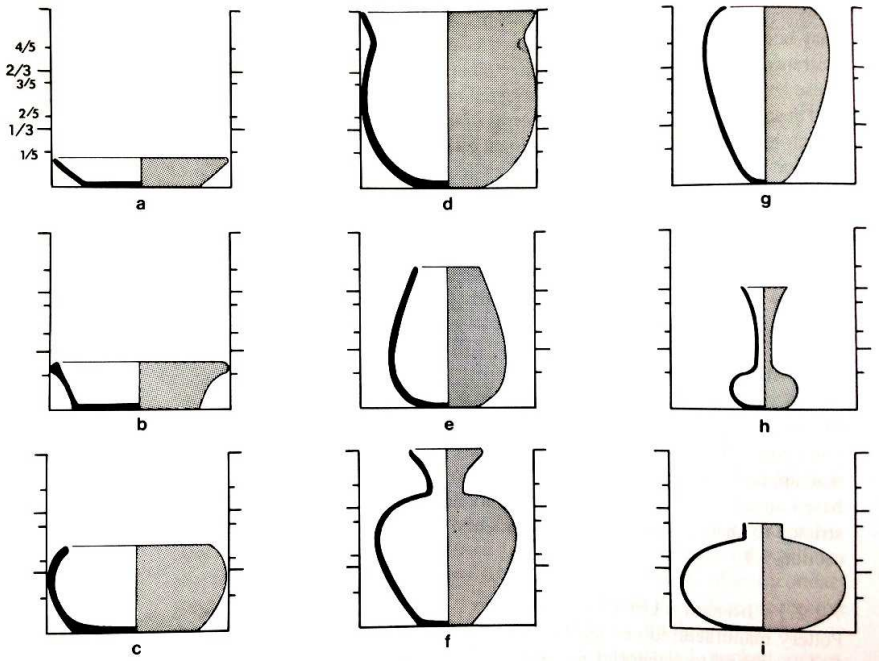


Figure 2.3 – Shape categories based on vessel proportions as a ratio of height to diameter and types of neck/orifice. (a) plate; (b) dish; (c) bowl; (d) bowl; (e) vase; (f) jar; (g) neckless jar or vase; (h) *florero* or jar; (i) jar. Rice (1987, Figure 7.4).

Sometimes the use of functional categories such as storage, cooking and serving are added to the shape class (e.g. storage jar, cooking pot) and albeit useful for expanding a classification system based only on shapes it can make assumptions on vessel use that are not always clear (Rice, 1987, p. 211-2). Other information like the context of finds, residue analysis and the mechanical and technical properties of the ceramic are important for the definition of vessel function (Hunt, 2017; Rice, 1987, p. 224-43).

Such diversity and sometimes inconsistency in vessel shape definitions in archaeology may cause some difficulties for ML methods and algorithms, the results must take this into account and this issue will be addressed in Chapter 5.

2.1.2 Attributes, variables and features

The concept of ‘characteristics or traits that can be observed in an object’ (Read, 2007, p. 110) is of fundamental importance to objects definition and the formation of classes of objects. This concept may have several terms associated to it,

sometimes used as synonyms (Bruce et al., 2020, p. 12), but there are some differences among them that will be briefly described.

Attribute and variable

According to Dunnell (1971, p. 49-50), attribute is the 'smallest qualitatively distinct unit involved in classification'. After defining the field of classification (e.g., archaeological artefacts) and the scale (e.g., assemblage of pottery vessels), the next step is the identification of the attributes that will become the potential criteria for classification (Dunnell, 1971, p. 49-50). There is a relation between the number of dimensions of attributes and the potential number of classes derived from them, leading to issues like lumping or splitting of attributes (Dunnell, 1971, p. 50). The color category 'brown', the temper category 'grit' and the height category '10.5' applied to a vessel sample are examples of attributes according to this definition (Dunnell, 1971, p. 51; Read, 2007, p. 110-13). In ML, an attribute is equivalent to a data type, (e.g., 'color'), but it can also be considered a synonym of feature (Géron, 2019, p. 8).

A variable is a category for attribute values, for instance color, temper and height are variables that may be associated with pottery vessels, each variable have its own set of possible attribute values. Variables may be qualitative (e.g., color) or quantitative (e.g., height), also referred to as categorical or numeric respectively (Bruce et al., 2020, p. 9-10; Read, 2007, p. 36-9, p. 110-3, p. 243-6).

Feature

This term is commonly used in both archaeology and ML, but with different meanings. In data science and ML, features are used to predict target values and for this reason a set of features is also called predictors (Bruce et al., 2020, p. 13; Géron, 2019, p. 8). A feature can also be considered either a synonym for attribute (e,g., 'color') or an attribute plus its value (e.g., 'color = brown') depending on the context (Géron, 2019, p. 8). In archaeology, feature can be defined as a 'separate archaeological unit that is not recorded as a structure, a layer, or an isolated artifact' (Kipfer, 2000, p. 186), such as walls, hearths and storage pits. The meaning of feature as predictors, the elements used to predict targets by ML algorithms, is the one used through this research, and also used to refer to the characteristics or traits that can be observed in an object (Chapters 3.1 and 3.3).

2.1.3 Approaches for artefact grouping

There are distinct approaches for artefact grouping (Bortolini, 2017, p. 658-60; Read, 2007, p. 27-8), one of these divides the alternatives in two methods (Figure 2.4): *top down*, associated with classification (and supervised learning in ML), and *bottom up*, associated with clustering (and unsupervised learning). According to Read (2007, p. 64, p. 135-8), groups must be 'internally coherent and externally isolated', meaning that members must be clearly identified as belonging to one specific group and not to others. Figure 2.4 is divided into two domains, ideational (without objective existence) and phenomenological, following concepts from Dunell (1971). One important detail is the question about a possible equivalence between two types of classes, explicit or implicit, involved in the grouping processes; this issue is addressed in Chapter 5.

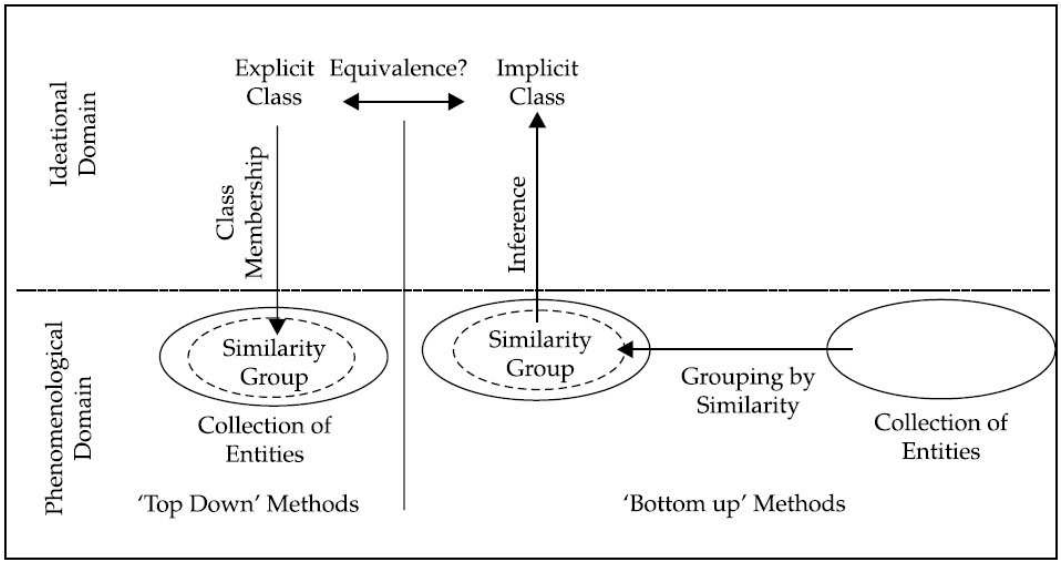


Figure 2.4 – Two approaches for artefact grouping: top down, associated with classification, and bottom up, associated with clustering (Read, 2007, Figure 1.1).

2.1.4 Classification strategies

There are several classification strategies, organised according to different approaches to the question. Level 2 in Figure 2.1 is the analytical level related to this subject (Santacreu et al., 2017, p. 183-5).

Intuitive vs. objective

Based on the applied level of formalism. The earlier approaches to classification are considered more intuitive since they were defined mainly on the analysts' perceptions of the differences and similarities in ceramic assemblages such as in Krieger (1944) and Rouse (1960), while the objective approaches are mainly defined on analytical and/or statistical methods. Some of the earlier examples are those from Kroeber (1940) and Spaulding (1953), the objective later became the dominant approach in archaeology (Read, 2007, p. 107-8; Santacreu et al., 2017, p. 183-4). Read (2007, p. 67-70, p. 107-8) draws attention to a reversion to the subjective approach in the work of Adams and Adams (1991), which emphasizes the human capacity of patterning identification and the importance of intuition on the identification of types.

Theoretical vs. practical/functional or emic vs. etic

Based on different ontologies. Theoretical or emic classifications are based on the empirical characteristics and conceptual systems of the object producers, either tangible or intangible. Attributes used in the classification have cultural saliency, they carry important cultural and historical meanings (Read, 2007, p. 39-42, p. 69-73; Santacreu et al., 2017, p. 184-5). Functional or etic classifications are based on the conceptual systems and technical criteria of the analyst who is attempting the classification. There is an understanding that the terminology and classificatory criteria used by object producers may be too complex to be perceived and replicated by foreigners (Read, 2007, p. 39-42, p. 69-73; Santacreu et al., 2017, p. 184-5).

Paradigmatic vs. taxonomic

Based on how the attributes are considered. In taxonomic classifications not all the attributes are considered to be of equal importance for all pottery being classified, and must be used in a sequential and hierarchical order according to different criteria to define the classes. Some classes may miss an entire attribute, for instance 'surface treatment'. Taxonomic classifications are usually represented in a hierarchical branching diagram as in Figure 2.8 (Dunnell, 1971, p. 70-6; Read, 2007, p. 81-3, p. 113-14, p. 241-2; Santacreu et al., 2017, p. 184). In paradigmatic classifications the classes are defined by each possible combination

of attributes, without hierarchy among the attributes. It is possible that some combination does not apply to some samples in the class (e.g. some do not have any surface treatment), but the attribute is still used to define the class (Dunnell, 1971, p. 76-84; Read, 2007, p. 81-3, p. 113-14, p. 241-2; Santacreu et al., 2017, p. 184).

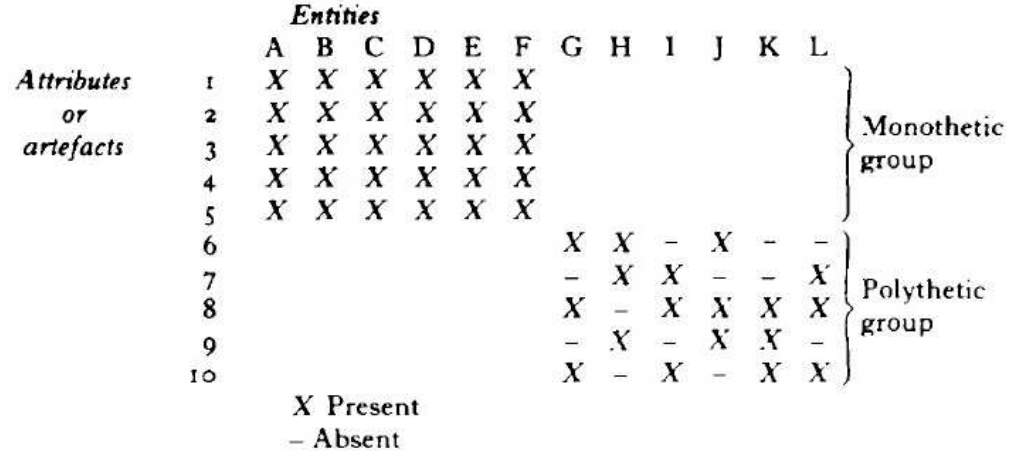


Figure 2.5 – Example of monothetic and polythetic groups of entities (classes) and attributes or artefacts. Clarke (1978, Fig. 3).

Concepts similar to paradigmatic and taxonomic are *polythetic* and *monothetic*, which are based on the degree of shared attributes between objects in a class (Figure 2.5). In monothetic classifications an object is a member of a class if it presents all attributes that compose the class, while in polythetic classifications an object is a member of a class if it presents a sufficient number of attributes from a set of possible attributes that compose the class (Bortolini, 2017, p. 658-60; Read, 2007, p. 134-5).

2.1.5 Applications of classification: typology and seriation

Classification of pottery vessels based on shape is fundamental for starting to answer a number of questions related to vessel function, place of origin or chronology, but in archaeology the more information available the better, therefore definition of shape classes must be whenever possible complemented by other specific techniques: residue analysis for determining contents and uses, ceramic petrography to identify vessel provenance, stratigraphy and absolute dating methods like TL for vessel chronology (Rice, 1987, p. 224-43, p. 435-46).

When such additional information is not available, shape can be of fundamental importance as in the case of the sequencing of Predynastic Egypt by Petrie (1899).

Petrie did not have stratigraphic information to help him to define a chronological sequence for the period before the firsts Egyptian dynasties (c. 3200 BC) since superposition of graves or burials was rarely found, then he used the grouping of similar vessel types according to their form and style/decoration (Figure 2.6) (Midant-Reynes, 2000).

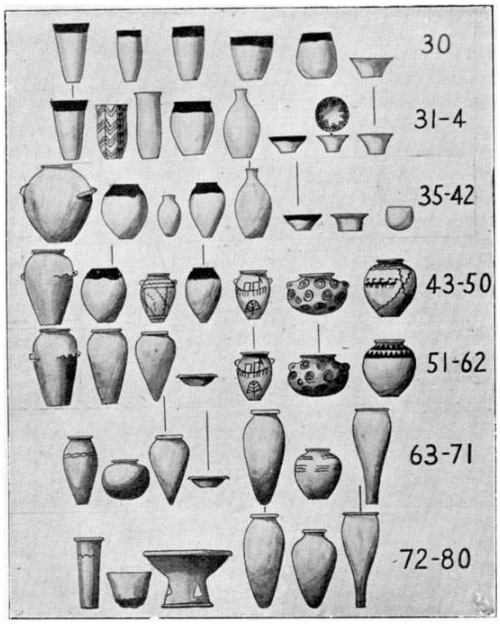


Figure 2.6 – Representative types of pottery of seven successive stages in Predynastic Egypt known as Naqada period. The numbers represent the sequence dates, and the vertical lines represent the vessel types that link one stage to the next one. Petrie (1899, Fig. 1).

The proportion of each group found in around 900 graves and the presence of specific types linked one stage to the next one, placing them in chronological order divided in seven stages and sub-stages (Midant-Reynes, 2000; Petrie, 1899).

One of the key elements identified by Petrie to define the sequence dating was the vessel handles, which seemed functional in earlier vessels and gradually became less functional and more decorative as it can be seen in some vessels in the left in Figure 2.6 (O'Brien & Lyman, 1999, p. 87). Additional elements such as vessel

handles and spouts may be relevant to identify vessel shape classes and are included in the dataset as categorical features as explained in Chapter 3.1.

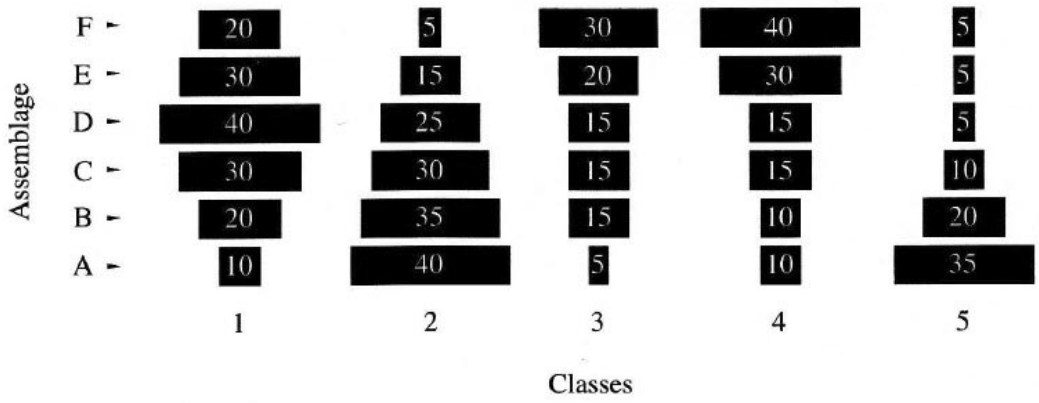


Figure 2.7 – Frequency seriation of six assemblages (A-F) using five artefact classes (1-5). After O'Brien and Lyman (2003, Figure 2).

The technique used by Petrie in Egypt is defined as phyletic or contextual seriation by O'Brien and Lyman (1999, p. 84-91, p. 111-14), in contrast with occurrence seriation and frequency seriation as created by Kroeber and further developed by Ford in the USA. The main difference is related to the use of quantitative information (relative abundance) associated with vessel features such as decoration and colors in different assemblages, and in the case of frequency seriation can be represented in graphs like the 'battleship' frequency curves (Figure 2.7), which are based on artefact classes and their frequency of occurrence on each assemblage (O'Brien & Lyman, 1999, p. 121-5).

A typology is associated with some question or aspect of interest about an artefact assemblage such as function, decoration, morphology, or chronology, or a combination of more than one aspect (O'Brien & Lyman, 2003, p. 23-4). A typology goes one step further in relation to classification; it is possible to create a classification of pottery vessels based on shape, but without defining a further goal, this can be defined later by a typology. This is also Santacreu's et al. (2017) approach, which is summarised in Figure 2.1; the first two analytical levels to study pottery form and classification are related to formal descriptions of form (qualitative or quantitative) and the classification itself, which can be of various aspects (e.g., intuitive or objective, paradigmatic or taxonomic). The next two levels are related to interpretation in various types of approaches (technological,

symbolic, chrono-cultural or functionalist), which are based on specific typologies (Santacreu et al., 2017).

To conclude this section, ceramic seriation remains an important source of information about chronology besides the increasing application of scientific dating techniques (Lipo et al., 2015; Peeples & Schachner, 2012; Porcic, 2013).

2.1.6 Quantitative classification

The method developed by Read (2007) for classification of archaeological assemblages is associated to the concepts of quantitative classification, recursive division and numerical taxonomy (Dunnell, 1971, p. 98-102; Read, 2007, p. 127-8, p. 199-240).

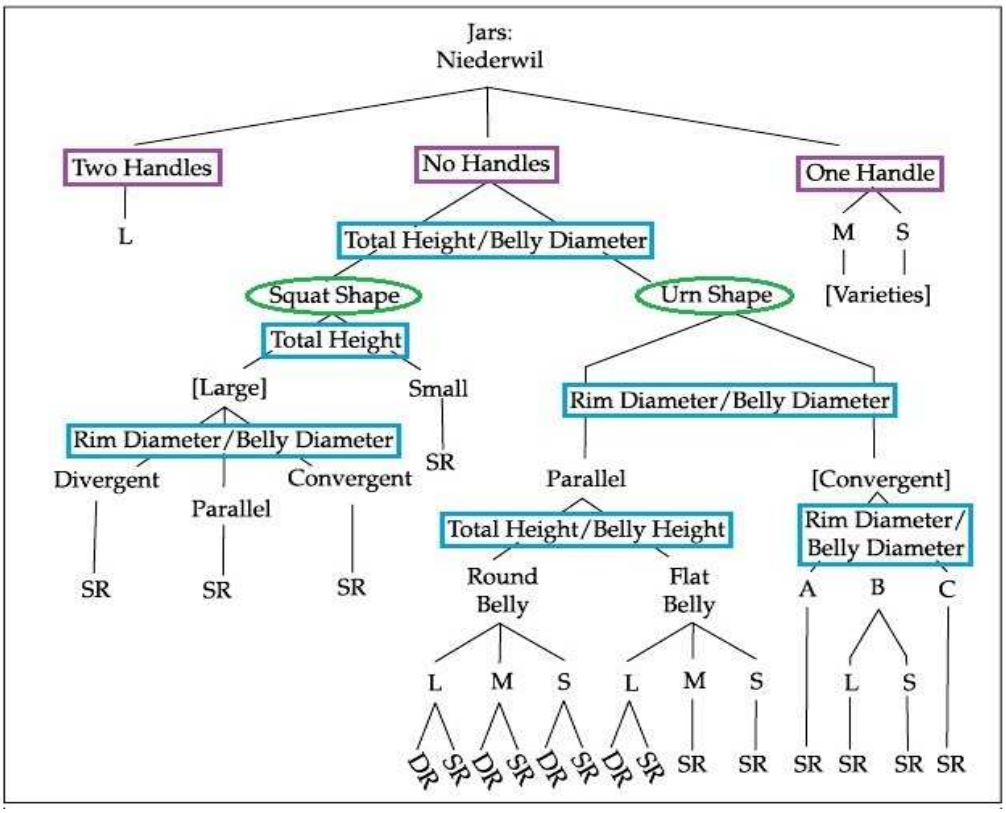


Figure 2.8 – Taxonomic structure of the method developed by Read (2007) for the classification of artefact assemblages, in this example pottery vessels from the late Neolithic site of Niederwil, Switzerland. Qualitative (categorical) features are highlighted in purple and quantitative features (basic and relative measurements) in blue; the two basic shapes are highlighted in green. A, B, C = Belly shapes (convergent); DR = Decorated rim; SR = Smooth rim; S = Small; M = Medium; L = Large. After Read (2007, Figure 8.23).

The method of recursive division starts with the identification of potential qualitative (categorical) features that have cultural salience in an assemblage and then use one of these features, or variables in Read (2007) terminology, to divide the vessels in groups. In the example shown in Figure 2.8, this feature is the vessel handle, or more precisely, the number of handles (0, 1 or 2). The method suggests starting with qualitative features if possible because these can be identified on individual artefacts without the need to identify patterns in the entire assemblage; these patterns are better identified in subgroups after an initial division is made (Read, 2007, 214-15).

After the first division based on the handle feature, one of these groups (the one without handles) is now divided based on a quantitative feature, the 'Total Height / Belly Diameter' ratio (more information on these types of features is presented in Section 3.1.8), which represents the overall vessel shape, and identifies two main groups of vessels: 'Squat shape' and 'Urn shape'. The method then continues subdividing the groups according to new features (qualitative or quantitative) until no more subgroups can be identified. The resulting groups can be seen in the last levels of the taxonomic structure (Figure 2.19). Some examples of these final classes are a 'large Urn shape vessel with round belly and decorated rim', and a 'small Squat shape vessel with smooth rim'.

2.2 Machine learning

As briefly commented in the introduction, one of the main characteristics of ML systems is that they use data and standard algorithms to reproduce human knowledge in contrast to extensively programmed systems (Alpaydin, 2016, p. 50-2). There still is need of some coding and understanding of basic data science concepts but it is a more straightforward approach than creating an entire knowledge system from scratch. The main challenges include choosing among the several alternatives of existing algorithms and the best alternatives of parameters for each of them; the choices will depend on the research questions and the characteristics of the dataset. One of the main advantages of ML systems is flexibility: if the pattern in the data changes it is not necessary to rewrite the system's rules, the changes are mostly associated to data, like updating the target

labels. The application of ML has been increasing in many areas of knowledge, including archaeology and cultural heritage (Bickler, 2021; Fiorucci et al., 2020). In this Chapter some of the main applications in these areas are summarised.

ML systems can be classified based on several criteria: according to the amount of human supervision during training, or whether the system can learn incrementally through online data, or how they generalise, through instance-based or model-based learning (Géron, 2019, p. 7-17). Here the distinct types of ML systems based on the amount of human supervision are briefly described.

2.2.1 Supervised learning

In supervised learning systems the objective is to predict a certain outcome (the target) from a given input (the features, the samples' attributes). During the training sessions the algorithms receive both types of information from the dataset, target labels and features (the training set) (Figure 2.9), in order to identify potential patterns and associations between them (Géron, 2019, p. 7-8; Müller & Guido, 2017, p. 27). The next step is to make predictions based on the same features from new, unknown data (the test/validation set) and the information obtained during the training sessions (Figure 2.10).

Figure 2.9 – Example of training dataset for classification. After Scikit (2021b).

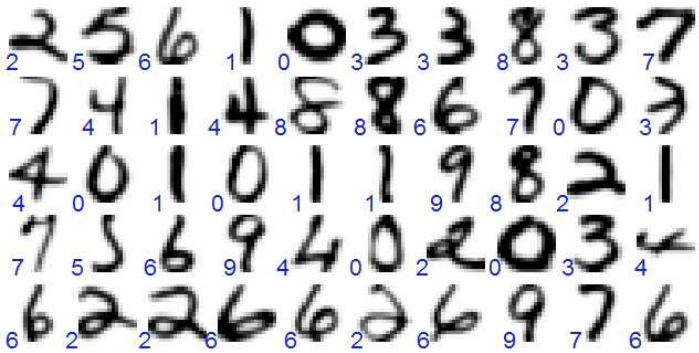


Figure 2.10 – Example of classification: identification of digits based on handwritten samples. After Scikit (2021b).

There are two main types of supervised learning algorithms, which deal with classification or regression problems (Géron, 2019, p. 7-8; Müller & Guido, 2017, p. 27).

In classification the goal is to predict a class label (the target) from a set of predefined values, which can be binary or multiple values. The binary classification is exemplified by the spam filter system: an email is either a spam or not. In the multiclass classification more than two classes form the target set, as in the system that identifies digits from 0 to 9 based on handwritten samples (Müller & Guido, 2017, p. 27; VanderPlas, 2017).

In regression the goal is to predict a continuous number (the target), for instance the approximate value of a house in a certain region based on information like house median age, number of rooms and locality. The value predicted could be any value within a predefined range (e.g., $200,000 to $999,999) (Géron, 2019, p. 8-9, 36; Müller & Guido, 2017, p. 27).

2.2.2 Unsupervised learning

In unsupervised learning systems there is no known output or target labels, knowledge must be extracted only by using input data. It can be more difficult to analyse the results, especially in the case of clustering algorithms, and for this reason this type of system is used in a more exploratory way when compared to supervised learning systems (Géron, 2019, p. 9-12; Müller & Guido, 2017, p. 133-4).

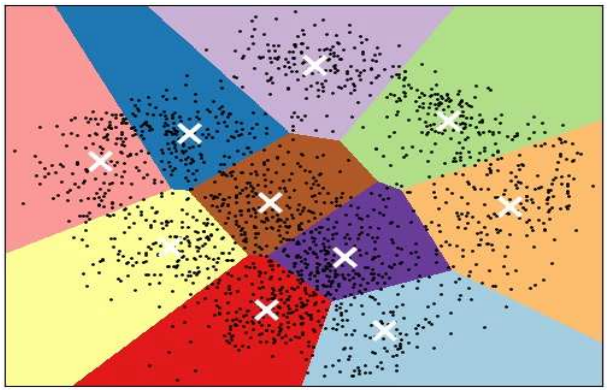


Figure 2.11 – Example of clustering based on the digits dataset; the X marks the centroids of each cluster. After Scikit (2021c).

Clustering is one of the most common types of unsupervised learning methods, other common types are the preprocessing and scaling algorithms, used for preparing the data for supervised learning algorithms (Müller & Guido, 2017, p. 134-42). Clustering algorithms divide the dataset into groups (clusters), aiming to create clusters where samples are similar within a cluster and different from other clusters (Müller & Guido, 2017, p. 169). There are many types of clustering algorithms, among them two are commonly used: k-Means and Hierarchical Clustering.

An example of the application of k-Means is shown in Figure 2.11, the same handwritten digits dataset used for classification in supervised learning now is used without target labels, as a result the algorithm creates clusters with the most similar samples together. In this example the result is easier to interpret because the total number of digits (from 0 to 9) is known in advance, so the number of clusters is equivalent. In the case of Hierarchical Clustering there is no initial parameter used for clustering criteria but there are useful tools for analysis like the dendrogram, described in Section 3.4.2.

2.2.3 Semi-supervised learning

It is an intermediate learning method where the data is partially labeled. Most samples are unlabeled as in the unsupervised learning methods, but some samples which can be labeled are used to better identify the shape of the data distribution and generalise to new samples (Géron, 2019, p. 12-13; Scikit, 2021d). Examples of semi-supervised learning are applications of photo identification, after the algorithm groups photos of the same person (an unsupervised learning process), the labeling of one of each person by the user is enough to propagate the label for other photos that belong to the same group (Géron, 2019, p. 12-13). Another example comes from Hörr et al. (2014), which is commented in Chapter 5.3.

2.2.4 Applications in archaeology

This section presents examples of applications of ML in archaeology and cultural heritage, grouped by the main subject of the research, in several levels: landscape, object/assemblage, ceramic materials, and chronology. Some of the

methods mentioned here involve artificial neural networks such as CNNs (Convolutional Neural Networks): these belong to a specific category of ML models inspired by networks of biological neurons, also associated to the Deep Learning concept, which involves the performing of large and highly complex ML tasks (Géron, 2019, p. 279, p. 289).

Landscape archaeology, automated object detection, remotely sensed data

Barberena et al. (2021) approach the human paleogeography and migrations in the Southern Andes between AD 800 and 1400 based on strontium isotopes analysis, the Random Forest regression algorithm and GIS analysis for construction of an isoscape (geological map of isotope distribution). In the area of archaeological survey, Lambers et al. (2019) integrate citizen science with the automated object detection in remotely sensed data based on CNNs to generate and validate the detection of archaeological objects (barrows, charcoal kilns, Celtic fields) in the Netherlands. Verschoof-van der Vaart and Lambers (2021) give continuity to the previous approach of archaeological survey through an automated object detection model (WODAN - Workflow for Object Detection of Archaeology in the Netherlands) and manual analysis in a way that both methods complement each other. Orengo and Garcia-Molsosa (2019) present an automated system for detection of pottery shards in the landscape based on an EE platform ML algorithm and high-resolution drone imagery, and Orengo et al. (2020) present an automated system for detection of archaeological mounds (Indus settlements from c. 3300 to 1500 BC) in Pakistan and the classification of satellite data using the Random Forest algorithm.

Identification of ceramic shapes through image analysis

Makridis and Daras (2012) present a technique for automatic classification of archaeological shards using k-Nearest Neighbors and feature selection algorithms, where a representative shard of each class is used as reference for the classification of the remaining ones through colour and texture features. Anichini et al. (2021) and Gualandi et al. (2021) present the ArchAIDE, a system/application for collection and automatic recognition of pottery through photos based on two complementary ML tools, one relies on the shard profile/outline and the other on decorative features. Núñez Jareño et al. (2021) make use of synthetic data (replicated features of the original objects) as a strategy

to make the Arch-I-Scan system viable for classification of Roman Fine Ware pottery, as datasets with limited size may cause difficulties for training of ML algorithms. MacLeod (2018) deals with the quantitative assessment of groups/types of North American Paleoindian projectile points through analyses of digital images and 3D scans using geometric morphometric data analysis and ML methods like PCA and Naïve Bayes. Pawlowicz and Downum (2021) present the application of CNNs on images of decorated ceramic for typology and classification of Tusayan White Ware from Northeast Arizona, USA.

Geochemical analysis of soil, ceramic chemical composition and petrography

Oonk and Spijker (2015) present a supervised ML approach to geochemical predictive modeling based on multi-element XRF results from archaeological features and background soils in the Netherlands using ML algorithms (k-Nearest Neighbors, Support Vector Machines) and artificial neural networks. Charalambous et al. (2016) present a method for classification of archaeological ceramics from the Early/Middle Bronze Age Cyprus through their chemical elements using ML algorithms (k-Nearest Neighbors, Decision Trees - C4.5) and the Learning Vector Quantisation (LVQ) method. Mikhailova et al. (2019) apply clustering algorithms (k-Means, DBSCAN and Hierarchical Clustering) to group ceramic and glass artefacts based on their chemical compositions, obtained through XRF analysis. Lyons (2021) uses CNNs for automatically recognise and classify ceramic fabrics from Honduras (AD 1000–1525) based on thin section samples.

Chronology

To conclude the examples of application of ML in archaeology, Klassen et al. (2018) present semi-supervised ML approaches (Multiple regression analysis and Graph-based SSL) for predicting the chronology of temples from medieval Angkor, Cambodia. The prediction is based on a dataset of temples with known architectural elements and artefacts that are used as a reference to estimate the date of most of other temples that are of unknown period.

3 DATA AND METHODS

This chapter provides detailed information on the dataset and ML methods and algorithms used in this research, which were briefly presented in Chapter 1. The methodology can be summarised as the application of supervised and unsupervised ML methods and quantitative classification concepts to a pottery assemblage dataset.

3.1 Dataset

This presentation of the dataset starts with an introduction to the archaeological sites, assemblages and vessel shapes that form the target classes of the ML model, followed by an overview of the dataset and descriptions of the features used to predict the target classes.

3.1.1 Archaeological sites and assemblages

This section provides information on the four archaeological sites that provided the pottery assemblages for this research. The codes starting with 'JZ' follow the Arcane project nomenclature for the sites located in the Jezirah region in Syria, the ancient Upper Mesopotamia region (Arcane, 2016). The location of the sites in the Khabur River plain, between the Tigris and the Euphrates, is shown in Figure 3.1. The information about phases and periods, probable origin/manufacture and contexts/functional categories correspond to the samples that were selected for the research dataset and do not represent the entirety of samples for that sites in the Arcane database.

JZ001 - Tell Brak

One of the largest ancient cities in Upper Mesopotamia, Brak (ancient Nagar) started to develop around 6000 BC as a small settlement, in the late 5th millennium BC (Late 'Ubaid/ Late Chalcolithic 1 periods) it became one of the earliest cities in the Near East (Oates, 2005; Tell Brak, 2013).

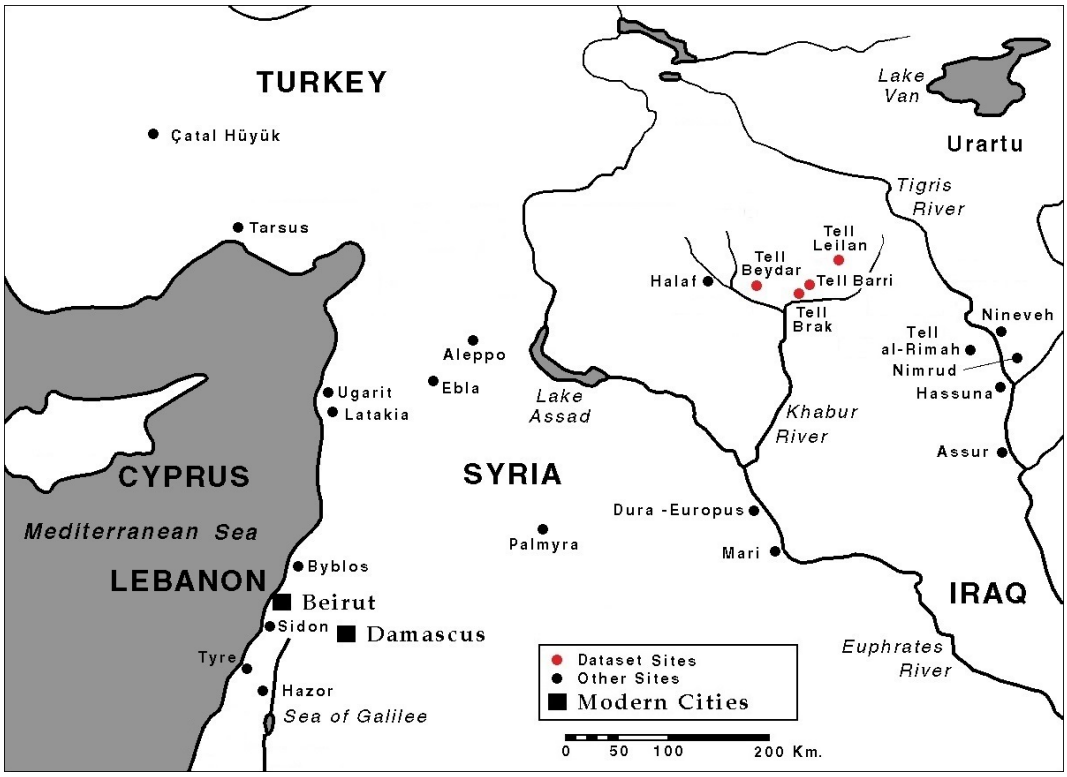


Figure 3.1 – Location of the archaeological sites in the Khabur River plain, Northeastern Syria (Upper Mesopotamia), which provided the pottery samples (dataset) for this research: Tell Beydar, Tell Brak, Tell Leilan and Tell Barri. After OI (2021).

In the 3rd millennium BC it was the dominant city in this part of Upper Mesopotamia, a strategic position in the Khabur River valley made Nagar an important connection among Anatolia, the Levant and Mesopotamia (Oates, 2005). The city was destroyed around 2300 BC and later rebuilt as a centre for provincial administration under the Akkadian kingdom. During the second millennium BC Nagar was under the rule of the Mitanni kingdom, and the Middle Assyrian period is the latest surviving occupation on the main tell (Oates, 2005; Tell Brak, 2013).

Dataset information. Phases and periods: from c. 3000 to 1950 BC. Mostly phases L (pre Akkadian) and N (post-Akkadian/ Hurrian), some phases H (post-Uruk/ pre Ninevite 5) and M (Akkadian) or undefined. Probable origin/manufacture: mostly local, some undefined. Contexts/functional categories: mostly domestic (tableware, storage or food processing), a few ritual or unspecified (Arcane, 2016). Number of samples: 270 (54%).

JZ002 - Tell Beydar

The occupation of Tell Beydar I (ancient Nabada) started c. 2900 BC and its greatest extension lasted until c. 2600 BC when the lower city was abandoned. From this period until c. 2340 BC, the urban settlement remained in the upper city when the its main structures were abandoned, reducing the local to a village until it was no longer occupied around 2100 BC (Pruss, 2013). The period between c. 2340 and c. 2000 BC was marked by urban settlements crisis in the Jezirah region, during this last period of occupation the village remained under the Akkadian control, and it was only briefly reoccupied again during the Hellenistic period (Pruss, 2013). Around the site of Tell Beydar I another settlement known as Tell Beydar II developed during the Mitanni period (c. 14th century BC) and rebuilt under the Neo-Assyrian period (Tell Beydar, 2016); this specific site and period are not represented in the pottery dataset.

Dataset information. Phases and periods: mostly IIIb, some IVa, a few II, IIIa, or IVb (c. 2775 to 2200 BC). Probable origin/manufacture: local. Contexts/functional categories: mostly domestic (storage, tableware, food processing, cosmetic), a few ritual, burial or unspecified (Arcane, 2016). Number of samples: 133 (27%).

JZ004 - Tell Leilan

The initial occupation of Tell Leilan is recorded during the late northern ‘Ubaid period, and continued through the Uruk period until c. 3200 BC, when the settlement started to decline (Weiss, 2013). Around 2600 BC a new phase of expansion started with the construction of monumental walls and public buildings, until its decline c. 2200 BC under the Akkadian rule (Weiss, 2013). After near two centuries of abandonment, the city redeveloped as an Amorite capital (Shubat-Enlil) c. 1950 BC. Its final ancient occupation was under the Mitanni rule during the 15th century BC (Weiss, 2013).

According to Weiss (2013, p. 109-10), the decline in the urban occupation in the Khabu River plains during a period of around 300 years, c. 2200 to 1900 BC (which affected also other cities like Tell Brak and Tell Beydar), might have been triggered by an abrupt decline in precipitation and cooling event known from paleoclimate proxy records.

Dataset information. Phases and periods: EJ II and EJ II/III (c. 2800 to 2500 BC). Probable origin/manufacture: local. Contexts/functional categories: domestic (storage, food processing) and burial (Arcane, 2016). Number of samples: 69 (14%).

JZ007 - Tell Barri

The initial occupation of Tell Barri (ancient Kahat) is recorded from the end of the 4th millennium BC with relevant phases of occupation during the 3rd millennium BC and the Middle Assyrian period, after the decline of important urban centres in the region such as Tell Brak and Tell Leilan (Palermo, 2019). After the collapse of the Assyrian empire, Tell Barri had a shorter record of occupation under the Achaemenid period and later nearly two centuries of occupation during the Hellenistic period, and records of occupation continued under the Parthian, Roman and Sasanian rules until the 4th century AD (Palermo, 2019).

Dataset information. Phases and periods: strata 37, 39, 41 (EJ II/IIIa to EJ IV, c. 2650 to 2170 BC). Probable origin/manufacture: local and imported. Contexts/functional categories: tableware and burial (Arcane, 2016). Number of samples: 24 (5%).

3.1.2 Vessel shape

In the Arcane database the ‘shape class’ feature is divided in two parts, one containing a code varying from ‘A’ to ‘Z’ and the other one containing a description of the vessel shape. In some cases different shape descriptions are assigned to vessels belonging to the same shape class, for instance the sample JZ001_P916 (T class) has ‘Bottle’ as shape description, whereas sample JZ001_P925 (same T class) has ‘Flask’ instead. Another cases are the G class, which has samples described either as ‘Beaker’ or ‘Cup/Mug’, and the K class, which has samples described either as ‘Jug/Tankard’ or ‘Juglet’. It is the shape code that is used as the label for classification and not the shape description, therefore this is not an issue, but the description may provide information about possible sub divisions in a class, as it will be commented in the discussion section.

The shapes are divided in three groups: open shapes, closed shapes and miscellaneous shapes. In this research the shapes that belong to the miscellaneous

group and some of the open and closed shapes will not be used, see Section 3.1.9 for details on sample selection criteria. Table 3.1 lists all the shape classes defined by the Arcane project and the ones used in this research.

Group	Shape Class	Shape Description	# of samples
Open Shapes	A	Plaque	1
	B	Dish/Plate, Platter, Pan	1
	C	**Shallow Bowl**	**10**
	D	Large Bowl	3
	E	**Bowl**	**182**
	F	Deep Bowl	7
	G	**Cup/Mug, Beaker**	**96**
	H	**Open Pot**	**20**
	J	Vat	2
Closed Shapes	**K**	**Jug/Tankard, Juglet**	**22**
	M	Closed Pot (rounded, squat)	9
	N	**Closed Pot (high)**	**34**
	P	**Jar (wide neck)**	**88**
	R	**Jar (restricted neck)**	**32**
	S	Pithos	4
	T	**Flask, Bottle**	**12**
Miscellaneous Shapes	L	Lamp	-
	V	Anthropomorphic or zoomorphic vessel	-
	W	Composite vessel	4
	X	Vessel with horizontal axis	-
	Y	Vessel without rotation axis	-
	Z	Stand, Andiron	7
			496

Table 3.1 – List of shape classes defined by the Arcane project (Arcane 2016) and number of pre-selected samples from the four sites. The classes used in this research are marked in bold.

The following specifications of open and closed shape classes present in the Arcane project were based on information provided by D. Meijer (personal communication, February 17, 2021). Some image examples are shown for the classes used in the research dataset. The dimensions specified for each shape (diameters, height) serve as a general reference and may not be strictly followed, therefore these dimensions alone cannot be used to characterise a shape class.

Open shapes:

A - Plaque

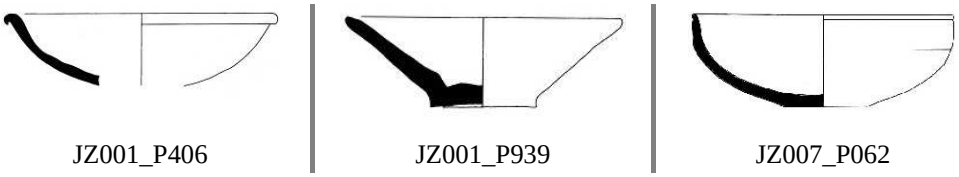
- Diameter is 12 or more times the vessel height

B - Dish, Plate, Platter, Pan

- Diameter is between 6 and 12 times the vessel height

C - Shallow Bowl

Examples:



JZ001_P406 | JZ001_P939 | JZ007_P062

Figure 3.2 – Examples of vessels from the ‘C – Shallow Bowl’ shape class. Images at different scales. After Arcane (2016).

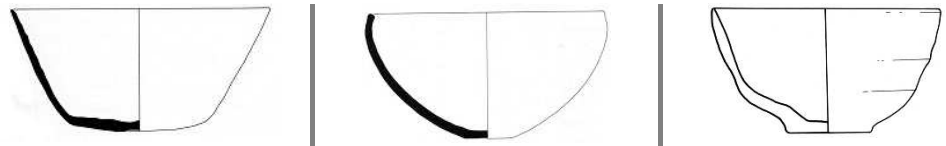
- Diameter is between 3 and 6 times the vessel height
- Maximum diameter is 30 cm

D - Large Bowl

- Diameter is between 3 and 6 times the vessel height
- Diameter is greater than 30 cm

E - Bowl

Examples:



JZ001_P015 | JZ001_P046 | JZ002_P028

Figure 3.3 – Examples of vessels from the ‘E –Bowl’ shape class. Images at different scales. After Arcane (2016).

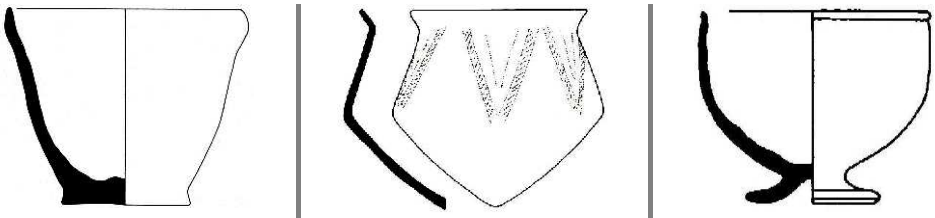
- Diameter is between 1.5 and 3 times the vessel height
- Maximum diameter is 30 cm
- Maximum height is 20 cm

F - Deep Bowl

- Diameter is between 1.5 and 3 times the vessel height
- Diameter is greater than 30 cm
- Shares some characteristics with class ‘J – Vat’

G - Cup/Mug, Beaker

Examples:



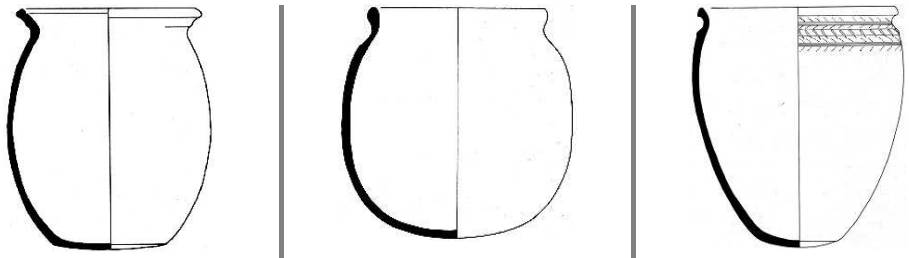
JZ001_P110 JZ004_P403 JZ004_P079

Figure 3.4 – Examples of vessels from the 'G – Cup/Mug/Beaker' shape class. Images at different scales. After Arcane (2016).

- Diameter is up to 1.5 times the vessel height
- Maximum diameter is 30 cm
- Maximum height is 20 cm

H - Open Pot

Examples:



JZ001_P073 JZ001_P087 JZ001_P252

Figure 3.5 – Examples of vessels from the 'H – Open pot' shape class. Images at different scales. After Arcane (2016).

- Diameter is up to 1.5 times the vessel height
- Height is greater than 20 cm

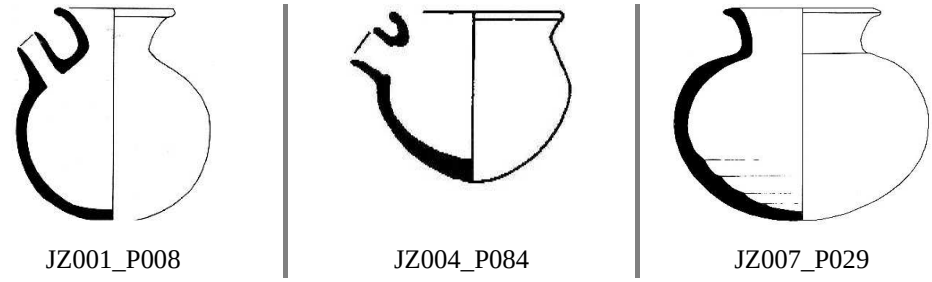
J - Vat

- Diameter is up to 1.5 times the vessel height
- Height is greater than 20 cm
- Shares some characteristics with class 'F - Deep Bowl'

Closed shapes:

K - Jug/Tankard, Juglet

Examples:



JZ001_P008 | JZ004_P084 | JZ007_P029

Figure 3.6 – Examples of vessels from the 'K – Jug/Tankard/Juglet' shape class. Images at different scales. After Arcane (2016).

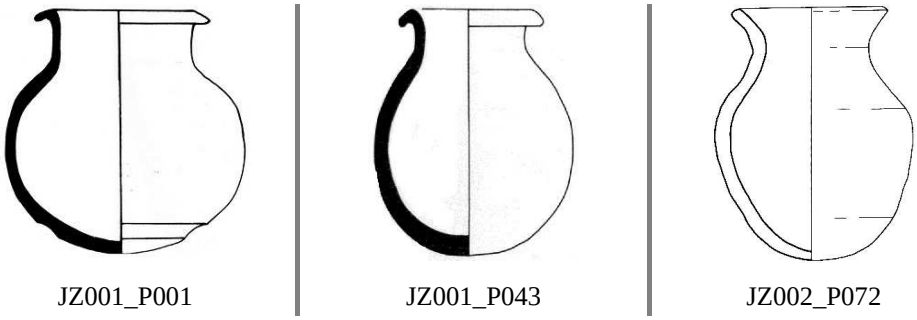
- Maximum height is 35 cm (15 cm for Juglet)
- Minimum diameter is between 20% and 60% of the Maximum diameter

M - Closed Pot (rounded, squat)

- Maximum height is 70 cm
- Height is less than or equal to maximum diameter
- Minimum diameter is between 60% and 80% of the Maximum diameter

N - Closed Pot (high)

Examples:



JZ001_P001 | JZ001_P043 | JZ002_P072

Figure 3.7 – Examples of vessels from the 'N – Closed Pot (high)' shape class. Images at different scales. After Arcane (2016).

- Maximum height is 70 cm
- Height is greater than or equal to maximum diameter
- Minimum diameter is between 60% and 80% of the Maximum diameter

P - Jar (wide neck)

Examples:

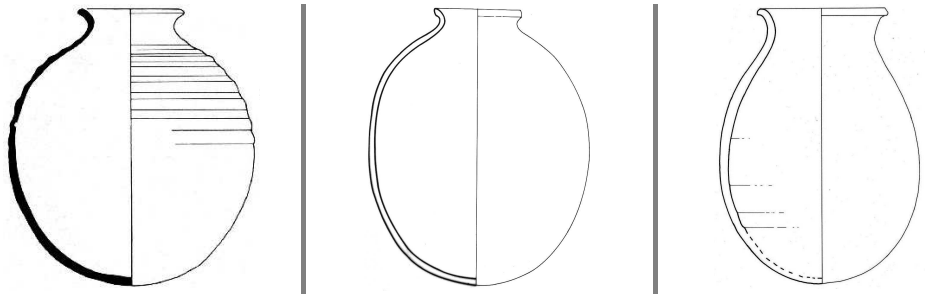
JZ001_P260 | JZ001_P923 | JZ004_P036

Figure 3.8 – Examples of vessels from the 'P – Jar (wide neck)' shape class. Images at different scales. After Arcane (2016).

- Height is between 35 and 70 cm
- Minimum diameter is between 40% and 60% of the Maximum diameter

R - Jar (restricted neck)

Examples:



JZ001_P272 | JZ002_P105 | JZ002_216

Figure 3.9 – Examples of vessels from the 'R – Jar (restricted neck)' shape class. Images at different scales. After Arcane (2016).

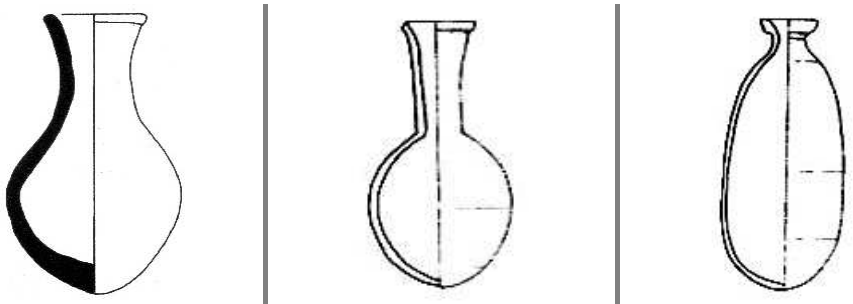
- Height is between 35 and 70 cm
- Minimum diameter is between 20% and 40% of the Maximum diameter

S - Pithos

- Height is greater than 70 cm
- Minimum diameter is between 20% and 80% of the Maximum diameter

T – Flask, Bottle

Examples:



JZ001_P925 | JZ002_P603 | JZ002_P618

Figure 3.10 – Examples of vessels from the 'T – Flask/Bottle' shape class. Images at different scales. After Arcane (2016).

- Minimum diameter is up to 20% of the Maximum diameter
- Maximum height is 15 cm for Flask

3.1.3 Dataset overview

Table 3.2 shows the summary information for the dataset used to train and test/validate the ML model. The detailed explanation for all features and the values associated with some of the features such as rim orientation and base typology are presented in the next sections.

Features are divided in two broad types, according to their scale of measurements: categorical [cat] and continuous [con]. Categorical features have a limited number of qualitative and mutually exclusive possible outcomes, the analyst or expert defines their possible contents (Fletcher & Lock, 2005, p. 1-5; VanPool & Leonard, 2011, p. 5-11) during the database specification for instance, or they may be defined by some predetermined rule (the case of the Vessel ID feature). The 'nominal' is the only categorical subtype used in the dataset. Continuous features are quantitative, they may represent either an interval (a sequence with fixed distances) or fixed distances with a datum (fixed zero) point (Fletcher & Lock, 2005, p. 1-5), which is the case of all continuous features in the dataset used in this research. The continuous features in the dataset represent either vessel measurements (cm or litres) or ratio between measurements.

#	Feature Name	Type	Description
1	Vessel ID	[cat]	Unique identification of the vessel composed by site code and sequence number
2	Shape class	[cat]	Shape of the vessel e.g. Jar, Bowl, Open pot, Closed pot represented by letters from A to Z
3	Rim orientation	[cat]	Orientation of vessel rim e.g. vertical, out-turned, in-turned, codified as A, B, C, D or E
4	Rim profile	[cat]	Profile of the vessel rim, codified from 00 to 22
5	Base typology	[cat]	Shape of the vessel base e.g. flat, pointed, rounded, codified from 00 to 15
6	Miniature vessel	[cat]	Indicates whether the vessel is a miniature form of a vessel shape. N = No; Y = Yes
7	Additional elements	[cat]	Indicates whether the vessel has any handle, lug or spout as an additional element. H = Handle(s); L = Lug(s); S = Spout; N = No elements
8	Total Height	[con]	Total height of the vessel (cm)
9	Diameter at opening	[con]	Diameter at vessel opening (cm) (Rim diameter)
10	Minimum diameter	[con]	Minimum vessel diameter (cm)
11	Maximum diameter	[con]	Maximum vessel diameter (cm)
12	Base diameter	[con]	Diameter of the vessel base (cm)
13	Capacity	[con]	Vessel capacity (litres)
14	Neck diameter	[con]	Diameter of the vessel neck (cm)
15	Belly diameter	[con]	Diameter of the vessel belly (cm)
16	Neck height	[con]	Height from vessel base to neck (cm)
17	Belly height	[con]	Height from vessel base to belly (cm)
18	H-Bd	[con]	Total Height / Belly diameter ratio
19	Bd-Rd	[con]	Belly diameter / Rim diameter ratio
20	Bd-Nd	[con]	Belly diameter / Neck diameter ratio
21	H-Bh	[con]	Total Height / Belly height ratio
22	H-Nh	[con]	Total Height / Neck height ratio
23	Bd-BaD	[con]	Belly diameter / Base diameter ratio

Table 3.2 – Vessel features used in the ML model. Detailed information for each feature is provided in the following sections. Types: [cat]=categorical; [con]=continuous.

The features are also divided in six groups (separated by bold lines in Table 3.2), according to their origin (if provided by the Arcane database or created specifically for the model) and purpose. Feature #1 (Vessel ID) has the purpose to uniquely identify each site and sample vessel and does not have any influence in the classification. Feature #2 (Shape class) is the information that will be used to evaluate the model (the target). In the supervised phase of the research, the model will attempt to assign the vessels to one of these pre-defined shapes; in the unsupervised phase the model will suggest potential new classes or sub-classes based on the vessel features.

Features #3 to #23 will be used as information for the classification model. The third group (features #3 to #7) is formed by categorical features that are provided by the Arcane database. The fourth group (features #8 to #13) is formed by features of continuous values, the vessel measurements that are provided by the database.

The last two groups include features that were not provided by the database but can be either approximately measured based on the pottery illustrations or calculated from basic measurements. These features/measurements were considered relevant for shape determination (Hörr et al., 2014; Orton et al., 1993, p. 152-65; Read, 2007). The fifth group (features #14 to #17) includes measurements, and the sixth group (features #18 to #23) includes ratios calculated from basic measurements.

The Arcane database provides a number of vessel features besides the ones shown in Table 3.2, however these will not be used in the classification model: probable origin/manufacture (local or non-local), functional category (e.g. domestic, storage, ritual), fabric (ware quality, hardness, inclusions), firing and building techniques, surface treatments, marks and decoration. Albeit some of these are relevant information for identification of vessel function and, in the case of decoration, for typology and relative dating, they would not contribute to the identification of vessel shape, which is the main reference for the classification model and the focus of this research.

Some categorical features (rim orientation, rim profile and base typology) have NA (null) values for some samples in the original Arcane database. Since scikit-learn ML algorithms do not process features with NA values (Scikit, 2021e) these values were replaced by codes meaning ‘Undefined’ in the research dataset as explained in the next sections.

3.1.4 Rim orientation and profile

Information about rims in the Arcane database are divided in two features: orientation and profile.

a) Orientation

There are three types of rim orientation in the research dataset (Figure 3.11, Table 3.3), 'B - Out-turned' being the most common (present in 79% of the samples). The code 'E – Undefined' was created for this research replacing the NA values from the Arcane dataset because the scikit-learn algorithms require that categorical data have no null (NA) values when using preprocessing encoders such as *OneHotEncoder* (Scikit, 2021e; Scikit, 2021f).

Type	Description	# of samples
A	Vertical	60
B	Out-turned	390
C	In-turned	43
D	Asymmetrical	-
E	Undefined	3
		496

Table 3.3 – List of rim orientations defined by the Arcane project (Arcane 2016). Only the first three types are present in the research dataset; the type 'E' – Undefined' was created to replace the NA values from the Arcane dataset.

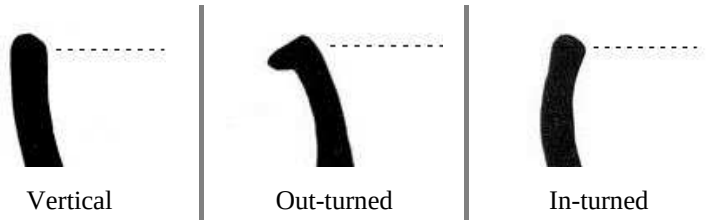


Figure 3.11 – Examples of the three types of rim orientation in the dataset. After Arcane (2016) images: JZ001_P004, JZ001_P003, JZ001_P046.

b) Profile

The two most common rim profiles are '01 – Thinned' and '03 – Rounded' (Table 3.4). Among the profiles that are folded, the outside folded is the prevailing type. The code '00 – Undefined' was created for this research replacing the NA values from the Arcane dataset for the same reason described for the rim orientation feature.

Type	Description	# of samples
01	Thinned	146
02	Squared	39
03	Rounded	125
04	Thickened	26
05	Bevelled outside	4
06	Bevelled inside	3
07	Round-folded outside	62
08	Round-folded inside	-
09	Horizontal folded outside	38
10	Horizontal folded inside	-
11	Thin-folded outside	15
12	Thin-folded inside	-
13	Square/flat-folded outside	10
14	Square/flat-folded inside	-
15	Angular-folded outside	10
16	Angular-folded inside	-
17	Moulded outside	8
18	Moulded inside	-
19	Gutter outside	-
20	Gutter inside	-
21	Hammer	-
22	Hammer moulded	-
00	Undefined	10
		496

Table 3.4 – List of rim profiles defined by the Arcane project (Arcane 2016). Some types are not present in the research dataset, and some rim profiles are undefined.

3.1.5 Base typology

Type	Description	# of samples
01	Pointed	47
02	Rounded	155
03	Flattened	58
04	Flat	136
05	Concave	12
06	Disk	31
07	Disk concave	10
08	Ring	-
09	Ring high	18
10	Pedestal	5
11	Ring folded	-
12	Ring protruding	-
13	Ring added	-
14	Button	-
15	Stump	5
00	Undefined	19
		496

Table 3.5 – List of base types defined by the Arcane project (Arcane 2016). Some types are not present in the research dataset, and some samples do not have a base preserved (undefined).

Among the several base types found in the dataset, five types are the most common: ‘01 – Pointed’, ‘02 – Rounded’, ‘03 – Flattened’, ‘04 – Flat’ and ’06 – Disk’ (Figure 3.12, Table 3.5). The type ‘00 – Undefined’ was created for this research replacing the NA values from the Arcane dataset for the same reason described for the rim orientation feature.

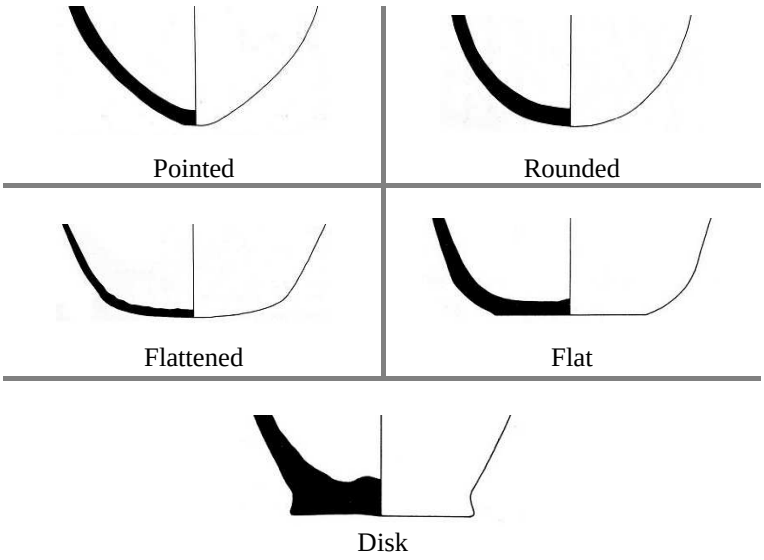


Figure 3.12 – The five most common base types in the dataset, together they are present in 85% of the samples. After Arcane (2016) images: JZ001_P209, JZ001_P009, JZ001_P014, JZ001_P005 and JZ001_P110

Depending on the vessel shape, especially for round-based vessels, it can be difficult to distinguish the base from the side of the vessel (Rice, 1987, p. 212-4). Some samples with pointed or rounded base types have low values (less than 1 cm) indicating a very small base or even a zero value, while vessels with similar shapes have greater values for the base diameter. Some samples have the base missing, in these cases the base diameter is recorded as NA.

The base diameter is one of the key measurements used to identify the vessel shape (Section 3.1.8), and if different criteria are used for similar shapes it may affect the performance of ML algorithms. This issue is discussed in Chapter 5. When the base diameter has a zero value, the relative measurement Bd-BaD (Belly diameter / Base diameter ratio) is equal to the belly diameter.

3.1.6 Miniature vessels

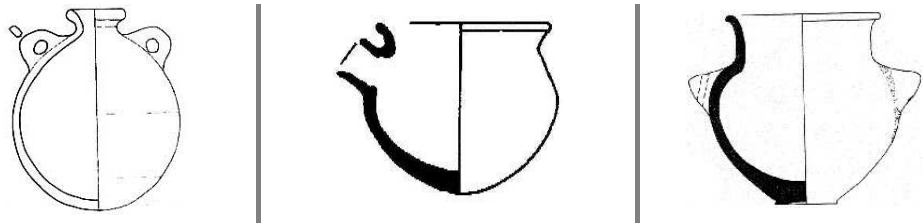
Some vessels are marked as miniatures in the Arcane database. There are only 17 samples of this type in the research dataset, belonging to six different shape classes, the ‘E – Bowl’ is the most common one with six samples, followed by ‘G – Cup/Mug’ with three samples. In the dataset most of miniature vessels have relatively small height (from 2.4 to 5.8 cm), small opening diameter (from 2.0 to 8.3 cm) and low capacity (from 0.02 to 0.22 litres) but no explicit criteria were found for classifying vessels as miniatures.

Some miniature vessels have distinct characteristics, but in general it seems that there are not enough differences from other vessels within the same shape classes that would justify classifying the miniatures as different classes. A particularly interesting case is the shallow bowl JZ002_P113, which has an opening diameter equal to 8 cm, and it is very similar to vessel JZ001_P126, classified as a bowl (opening diameter equal to 11.8 cm), and not marked as a miniature. Both vessels have a peculiar shape and come from different sites.

The use of this feature in the research is related to the investigation of the relevance of the size of vessels in the definition of their shape classes.

3.1.7 Additional elements

Some vessels may have additional elements, in the case of the research dataset these can be handles, lugs or spouts (Table 3.6, Figure 3.13).



JZ002_P622 JZ004_P084 JZ001_P307

Figure 3.13 – Examples of vessels with additional elements. Images at different scales. Left: Jug with handles; center: Jug with spout; right: Jar (wide neck) with lugs. After Arcane (2016).

The types are adapted and summarised from the Arcane project database, which includes more details such as the position (upper, middle or lower part of the vessel body, or in the vessel rim), quantity and orientation (vertical/horizontal) of these elements. Since there are only 16 vessels that have additional elements in the research dataset (around 3% of the samples), only the most basic information (element type) was kept in order to not add unnecessary complexity to the ML model.

Type	Description	# of samples
H	Handle(s)	1
L	Lug(s)	9
S	Spout	6
N	Without additional elements	480
		496

Table 3.6 – List of additional elements that may be present in some vessels.

3.1.8 Vessel measurements

There are three types of vessel measurements used in this research: i) original measurements from the Arcane project database; ii) additional measurements, based on vessel drawings from the Arcane project; and iii) relative measurements, which are calculated on the basis of the other two types of measurements.

Arcane measurements

- Height (cm)
- DO - Diameter at Opening (cm)
- D min - Minimum Diameter (cm)
- D max - Maximum Diameter (cm)
- D base - Diameter at Base (cm)
- Capacity (litres)

Additional measurements

These measurements (Figure 3.14) are based on Read (2007) and apply mostly to closed shapes because they are based on the concepts of neck and belly vessel parts (many open shapes vessels do not have these parts clearly defined), nevertheless the method was adapted to include also the open shape vessels as explained in the next paragraphs.

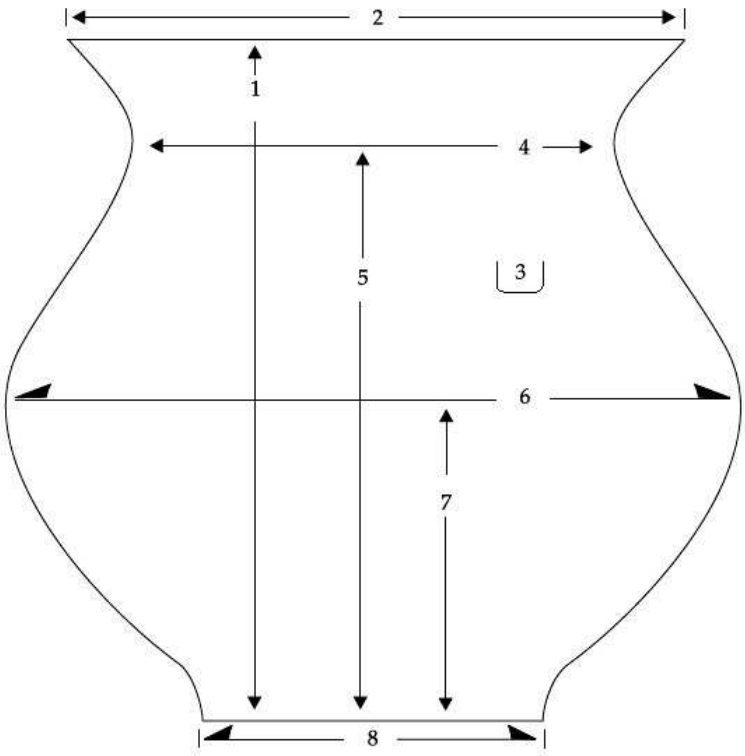


Figure 3.14 – Vessel basic measurements. After Read (2007, Figure 8.14). (1) total height; (2) rim diameter; (4) neck diameter; (5) height to neck; (6) belly diameter; (7) height to belly; (8) base diameter. In this research the capacity in litres replaces the original measurement (3) square root of cross sectional area.

- *Vessel height (cm)*
- *Rim diameter (cm)*
- Neck diameter (cm)
- Height to Neck (cm)
- Belly diameter (cm)
- Height to Belly (cm)
- *Base diameter (cm)*

For this research purposes, three of these seven measurements (in italic) are exactly the same (vessel height, base diameter) or equivalent (diameter at opening/rim diameter) as three of the Arcane measurements. The other four measurements were obtained based on the vessel drawings available in the Arcane project using the ImageJ software, some of these drawings are shown in Sections 3.1.2 and 4.2.2. The combination of the six Arcane measurements with these four additional measurements resulted in the ten basic measurements listed in Table 3.2. The relative measurements, which are obtained through the basic ones, are described in the next section.

Many samples from the open shape group, especially bowls and some types of beakers, do not have a neck or do not have neither belly nor neck, such as the examples in Figure 3.15. In this case, these shapes do not fit exactly in the method proposed by Read (2007) and used in this research, nevertheless there are some alternatives to characterise the vessel parts or anatomy (Rice 1987, p. 211-22).

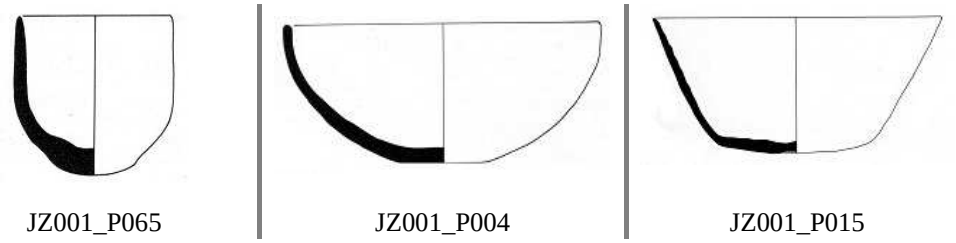


Figure 3.15 – Examples of open shape vessels without neck: beaker (left), and without neck and belly: shallow bowl (center) and bowl (right). Images at different scales. After Arcane (2016).

The alternatives for this research were to use:

- Vessel height in place of the height to neck measurement (Figure 3.14) in the case of vessels without neck;
- Vessel height in place of both the height to neck and height to belly measurements in the case of vessels without neck and belly;
- Diameter at opening (rim diameter) in place of the neck and belly diameters.

The measurements neck diameter and belly diameter (taken with ImageJ) are, for closed shapes vessels, in most cases coincident with the measurements minimum diameter and maximum diameter from the Arcane database. The equivalence

between neck diameter and minimum diameter measurements is 96% in average, and between belly diameter and maximum diameter is 98% in average, for closed shapes in the research dataset. The neck diameter and belly diameter measurements are preferred in the relative measurements formulas because of the significant number of minimum diameter and maximum diameter measurements which are missing (have NA values) in the Arcane database, and also to keep the original formulas proposed by Read (2007) as shown in the next section.

Relative Measurements

Since the neck diameter and belly diameter measurements (taken with ImageJ) are used in the relative measurements calculations, the other most relevant measurements (taken from the Arcane database) are height, diameter at opening and diameter at base. Both height and diameter at opening are recorded for all samples used in the research dataset. Diameter at base is missing for a number of samples mainly because the vessel base is not preserved; in these cases the formula Bd-BaD (Belly diameter / Base diameter ratio) is recorded as NA (not available) in the research dataset. When 'Diameter at base' is zero, the Bd-BaD value is equal to the belly diameter.

These following measurements are based on Read (2007); a similar use of measurement-based classifications is mentioned in Orton et al. (1993, p. 155-8).

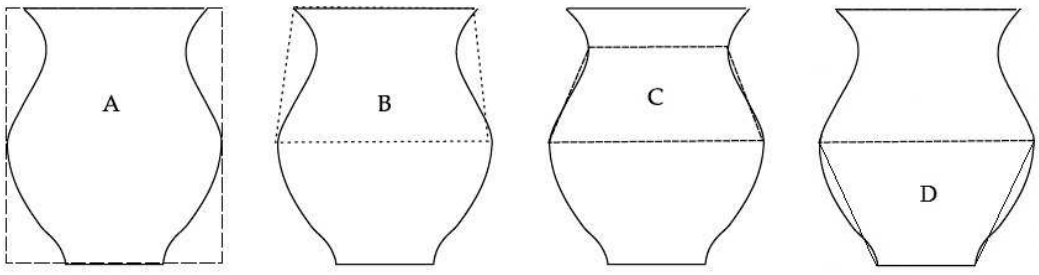


Figure 3.16 – Vessel relative measurements: (A) overall vessel shape based on the total Height / Belly diameter ratio; (B) upper portion, based on the Belly diameter / Rim diameter ratio; (C) neck portion, based on the Belly diameter / Neck diameter ratio and the relative location of the neck in the vessel height; (D) lower portion, based on the Belly diameter / Base diameter ratio. After Read (2007, Figure 8.15).

Overall shape:

- Total Height / Belly diameter ratio (Figure 3.16-A).

Upper portion:

- Belly diameter / Rim diameter ratio (Figure 3.16-B). If ratio > 1 then the upper portion is convergent, divergent if ratio < 1 or parallel if ratio = 1.
- Belly diameter / Neck diameter ratio (Figure 3.16-C). Identifies the vessel shape located between the belly and the neck.
- Total height / Neck height ratio: relative location of the neck in the vessel vertical dimension.

Lower portion:

- Belly diameter / Base diameter ratio (Figure 3.16-D). Identifies the vessel shape located between the belly and the base.
- Total height / Belly height ratio: relative location of the belly in the vessel vertical dimension.

3.1.9 Sample selection

The criteria to select the samples in the Arcane database were:

1. They must belong to archaeological sites that are culturally related, located in the same region and within the same broad time period.

2. They must be enough well preserved so that their shape can be identified through a minimum of basic measurements and categorical features, and preferably have also images (drawings).

3. They must belong to the open shapes or closed shapes groups. Shapes from the miscellaneous shapes group have no measurement parameters for defining the shape classes, have very particular shapes and there are few or none samples among the selected sites (except for the Z class).

4. There must be at least 10 samples of the shape class among the different sites. During the dataset split into training and test parts, the test will have at least tree samples from any shape class (¼ of the class total samples, that's 2.5 rounded up to 3 when using stratified distribution). Classes that have too few samples may be problematic for pattern recognition by ML algorithms (Géron, 2019, p. 22-6).

As a result of these selection criteria, the final dataset used in this research is composed by a total of 496 samples divided in nine shape classes.

3.2 Software

The toolkit used to create the ML model was chosen based on recommendations from A. Brandsen, researcher in Digital Archaeology at the Faculty of Archaeology, giving preference to Open Source software. The possibility for contribution to Open Science practices was also considered in this research. The dataset and the Jupyter notebooks created to run one supervised learning training session (Section 4.1.3) and the clustering analyses with k-Means (Section 4.2.1) are available in the Zenodo repository (Zenodo, 2013). Instructions on how to access these resources can be found in Appendix C.1.

3.2.1 Machine learning toolkit

- *Anaconda*: a distribution of the Python and R programming languages and a platform for data science and ML applications (Anaconda, 2021), allows the execution of packages and libraries such as *scikit-learn* and *Jupyter* used in this research. Version: 3-2020.11 (Windows)
- *scikit-learn*: an open source ML library of algorithms for supervised and unsupervised learning methods such as classification and clustering (Pedregosa et al., 2011; Scikit, 2021a). Version: 1.0
- *Jupyter Notebook*: an open source, interactive web-based environment (Jupyter, 2021), used to define the training/test datasets, set parameters and control the execution of ML algorithms. Version: 6.1.4

3.2.2 Additional software

- *ImageJ*: a public domain Java image-processing program (ImageJ, 2021), used for taking vessels measurements not provided by the Arcane database. Version: 1.53k

- *SciPy*: an open source library of algorithms for scientific computing in Python (Scipy, 2021a), used for the dendrogram generation. Version: 1.8.0
- Spreadsheet for creating the research dataset, based on information obtained from the Arcane database application. The dataset is further converted to .CSV format and loaded into the Jupyter Notebook.

3.3 Supervised learning methods and algorithms

The supervised learning part of this research is based on classification methods and algorithms, which are detailed in this Chapter, starting with information about the dataset structure. The research dataset is divided in four parts: the target classes are separated from the features used to characterise the classes, and the samples used to train the ML model are separated from the samples used to test it.

3.3.1 Target classes and features

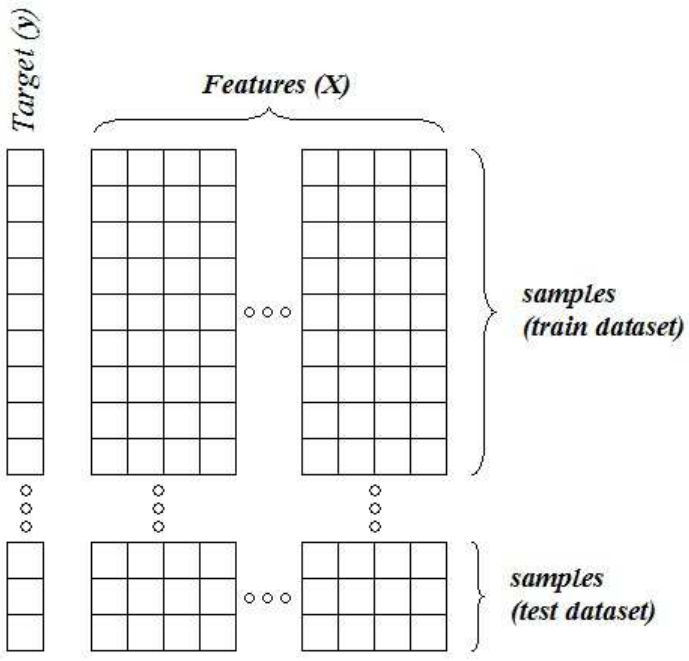


Figure 3.17 – The dataset divided into four parts: target classes vs. features, train vs. test dataset. After VanderPlas (2017, Figure 5-13).

In scikit-learn the dataset is implemented by a matrix of features and samples (by convention represented as *X*) and a target array (represented as *y*); the features are

the columns and the samples are the rows in the X matrix (Figure 3.17) (VanderPlas, 2017). The features must be numeric (continuous or discrete-valued) in scikit-learn, while the target array may be numeric or categorical, representing discrete classes/labels (VanderPlas, 2017), depending on the type of supervised learning main problem, regression or classification. Because of this scikit-learn requirement, the categorical features in the research dataset must be processed through the encoding method as detailed in Section 3.3.3.

3.3.2 Training and test datasets

In addition to the division between target classes and features, in supervised learning problems the dataset is divided into training and test (or validation) parts (Figure 3.17). The training set is used to build the ML model, while the test set is used to assess how the model will *generalise*, or how well the model works on unknown data (Müller & Guido, 2017, p. 17). The model uses the knowledge provided by the associations between features (*X_train*) and target classes (*y_train*) in the training set to predict the target classes in the test set based on its features (*X_test*); the predicted classes (*y_pred*) are then compared to the test target classes (*y_test*) to assess the model accuracy.

The dataset is split in the following proportions, following the scikit-learn default (Müller & Guido, 2017, p. 17): 75% of the samples are used for training, and 25% of the samples are used for test/validation of the model. Other proportions such as 80/20 or 70/30 could also have been used. In order to guarantee reproducibility of the ML sessions and results, the same value for the 'random_state' parameter (used by the 'train_test_split' method) is used across all the algorithms and training sessions (Müller & Guido, 2017, p. 17-18).

A final important parameter is related to whether the distribution of samples during the application of the 'train_test_split' method is *stratified* or *not stratified*. In the stratified variation, the distribution of samples for each shape class in the test dataset is proportional to their quantities in the full dataset; when the distribution is not stratified, this proportion is not taken into consideration when the test dataset is created.

3.3.3 Encoding

Scikit-learn ML algorithms do not process categorical features directly (Scikit, 2021f). It is necessary to convert them in some numerical form before using the dataset in the training process. There are two alternatives available in the scikit-learn library (Géron, 2019, p. 65-7; Müller & Guido, 2017, p. 213-24): OrdinalEncoder and OneHotEncoder. The first method simply converts the categorical values into numeric ones, whereas OneHotEncoder creates one binary feature per category (for each sample and category, only one feature will be equal to 1 and the others will be equal to 0). The feature rim orientation is used as an example in Table 3.7.

(1) OrdinalEncoder:

	ORIGINAL DATA	ENCODED DATA
Vessel ID	**Rim orientation**	**Rim orientation**
JZ001_P001	B	2
JZ001_P004	A	1
JZ001_P037	B	2
JZ001_P038	C	3

(2) OneHotEncoder:

	ORIGINAL DATA	ENCODED DATA		
		Rim orientation		
Vessel ID	**Rim orientation**	**A**	**B**	**C**
JZ001_P001	B	0	1	0
JZ001_P004	A	1	0	0
JZ001_P037	B	0	1	0
JZ001_P038	C	0	0	1

Table 3.7 – Examples of results from different scikit-learn data encoding methods based on the rim orientation feature.

The main issue with the OrdinalEncoder method is that ML algorithms will assume that the categories are ordered, and two nearby values would be more closely related than distant values (Géron, 2019, p. 66; Scikit, 2021f). For instance, the rim orientation ‘A’ (encoded as 1) would be considered more related to ‘B’ (encoded as 2) than to ‘C’ (encoded as 3), which is not true for this specific feature and the other categorical features in the research dataset. One disadvantage of OneHotEncoder is when the feature has a large number of possible categories, resulting in a large number of input features in the ML model, which may create performance issues during the training sessions (Géron 2019, 67). Since none of

the categorical features in the dataset have a large number of categories and the dataset is not large, this is not a problem here and the OneHotEncoder was selected for the dataset preparation for the ML algorithms. All the categorical features, except shape class (the model target), are encoded to numerical format before processing by ML algorithms. The features rim profile and base typology, despite being codified as numbers, are also encoded because of the ordering issue already mentioned.

Values from categorical features that have very few occurrences in the dataset (e.g., rim_profile[6] 'Bevelled inside' occurs only three times) may cause problems after the dataset is split into train and test parts. If the test part does not have any occurrence of the categorical feature value (which is present in the train dataset), scikit-learn algorithms return an error because the number of features between the two datasets parts does not match. A possible alternative in case this situation occurs is to change the 'random_state' parameter used by the 'train_test_split' method (Section 3.3.2) and by some algorithms, which causes a new distribution of samples between the train and test datasets.

3.3.4 Missing values

Scikit-learn ML algorithms do not process features with NA (null) values (Scikit, 2021e). In the case of categorical features, one solution adopted was to replace the null values with codes representing 'undefined' values in the research dataset, before the processing by ML algorithms (Sections 3.1.4 and 3.1.5). In the remaining (continuous) features, the solution was the application of the scikit-learn *impute* method, which replaces the null values with a choice of alternatives (the feature mean, median, most frequent or a constant value such as zero). In this research the choice was to replace all the null values in the continuous features with zeroes.

3.3.5 Classification algorithms

For the supervised learning part of this research, six classification algorithms were selected, including two ensemble methods, which combine the predictions of several base algorithms (Scikit, 2021g). The four base algorithms follow distinct

principles for classification, they all have strengths and weaknesses and their performance will depend on the research problem, the dataset characteristics (e.g., type and quantity of features, number and quality of samples), and the types and values of parameters used (Müller & Guido, 2017, p. 31). For some algorithms there are classification and regression versions, here only the classification versions are considered. The images that illustrate the algorithms are based on simple models with generic features and classes. It is easier to visualise the basic mechanism of the algorithms through these models using only two features and a few classes than using more complex models.

k-Nearest Neighbors

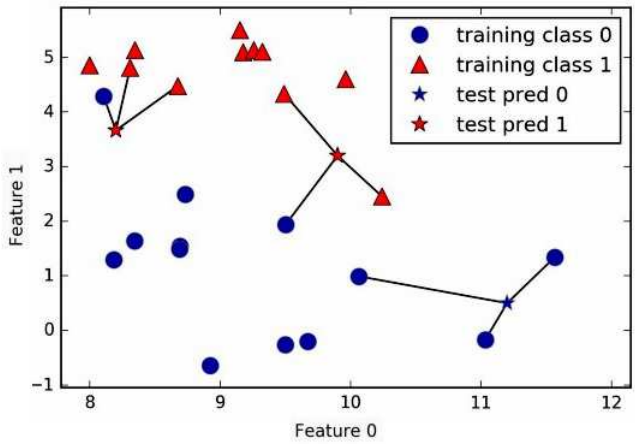


Figure 3.18 – k-Nearest Neighbors algorithm predictions for two classes and the parameter set to three nearest neighbours. After Müller & Guido (2017, Figure 2-5).

KNN is the simplest classification algorithm used in this research (Müller & Guido, 2017, p. 36). To make a prediction the algorithm finds the closest data points (the nearest neighbours) in the training set. The number of neighbours to be used (starting with one) can be defined as a parameter in the algorithm (Müller & Guido, 2017, p. 36-7). In the example shown in Figure 3.18 three neighbours are used to predict the class of new instances (the blue and red stars), in this case a voting system is used, counting how many classes 0 (circle) and 1 (triangle) the new instances can be associated, and the instance is assigned to the class with the greatest neighbours count (Müller & Guido, 2017, p. 36-7).

Logistic Regression

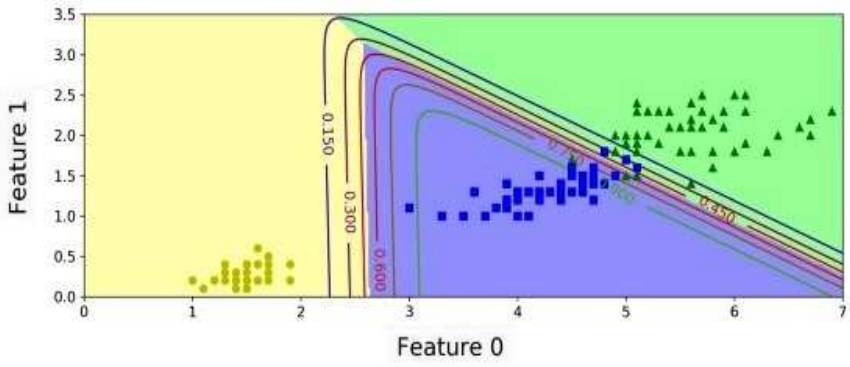


Figure 3.19 – Decision boundaries of Logistic Regression algorithm with three classes. The curved lines represent the probabilities of the instances belong to the blue square class. After Géron (2019, Figure 4-25).

Despite the 'regression' on its name, this is a classification and not a regression algorithm (Müller & Guido, 2017, p. 58). Logistic (or Logit) Regression estimates the probability that an instance belongs to a particular class. In its basic form, binary classification, this algorithm returns the logistic (an S-shaped function that produces a number between 0 and 1) of the result, based on a weighted sum of the features (Géron, 2019, p. 142-3). In the multiclass version, also known as Softmax or Multinomial Logistic Regression, this algorithm computes a score of every class and estimates the probabilities that an instance belongs to each class applying the softmax (normalised exponential) function (Géron, 2019, p. 147-8). The linear decisions boundaries are accompanied by probabilities of the instances belong to each class as shown by the curved lines in Figure 3.19 (Géron, 2019, p. 150-1).

Support Vector Machines – SVM

SVMs include a range of algorithms for linear and non-linear classification and regression problems. Scikit-learn implementations are Linear SVC and SVC algorithms for classification, and Linear SVR and SVR algorithms for regression (Scikit, 2021a). In this research the SVC algorithm, which supports both linear and non-linear (polynomial) variants, and allows the utilisation of several parameters to control the overfitting or underfitting of the model (Géron, 2019, p.

157-8), was selected. The basic concept of SVM is to create a boundary to keep the training instances from different classes as distant as possible using specific instances (the support vectors) located on the edges of the classes as reference (Figure 3.20-A) (Géron, 2019, p. 153-4).

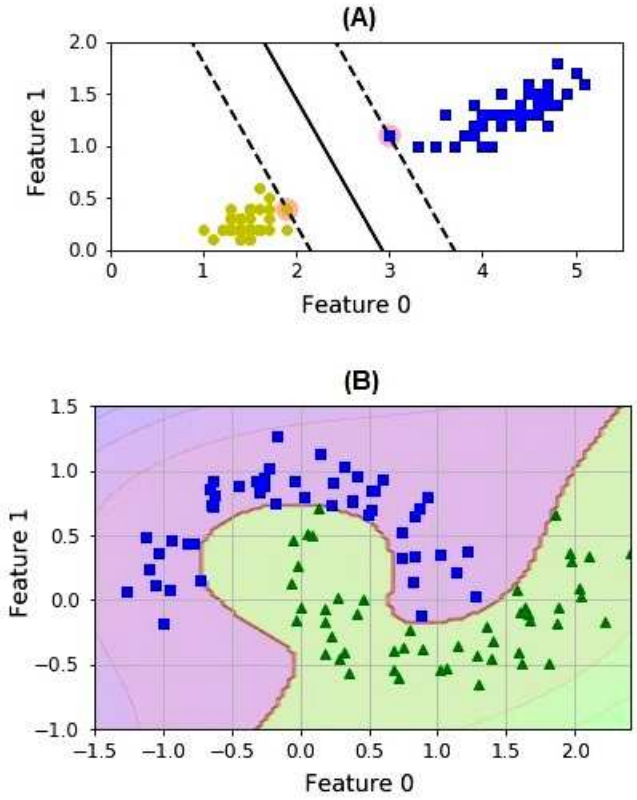


Figure 3.20 – (A) linear and (B) polynomial decision boundaries of SVM algorithms for two classes; (A) shows the support vectors (outlined circles and squares) for each class. After Géron (2019, Figure 5-1, Figure 5-7).

The non-linear classification is an alternative for more complex datasets that cannot be separated by linear functions, in this case a polynomial function is used (Figure 3.20-B) (Géron, 2019, p. 157-8). SVMs may need pre-processing (feature scaling) in case the features are of completely different orders of magnitude (Müller & Guido, 2017, p. 103-4), which is not the case of this research dataset since the numeric values are mostly in centimetres or defined as a relation between measurements.

Decision Tree Classifier

The Decision Tree Classifier algorithm searches for all possible tests based on the dataset features, and defines the test that results in the most significant information to define a boundary between classes. This process is repeated recursively until some termination criterion is reached, for instance until the tree reaches a certain depth (Müller & Guido, 2017, p. 73-7). If no explicit criterion is determined, the process continues until the leaves are pure (all the instances in a leaf share the same target classes), which can result in higher accuracy in the training set but causes a lower accuracy with unknown data (model overfitting) (Müller & Guido, 2017, p. 75-7).

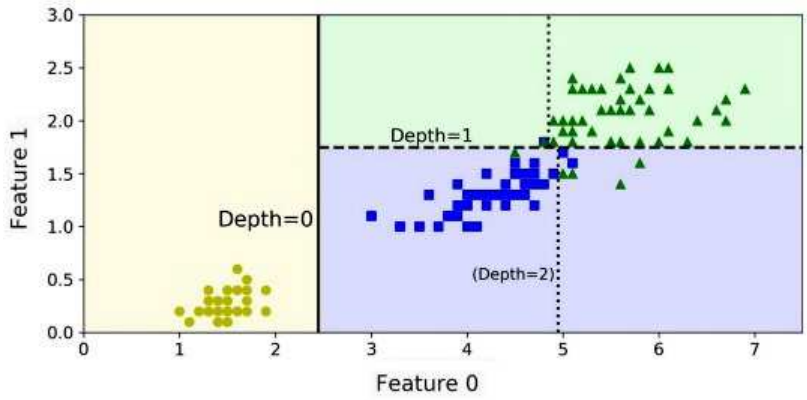


Figure 3.21 – Decision Tree algorithm boundaries according to different tree depths. After Géron (2019, Figure 6-2).

The vertical line in Figure 3.21 represents the decision boundary in the root node (depth = 0), which separates the yellow circle class from the other two classes; at least one more boundary (depth = 1), represented by the horizontal dotted line, is necessary to separate the green triangle and the blue square classes, and if the depth parameter is set to 2, an additional boundary is defined (Géron, 2019, p. 177). It is possible to notice that some instances are left outside the class boundaries and therefore associated with the wrong classes (misclassification) in this example.

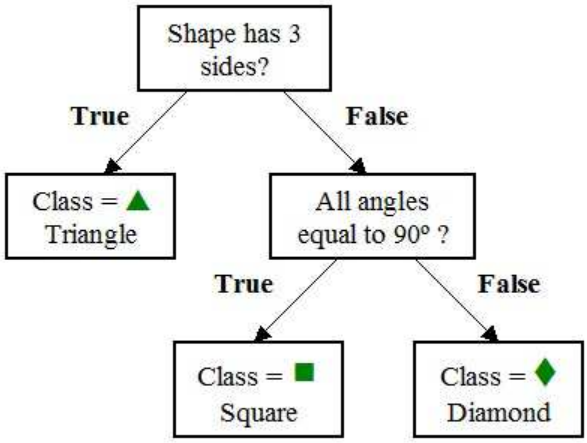


Figure 3.22 – Example of simple decision tree based on three geometric shapes.

Depending on the features used to characterise the target classes (the geometric shapes in the example of Figure 3.22) and the criteria used by the algorithm to identify them, the classification can be more or less accurate. The shape ◆ is a square according to the geometric definition (all angles equal to 90º), but because of its rotated position it could be misclassified as a diamond if a non-exact criterion was used.

The *Classification and Regression Tree* (CART) algorithm implemented in scikit-learn always produces binary trees, where non-leaf nodes have two children based on True/False questions using a single feature and a threshold value (Géron, 2019, p. 177-8). It is possible to observe some similarities between this classifier and the taxonomic structure presented in Section 2.1.6. Some of the main differences are that the algorithm implemented in the scikit-learn library uses only quantitative features (the qualitative features must be converted to a quantitative representation) and makes only binary tests (whether the feature is less or equal than, or greater than a certain value).

3.3.6 Ensemble methods

Ensemble methods aggregate the results from a group of predictors (classifiers or regressors), aiming to obtain an overall improved performance than provided by individual predictors (Géron, 2019, p. 189-90). Scikit-learn provides several implementations of ensemble methods, divided in two groups: *averaging methods*

and *boosting methods*. In averaging methods the ensemble applies the average of several independent predictors resulting in a final variance reduction, while in boosting methods the ensemble applies several predictors sequentially, reducing the bias based on the results provided by the preceding predictor (Scikit, 2021g). For this research two ensembles of the averaging method were selected: Random Forest Classifier and Voting Classifier.

Random Forest Classifier

The Random Forest Classifier is a special type of bootstrap aggregating method (shortened to *bagging*) that uses the Decision Tree Classifier as the base training algorithm. The bagging method uses the same algorithm for several predictions using different random subsets of the training set (Géron, 2019, p. 192-7). The ensemble aggregates the predictions from all training subsets and then associates the class with the most frequent prediction to a new instance, that way reducing the variance when compared to an algorithm trained only on the original training set (Géron, 2019, p. 192-3). An additional characteristic of the Random Forest is that it searches for the best feature among a random subset of features when splitting a node in the decision tree (Géron, 2019, p. 196).

In summary, while the Decision Tree algorithm generates one tree based on the training set, Random Forest returns the tree with the best performance among several alternative trees.

Voting Classifier

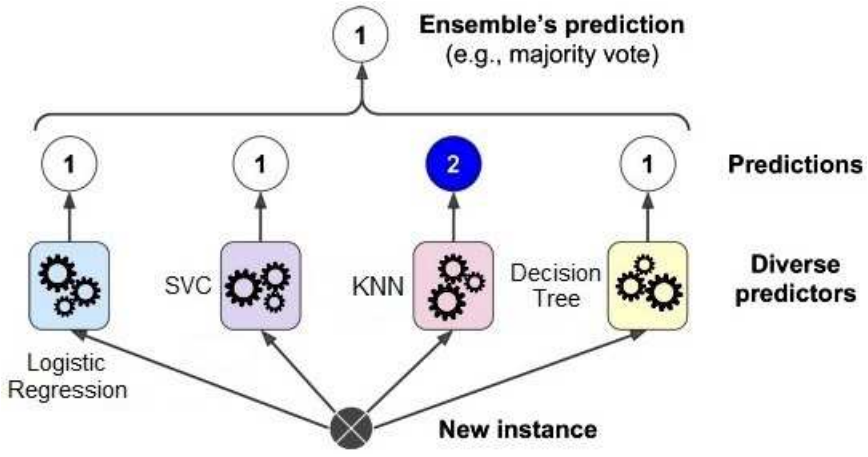


Figure 3.23 – Voting Classifier mechanism. After Géron (2019, Figure 7-2).

A Voting Classifier aggregates the predictions of diverse independent classifiers such as Logistic Regression, SVC and KNN, and associates an instance to the class that gets the majority of votes (Géron, 2019, p. 189).

In the example shown in Figure 3.23, class '1' is the ensemble's prediction because it received the votes from three algorithms, while class '2' received the vote from only one. This basic system is called *hard voting*, a variation of it that gives more weight to highly confident votes (highest class probability) is called *soft voting* (Géron, 2019, p. 192). Depending on the types of the classifiers involved (the more diverse algorithms the better) and the results provided by them, the Voting Classifier is capable to provide a higher performance than all the individual classifiers (Géron, 2019, p. 189-91).

3.3.7 Grid search and cross-validation

Grid search (parameters selection) and cross-validation are two techniques used to evaluate, fine-tune and improve the ML model (Géron, 2019, p. 73-77; Müller & Guido, 2017, p. 270-79), summarised in Figure 3.24. Trough the scikit-learn implementation GridSearchCV (Scikit, 2021h) it is possible to apply both techniques as a single method that behaves like a classifier (Müller & Guido, 2017, p. 272-4).

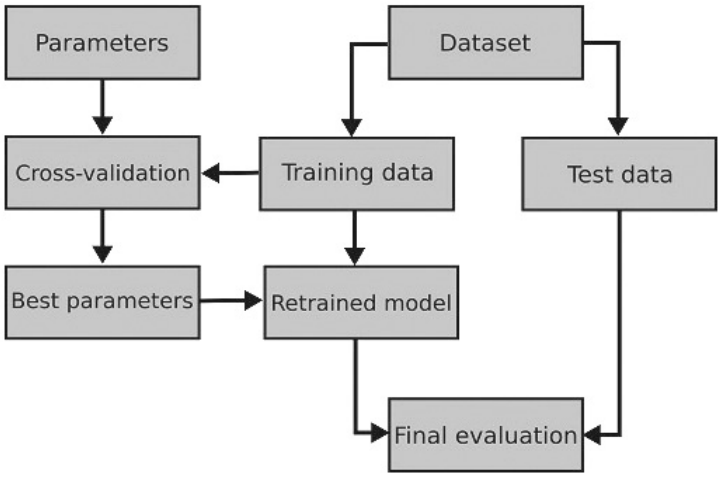


Figure 3.24 – Overview of grid search (parameters selection) and cross-validation workflow. After Müller and Guido (2017, Figure 5-7) and Scikit (2021i).

This method is applied in the third training session (Section 4.1.4), this is a way to verify that the best possible combinations of parameters were applied during the previous training session.

In c*ross-validation* the dataset is split in several different ways instead of a unique split into training and test sets, resulting in multiple models being trained (Müller & Guido, 2017, p. 258). In the *k-fold* cross-validation method, the dataset is partitioned into *k* parts (folds), where most folds are used as the training set and the remaining ones as the test set, and the model performance is evaluated in this configuration. In the next iterations (defined by the parameter *k*) the folds are changed, the model is trained again and, at the end of the process, an average performance is obtained (Müller & Guido, 2017, p. 258).

Grid search is a technique used to test several possible combinations of the parameters used in the algorithms in order to identify the optimal combination, the one that produces the higher model performance. For instance, two important parameters in SVC algorithm are *gamma* (kernel bandwidth) and *C* (regularisation); if each parameter receives six different values, there are 36 possible combinations (Müller & Guido, 2017, p. 267). The parameter requirements are specific for different algorithms, a few parameters are mandatory while most are optional, and some parameters have a broader range of values than others. The grid search technique facilitates the evaluation of a higher number of parameters alternatives (Müller & Guido, 2017, p. 267).

3.3.8 Feature importance

Decision Tree algorithms and some ensemble methods based on decision trees such as Random Forest have a useful resource that helps to summarise and interpret the tree mechanism, the *feature importance* property. This property rates how important or informative each feature is for the decision making process that builds the tree, using a proportional value between 0 (feature not used) and 1 (feature perfectly predicts the target), the values of each feature are added together resulting in 1 for the complete tree (Müller & Guido, 2017, p. 79). Features with a low rate are not necessarily uninformative, it can happen that more than one

feature codifies similar information and the algorithm uses only one of them (Müller & Guido, 2017, p. 79).

The feature importance is relevant also for feature selection, the utilisation of those features that contribute the most to increase the ML model performance without adding unnecessary complexity to it.

3.3.9 Confusion matrix

The *confusion matrix* is one of the most effective resources to evaluate the performance of a classifier algorithm: it counts the occurrences of misclassifications by the model, the times the instances of a certain class are classified as another class (Géron, 2019, p. 90). Each row represents an actual class, while each column represents a predicted class; the *true positives* and *true negatives* classifications are shown in the main diagonal (top left to bottom right), while *false positives* and *false negatives* are shown outside the diagonal (Géron, 2019, p. 90-1). These terms are described below:

- **TP** = True Positives: instances of class ‘C’ correctly classified as class ‘C’ (value in the diagonal in the class row)
- **FP** = False Positives: instances that do not belong to class ‘C’ incorrectly classified as class ‘C’ (values in the class column outside the diagonal)
- **FN** = False Negatives: instances of class ‘C’ incorrectly classified as a class different from ‘C’ (values in the class row outside the diagonal)
- **TN** = True Negatives: instances that do not belong to class ‘C’ correctly classified as a class different from ‘C’ (all other values in the diagonal)

Classified as →	■	♦	▲
■ Square	**15**	3	-
♦ Diamond	2	**20**	-
▲ Triangle	-	-	**10**

Table 3.8 – Example of confusion matrix based on three simple geometric shapes.

Table 3.8 shows an example of confusion matrix based on three simple geometric shapes with 50 instances in total. The class ‘Square’ has 15 TP (true positives) and three FN (false negatives), misclassified as ‘Diamond’, these are visualised in the ‘Square’ row; the FP (false positives) for class ‘Square’ are visualised in the ‘Square’ column: two ‘Diamond’ instances are misclassified as ‘Square’. The class ‘Square’ has also 30 TN (true negatives). The class ‘Triangle’ has all instances correctly classified: 10 TP (true positives).

3.3.10 Accuracy and other metrics

This section presents some of the most common ML metrics for classification models, based on the confusion matrix counts described in the previous section: **TP** = True Positives, **FP** = False Positives, **FN** = False Negatives, and **TN** = True Negatives.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Figure 3.25 – Common ML metrics for classification models. After Müller and Guido (2017, p. 289-90).

Accuracy is the number of correct classifications made by the model divided by the number of instances used in the model, which is equal to the sum of all entries in the confusion matrix (Müller & Guido, 2017, p. 288-9). This is therefore a metric used for the entire set, it is not used specifically for a class.

Precision measures how many instances associated to a class actually belong to that class (Müller & Guido, 2017, p. 289).

Recall, also known as sensitivity or true positive rate, measures how many instances of a class are actually associated to that class (Müller & Guido, 2017, p. 289).

The *F_1-Score* is the harmonic mean of precision and recall, therefore it shows a balanced result of the classification model since there is usually a trade-off between optimising precision and recall; deciding which metric is more important depends on the research objectives (Géron, 2019, p. 91-6; Müller & Guido, 2017, p. 289-90). For this research, both metrics are considered equally important, therefore the F_1-Score is used as the reference metric for the classes in the results chapter, in addition to accuracy as an overall metric. The F_1-Score can be also calculated for the entire set.

The following is an example of metrics calculation for class 'Square' based on the confusion matrix in Table 3.8:

TP = 15; TN = 30; FP = 2; FN = 3
Accuracy (model) = (15 + 30) ÷ (15 + 30 + 2 + 3) = 45 ÷ 50 = 0.90
Precision (class) = 15 ÷ (15 + 3) = 0.83
Recall (class) = 15 ÷ (15 + 2) = 0.88
F_1 (class) = 2 × (0.83 × 0.88) ÷ (0.83 + 0.88) = 0.85

Other metrics provided by scikit-learn and used in the summary of results (Section 4.1.1) are the *macro* average and *weighted* average. The macro average calculates the score giving equal weight to all classes, while the weighted average calculates the score based on the class support (the number of instances that belong to the class) (Müller & Guido, 2017, p. 304-5).

3.3.11 Training sessions procedure

The first two supervised learning sessions follow this procedure.

The third training session performs the same steps #1 to #8, in step #9 the difference is that the technique of grid search with cross-validation is applied for each base algorithm (ensembles excluded) and step 9.6 is not executed.

Procedure for Supervised Learning sessions

1. Start Anaconda navigator
2. Launch Jupyter Notebook
3. Import Python and scikit-learn libraries
4. Load the dataset (.csv file)
5. Check dataset structure and summaries
6. Define variables and parameters used through the session
7. Prepare the dataset:
    7.1 Separate dataset into target classes and features
    7.2 Split dataset into training (75%) and test (25%) datasets
8. Apply features transformations:
    8.1 Encoding of categorical features
    8.2 Imputing in continuous features with NA (missing) values
9. For each one of the ML algorithms:
    9.1 Create a model instance
    9.2 Fit the model with training data
    9.3 Predict the results with test data
    9.4 Print algorithm accuracy and metrics
    9.5 Print confusion matrix
    9.6 For Decision Tree Classifier execute these additional steps:
        i. Generate a decision tree in graphic format
        ii. Generate a decision tree in text format
        iii. Print feature importance values
        iv. Plot feature importance graphics
10. Record the results and end session

3.4 Unsupervised Learning Methods and Algorithms

This unsupervised learning part of this research is based on the clustering method, represented by two algorithms: k-Means and Hierarchical Clustering. For unsupervised learning there are no comparable metrics to those used in supervised learning that allow to assess how well the cluster correspond to an element of reference (the equivalent of the target classes in supervised learning), the best alternative is to analyse the clusters manually (Müller & Guido, 2017, p. 196), and in a complementary way to perform analyses on reduced parts of the dataset through the dendrogram.

3.4.1 Clustering algorithms

k-Means

This algorithm is based on the identification of cluster centers, which are located as a mean of the instances associated to the clusters. Two steps are performed iteratively until no modifications to the clusters are identified: i) the assignment of instances to the closest cluster center, and ii) the recalculating of the center (Müller & Guido, 2017, p. 170-1). The number of clusters to be used is defined as a parameter (*k*) of the algorithm, and the optimal results will depend on the dataset characteristics. Figure 3.26 shows the same instances associated to two and five clusters, in this example two clusters do not seem enough to identify all the potential groups and, while five clusters might perform better than two, an alternative of three clusters could also have been considered. k-Means may result in lower performances when the clusters have different densities or have non-spherical shapes (Müller & Guido, 2017, p. 175-83).

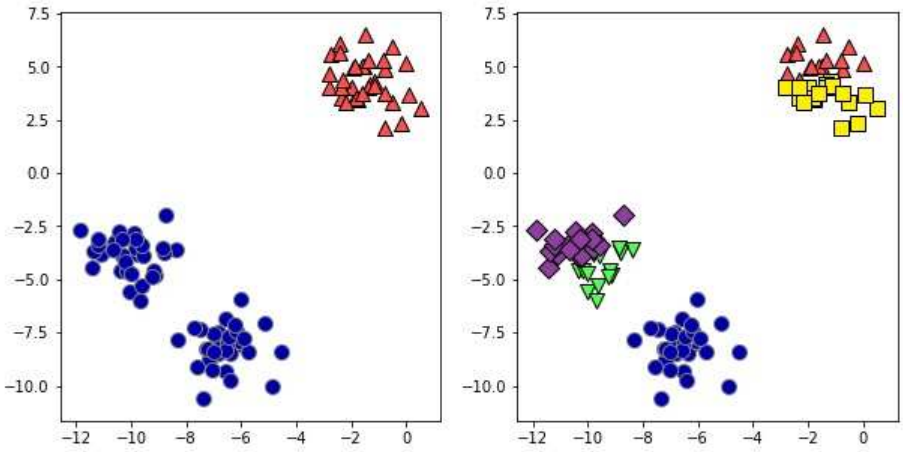


Figure 3.26 – k-Means cluster assignments based on two clusters (left) and five clusters (right). Müller and Guido (2017, Figure 3-26).

The main objectives of clustering and classification are equivalent: to associate similar instances in groups (either clusters or classes) according to their features where each instance receives a label, however in the case of clustering the labels do not have a pre-defined meaning, there is no ground truth which to compare the results (Müller & Guido, 2017, p. 173). In addition, in the k-Means algorithm the

analyst must define the number of clusters as a parameter and evaluate which one provides the most meaningful results.

Hierarchical Clustering

Hierarchical clustering is a type of agglomerative clustering, where one of the iterative steps, the assignment of instances to a cluster, is similar to the k-Means algorithm, but the next step consist in the merging of the two most similar clusters based on one of three methods, implemented by scikit-learn: ward, average or complete (Géron, 2019, p. 258; Müller & Guido, 2017, p. 183-88). The difference among the methods is how they measure cluster similarity: the *ward* method considers the least variance within all clusters, the *average* considers the smallest average distance between all clusters instances, and the *complete* or *maximum* method considers the smallest maximum distance between all clusters instances (Müller & Guido, 2017, p. 183-4). Figure 3.27 shows an example of a hierarchical clustering based on the agglomerative clustering method.

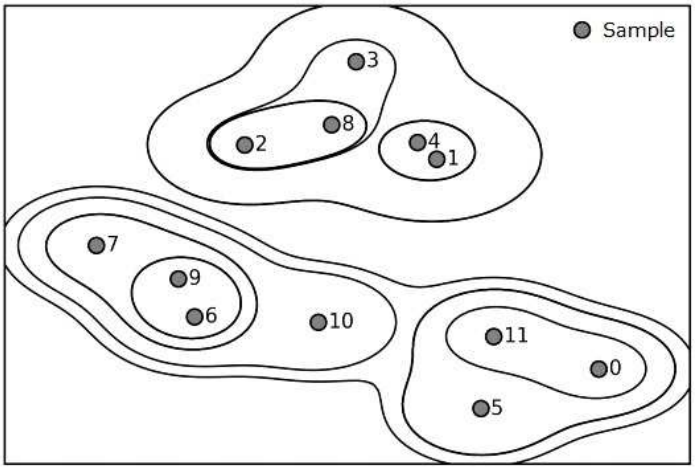


Figure 3.27 – Agglomerative clustering. After Müller and Guido (2017, Figure 3-35).

3.4.2 Dendrogram

Scikit-learn provides an algorithm for hierarchical clustering, the Agglomerative Clustering, but it does not provide a method for generating a dendrogram, which is the best tool to visualise the grouping of clusters (Müller & Guido, 2017, p. 185-88), the alternative used in this research is the dendrogram algorithm from the

SciPy library (Scipy, 2021b). Figure 3.28 shows the dendrogram version of the hierarchical clustering shown in Figure 3.27. A further difference between the k-Means algorithm and hierarchical clustering is that the latter does not require the information of the number of clusters as a parameter.

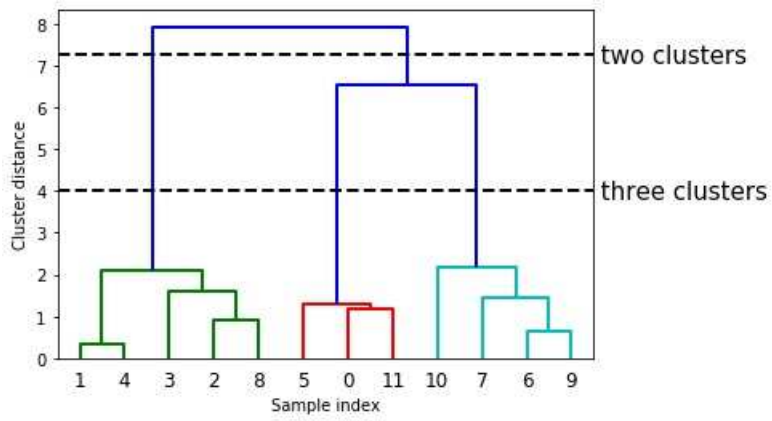


Figure 3.28 – Dendrogram of the agglomerative clustering. Müller and Guido (2017, Figure 3-36).

3.4.3 Silhouette score

When the number of pre-defined clusters is too small for one particular dataset, the clusters that would be better separated end up being merged and, on the other hand, when the number is too large some clusters may be inappropriately separated (Géron, 2019, p. 245). The *silhouette score* is a metric that provides a starting point for the analysis of clusters, through the identification of the range that may provide the optimal number of clusters. This method measures the mean distances among instances in the same cluster and the mean distances to the instances of the next closest cluster. The score ranges from –1 to 1, if the values are closer to 1 then the instances are stronger related to their own clusters and far from other clusters (Géron, 2019, p. 245-8).

4 RESULTS

The results of the training sessions are presented separately for supervised and unsupervised learning. An integrated summary based on the shape classes is presented next, and results from both approaches are compared in Chapter 5.

4.1 Supervised learning

The supervised learning process was divided in three sessions, each session was subdivided in two variations:

- **First session**: uses the complete set of features described in Chapter 3.1 and the six algorithms described in Section 3.3.5.
- **Second session**: the features miniature, min_diam (minimum diameter), max_diam (maximum diameter) and capacity were removed, using the same six algorithms.
- **Third session**: uses the same dataset of the second session with the grid search and cross-validation methods (Section 3.3.7).

The variation within each session is related to whether the distribution of samples during the application of the 'train_test_split' method is stratified or not stratified. Table 4.1 shows the difference between the two approaches.

Shape class	Full dataset	Test dataset (25%)	
		Stratified	Not stratified
C	10	3	1
E	182	46	47
G	96	24	22
H	20	5	7
K	22	5	4
N	34	8	8
P	88	22	21
R	32	8	10
T	12	3	4
TOTAL	496	124	124

Table 4.1. Differences between the stratified and not stratified distribution of samples in the test dataset.

The results presented in the training sessions are always from the stratified variation since this is the one that provided the best overall results.

4.1.1 Summary of results

The summaries of results for supervised learning are presented in three tables and one figure. Table 4.2 is a summary by shape classes of F_1-Scores of the five algorithms that achieved the highest overall performances (≥ 0.80) in both accuracy (Acc) and F_1-Score (F_1). Table 4.3 is a summary of feature importance averages from Decision Tree Classifier algorithm, and Table 4.4 is a summary of accuracy, precision, recall and F_1-Score metrics for all algorithms and training sessions.

The highest scores taking into account all sessions were provided by the ensemble Voting Classifier (Acc = 0.87, F_1 = 0.86), closely followed by Logistic Regression and SVC (Acc = 0.86, F_1 = 0.86). The algorithm with the lowest scores was k-Nearest Neighbors (Acc = 0.77, F_1 = 0.74). The Decision Tree Classifier (Acc = 0.81, F_1 = 0.81) and the ensemble Random Forest (Acc = 0.83, F_1 = 0.81) provided intermediate scores.

Shape class		F_1-Score					
		Logit	SVC	DT	RF	VC	Avg
C	Shallow bowl	0.50	0.50	0.50	0.00	0.50	0.40
E	Bowl	0.93	0.92	0.94	0.95	0.96	0.94
G	Cup/Mug or Beaker	0.92	0.90	0.84	0.92	0.92	0.90
H	Open pot	0.89	0.89	0.73	0.89	0.89	0.86
K	Jug/Tankard or Juglet	0.77	0.73	0.60	0.60	0.67	0.67
N	Closed pot (high)	0.67	0.67	0.50	0.55	0.67	0.61
P	Jar (wide neck)	0.77	0.84	0.86	0.73	0.81	0.80
R	Jar (restricted neck)	0.88	0.82	0.63	0.62	0.82	0.75
T	Flask or Bottle	0.67	0.67	0.33	0.80	0.67	0.63
	Weighted average	0.86	0.86	0.81	0.81	0.86	

Table 4.2 – Summary of F_1-Scores of five algorithms, second training session with stratified distribution, highlighting the best results for each shape class. Logit = Logistic Regression; SVC = Support Vector Machine for Classification; DT = Decision Tree Classifier; RF = Random Forest Classifier; VC = Voting Classifier; Avg = Average.

Feature	Description	Occur-rences	Average importance
H-Bd	Height / Belly diameter ratio	5	0.368
Bd-Nd	Belly diameter / Neck diameter ratio	5	0.218
base_diam	Base diameter	5	0.096
neck_diam	Neck diameter	4	0.064
Bd-Rd	Belly diameter / Rim diameter ratio	5	0.047
belly_height	Belly height	5	0.045
rim_profile[7]	Rim rounded-folded outside	4	0.041
belly_diam	Belly diameter	4	0.038
H-Nh	Height / Neck height ratio	5	0.033
height	Vessel height	4	0.022
base_type[2]	Rounded base	5	0.019
Bd-BaD	Belly diameter / Base diameter ratio	5	0.016
additional_elem[S]	Spout(s)	3	0.015
neck_height	Neck height	1	0.014
rim_orient[B]	Out-turned rim	1	0.012
H-Bh	Height / Belly height ratio	4	0.012
additional_elem[N]	No additional element	1	0.011
base_type[1]	Pointed base	2	0.009
opening_diam	Opening/rim diameter	3	0.009
rim_profile[3]	Rounded rim	4	0.008
rim_profile[1]	Thinned rim	2	0.006
base_type[4]	Flat base	1	0.006

Table 4.3 – Feature importance in Decision Tree Classifier algorithm, average value based on the second training session using five different combinations of parameters max_depth and min_samples_leaf: (5-1, 6-1, 6-2, 6-4, 7-1). *Occurrences* indicate in how many combinations the feature appeared.

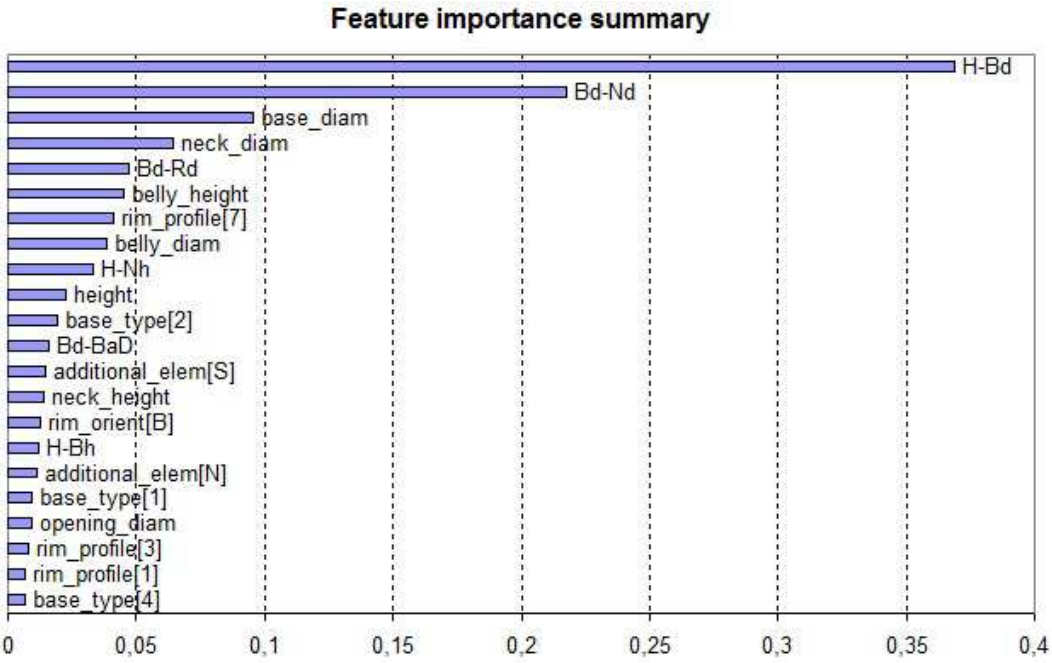


Figure 4.1 – Feature importance in Decision Tree Classifier algorithm, average value based on the second training session using five different combinations of parameters max_depth and min_samples_leaf: (5-1, 6-1, 6-2, 6-4, 7-1). Based on data from Table 4.3.

Algorithm		Training session	Accu-racy	Precision		Recall		F_1-Score	
				M	W	M	W	M	W
KNN		1.1	0.71	0.56	0.70	0.49	0.71	0.50	0.69
		1.2	0.75	0.61	0.75	0.53	0.75	0.53	0.73
		2.1	0.73	0.49	0.72	0.49	0.73	0.47	0.71
		2.2	**0.77**	**0.59**	**0.74**	**0.55**	**0.77**	**0.54**	**0.74**
		3.1	0.69	0.49	0.69	0.48	0.69	0.46	0.68
		3.2	0.75	0.59	0.73	0.55	0.75	0.55	0.73
Logit		1.1	0.80	0.70	0.80	0.64	0.80	0.66	0.79
		1.2	0.80	0.78	0.80	0.69	0.80	0.72	0.79
		2.1	0.84	0.75	0.85	0.67	0.84	0.69	0.83
		2.2	**0.86**	**0.86**	**0.88**	**0.76**	**0.86**	**0.78**	**0.86**
		3.1	0.83	0.76	0.85	0.69	0.83	0.71	0.83
		3.2	0.86	0.86	0.88	0.76	0.86	0.78	0.86
SVC		1.1	0.81	0.67	0.80	0.65	0.81	0.66	0.80
		1.2	0.83	0.73	0.83	0.75	0.83	0.73	0.83
		2.1	0.79	0.75	0.81	0.73	0.79	0.71	0.79
		2.2	**0.86**	**0.86**	**0.87**	**0.75**	**0.86**	**0.77**	**0.86**
		3.1	0.78	0.67	0.80	0.63	0.78	0.63	0.78
		3.2	0.86	0.86	0.87	0.75	0.86	0.77	0.86
DT		1.1	0.77	0.65	0.79	0.63	0.77	0.63	0.78
		1.2	0.80	0.69	0.81	0.60	0.80	0.61	0.79
		2.1	0.79	0.67	0.81	0.65	0.79	0.65	0.79
		2.2	**0.81**	**0.72**	**0.82**	**0.65**	**0.81**	**0.66**	**0.81**
		3.1	0.77	0.66	0.79	0.63	0.77	0.63	0.78
		3.2	0.78	0.65	0.80	0.64	0.78	0.63	0.78
Ensembles	RF	1.1	0.80	0.71	0.82	0.59	0.80	0.62	0.79
		1.2	**0.84**	**0.76**	**0.84**	**0.65**	**0.84**	**0.67**	**0.82**
		2.1	0.81	0.74	0.84	0.63	0.81	0.66	0.81
		2.2	0.83	0.76	0.83	0.64	0.83	0.67	0.81
	VC	1.1	0.82	0.71	0.84	0.67	0.82	0.68	0.82
		1.2	0.84	0.87	0.84	0.70	0.84	0.75	0.83
		2.1	0.83	0.72	0.84	0.69	0.83	0.69	0.83
		2.2	**0.87**	**0.85**	**0.88**	**0.75**	**0.87**	**0.77**	**0.86**

Table 4.4. Summary of training sessions' results.

Algorithms: KNN = k-Nearest Neighbors; Logit = Logistic Regression; SVC = Support Vector Machine for Classification; DT = Decision Tree Classifier; RF = Random Forest Classifier; VC = Voting Classifier. When more than one parameter was used, the result which achieved the greater accuracy is shown.
*Training session*s: the best results for each algorithm are highlighted.
Metrics: M = Macro average; W = Weighted average.
Support (quantity of samples in the test dataset) = 124 for all training sessions.

The following sections detail the results of the training sessions, which are shown through confusion matrices and other information (feature importance and decision tree for DT algorithm), for those algorithms that achieved the highest performances, in both Accuracy (Acc) and F_1-Score (F_1).

4.1.2 First training session

In the first training session the algorithms Voting Classifier (Acc = 0.84, F_1 = 0.83), Random Forest (Acc = 0.84, F_1 = 0.82) and SVC (Acc = 0.83, F_1 = 0.83) provided the highest scores. Table 4.5 shows the confusion matrix for the Random Forest algorithm and Figure 4.2 shows the feature importance for continuous features in the Decision Tree Classifier. The Random Forest performance was the only case an algorithm in the first session provided a higher score compared to the second or third sessions, the reasons for this behaviour are unclear.

Random Forest

	Classified as →	C	E	G	H	K	N	P	R	T	F_1
C	Shallow bowl	**-**	3	-	-	-	-	-	-	-	0.00
E	Bowl	-	**43**	3	-	-	-	-	-	-	0.93
G	Cup/Mug or Beaker	-	-	**24**	-	-	-	-	-	-	0.91
H	Open pot	-	-	1	**3**	-	1	-	-	-	0.75
K	Jug/Tankard or Juglet	-	-	1	-	**4**	-	-	-	-	0.89
N	Closed pot (high)	-	-	-	-	-	**4**	4	-	-	0.57
P	Jar (wide neck)	-	-	-	-	-	1	**21**	-	-	0.81
R	Jar (restricted neck)	-	-	-	-	-	-	4	**3**	1	0.55
T	Flask or Bottle	-	-	-	-	-	-	1	-	**2**	0.67

Table 4.5 – Confusion matrix resulting from the Random Forest algorithm, which provided the highest scores in the first training session with stratified distribution, together with Voting Classifier and SVC.

Correct classifications = 104
Total classifications = 124
General accuracy = 0.84
F_1-Score, weighted average = 0.82

The shape classes with the highest scores ($F_1 \geq 0.80$) are ‘E - Bowl’, ‘G – Cup/Mug’, ‘K – Jug/Tankard’ and ‘P – Jar (wide neck)’, in that order. The class with the lowest score is ‘C – Shallow bowl’, the three samples of this class were misclassified as ‘E – Bowl’, resulting in an unusual score of 0.00. The Decision Tree Classifier (Acc = 0.80, F_1 = 0.79) used nearly all the continuous features to build the decision tree (Figure 4.2), with the exception of min_diam (minimum diameter) and opening_diam (diameter at opening). H-Bd (total Height / Belly

diameter ratio) and Bd-Nd (Belly diameter / Neck diameter ratio) are the features with highest importance.

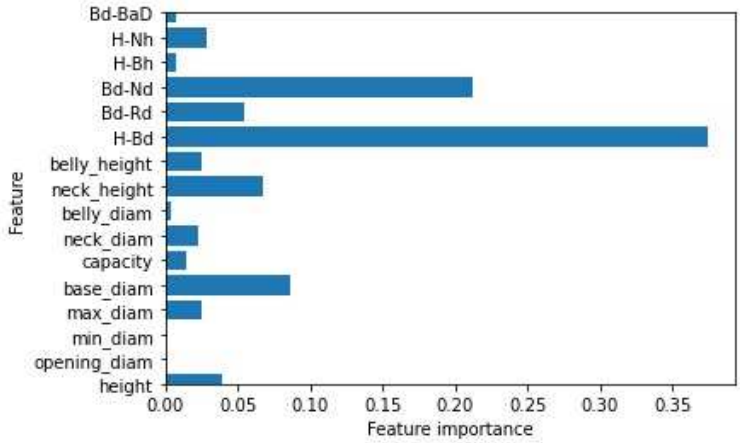


Figure 4.2 – Feature importance of continuous features in the Decision Tree Classifier, first training session, using parameters max_depth = 6 and min_samples_leaf = 1.

The next training session will be analysed in more detail since it provided the best results among all sessions.

4.1.3 Second training session

In the second training session the algorithms Voting Classifier (Acc = 0.87, F_1 = 0.86), Logistic Regression (Acc = 0.86, F_1 = 0.86) and SVC (Acc = 0.86, F_1 = 0.86) provided the best results. Tables 4.6 to 4.10 shows the confusion matrix for these algorithms and also for the Random Forest and Decision Tree Classifier.

One modification in the dataset was made for the second training section, the removing of four features: one categorical (miniature) and three continuous (min_diam, max_diam, capacity). The reason for this modification was to reduce the dataset complexity, allowing the ML algorithms to focus on the most relevant features (Géron, 2019, p. 26-7). These specific continuous features were chosen because there is some overlap among them and the measurements obtained through the Arcane images (neck_diam, belly_diam, neck_height, belly_height), these were preferred over the original Arcane measurements because of the absence of NA values and also because these measurements are used in the formulas of the relative measurements. The features removed have low values of

feature importance: miniature = 0, min_diam = 0, max_diam = 0.025, capacity = 0.014, in a scale from 0 to 1 (Figure 4.2), and two have high amount of NA values (min_diam = 253, max_diam = 131, in a total of 496 samples), which can potentially influence the results as commented in the discussion chapter.

These modifications resulted in increasing accuracies compared to the first training session for all algorithms (Table 4.4), with the exception of Random Forest with stratified distribution, which slightly decreased from 0.84 to 0.83.

Logistic Regression

	Classified as →	C	E	G	H	K	N	P	R	T	F_1
C	Shallow bowl	**1**	2	-	-	-	-	-	-	-	0.50
E	Bowl	-	**43**	3	-	-	-	-	-	-	0.93
G	Cup/Mug or Beaker	-	-	**24**	-	-	-	-	-	-	0.92
H	Open pot	-	-	1	**4**	-	-	-	-	-	0.89
K	Jug/Tankard or Juglet	-	-	-	-	**5**	-	-	-	-	0.77
N	Closed pot (high)	-	-	-	-	-	**4**	4	-	-	0.67
P	Jar (wide neck)	-	1	-	-	2	-	**17**	1	1	0.77
R	Jar (restricted neck)	-	-	-	-	1	-	-	**7**	-	0.88
T	Flask or Bottle	-	-	-	-	-	-	1	-	**2**	0.67

Table 4.6 – Confusion matrix resulting from the Logistic Regression algorithm, which provided the second highest scores in the second training session with stratified distribution, together with the SVC algorithm.

Correct classifications = 107
Total classifications = 124
General accuracy = 0.86
F_1-Score, weighted average = 0.86

The individual results from the Logistic Regression and SVC are very similar (Tables 4.6 and 4.7). The three shape classes with the highest scores ($F_1 \geq 0.80$) are ‘E - Bowl’, ‘G – Cup/Mug/Beaker’, ‘H – Open pot’, in that order. The class with the lowest score is ‘C – Shallow bowl’, two of the three samples of this class were misclassified as ‘E – Bowl’. The main difference between the two algorithms are the results from the ‘P – Jar (wide neck)’ and ‘R – Jar (restricted neck)’ classes. While both classes resulted in high ($F_1 \geq 0.80$) and similar scores in SVC, the result for the R class was considerably better than the P class in the case of Logistic Regression.

SVC

	Classified as →	C	E	G	H	K	N	P	R	T	F_1
C	Shallow bowl	**1**	2	-	-	-	-	-	-	-	0.50
E	Bowl	-	**43**	3	-	-	-	-	-	-	0.92
G	Cup/Mug or Beaker	-	1	**23**	-	-	-	-	-	-	0.90
H	Open pot	-	-	-	**4**	1	-	-	-	-	0.89
K	Jug/Tankard or Juglet	-	-	1	-	**4**	-	-	-	-	0.73
N	Closed pot (high)	-	-	-	-	-	**4**	4	-	-	0.67
P	Jar (wide neck)	-	1	-	-	-	-	**19**	1	1	0.84
R	Jar (restricted neck)	-	-	-	-	1	-	-	**7**	-	0.82
T	Flask or Bottle	-	-	-	-	-	-	-	1	**2**	0.67

Table 4.7 – Confusion matrix resulting from the SVC algorithm, which provided the second highest scores in the second training session with stratified distribution, together with the Logistic Regression algorithm.

Correct classifications = 107
Total classifications = 124
General accuracy = 0.86
F_1-Score, weighted average = 0.86

Voting Classifier

	Classified as →	C	E	G	H	K	N	P	R	T	F_1
C	Shallow bowl	**1**	2	-	-	-	-	-	-	-	0.50
E	Bowl	-	**45**	1	-	-	-	-	-	-	0.96
G	Cup/Mug or Beaker	-	-	**24**	-	-	-	-	-	-	0.92
H	Open pot	-	-	1	**4**	-	-	-	-	-	0.89
K	Jug/Tankard or Juglet	-	-	1	-	**4**	-	-	-	-	0.67
N	Closed pot (high)	-	-	-	-	1	**4**	3	-	-	0.67
P	Jar (wide neck)	-	1	1	-	1	-	**17**	1	1	0.81
R	Jar (restricted neck)	-	-	-	-	1	-	-	**7**	-	0.82
T	Flask or Bottle	-	-	-	-	-	-	-	1	**2**	0.67

Table 4.8 – Confusion matrix resulting from the Voting Classifier algorithm, which provided the highest scores in the second training session with stratified distribution.

Correct classifications = 108
Total classifications = 124
General accuracy = 0.87
F_1-Score, weighted average = 0.86

The Voting Classifier (VC) algorithm achieved the highest scores considering all the training sessions (Table 4.8), albeit with a minimum difference from Logistic Regression and SVC. These results are coherent with the algorithm mechanism explained in Section 3.3.6. From the nine shape classes, five resulted in high ($F_1 \geq 0.80$) scores, with three classes near or above 0.90. The independent classifiers used as parameters for VC were Logistic Regression, SVC, Decision Tree Classifier and KNN, using the 'hard' voting variation. Other combinations of classifiers and variations were tested, but this one returned the highest scores.

Random Forest

	Classified as →	C	E	G	H	K	N	P	R	T	F_1
C	Shallow bowl	**-**	3	-	-	-	-	-	-	-	0.00
E	Bowl	-	**44**	2	-	-	-	-	-	-	0.95
G	Cup/Mug or Beaker	-	-	**24**	-	-	-	-	-	-	0.92
H	Open pot	-	-	1	**4**	-	-	-	-	-	0.89
K	Jug/Tankard or Juglet	-	-	1	-	**3**	-	1	-	-	0.60
N	Closed pot (high)	-	-	-	-	-	**3**	5	-	-	0.55
P	Jar (wide neck)	-	-	-	-	2	-	**19**	1	-	0.73
R	Jar (restricted neck)	-	-	-	-	-	-	4	**4**	-	0.62
T	Flask or Bottle	-	-	-	-	-	-	1	-	**2**	0.80

Table 4.9 – Confusion matrix resulting from the Random Forest algorithm in the second training session with stratified distribution.

Correct classifications = 103
Total classifications = 124
General accuracy = 0.83
F_1-Score, weighted average = 0.81

The Random Forest Classifier achieved a middle-range performance (Table 4.9) if compared against other algorithms. It performed slightly better than the Decision Tree Classifier (Table 4.10), but worse than its own score in the first training session (Table 4.5). When compared to the DT, it can be noted that the RF concentrated the misclassifications in fewer shape classes, for instance the N class in RF was misclassified as the P class only, whereas in the DT the N class was misclassified as four other classes; similar results are visible in the P and R shape classes.

Decision Tree

	Classified as →	C	E	G	H	K	N	P	R	T	F_1
C	Shallow bowl	**1**	2	-	-	-	-	-	-	-	0.50
E	Bowl	-	**44**	2	-	-	-	-	-	-	0.94
G	Cup/Mug or Beaker	-	2	**21**	1	-	-	-	-	-	0.84
H	Open pot	-	-	1	**4**	-	-	-	-	-	0.73
K	Jug/Tankard or Juglet	-	-	1	-	**3**	-	-	1	-	0.60
N	Closed pot (high)	-	-	-	1	1	**3**	2	1	-	0.50
P	Jar (wide neck)	-	-	1	-	-	1	**18**	1	1	0.86
R	Jar (restricted neck)	-	-	-	-	1	-	-	**6**	1	0.63
T	Flask or Bottle	-	-	-	-	-	-	-	2	**1**	0.33

Table 4.10 – Confusion matrix resulting from the Decision Tree algorithm in the second training session with stratified distribution.

Correct classifications = 101
Total classifications = 124
General accuracy = 0.81
F_1-Score, weighted average = 0.81

Despite not being among those with the best overall results, the Decision Tree Classifier is one of the most important algorithms in this research because it provides a range of useful information based on the visualisation of the decision tree and the importance of the features used in the tests that generate the tree. This information can be used for the improvement of the dataset and the final ML model.

k-Nearest Neighbors

This algorithm achieved its highest performance in the second training session (Acc = 0.77, F_1 = 0.74). KNN was the only algorithm that did not achieved a minimum score of 0.80 in accuracy or F_1-Score, and for that reason its results are not analysed in detail. It was already commented about the KNN being known as a simpler ML algorithm but that does not mean it is not efficient and should be excluded from other studies, its usefulness will depend on the research problems, and its performance on the characteristics of the dataset (this is true for other ML algorithms as well). For instance, KNN was important as a parameter for the VC classifier, and its performance was superior to other two algorithms as described in a related research case (Chapter 5.3).

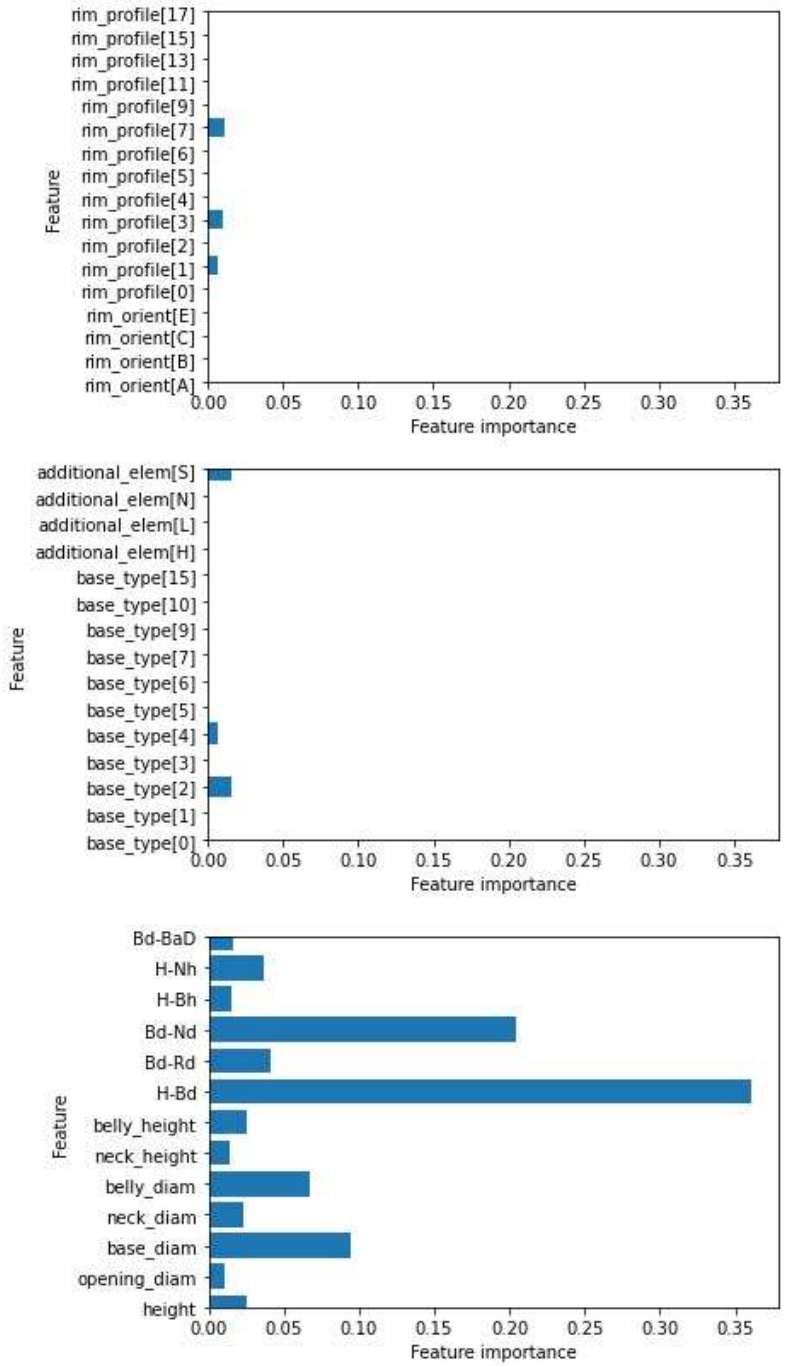


Figure 4.3 – Feature importance in the Decision Tree Classifier, second training session, using the parameters max_depth = 6 and min_samples_leaf = 1. Top and middle images: categorical features; bottom image: continuous features.

The feature importance is a very useful resource to understand how the features of the dataset contribute to the identification of the shape classes. It is clear in the graph in Figure 4.3 that the two most relevant features are H-Bd (total Height / Belly diameter ratio), with more than one third of the total score in the Decision Tree Classifier, and Bd-Nd (Belly diameter / Neck diameter ratio), with around one fifth of the total score. Together, these two features represent almost 60% of the total score for the classifier, however the utilisation of other features is also of importance as will be described in the analysis of the decision tree.

Parameters		**Results**	
max_ depth	min_ samples_ leaf	# of features	Accuracy
4	1	7	0.73
4	2	7	0.73
5	1	11	0.78
5	2	11	0.78
5	3	11	0.77
6	**1**	**19**	**0.81**
6	2	16	0.77
6	3	15	0.77
6	4	16	0.78
7	1	19	0.81
7	2	17	0.75
7	3	15	0.76
8	1	23	0.78
8	2	19	0.73

Table 4.11 – Summary of important features and accuracies for different combinations of parameters from Decision Tree Classifier, second training session. The max_depth and min_samples_leaf combination used in this research (6-1) is the one that returns the highest accuracy and at the same time it is less complex than other combinations with similar score (such as 7-1).

The results for the Decision Tree Classifier were based on a combination of different parameters (Table 4.11). This combination also reflects how the features are used, as can be seen in Figure 4.4 compared to Figure 4.3 (bottom). The algorithm variation with parameter max_depth = 5 uses less features than the variation with max_depth = 6, however the accuracy is lower because not enough tests were done to correctly associate the classes with as many samples as possible.

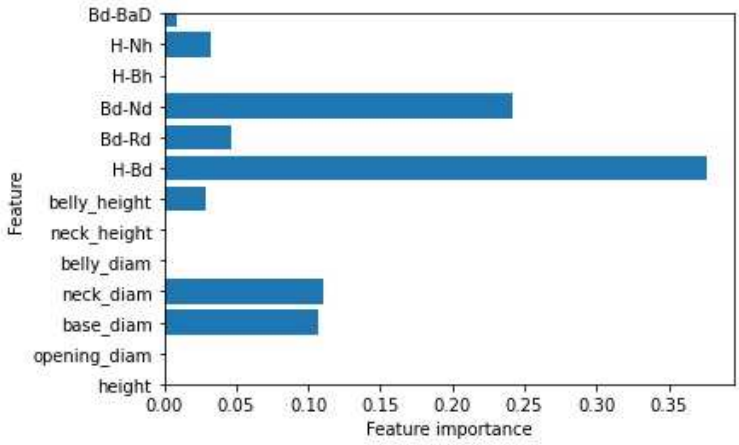


Figure 4.4 – Feature importance of continuous features in the Decision Tree Classifier, second training session, using parameters max_depth=5 and min_samples_leaf =1.

Since the combination of parameters max_depth = 6 and min_samples_leaf = 1 was the best result for the Decision Tree Classifier, this specific decision tree generated by the algorithm will be analysed in this section.

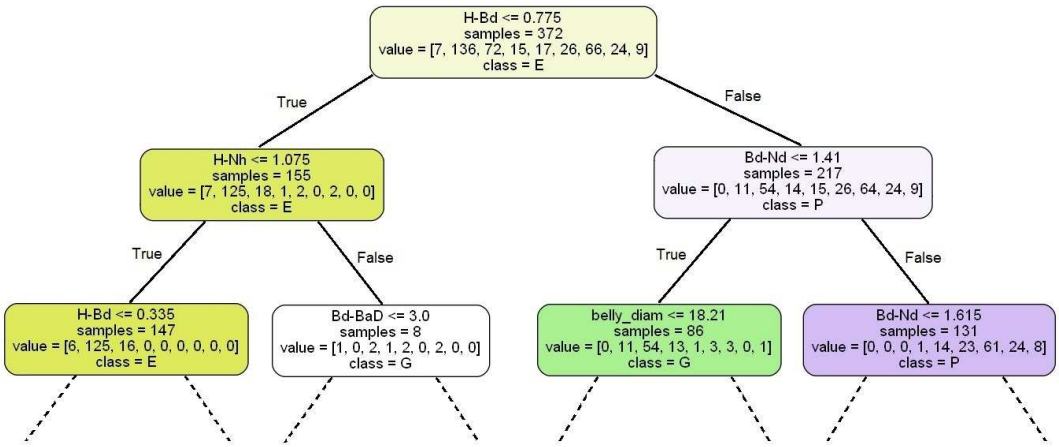


Figure 4.5 – Root node and first two levels of the tree generated by the Decision Tree Classifier algorithm in the second training session. The complete tree with six levels (max_depth = 6) is shown in text format in Figures 4.6 and 4.7.

```
|--- H-Bd <= 0.77
|   |--- H-Nh <= 1.07
|   |   |--- H-Bd <= 0.34
|   |   |   |--- base_diam <= 5.25
|   |   |   |   |--- class: C                          (5/5 = 100%)
|   |   |   |--- base_diam >  5.25
|   |   |   |   |--- class: E                          (2/2 = 100%)
|   |   |--- H-Bd >  0.34
|   |   |   |--- H-Bd <= 0.67
|   |   |   |   |--- Bd-BaD <= 12.07
|   |   |   |   |   |--- H-Bh <= 2.08
|   |   |   |   |   |   |--- class: E                  (104/108 = 96%)
|   |   |   |   |   |--- H-Bh >  2.08
|   |   |   |   |   |   |--- class: E                  (2/4 = 50%)
|   |   |   |   |--- Bd-BaD >  12.07
|   |   |   |   |   |--- class: G                      (1/1 = 100%)
|   |   |   |--- H-Bd >  0.67
|   |   |   |   |--- rim_profile[7] <= 0.50
|   |   |   |   |   |--- rim_profile[3] <= 0.50
|   |   |   |   |   |   |--- class: E                  (15/18 = 83%)
|   |   |   |   |   |--- rim_profile[3] >  0.50
|   |   |   |   |   |   |--- class: G                  (4/6 = 67%)
|   |   |   |   |--- rim_profile[7] >  0.50
|   |   |   |   |   |--- class: G                      (3/3 = 100%)
|   |--- H-Nh >  1.07
|   |   |--- Bd-BaD <= 3.00
|   |   |   |--- H-Nh <= 1.12
|   |   |   |   |--- class: G                          (2/2 = 100%)
|   |   |   |--- H-Nh >  1.12
|   |   |   |   |--- base_type[4] <= 0.50
|   |   |   |   |   |--- class: K                      (2/2 = 100%)
|   |   |   |   |--- base_type[4] >  0.50
|   |   |   |   |   |--- opening_diam <= 14.60
|   |   |   |   |   |   |--- class: C                  (1/1 = 100%)
|   |   |   |   |   |--- opening_diam >  14.60
|   |   |   |   |   |   |--- class: H                  (1/1 = 100%)
|   |   |--- Bd-BaD >  3.00
|   |   |   |--- class: P                              (2/2 = 100%)
```

Figure 4.6 – First half (H-Bd ≤ 0.77) of the decision tree in text format, second training session. The numbers in brackets show the positive identifications and the total amount of samples involved in the test. 100% means the leaf is pure (all samples belong to the same class). The lines in red show a limitation of the algorithm: the class is the same regardless the test result. The first two levels of the tree are represented graphically in Figure 4.5.

This tree achieves an accuracy of 0.90 based on the training dataset, and when applied to the test dataset the accuracy decreases to 0.81. This issue is related to the concept of generalisation: a ML model is built on training data and the goal is to maximise the accuracy in the test/validation data, which is previously unknown to the model, that way the model will have an optimal performance when dealing with new, unknown data. The ML model based on the Decision Tree Classifier can achieve an accuracy of 1.00 (100%) in the training dataset if the algorithm does not receive any pruning parameter such as max_depth or min_samples_leaf, however that way it becomes too complex and too specific, and behaves poorly when faced with unknown data.

```
|--- H-Bd >  0.77
|   |--- Bd-Nd <= 1.41
|   |   |--- belly_diam <= 18.21
|   |   |   |--- base_diam <= 5.85
|   |   |   |   |--- H-Bd <= 0.81
|   |   |   |   |   |--- H-Bd <= 0.80
|   |   |   |   |   |   |--- class: G                (6/8 = 75%)
|   |   |   |   |   |--- H-Bd >  0.80
|   |   |   |   |   |   |--- class: E                (3/3 = 100%)
|   |   |   |   |--- H-Bd >  0.81
|   |   |   |   |   |--- height <= 12.50
|   |   |   |   |   |   |--- class: G                (43/44 = 98%)
|   |   |   |   |   |--- height >  12.50
|   |   |   |   |   |   |--- class: G                (2/5 = 40%)
|   |   |   |--- base_diam >  5.85
|   |   |   |   |--- height <= 7.55
|   |   |   |   |   |--- class: E                    (6/6 = 100%)
|   |   |   |   |--- height >  7.55
|   |   |   |   |   |--- class: G                    (3/3 = 100%)
|   |   |--- belly_diam >  18.21
|   |   |   |--- neck_diam <= 16.50
|   |   |   |   |--- class: P                        (2/2 = 100%)
|   |   |   |--- neck_diam >  16.50
|   |   |   |   |--- opening_diam <= 15.90
|   |   |   |   |   |--- class: N                    (1/1 = 100%)
|   |   |   |   |--- opening_diam >  15.90
|   |   |   |   |   |--- rim_profile[1] <= 0.50
|   |   |   |   |   |   |--- class: H                (12/13 = 92%)
|   |   |   |   |   |--- rim_profile[1] >  0.50
|   |   |   |   |   |   |--- class: N                (1/1 = 100%)
|   |--- Bd-Nd >  1.41
|   |   |--- Bd-Nd <= 1.62
|   |   |   |--- base_diam <= 6.20
|   |   |   |   |--- base_type[2] <= 0.50
|   |   |   |   |   |--- H-Bh <= 2.06
|   |   |   |   |   |   |--- class: P                (3/3 = 100%)
|   |   |   |   |   |--- H-Bh >  2.06
|   |   |   |   |   |   |--- class: H                (1/2 = 50%)
|   |   |   |   |--- base_type[2] >  0.50
|   |   |   |   |   |--- class: N                    (5/5 = 100%)
|   |   |   |--- base_diam >  6.20
|   |   |   |   |--- class: N                        (11/11 = 100%)
|   |   |--- Bd-Nd >  1.62
|   |   |   |--- base_diam <= 0.25
|   |   |   |   |--- belly_height <= 3.82
|   |   |   |   |   |--- neck_diam <= 3.76
|   |   |   |   |   |   |--- class: T                (7/7 = 100%)
|   |   |   |   |   |--- neck_diam >  3.76
|   |   |   |   |   |   |--- class: K                (1/2 = 50%)
|   |   |   |   |--- belly_height >  3.82
|   |   |   |   |   |--- H-Bd <= 0.98
|   |   |   |   |   |   |--- class: K                (7/9 = 78%)
|   |   |   |   |   |--- H-Bd >  0.98
|   |   |   |   |   |   |--- class: R                (12/18 = 67%)
|   |   |   |--- base_diam >  0.25
|   |   |   |   |--- Bd-Rd <= 2.45
|   |   |   |   |   |--- additional_elem[S] <= 0.50
|   |   |   |   |   |   |--- class: P                (52/60 = 87%)
|   |   |   |   |   |--- additional_elem[S] >  0.50
|   |   |   |   |   |   |--- class: K                (2/3 = 67%)
|   |   |   |   |--- Bd-Rd >  2.45
|   |   |   |   |   |--- neck_height <= 19.37
|   |   |   |   |   |   |--- class: P                (2/2 = 100%)
|   |   |   |   |   |--- neck_height >  19.37
|   |   |   |   |   |   |--- class: R                (9/9 = 100%)
```

Figure 4.7 – Second half (H-Bd > 0.77) of the decision tree in text format, second training session. The numbers in brackets show the positive identifications and the total amount of samples involved in the test. 100% means the leaf is pure (all samples belong to the same class). The lines in red show a limitation in the algorithm: the class is the same regardless the test result. The first two levels of the tree are represented graphically in Figure 4.5.

A decision tree is generated based on the training dataset (in this case, n_samples = 372 in the first level, or 75% of the full dataset); the algorithm then uses this decision tree to predict the classes in the test dataset. The number of samples in each class (training dataset) is shown in the 'value' line (Figure 4.5). In this example they are [7, 136, 72, 15, 17, 26, 66, 24, 9], corresponding to the classes [C, E, G, H, K, N, P, R, T]. The class with the greatest number of samples on the node is shown as the reference class (they are E in the tree root, and E and P in the first level). The values for the complete tree are shown in Figures 4.6 and 4.7. The tree can be broadly divided in four parts, which are analysed in the next paragraphs. Only the analysis of the top levels and some specific tests in the lowest levels are described, enough to explain the logic behind the algorithm, the concepts of open/closed shapes and the relevance of the features.

The root level split – H-Bd feature

The algorithm uses the H-Bd feature (total Height / Belly diameter ratio) as the first test (H-Bd ≤ 0.77), and uses the result (true or false) to split the tree in two parts. All samples from the 'C – Shallow bowl' class and the majority of the samples (125 out of 136) of the 'E – Bowl' class were placed in the left branch. Nearly all (64 out of 66) of the 'P – Jar' and 'K – Jug/Juglet' (15 out of 17), and all samples from the N, R and T classes were placed in the right branch, completing the set of closed shapes group. Most samples of other classes that belong to the open shapes group ('G – Cup/Mug/Beaker' and 'H – Open pot') were also placed in the right branch, but these shapes are going to be dealt later in the decision tree.

The H-Bd (Fig. 3.16-A) is the most important feature to differentiate the two broad groups of classes, open and closed shapes. This ratio identifies the overall vessel shape, values much greater than 1.00 indicate higher or closed shapes such as jars, and values much lower than 1.00 indicate wider or open shapes such as the majority of bowls, but there are shapes that are closer to the 1.00 ratio such as beakers and open/closed pots. The algorithm then proceeds with the next tests to refine the association of samples to the shape classes.

First level split (right branch) – Bd-Nd feature

The test using the Bd-Nd feature (Belly diameter / Neck diameter ratio) (Figure 4.7) further differentiates the two broad groups of classes, open shapes and closed shapes. As the ratio gets closer to 1.00 the shapes become more of the open type (Figure 3.16-C), since some open shape samples such as bowls and beakers do not have bellies or necks (Figure 3.15) and both have the same measurements for these vessel parts, which are equal to the rim/opening diameter. In this case, the samples where the Bd-Nd ratio is greater than 1.41 were associated to the closed shapes, whereas the remaining ones (ratio ≤ 1.41), from the E, G and H classes, were associated to the open shapes group.

There still remained samples from the closed group in the 'open' branch of the tree. The belly_diam (belly diameter) feature, with a value of 18.21 cm, was used to identify most of them, separating samples of the E and G classes from samples of the H, N and P classes. After further divisions and utilisation of other features for testing, eight samples from the closed shapes group remained in the final level of the tree in this branch. As a final result, two high level 'open' branches can be identified, one with E and G samples and one with H samples (Figure 4.8), with residual closed shape samples. The residual samples are located in leaves that are not 100% pure, as shown in Figures 4.6 and 4.7.

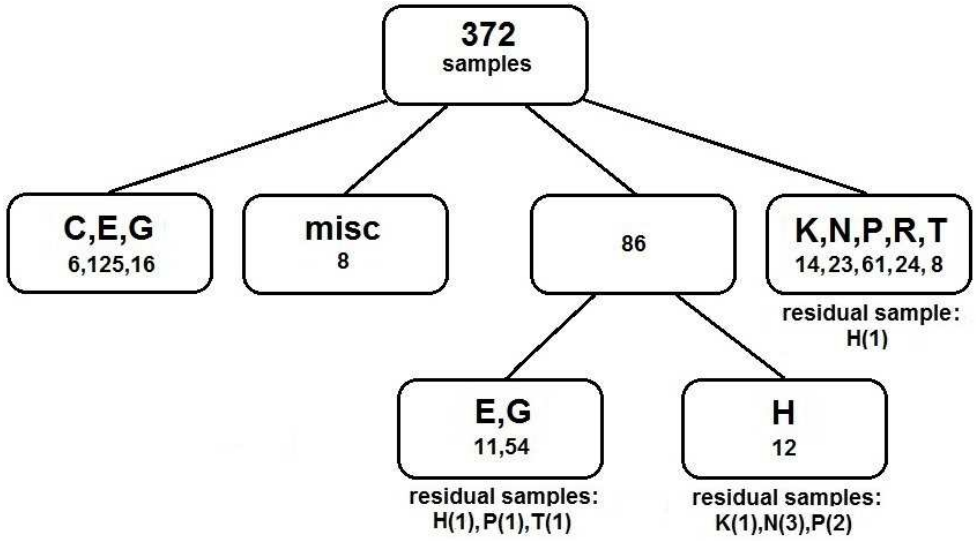


Figure 4.8 – Summary of the tree generated by the Decision Tree Classifier algorithm. The number of samples in the root (372) is from the training dataset (75% of the total), the numbers under each shape class represent the amount of samples of that class in each node.

In the ‘closed’ branch of the tree (Bd-Nd > 1.41), where all the closed shapes are present, only one sample of the open shapes, of the ‘H – Open pot’ class, remained.

First level split (left branch) – H-Nh feature

After the first (root) split, which placed all C class, most of E class (125/136), part of G class (18/72) and a few samples from other shapes in the left branch of the tree, a test using the H-Nh (total Height / Neck height) ratio was used to further differentiate the samples (Figure 4.6). This measurement indicates the relative location of the neck in the vessel vertical dimension (Section 3.1.8). Samples with H-Nh ≤ 1.075 included all E class in this branch, and nearly all C (6/7) and G (16/18) classes. In the right branch eight samples from the C, G, H, K and P classes remained (Figure 4.8, ‘miscellaneous’). It is possible to consider these samples as more atypical members of their classes. This specific test (H-Nh ≤ 1.075) was less decisive on separating the classes, since the majority of samples in this branch had already been identified in the previous test (H-Bd ≤ 0.775), this is suggested also by the value of H-Nh in the feature importance graphic (Figure 4.3), which is in the average only. Usually the features that are either used in the top levels of the tree or used in a number of different tests are considered of higher importance, but this is not exactly the case with H-Nh.

Other tests with continuous features

The H-Bd (total Height / Belly diameter ratio) feature was used in five other tests in different levels beside the root level split, using different values to differentiate between C and E, between E and G and between K and R classes.

Some continuous features beyond the ones used in the top levels of the tree were considered of higher importance by the Decision Tree classifier (Figure 4.3).

The Bd-Rd (Belly diameter / Rim diameter, Figure 3.16-B) identifies the shape of the upper portion of the vessel: it is convergent if the ratio is > 1, divergent if ratio < 1 or parallel if ratio = 1. This feature was used to separate samples of the K and P classes from samples of the R class, which has a more restricted neck and a smaller opening/rim diameter. In these cases they are all convergent, but the algorithm used the ratio = 2.445 as the reference to identify the shapes.

The features base_diam (base diameter) and belly_diam (belly diameter) had the third and fourth higher scores respectively (Figure 4.3). The test belly_diam ≤ 18.21 was used in the second level of the tree (right branch, Figure 4.7) to separate samples of the E and G classes from samples of the H, N and P classes. Three different tests with base_diam were used, the separation of samples of the N class from other closed shapes, the separation of samples of the P class from samples of the K, R and T classes (Figure 4.7), and the separation of C and E classes (Figure 4.6).

All the continuous features were used in at least one test by the Decision Tree classifier in this parameters configuration (max_depth = 6 and min_samples_leaf = 1). In some cases, specific tests in the lower levels of the tree were necessary to identify samples belonging to similar classes or samples that are atypical within the class, which can not be identified by the more broad tests in the highest levels of the tree. Some absolute measurements show a low score in the feature importance graph but it does not necessarily mean they are not relevant; they are used indirectly as a base for the relative measurements. This is the case of open_diam (opening/rim diameter), used in the Bd-Rd ratio, neck_diam (neck diameter), used in the Bd-Nd ratio, and height (total height), used in the H-Bd ratio.

Categorical features

The categorical features used in this research (rim orientation, rim profile, base typology, miniature vessel and additional elements) were used in a secondary way by the Decision Tree algorithm when compared to the continuous features. In this parameters configuration (max_depth = 6 and min_samples_leaf = 1) of the algorithm, three types of rim profile, two base types and one additional element were used in tests in the lowest levels of the tree (Figures 4.6 and 4.7). Despite their secondary utilisation, in some cases they may be the only alternative to identify a vessel shape class. This is the case of the test additional_elem[S] ≤ 0.5 which means that the test result is true if there is no additional element ‘spout’ in the sample, or the result is false when there is an additional element ‘spout’, according to the encoding method applied (Section 3.3.3). This test is used to separate samples of the ‘K – Jug/Juglet’ class, which have spouts, from samples of the ‘P – Jar (wide neck)’, which are similar in shape but don’t have these

elements. In the research dataset, only samples of the K and N classes have spouts.

The base_type feature was used twice by the algorithm: base_type[2] (rounded) and base_type[4] (flat), which are the two most common base types in the dataset, were used to separate samples of the N class from the P class, and to separate samples of the K class from samples of the C and H classes, respectively.

The rim_profile feature was used three times by the algorithm: rim_profile[1] (thinned), rim_profile[3] (rounded) and rim_profile[7] (rounded-folded outside), which are the three most common rim profile types in the dataset, were used to separate samples of the H class from the N class (rim_profile[1]), and to separate samples of the E class from the G class (rim_profile[3] and [7]).

Decision Tree summary

Figure 4.8 illustrates the summary of the decision tree analysis. From the four main branches of the tree, two represent the open shapes group, one represents the closed shapes group and one is composed by samples from both groups (miscellaneous). In the closed shapes group (K, N, P, R and T classes), only one open shape sample (H class) is present. Only open shapes (C, E and G) compose the leftmost of the two open shapes branches, while the other branch (E, G and H) include eight residual samples from the closed shapes group. Eight samples, four open and four closed shapes, compose the last branch (miscellaneous).

The open shapes branch E-G-H was divided in two parts: the E-G part included 65 samples of E and G classes and three samples of H, P and T classes, while the H part included 12 samples of H class and other six closed shapes samples, suggesting some similarity among the H class and the closed shapes. This issue will be discussed in Chapter 5.

4.1.4 Third training session

The main goal of the third and last training session was to test an alternative technique that might improve the accuracy results, the grid search with cross-validation. This technique was applied in all algorithms with the exception of the ensemble group.

Two algorithms returned the highest scores: Logistic Regression (Acc = 0.86, F_1 = 0.86) and SVC (Acc = 0.86, F_1 = 0.86). These results and also the confusion matrices are equal to those from the second training session (Tables 4.6 and 4.7). It was expected that the confusion matrices would show slightly different results when compared to the second training session because of the cross-validation mechanism of the GridSearchCV method, explained in Section 3.3.7.

Algorithm	Fixed parameters	Parameter values returned by GridSearchCV	Best parameter values (2nd training session)
KNN		n_neighbors = 3	n_neighbors = 4
Logit	solver = 'sag', max_iter = 1000	C = 0.01, penalty = 'none'	Same as GridSearchCV
SVC		C = 0.1, gamma = 'scale', kernel = 'linear'	Same as GridSearchCV
DT		max_depth = 6, min_samples_leaf = 4	max_depth = 6, min_samples_leaf = 1

Table 4.12 – List of parameters used in the GridSearchCV method, with the values returned by the method and the ones that provided the best performances in the second training session. *Algorithms*: KNN = K-Nearest Neighbors; Logit = Logistic Regression; SVC = Support Vector Machine for Classification; DT = Decision Tree Classifier.

Table 4.12 shows the parameter values returned by the method and the values that provided the best performances in the second training session. Since this method uses different combinations of the training and test datasets (the parameter was set to five different combinations) it was unlikely that the same combination of training/test datasets used in the second session would be repeated here, nevertheless the results suggest that. In addition, the possibility that some unidentified problem occurred in the application of the method by the author cannot be ruled out.

The grid search part of the GridSearchCV method was very useful despite this issue with the cross-validation. The parameters that were considered the best ones by the method were tested in the second training session script, following the method's cyclic workflow (Figure 3.24) and the Logistic Regression and SVC algorithms' performance increased. On the other hand the parameter values suggested by GridSearchCV provided a lower performance for k-Nearest Neighbors and Decision Tree Classifier when compared to the values used in the second training session (Table 4.4).

Under the column ‘Fixed parameters’, two parameters that were tested a number of times through the GridSearchCV method are shown, they always appeared among the best results, therefore they were fixed in the Logistic Regression parameter’s list and were not tested anymore under the GridSearchCV method. There are a number of other parameter options beside the ones presented in Table 4.12, and other combinations not shown here were tested. It is beyond the scope of this research to provide more detailed information on algorithms’ parameters, this is available in the scikit-learn documentation. The objective here is to draw attention to this important aspect of ML and indicate the resources available that help to arrive at the best possible solution.

A final observation about the kernel = ‘linear’ parameter in SVC algorithm. Other options beside the linear were tested, including the polynomial, but resulted in lower scores. This suggests that linear functions work best for the dataset used in this research, this is consistent with the results from Logistic Regression, an algorithm based on linear functions (Müller & Guido, 2017, p. 46-69).

4.2 Unsupervised learning

The results of the unsupervised training sessions are presented separately for k-Means and Hierarchical Clustering.

4.2.1 Clustering with k-Means

As commented in Chapter 3.4 the best alternative to interpret unsupervised learning results is the manual analysis of the clusters, which is done in tables 4.13 to 4.16 and through the dendrogram (Figure 4.10). The dendrogram is a very useful tool however the number of samples must be limited in order to allow a visual analysis.

The silhouette scores provided a starting point for determining the optimal number of clusters, but the number of clusters that are meaningful for the analysis limited its practical application. Figure 4.9 shows the scores for three different versions of the dataset. The first version is the full dataset with 496 samples, the second version is a reduced dataset with 10% of the samples obtained using the

scikit-learn ‘train_test_split’ method with stratified sampling, and the third version is a reduced dataset with five to six samples of each shape class, resulting in 50 samples. In the third version the samples were randomly selected; for the shape classes with greater number of samples (E, G, P, N and R), six samples were selected, and for the remaining shape classes, five samples were selected. The third version was used also for analysis in the dendrogram.

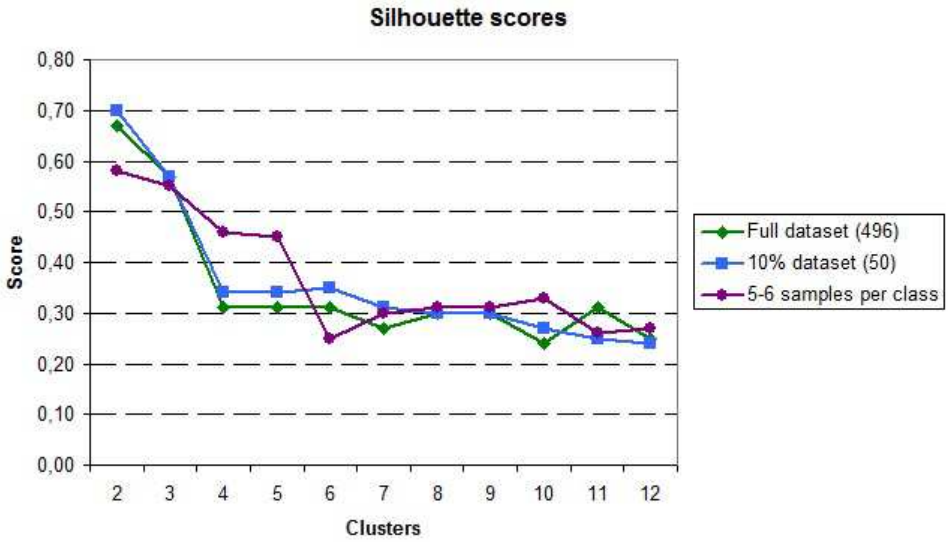


Figure 4.9 – Silhouette scores for three different versions of the research dataset, based on the k-Means algorithm: the full dataset with 496 samples; reduced dataset with 10% of the samples (proportional number of shape classes); reduced dataset with five to six samples of each shape class (50 samples in total).

Results in the silhouette scores look better when the number of clusters is low (two or three) because the samples are less dispersed across clusters, but these numbers of clusters are less useful for the analysis. Above three clusters the silhouette score vary according to the version of the dataset used, in general the full dataset and the 10% dataset versions perform in a more similar way. Despite the silhouette scores, the more useful range to analyse the research dataset in search of meaningful clusters is between four and six clusters, above that number the samples become too dispersed across the clusters.

The results based on four and six clusters are presented here, and the other clusters created for this research (2, 3, 5 and 8 clusters) can be consulted in Appendix A.

	Clusters				
Shape	**0**	**1**	**2**	**3**	**Total**
C	3		**7**		10
E	56	1	**125**		182
G	**75**		21		96
H	1	**13**	2	4	20
K	**15**	2	5		22
N	10	**16**	1	7	34
P	**56**	22	1	9	88
R	8	**19**		5	32
T	**12**				12
Total	236	73	162	25	496

Table 4.13 – Summary of samples divided into four clusters (0-3) and the corresponding shape classes, based on the k-Means algorithm. Cells in grey indicate the prevailing cluster for each shape class, cells in light blue indicate significant amounts of samples associated to secondary clusters.

	Clusters						
Shape	**0**	**1**	**2**	**3**	**4**	**5**	**Total**
C	**7**				3		10
E	**124**	1			56	1	182
G	22				**72**	2	96
H	1	**9**	4			6	20
K	1				**14**	7	22
N	1	10		5	8	10	34
P		8		9	**42**	29	88
R		**12**		5	7	8	32
T					**11**	1	12
Total	156	40	4	19	213	64	496

Table 4.14 – Summary of samples divided into six clusters (0-5) and the corresponding shape classes, based on the k-Means algorithm. Cells in grey indicate the prevailing cluster for each shape class, cells in light blue indicate significant amounts of samples associated to secondary clusters; shape N has two prevailing clusters.

(A)

	Clusters						
Shape	**0**	**1**	**2**	**3**	**4**	**5**	**Total**
C		**1**					1
E	1	**16**			1		18
G	2	**8**					10
H			1		1		2
K	1	1					2
N	1			**3**			4
P	**6**	2				1	9
R	1			**2**			3
T	**1**						1
Total	13	28	1	5	2	1	50

(B)

	Clusters						
Shape	**0**	**1**	**2**	**3**	**4**	**5**	**Total**
C					**5**		5
E				1	**5**		6
G	**4**				2		6
H	1	2	2				5
K	**4**				1		5
N	1	**3**		2			6
P	**4**			2			6
R	1	2		2		1	6
T	**5**						5
Total	20	7	2	7	13	1	50

Table 4.15 – Summary of samples divided into six clusters (0-5) and the corresponding shape classes, based on the k-Means algorithm. Table (A): reduced dataset with 10% of the samples; Table (B): reduced dataset with 5 to 6 samples of each shape class. Cells in grey indicate the prevailing cluster for each shape class, cells in light blue indicate significant amounts of samples associated to secondary clusters or no prevailing cluster. In the more balanced division of (B) is easier to visualise a general trend equivalent to the full dataset (Tables 4.13 and 4.14).

Both Tables 4.13 and 4.14 show a similar distribution of the shape classes across the clusters. The arrangement of clusters groups the majority of samples of shapes ‘C – Shallow bowl’ and ‘E – Bowl’ into one cluster, shapes ‘H – Open pot’ and ‘R – Jar (restricted neck)’ into a second cluster and ‘G – Cup/Mug/Beaker’, ‘K – Jug/Juglet’, ‘P – Jar (wide neck)’ and ‘T – Flask/Bottle’ into a third cluster. Shape ‘N – Closed pot (high)’ is more clearly associated to the same cluster as H and R in the four clusters version (Table 4.13), but an association among these shapes can also be seen in the six clusters version (Table 4.14). In the four clusters version, the fourth cluster (#3) is formed mainly by residual samples of the shape classes. In the six clusters version, four clusters concentrate the majority of samples and two clusters (#2 and #3) are formed by residual samples, but even in this version the division in three main clusters is visible.

Distribution across 6 clusters (496 samples)						
Shape	**Main**	**2nd**	**3rd**	**4th**	**Resi-dual**	**Total**
C	0.70	0.30				1.00
E	0.68	0.31			0.01	1.00
G	0.75	0.23			0.02	1.00
H	0.45	0.30	0.20		0.05	1.00
K	0.64	0.32			0.05	1.00
N	0.29	0.29	0.24	0.15	0.03	1.00
P	0.48	0.33	0.10	0.09		1.00
R	0.38	0.25	0.22	0.16		1.00
T	0.92	0.08				1.00

Table 4.16 – Distribution of shape classes across the clusters based on table 4.14. Shape T is the most uniform, with 92% of samples in the same cluster, followed by G and C. Shapes C, E, G and K are divided into two clusters, one of them being the prevailing one. The shapes H, N, P and R are distributed across three or four clusters.

Table 4.16 shows the distribution of the shape classes across the six clusters version (Table 4.14). Apart from shape ‘T – Flask/Bottle’ which has 92% of its samples associated to one cluster, the other shape classes show a distribution of samples in at least two and, in some cases, three or four clusters. Shapes C, E, G and K are divided basically into two clusters (with few residual samples), one of them being the prevailing cluster. Shapes H, N, P and R are distributed across three or four clusters.

The second version of the dataset (10% of the samples, Table 4.15-A), shows fewer meaningful associations most likely because the small amount of samples for some shape classes; four shapes have only one or two samples and this limits the utility of the results. The most visible associations are the grouping of classes P and T (cluster #0), classes C, E and G (cluster #1) and classes N and R (cluster #3). One relevant aspect here is the association among shapes of the open group (cluster #1) and among shapes of the closed group (clusters #0 and #3). The shapes that are not clearly associated to one of these groups, H and K, are the shapes that share most characteristics of both groups (open and close).

This reduced version was intended to be used in the dendrogram analysis, however because of the issue of sample distribution, a third version of the dataset, with five to six samples for each shape class, was created (Table 4.15-B). This version shows a similar pattern in the grouping of shape classes to the one presented by the full dataset with six clusters. As a matter of clarification there is no direct relation in the cluster numberings (#0 to #3, or #0 to #5) between the different versions of the dataset, or different cluster numbers within the same dataset; for instance, cluster #0 in Table 4.13 is the equivalent of cluster #4 in Table 4.14, and cluster #0 in Table 4.14 is the equivalent of cluster #4 in Table 4.15-B.

In Table 4.15-B the shapes G, K, P and T form a cluster like in Table 4.13, and shapes C and E form a second clear cluster. Two other clusters are less clearly formed: cluster #1 includes shapes H, N and R, and cluster #3 includes shapes N, P, R and a residual sample of E shape. The relationships among these shapes will be analysed in detail through the dendrogram.

4.2.2 Hierarchical Clustering

The Hierarchical Clustering algorithm using the Ward linkage method, based on the dataset version with 5 to 6 samples of each class, generated the dendrogram shown in Figure 4.10. In this algorithm the number of clusters is not informed by the analyst, the level of detail is higher and it is possible to identify sub-clusters in addition to the main clusters provided by the k-Means algorithm.

An important observation is necessary at this point: despite the fact that shape classes have been used as a way to assess the results of clustering algorithms, both k-Means and Hierarchical Clustering results (the clusters) are independent of shape classes, the results are based only on categorical and continuous features present in the research dataset, and the proposed clusters in fact indicate different criteria to shape classification which might be or not related to the original shape classes associated to the samples. It is necessary therefore to consider the possibility that other patterns of grouping exist beyond the ones provided by the experts and supervised learning algorithms.

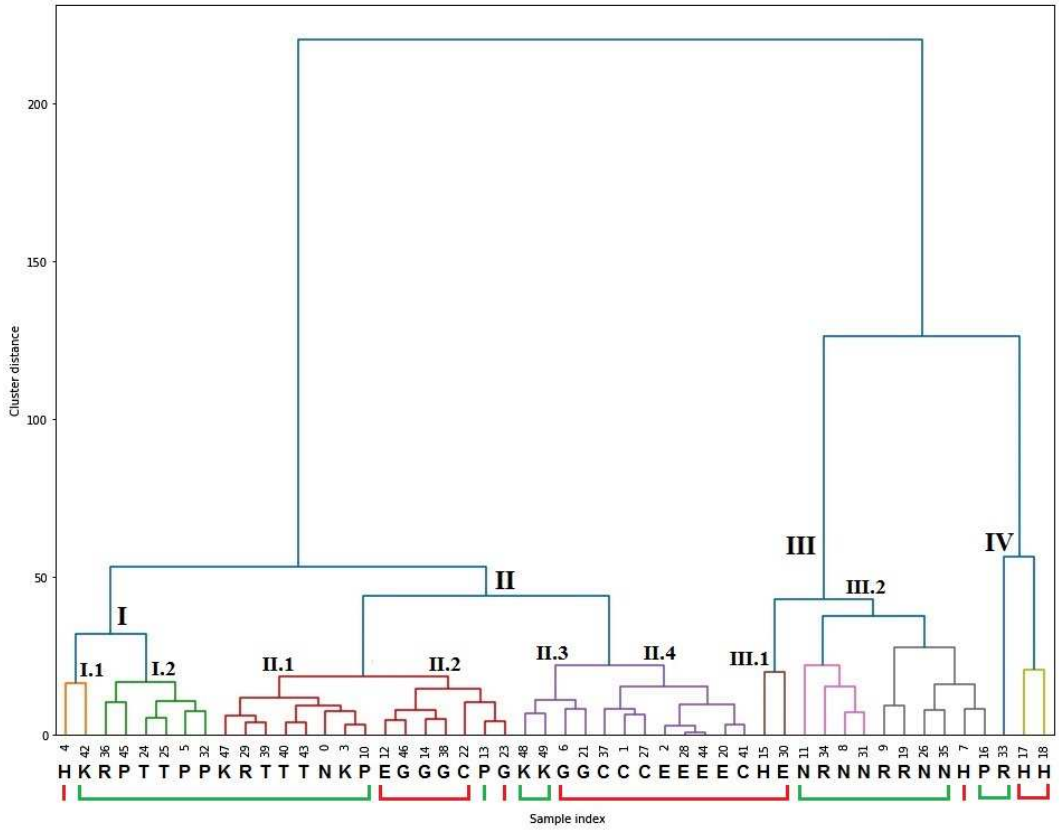


Figure 4.10 – Dendrogram from the Hierarchical Clustering algorithm based on five to six samples of each shape class. Roman numbers identify the main clusters and sub-clusters. The numbers below the bars identify the individual samples (Sample index), each sample is associated to one shape class represented by letters. The red and green bars below the letters indicate whether the class belongs to the open (red) or closed (green) shape groups.

In Figure 4.10 all the 50 samples used in the analysis are shown in the bottom (lowest level) of the graph. It is possible to identify four main clusters (indicated in Roman numerals) and corresponding sub-clusters; Figure 4.11 shows the top-level clusters only, here it is visible the unbalanced distribution of samples across the main clusters: cluster II includes more than half of the samples, and cluster IV includes only 3 samples.

The clustering patterns become clearer when the sub-clusters are analysed, but even in this level some unidentifiable patterns emerge when the hierarchical clustering is taken into consideration. The cluster distance, measured in the vertical axis in the graph, indicates how close the relation among clusters is. According to this information, clusters II.1 and II.2 are more closely related than clusters III.1 and III.2 for instance. However, when applying the criteria of open vs. closed shapes (shown in red and green lines below the shape classes in Figure 4.10), clusters II.1 and II.2 do not seem to have the closer relation between them, since cluster II.1 is formed by closed shapes and II.2 by open shapes, except for one sample (#13). On the other hand, clusters I.2 and II.1 seem to be more closely related because they contain only closed shapes. Another example is cluster II.4, which can be related to its neighbour clusters II.3 and III.1 by the presence of open shapes. In the case of cluster II.3, the presence of two samples of the 'K – Jug/Juglet' shape, which is the closed shape closest to the open shape group, reinforces this observation. Figures 4.12 to 4.15 illustrate the samples that belong to each cluster.

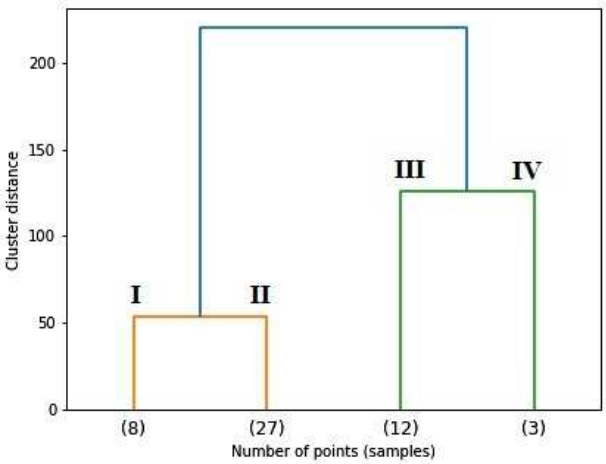


Figure 4.11 – Same dendrogram from Figure 4.10 showing the four top-level clusters only.

There are similarities and differences in the results from the Hierarchical Clustering and the k-Means algorithms. Comparing the data from Tables 4.13 to 4.15 with the dendrogram in Figure 4.10, it is evident the grouping of shape classes C and E, and P and T. Depending on the version of the dataset used in the k-Means algorithm (represented by Tables 4.13 to 4.15), the G class is joined either to the C-E group or to the P-T group. In the dendrogram, samples of the G class are all associated to the open shapes group (clusters II.2, II.3 – partial, and II.4). Samples of the H class are clearly associated to samples of N and R classes in Table 4.13 and also in the dendrogram (clusters III and IV), but samples of H class do not follow a clear pattern in other versions of k-Means algorithm (Tables 4.14 and 4.15). Samples of the K class are associated to the P-T group in two of the three versions of the k-Means algorithm (Tables 4.14 and 4.15-B) but are divided into three clusters in the dendrogram (I, II.1 and II.3), which includes both open and closed shapes. Shapes of the N and R classes are associated in the dendrogram (cluters III and IV) but are clearly associated in k-Means clusters only in Table 4.15-A.

The criteria of open vs. closed shapes as defined by the Arcane project is being applied here for cluster analysis but, as it was commented previously in this chapter, this is not the only criteria and it is possible that another unidentified patterns of similarity among clusters exists. The next set of figures (Figure 4.12 to Figure 4.15) illustrates the samples included in the dendrogram (Figure 4.10). The figures are divided basically according to the four high level clusters (I to IV), with cluster II divided in two figures and clusters III and IV included in the same figure. Through these associations of shapes and the samples illustrations it is possible to identify some patterns that are not clear in tables or graphs.

Cluster I

The main characteristic of the first association of shape classes (Figure 4.12), which includes clusters I.1 and I.2, is the predominance of closed shapes, except for one sample of the ‘H – Open pot’ class. Two samples from the ‘T – Flask/Bottle’ class, which are clearly different from those included in cluster II.1 (Figure 4.13) are included here, as well half of the samples of the ‘P – Jar (wide neck)’ class. The other samples of the P class are divided in different clusters.

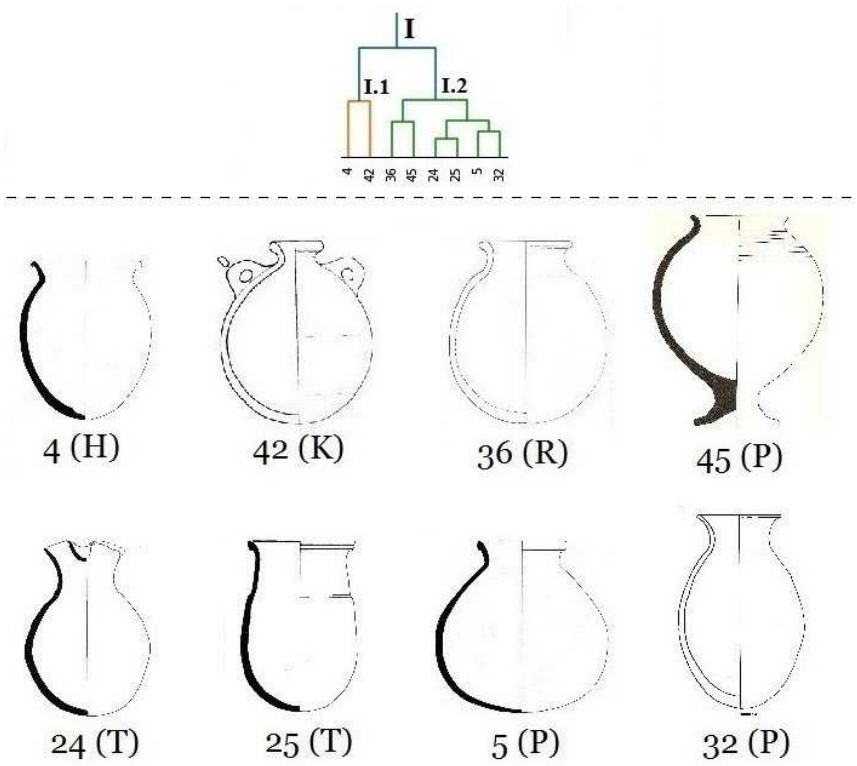


Figure 4.12 – Samples belonging to cluster I.1 and I.2 (complete dendrogram is shown in Figure 4.10). The number indicates the sample random number (from 0 to 49) and the letter in brackets indicate the sample shape class. First row: JZ001_P010, JZ002_P622, JZ002_P204 and JZ004_P037; second row: JZ001_P916, JZ001_P930, JZ001_P011 and JZ002_P081. Images at different scales. After Arcane (2016).

Clusters II.1 and II.2

The second association of shape classes (Figure 4.13) includes cluster II.1 and II.2 and it is characterised by the differences between them. In cluster II.1 all samples belong to the closed shapes group, with a predominance of samples of the 'T – Flask/Bottle' and 'K – Jug/Juglet' classes. In cluster II.2 there is a predominance of open shapes, especially the majority of the samples of the 'G – Cup/Mug/Beaker' class. One sample from the 'P – Jar (wide neck)' class is included here (#13). This sample is an uncommon representative of this class and, despite being a closed shape, it has similarities with some samples of the G class.

It seems clear that the samples from cluster II.2 are visually closer to the samples from clusters II.3 and II.4 (Figure 4.14) than to the samples in cluster II.1, whereas the Hierarchical Clustering algorithm joined the clusters II.1 and II.2 first, and then joined them to clusters II.3 and II.4 later (Figure 4.10). It is likely that the algorithm recognised an unidentified pattern between the two clusters,

without using the open/closed shapes criteria, but the hypothesis that the agglomerative clustering technique was inefficient in this case cannot be ruled out.

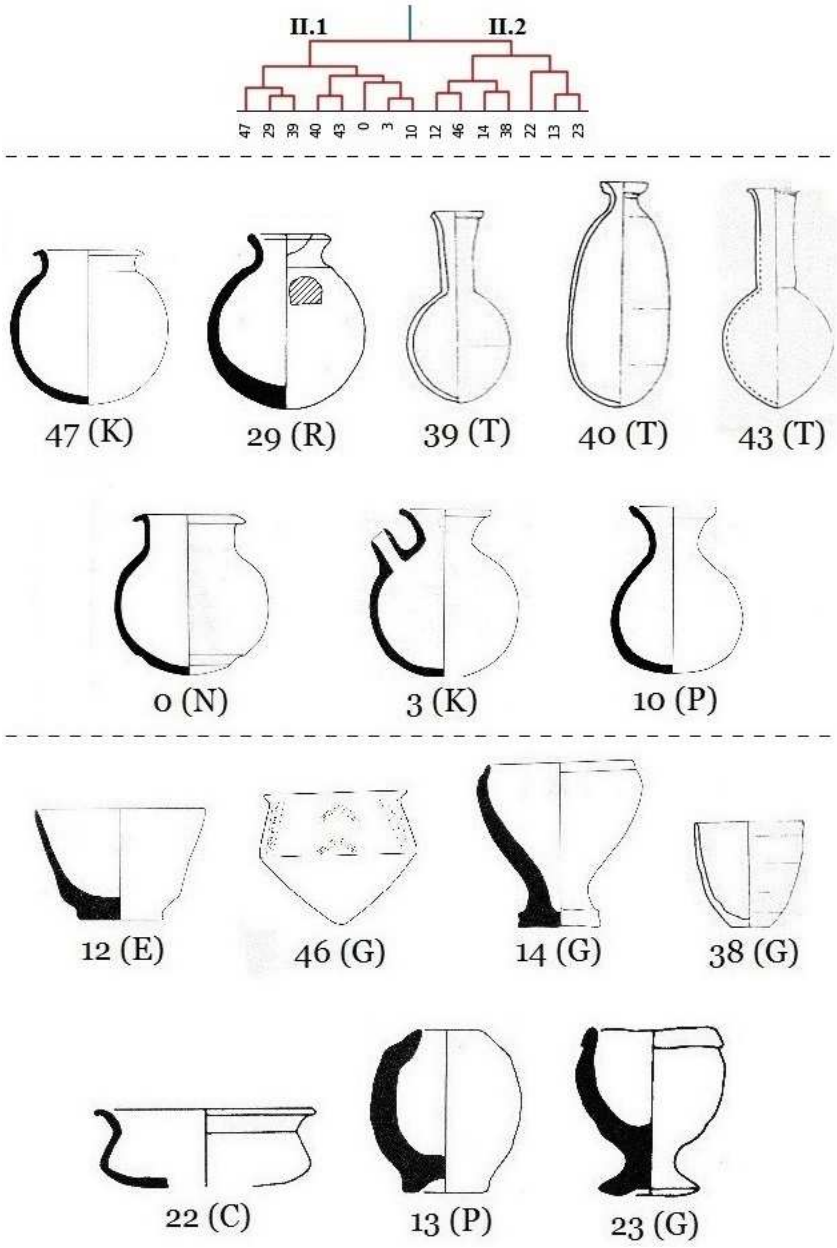


Figure 4.13 – Samples belonging to cluster II.1 and II.2 (complete dendrogram is shown in Figure 4.10). The number indicates the sample random number (from 0 to 49) and the letter in brackets indicate the sample shape class. First row: JZ007_P001, JZ001_P947, JZ002_P603, JZ002_P618 and JZ002_P624; second row: JZ001_P001, JZ001_P008 and JZ001_P083; third row: JZ001_P119, JZ004_P053, JZ001_P192, JZ002_P602; fourth row: JZ001_P485, JZ001_P182 and JZ001_P492. Images at different scales. After Arcane (2016).

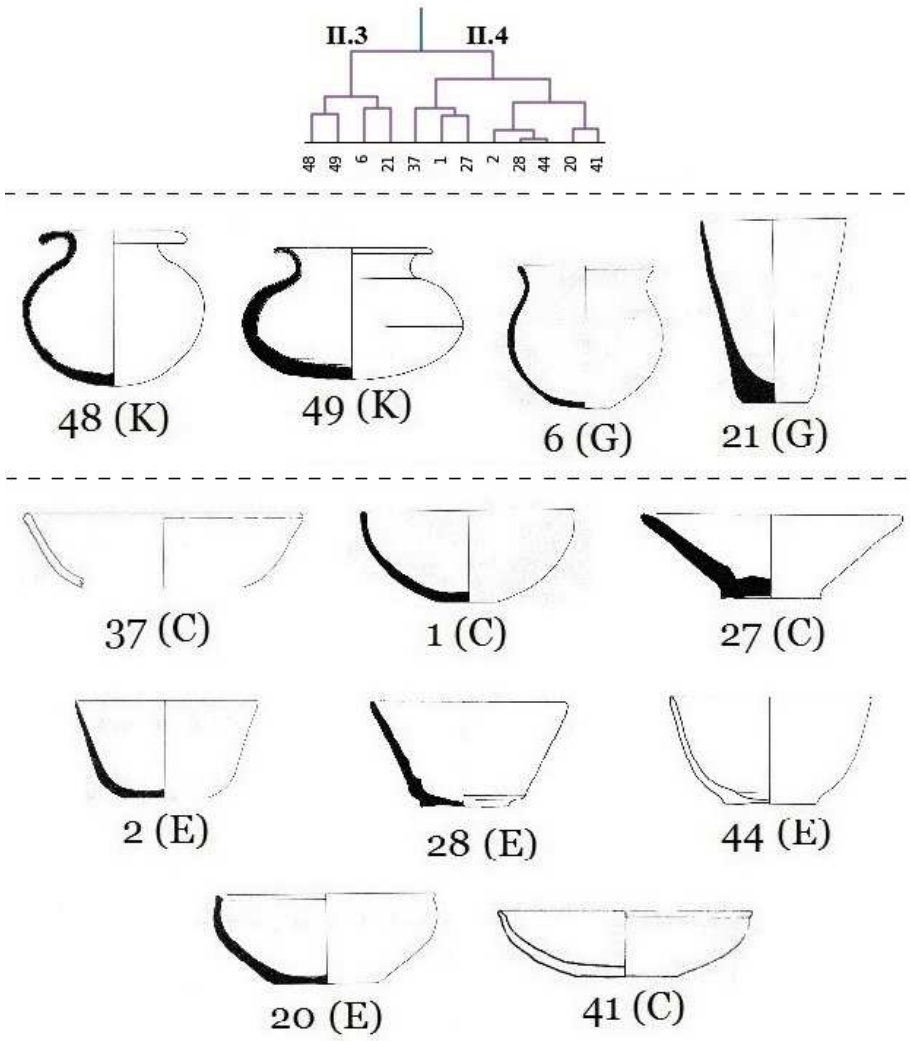


Figure 4.14 – Samples belonging to clusters II.3 and II.4 (complete dendrogram is shown in Figure 4.10). The number indicates the sample random number (from 0 to 49) and the letter in brackets indicate the sample shape class. First row: JZ007_P012, JZ007_P026, JZ001_P024 and JZ001_P295; second row: JZ002_P243, JZ001_P004 and JZ001_P939; third row: JZ001_P005, JZ001_P942 and JZ002_P693; fourth row: JZ001_P292 and JZ002_P621. Images at different scales. After Arcane (2016).

Clusters II.3 and II.4

The third association of shape classes (Figure 4.14), which includes clusters II.3 and II.4, is composed mostly by samples of the open shapes group, including the almost totality of samples of the 'C – Shallow bowl' and 'E – Bowl' classes, and the two remaining samples of the 'G – Cup/Mug/Beaker' class. Two samples of the 'K – Jug/Juglet', a closed shape, are included in this association. The fact that samples of this class that have a higher H-Bd ratio were joined with closed shapes

(cluster II.1) and those with a lower ratio were joined with open shapes will be commented in the next Chapter.

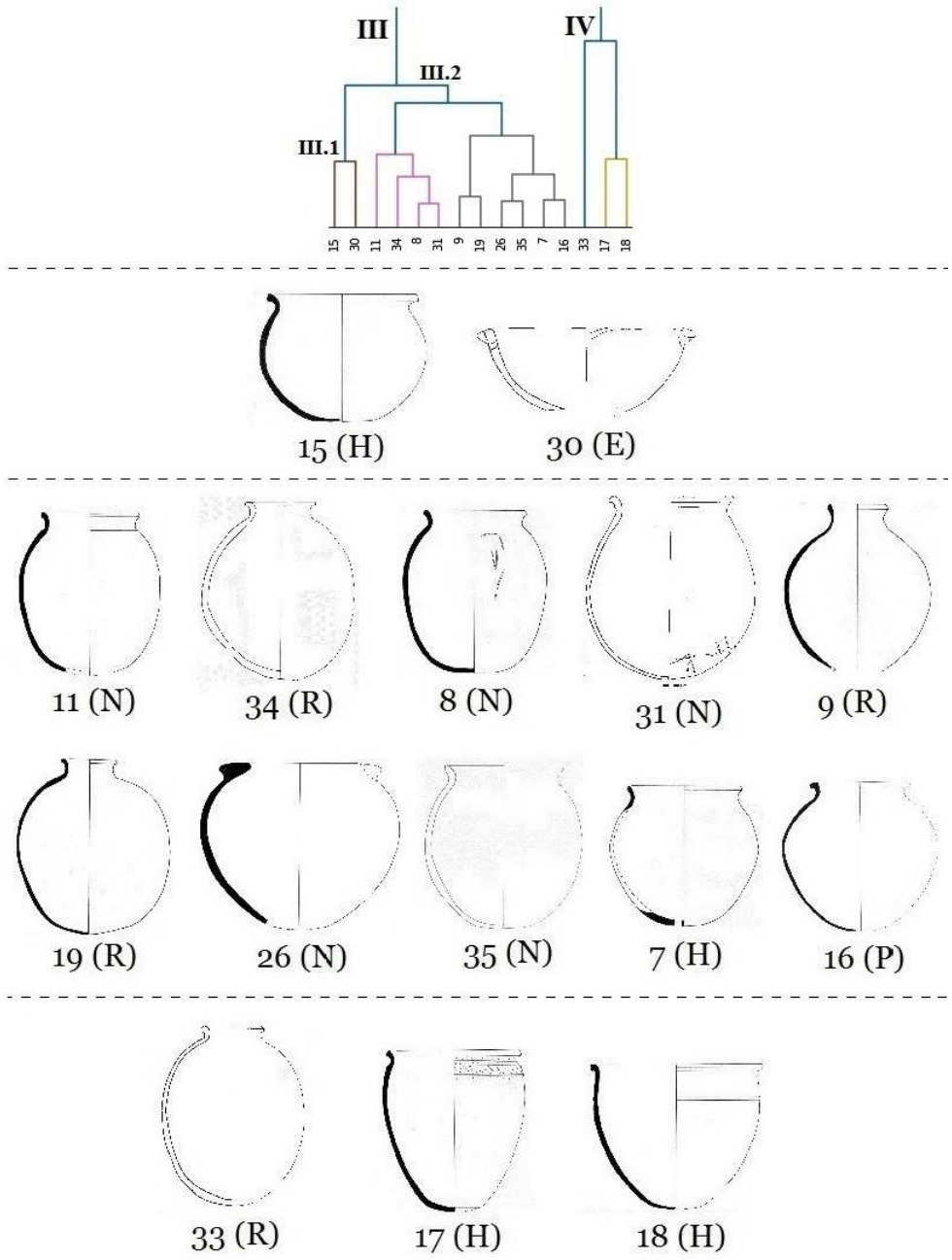


Figure 4.15 – Samples belonging to clusters III and IV (complete dendrogram is shown in Figure 4.10). The number indicates the sample random number (from 0 to 49) and the letter in brackets indicate the sample shape class. First row: JZ001_P203 and JZ002_P060; second row: JZ001_P088, JZ002_P101, JZ001_P052, JZ002_P068 and JZ001_P072; third row: JZ001_P283, JZ001_P932, JZ002_P203, JZ001_P051 and JZ001_P241; fourth row: JZ002_P084, JZ001_P252 and JZ001_P273. Images at different scales. After Arcane (2016).

Clusters III and IV

The predominance of closed shapes, mainly samples of the 'N – Closed pot (high)' and 'R – Jar (restricted neck)' classes, characterises the fourth association of shapes, represented by the clusters III and IV (Figure 4.15). The presence of one sample of the 'P – Jar (wide neck)' class occurs naturally here, apparently unusual is the presence of four samples (from a total of five) of the 'H – Open pot' class. There may be an explanation for this association, which will be discussed in the next Chapter. A final sample (#30) that could be considered an outlier in this association is a member of the 'E – Bowl' class. The main difference between this one and other common samples of the E class is the presence of lugs, this is the only sample in the class that has this additional element.

4.3 Summary of results by shape

This summary aims to provide an overview of the results from both supervised (classification) and unsupervised (clustering) methods focused on the individual shape classes. Each set of information (Figures 4.16 and 4.17) is divided into three parts: i) a generic illustration of the shape class; ii) a summary of metrics for supervised learning, indicating the score of the VC (Vote Classifier), the algorithm with the best overall results, and the highest score considering all algorithms; iii) the relationship among the shape classes, taking both approaches (classification and clustering) into consideration.

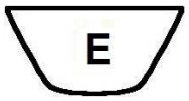
In the third part the symbols ●/○ indicate whether the relationship between two shape classes is stronger (●) or weaker (○), or the cell is left blank if there is no clear relationship. For classification, the criteria are the results provided by the confusion matrices in the second training session (Section 4.1.3). If among the algorithms five or more misclassifications occur for one particular shape, then the relationship between the shapes is stronger. If there are two to four occurrences, the relationship is weaker. For clustering, the criteria are the results provided by the k-Means (Tables 4.13 to 4.15) and Hierarchical Clustering (Figure 4.10). If the shapes are part of the same cluster in 4-5 results, the relationship among the shapes is stronger. If the shapes are part of the same cluster in 2-3 results, the relationship is weaker.

4.3.1 Open shapes

Shallow bowl

Classification	
Total samples	**10**
Test samples	**3**
F_1-Score - VC	**0.50**
F_1-Score - highest	**0.50**

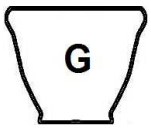
Relation to other shapes		
Shape	Classif.	Clust.
E	●	●
G		○



Bowl

Classification	
Total samples	**182**
Test samples	**46**
F_1-Score - VC	**0.96**
F_1-Score - highest	**0.96**

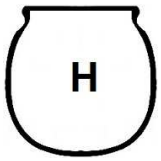
Relation to other shapes		
Shape	Classif.	Clust.
C	●	●
G	●	○
P	○	



Cup/Mug or Beaker

Classification	
Total samples	**96**
Test samples	**24**
F_1-Score - VC	**0.92**
F_1-Score - highest	**0.92**

Relation to other shapes		
Shape	Classif.	Clust.
C		○
E	●	○
H	●	
K	○	●
P		○
T		○



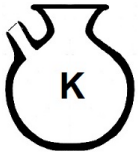
Open pot

Classification	
Total samples	**20**
Test samples	**5**
F_1-Score - VC	**0.89**
F_1-Score - highest	**0.89**

Relation to other shapes		
Shape	Classif.	Clust.
G	●	
N		●
R		●

Figure 4.16 – Summary of results: open shapes. Images drawn after Arcane (2016) samples: JZ001_P939, JZ001_P015, JZ001_P110 and JZ001_P087.

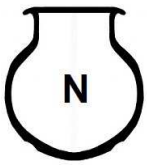
4.3.2 Closed shapes



Jug/Tankard or Juglet

Classification	
Total samples	**22**
Test samples	**5**
F_1-Score - VC	**0.67**
F_1-Score - highest	**0.89**

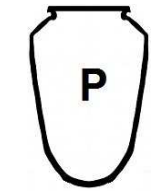
Relation to other shapes		
Shape	Classif.	Clust.
G	○	●
P	●	●
R	●	○
T		●



Closed pot (high)

Classification	
Total samples	**34**
Test samples	**8**
F_1-Score - VC	**0.67**
F_1-Score - highest	**0.67**

Relation to other shapes		
Shape	Classif.	Clust.
H		●
P	●	○
R		●



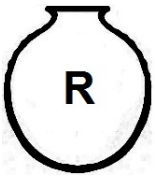
Jar (wide neck)

Classification

Total samples	**88**
Test samples	**22**
F_1-Score - VC	**0.81**
F_1-Score - highest	**0.86**

Relation to other shapes

Shape	Classif.	Clust.
E	○	
G		○
K	●	●
N	●	○
R	●	○
T	●	●



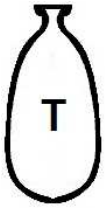
Jar (restricted neck)

Classification

Total samples	**32**
Test samples	**8**
F_1-Score - VC	**0.82**
F_1-Score - highest	**0.88**

Relation to other shapes

Shape	Classif.	Clust.
H		●
K	●	○
N		●
P	●	○
T	●	○



Flask/Bottle

Classification

Total samples	**12**
Test samples	**3**
F_1-Score - VC	**0.67**
F_1-Score - highest	**0.80**

Relation to other shapes

Shape	Classif.	Clust.
G		○
K		●
P	●	●
R	●	○

Figure 4.17 - Summary of results: closed shapes. Images drawn after Arcane (2016) samples: JZ001_P008, JZ001_P001, JZ001_P923, JZ001_P272 and JZ002_P618.

5 DISCUSSION

The research results based on the supervised and unsupervised approaches will be assessed with a focus on the main themes directly related to the research questions, on general issues of the application of ML in archaeology, and how the research results compare to selected related studies.

5.1 Main themes

5.1.1 Artefact features

The Decision Tree Classifier (DT) was not among the algorithms with the higher performance, nevertheless it was very useful for the analysis of results through the generated decision tree and the relevance of the features used in the tree. The analysis of the decision tree, either in graphic or textual form, makes easier to understand the criteria used by the algorithm for the classification and, based on this information, the traditional classification may be evaluated and provide new insights into the applied methods. Despite the feature importance being a property of algorithms based on decision trees like DT and Random Forest, the information provided by them can be used to improve the dataset as a whole and consequently improve the results of other algorithms.

As summarised in Table 4.3 and Figure 4.1, the most important features are two relative measurements, H-Bd (Total Height / Belly diameter ratio) and Bd-Nd (Belly diameter / Neck diameter ratio), followed by two absolute measurements, base diameter and neck diameter. The most relevant categorical feature is rim_profile[7] (round-folded outside), and the next one is base_type[2] (rounded base). If all categorical features were considered not individually, but as a unit, without the division created by the OneHotEncoder method, they would have the following average importance based on Table 4.3: rim_profile = 0.055, base_type = 0.034, additional_elem = 0.026, and rim_orient = 0.012. The rim_profile feature would be the 5^{th} in importance, and the base_type the 10^{th}. These are just estimative, the actual values would only be revealed if the encoder had not processed these features.

The identification of the most relevant artefact features and how these contribute for the identification of vessel shape is one of the strengths of ML. The following case, based on the tree generated by the DT algorithm during the second training session (Section 4.1.3), illustrates at the same time the importance of feature definitions and the exactness of the data used to build the ML model.

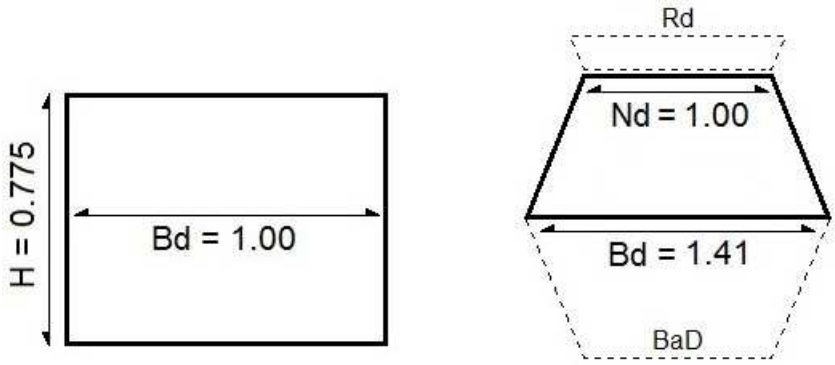


Figure 5.1 – Vessel measurements used by the Decision Tree classifier in the root (H-Bd ≤ 0.775) and first level (Bd-Nd ≤ 1.41) tests. Left: shape proportions that separated the majority of bowls from other shapes; Right: proportions that separated the majority of closed shapes samples from the open shapes. H = total Height, Bd = Belly diameter; Nd = Neck diameter; Rd = Rim/opening diameter; BaD = Base diameter.

Figure 5.1 illustrates the vessel measurements used by the Decision Tree classifier in the root (H-Bd ≤ 0.775) and first level (Bd-Nd ≤ 1.41) tests. Both tests applied the relative measurements defined in Section 3.1.8. The left image shows the shape proportions that separated the majority (92%) of 'E – Bowls' samples from other shapes in the training part of the dataset. The right image shows the proportions that separated the majority (91%) of closed shapes samples from the open shapes samples in the training part of the dataset.

The samples of the 'E – Bowl' class were separated into two initial branches, and then further divided in the deeper level branches, as described in Section 4.1.3. The focus here is the first division, especially the separated branch that included the minority of E class samples. Figure 5.2 illustrates the samples that were included into this branch by the DT classifier, which represent bowls that are deeper than the more common bowl types.

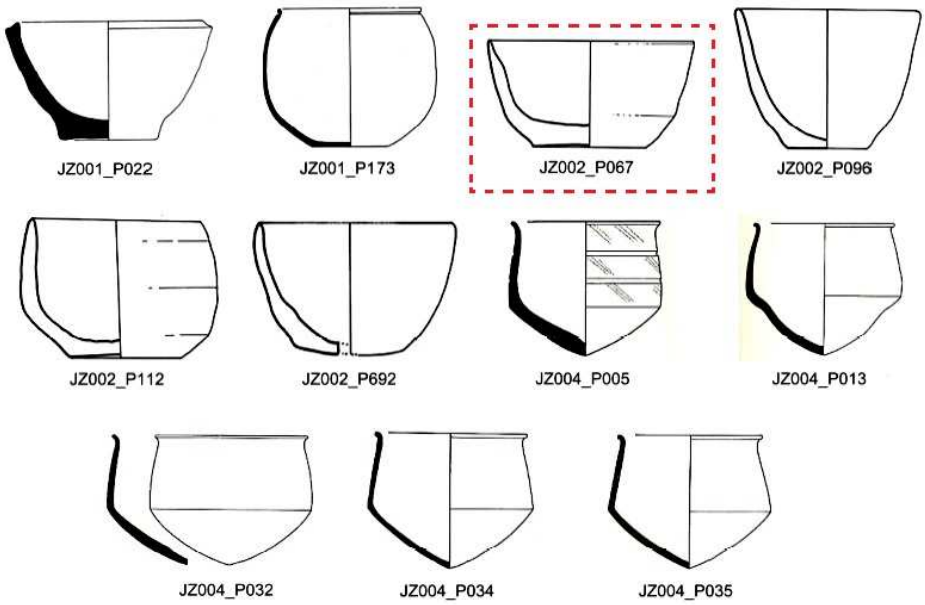


Figure 5.2 – Samples of 'E – Bowl' shape class that were classified in a separated branch by the Decision Tree classifier (Section 4.1.3), under the condition H-Bd > 0.775. The highlighted sample JZ002_P067, which originally would not match this condition, was included in this branch by the algorithm because of a measurement error in the Arcane database. Images at different scales. After Arcane (2016).

The sample JZ002_P067 (highlighted in Figure 5.2) was included in this selection by the DT algorithm because an error in the Arcane database measurements. After analysing these results, the vessel was measured through ImageJ using the scale provided in the original Arcane image. It was possible to conclude that its height (6.0 cm) and other measurements are correct, however the diameter at opening was recorded as 5.70 cm, when the correct measurement should be around 11.0 cm. The diameter at opening is used in the relative measurement H-Bd in the place of belly diameter when the vessel has no belly and neck, as explained in Section 3.1.8. As a consequence of this error, the shape recognised by the algorithm did not reflect the original shape of the vessel.

The next issue is related to NA (null) values in features, in this case the base_diam (base diameter) feature, extending to the Bd-BaD (Belly diameter / Base diameter ratio) feature. Both features have a number of samples with NA values (89 in 496 samples). As commented in Section 3.3.4, scikit-learn algorithms does not process NA values and these must be converted to zero or other calculated value. This may not be a big problem for those vessels that have a pointed or small base in

relation to the body, e.g., the samples starting with ‘JZ004’ in Figure 5.2, but for vessels with larger proportional bases, the vessel shape can be distorted and consequently being misclassified by the algorithms.

Alternatives to this issue could be the non-utilisation of these features in the ML model (as it was done with min_diam and max_diam after the first training session), or the removal of samples that have NA values for these features. It was decided to keep these features and samples in this research because completely preserved pottery vessels in archaeology are relatively rare, the most usual is the preservation of certain parts of the vessel, which in some cases can lead to identification of missing parts and the ‘visualisation’ of the complete vessel shape based on its measurements (Orton et al. 1993, p. 76-80). The occurrence of incomplete samples is important to assist in the identification of potential issues and limitations in the ML model, in order to develop techniques that may improve the results taking these limitations into consideration.

5.1.2 Shape classes

Some shape classes may be more complicated to classify than others, apart from the sample size of the class. It is for instance the case of ‘N – Closed pot (high)’, which has a sample size greater than other shape classes (H, K and R) but returned lower scores than these classes. In the second training session (Tables 4.6 to 4.10), nearly half of the samples of the N class were misclassified as ‘P – Jar (wide neck)’ or other shapes. The N shape is also the one with greatest dispersal in clustering (Table 4.16, almost equally divided into three clusters), followed by the ‘R – Jar (restricted neck)’ class.

The class ‘G – Cup/Mug/Beaker’ has samples with significant differences among each other (Figures 3.4 and 4.13), especially regarding the base type, nevertheless it is one of the shape classes with higher scores in classification (0.92), and it is divided basically into only two clusters (Table 4.16). A similar comment can be made about the class ‘K – Jug/Juglet’, a high score in classification (0.89) and only two basic clusters. Some of the differences in the case of samples of the K class are related to the presence of additional elements, especially spouts, which were correctly identified by the Decision Tree algorithm (Figure 4.7).

From the five samples of the ‘H – Open pot’ class present in the dendrogram (Figure 4.10), four are associated with clusters represented by closed shapes. One sample of the H class is associated with samples of the K, P, R and T classes (Figure 4.12), while other three H samples are associated with samples of the N and R classes (Figure 4.15). The decision tree generated in the second training session (Figure 4.8) also indicates a proximity between the H class and those of the closed shape group, however the closer shape to H according to supervised learning in general is the ‘G – Cup/Mug/Beaker’ class (Figure 4.17). These results suggest that the H class can be considered a ‘hybrid’ class, having characteristics from both open and closed groups. From the open shapes the most evident is the wider opening/rim diameter, and from the closed shapes, the presence of a neck and the rim profile/orientation.

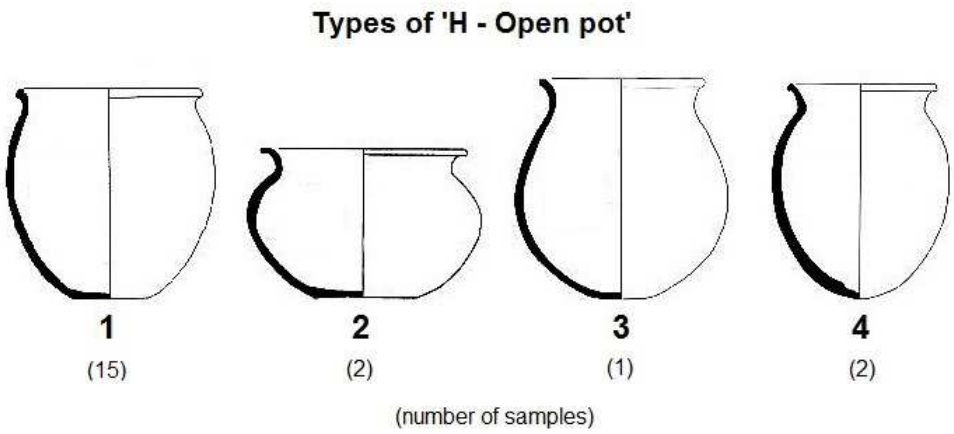


Figure 5.3 – Types of ‘H – Open pot’ shape classes identified through analysis of the tree generated by the Decision Tree classifier. *Vessels*: 1 = JZ001_P258; 2 = JZ004_P092; 3 = JZ001_P049; 4 = JZ001_P010. Images at different scales. After Arcane (2016).

The Decision Tree classifier can also be useful in the definition of potential sub-classes/types, or in the identification of outliers (atypical class members). Figure 5.3 shows four types of vessels of the H class, and Appendix B.1 shows the structure that identified these types based on the DT classifier. The tree mechanism is the same from the original decision tree (Figures 4.5 to 4.7), the difference is the focus in just one shape class. Another difference is that the original tree used only the 15 samples that are part of the training dataset, in Figure 5.3 and in the appendix all 20 samples of the H class are considered. The

vessel type #1 is the most common type of the H class in the dataset, 15 samples of this type have at the same time H-Bd > 0.77, Bd-Nd ≤ 1.41 and belly_diam > 18.21. Other 5 vessels have at least one feature that has different values from these parameters defined by the algorithm: type #2 is shorter, type #3 has a more restricted neck and type #4 has narrower belly diameter when compared to the average vessels of the H class. This is a simplified classification, a more complete one would also apply categorical features (rim profile/orientation, base type) to refine it. The tests applied by the DT classifier using these features and values were created to process all shape classes in the dataset, in the case of a dataset where only samples of H class were used, it is possible that different criteria would be applied.

According to the specifications in Section 3.1.2, the main difference between jugs and juglets is the vessel height (up to 15 cm for juglets). From the five samples of the K class included in the dendrogram, four are defined as juglets in the Arcane database. Three samples are associated with closed shapes (Figures 4.12 and 4.13) and two are associated with open shapes, closer to the 'G – Cup/Mug/Beaker' class (Figure 4.14). It is possible to identify that the two samples associated with open shapes have a lower H-Bd ratio compared to the other three, but there is no relation between this information and the division between jug and juglet, since three juglets were associated to the closed shapes, which have a higher H-Bd ratio. It would be necessary to expand this analysis to a greater number of samples in order to identify a clearer pattern for the K class, and the criteria used by the clustering algorithms to group its samples. Furthermore, the presence of additional elements should be considered since only samples of K and N classes have spouts, and the only sample with handles in the dataset is a jug (Figure 4.12, sample #42).

The 'P – Jar (wide neck)' class is peculiar in the sense that it produced high scores (up to 0.86) but at the same time it was the class that produced the most variety in misclassifications (Table 5.1), with three to four related shapes in average, five in the case of Voting Classifier. The positive side of the results is probably due to the high number of samples (the third highest in the dataset), an opposite situation when compared to the C and T classes. In terms of cluster analysis, the P shape was divided into four clusters, but only two are the prevailing ones, with 81% of samples (Table 4.16).

During the development of this research it was not possible to access the complete Arcane project documentation, which is available in printed volumes (Arcane, 2016b). For this reason there is limited information about potential criteria used in the assemblages classification and how the vessels are associated to shape classes, beyond the general guidance presented in Section 3.1.2. These specifications work as a general reference and may not necessarily be rigorously followed since the final definition of shape classes may be a subjective matter.

Taking this limitation into consideration, one explanation for associations described in the previous paragraphs may be related to the concepts of open and closed shapes and how the shape classes are defined. In principle, two possibilities can be considered regarding the process of defining the vessel shape class, which are also related to the top down and bottom up approaches described in Section 2.1.3. The first alternative is broadly classifying a vessel as either open or closed (or some of the other groups listed in Table 3.1), and then comes the decision of what specific class the vessel belongs to. The second alternative is to directly define the vessel shape class, and it is then automatically defined as belonging to either the open or closed group.

The criteria for defining the shape classes are different for open and closed shapes (Section 3.1.2). In the case of open shapes, both the relation between vessel diameter and vessel height, and the absolute diameter and height, are used as reference. For closed shapes, the relation between the minimum and the maximum vessel diameter, and the absolute height are used. Since there are different criteria for both groups and these criteria are not mutually exclusive, there is an overlap in the classification criteria between the two groups. In order to avoid this problem, it would be necessary to define first whether a vessel belongs to the open or close group, and then apply the classification criteria to define the shape class under one of these groups.

Finally, the possibility of misclassifications caused by occasional data input errors must be considered. In one specific case, identified during the sample selection in the Arcane database, the vessel JZ001_P175, classified as a beaker (G class), clearly does not belong to this class, according to the vessel measurements (height of 95 cm and capacity of 298 litres) and illustration, it could be classified as a large storage jar (possibly a ‘S – Pythos’?).

Voting Classifier

	Classified as →	C	E	G	H	K	N	P	R	T	F_1
C	Shallow bowl	**1**	2	-	-	-	-	-	-	-	0.50
E	Bowl	-	**45**	1	-	-	-	-	-	-	0.96
G	Cup/Mug or Beaker	-	-	**24**	-	-	-	-	-	-	0.92
H	Open pot	-	-	1	**4**	-	-	-	-	-	0.89
K	Jug/Tankard or Juglet	-	-	1	-	**4**	-	-	-	-	0.67
N	Closed pot (high)	-	-	-	-	1	**4**	3	-	-	0.67
P	Jar (wide neck)	-	1	1	-	1	-	**17**	1	1	0.81
R	Jar (restricted neck)	-	-	-	-	1	-	-	**7**	-	0.82
T	Flask or Bootle	-	-	-	-	-	-	-	1	**2**	0.67

Table 5.1 – Confusion matrix resulting from the Voting Classifier algorithm (reproduced from Table 4.8), with some misclassifications highlighted in blue. The sample highlighted in orange is JZ004_P064 vessel (Figure 5.4).

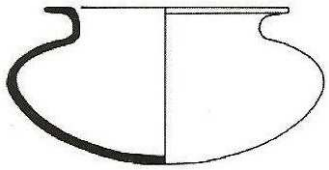


Figure 5.4 – JZ004_P064, vessel of the 'P – Jar (wide neck)' class misclassified as 'E – Bowl'. After Arcane (2016).

This sample was not included in the research dataset because it would just provide wrong information for the training of ML algorithms. If an error like this was not identified during the sampling selection process and a ML model processes a problematic sample, the results provided by tools such as the confusion matrix (e.g., an association between two completely distinct classes) would raise the attention of the analyst for a potential mistake in the dataset. This is one of the benefits of using ML in pottery classification.

This issue can also be exemplified through the confusion matrix provided by the Voting Classifier algorithm (Table 4.8), and reproduced here (Table 5.1). A characteristic of ML algorithms that achieved the highest scores in this research is to produce misclassifications of shapes that are closer to the original shape. The misclassifications are mostly adjacent or near to the main diagonal, shown in pale blue in Table 5.1. Because of the shape classes characteristics, it is more

reasonable to expect samples of the C class misclassified as E class, or samples of the N class misclassified as P class than, for instance, samples of the C class misclassified as T class. The cell highlighted in orange in Table 5.1 shows a sample of the 'P – Jar (wide neck)' class misclassified as a sample of the 'E – Bowl' class. These classes are not close to each other. After a more detailed analysis on this particular case, the sample was identified as the JZ004_P064 vessel.

This vessel (Figure 5.4) looks like an uncommon member of the P class, it is possible to identify similarities in the vessel body with some samples of the E class, and it was interpreted that way by the VC and other ML algorithms. This vessel is also similar to some samples of the 'K – Jug/Juglet' class (e.g., sample #49 in Figure 4.14), the main difference is in the base type. It is not possible to know if this was a misclassification in the Arcane database, this vessel was not originally classified as a bowl likely because of certain details beyond the basic shape, like the neck and the rim profile. In this particular case, the vessel measurements and the categorical features were not enough to identify it as a sample of the P class (or other class, in the case of an original misclassification in the Arcane database). In any case, the analysis of the confusion matrix result would draw the analyst's attention to a revision in the classification of the sample.

5.1.3 Classification and clustering

A ML model can be used as a tool to assist pottery experts in their decision while performing the task of vessel classification. In the supervised learning approach, the overall performance of 0.87 in accuracy achieved by the Voting Classifier algorithm indicates that ML may produce results that can be useful to the expert, and it is theoretically possible that a more balanced dataset, with a more uniform distribution of shape classes, could provide highest results. This could be a question for future research.

The benefits of using unsupervised learning methods are not so clear initially, as it is the case of supervised learning, mostly because there is no straightforward way to assess the results since there are no target classes to be compared. On the other hand, clustering methods such as k-Means and Hierarchical Clustering have the

benefit to create clusters, which can become potential classes, based on different criteria and therefore providing new insights into the pottery assemblage. Clustering methods also allow a new analysis on previously classified collections in accordance with new perspectives and approaches. The basic classification criteria used in this research, dividing the shape classes mainly in open and closed shapes, was only followed partially by the clustering algorithms. This cannot be considered a problem, on the contrary, because one of the goals of unsupervised learning is to identify possible new patterns and relationships among artefacts.

Groups of artefacts (either classes or clusters) should be internally coherent and externally isolated (Read, 2007, p. 64, p. 135-8), and the groups should be precisely defined to allow the classification to be externally reproduced (Orton et al., 1993, p. 152). Members (the samples in the dataset) must be clearly identified as belonging to one group and not to others. In practice this may not be easily achieved, as the results from ML algorithms show. Samples from the same shape class are split into two or more clusters, and misclassifications occur in all shape classes, with different degrees of precision/recall. Nevertheless, it is possible to identify patterns in the grouping of shape classes.

Five shapes clearly relate to each other in terms of both classes and clusters, based on the results: C-E, K-P and P-T. Some shape classes show stronger relations with other classes in terms of classification: E-G, G-H, N-P, P-R and R-T, while other shape classes show stronger relations with other classes in terms of clustering: G-K, H-N, H-R, K-T and N-R.

The creation of a complete taxonomic structure for this assemblage like the example shown in Figure 2.8 is beyond the scope of this research, nevertheless it is possible to provide some examples of structures based on the results from both supervised and unsupervised learning. The experts from the Arcane project somehow created a specialisation for some shape classes as commented in Section 3.1.2, it is the case of the G class, divided in samples described as ‘Cup/Mug’ or ‘Beaker’, the K class, divided in ‘Jug/Tankard’ or ‘Juglet’, and the T class divided in ‘Flask’ or ‘Bottle’. In these cases the size of the vessel usually indicates the differences in the description, but the shape class (the target) is the same, therefore there is no way for ML algorithms to recognise any difference between them.

The taxonomic structures presented here are valid for the assemblage used in this research only, and do not represent other archaeological sites or cultures recorded in the Arcane database, for this reason 'Research assemblage' is on the top of the structures. In addition, not all potential subclasses or super classes are shown in these structures.

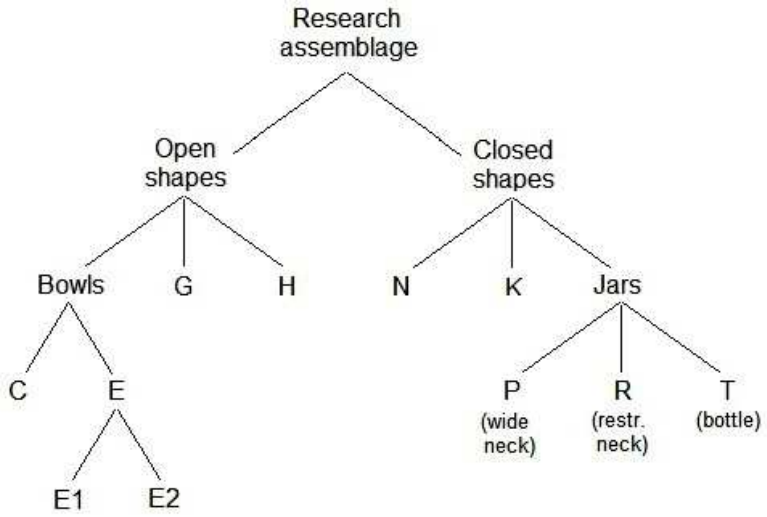


Figure 5.5 – Possible hierarchy of classes (taxonomic structure) based on supervised learning algorithms results. C = Shallow bowl; E = Bowl; G = Cup/Mug/Beaker; H = Open pot; K = Jug/Juglet; N = Closed pot (high); P = Jar (wide neck); R = Jar (restricted neck); T = Flask/Bottle.

Through supervised learning results it is possible to identify a closer relation among certain shape classes, especially C-E in the open shapes group, and P-R-T in the closed shapes group. These shape classes could be grouped to form two super classes, Bowls and Jars (Figure 5.5).

The 'T – Flask/Bottle' class could be considered a special case of the 'Jars' class, since the main differences between members of the T class and those of the P and R classes is the more restricted neck, and the lower height in the case of flasks. On the opposite way, the E class could be separated in two subclasses, as identified by the Decision Tree classifier. The bowls with a higher H-Bd (total Height / Belly diameter) relation could compose one of these subclasses, not necessarily based on the H-Bd value greater than 0.775 as in the DT test, but a similar value

could be used as a reference. This subclass of 'deeper' bowls would be an intermediate between the 'E – Bowl' and the 'F – Deep bowl' classes.

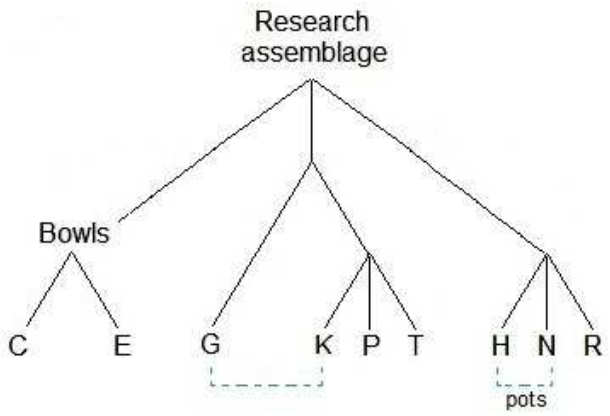


Figure 5.6 – Possible hierarchy of classes (taxonomic structure) based on unsupervised learning algorithms results. C = Shallow bowl; E = Bowl; G = Cup/Mug/Beaker; H = Open pot; K = Jug/Juglet; N = Closed pot (high); P = Jar (wide neck); R = Jar (restricted neck); T = Flask/Bottle. The dotted lines indicate shape proximities.

Through unsupervised learning results it is also possible to identify a closer relation between C-E shape classes as in the case of supervised learning, but the clustering analysis revealed different groupings in the case of the closed shapes group (Figure 5.6). One group is the cluster resulting from the K-P-T classes (with the possible inclusion of the G class), and the other one is the cluster resulting from the H-N-R classes. This cluster has the peculiarity of including the H class, an open shape class, together with two closed shapes, N and R. The criteria used in the Arcane project to separate the classes in open and closed groups (which is based partially on the relation between belly and neck diameters) was not followed by the clustering algorithms. It is easier to identify a similarity between the H and N shapes, since both are described as 'pots'; it must be also observed the Arcane project division of closed pots in two classes, 'N – Closed pot (high)' and 'M – Closed pot (squat)', which is not used in this research because the motives explained in Section 3.1.9.

The relation of both shapes, H and N, with the 'R – Jar (restricted neck)', is more complex to identify through the information provided by the clustering algorithms, and it would require an individual analysis of most samples from these

classes. It is possible however to identify some similarities through the vessels shown in Figure 4.15, which contains most samples from the H, N and R classes selected for the dendrogram. In this figure it is possible to identify the more restricted neck of the samples belonging to the R class, and therefore the neck diameter was probably not among the main criteria to create these clusters.

Based on a visual and exploratory analysis, since unsupervised learning algorithms do not provide information about the importance of features, it is possible to suggest that the overall vessel shape (represented by the H-Bd feature) was more decisive in this case, and also the rim orientation/profile in a secondary way. The majority of vessels in Figure 4.15 (excluding the ones in the first row, samples #15 and #30) have a similar overall shape, with an H-Bd range of 0.83 to 1.27 (mean = 1.06), and prevalence of rim profiles 03 (Rounded), 04 (Thickened) and 09 (Horizontal folded outside). If these vessels are compared with the ones from Figures 4.13 and 4.14, this idea is reinforced. On the other hand, the vessels in Figure 4.15 are more similar to those in Figure 4.12, which belong to a distant cluster. One difficulty here is to name the clusters (except the Bowls), the K-P-T and the H-N-R clusters do not have a clear term that could be used to represent all the shape classes in each cluster.

5.2 Machine learning issues

Dataset context

The results provided by the ML model are valid at first only for pottery assemblages that are culturally related in some way to the assemblage used for this research. The four archaeological sites, selected among 168 sites available in the Arcane dataset, are from the same region (Northeast Syria), approximately from the same period (Third millennium BC) and the contexts of finds are similar (domestic/storage) for the great majority of the samples (Section 3.1.1). The model trained with information about this dataset would probably not generalise well for pottery associated with other cultures or periods, but tests should be made in order to quantify this hypothesis.

The decision to use an assemblage from four different sites (even if culturally related) was made in order to provide a minimal amount of well-preserved

samples to allow the training of ML algorithms. It is possible that this choice contributed to the variation of shapes within certain shape classes and, consequently, to limit the ML model performance. The site with greater amount of samples, Tell Brak, is also the one with greater time range, however the great variety in shapes is probably from Tell Leilan, since part of the samples are from funerary contexts and not from domestic/storage contexts as the majority of samples in the dataset.

In respect to the dataset structure (the features used to obtain the target classes) and the procedure used in the training sessions (Section 3.3.11), they are generic enough and at first they could be used to other types of assemblages, therefore the model could be reutilised with new samples and trained with them without significant modifications.

Benchmark

One of the most relevant aspects observed in the results is related to how consistent is the benchmark used to assess the performance obtained from the training sessions. The results were assessed analytically through the confusion matrices and synthetically through accuracy and F_1-Score metrics, but their utility is limited by the quality of the benchmark.

As explained in Section 5.1.2, the access to the completed Arcane project documentation was not possible, limiting the hypotheses about the original classification process used in the project. It is remarkable that all vessels in the database (more than 8200 records from 168 sites) were classified under the same shape classes' scheme, allowing the comparison of different sites and periods. It is possible, however, that different experts were responsible for defining the vessels' shape classes in different sites, using slightly different criteria when classifying samples for different sites or different contexts within sites. There is a section in the Arcane pottery database where some information about the users responsible for specific actions is found: entering and editing data, scientific validation, and technical validation (which is software related). In the case of the four sites used in this research, only JZ002 – Beydar has information about a scientific validation task, and the users responsible for entering data are different for each one of the four sites.

Even if the four selected sites and their assemblages are reasonable similar among each other, it is possible to expect some differences in the classification criteria applied by the experts, which would impact the learning process and consequently the performance during prediction.

Algorithms parameters

Some of the algorithms that provided better results than DT are less transparent in their mechanisms (Logistic Regression, SVC) and more sensible to the combination of input parameters, and in those cases the utilisation of the grid search and cross-validation methods is important to fine tune the ML model and obtain the best possible performance. Albeit it is possible to test a combination of parameters manually in the algorithms, the utilisation of methods such as GridSearchCV in scikit-learn makes this task easier. Even when applying this method it may be necessary to test different parameters combinations since some algorithms, especially SVC, have a considerable alternative of parameters, and using a large number of them at the same time may slow down the running of training scripts. This was not a problem in this research particularly, since the dataset is relatively small, and the longest time for a script to run the GridSearchCV method was limited to a few minutes.

The DT classifier has some disadvantages when compared to other ML algorithms used in this research, among them is the sensitivity to small variations in the training data, and unbalanced datasets (certain classes are dominant in terms of sample quantity) (Géron, 2019, p. 185-6; Scikit, 2021j). Another issue is related to the necessity to limit the growth of the tree to avoid the model overfitting, which means that the model may perform well on the training data but it does not generalise well for unknown data (e.g., the test dataset) (Géron, 2019, p. 27-9, p. 180-1; Müller & Guido, 2017, p. 28-30). When the DT classifier is not limited in depth, it can generate a highly complex tree that produces a very high performance on the training dataset (accuracy can be equal to 1.00), but produces a much lower performance on the test dataset. After limiting the tree growth (in this research the limit in 5 or 6 levels produced the best results, but that depends on the dataset characteristics), the generated tree based on the training dataset produced a lower performance but on the other hand the performance increased for the test/unknown dataset, which is the goal of the ML models.

The issue of overfitting is valid for other algorithms beside DT classifier, how they will perform on both the training and test datasets is controlled by a series of parameters, some algorithms requires the use of none or few parameters, while others requires an optimal combination of several parameters to acquire the maximum performance. For that reason the technique of grid search with cross-validation was applied in the third training session. Another relevant issue related to parameters is the optional randomness during the creation of training and test datasets through the 'train_test_split' method. It is preferable to use the option for not generating a random splitting of the train and tests datasets each time the training session runs, that way the samples will be splitted the same way every time and the results can be reproduced in order to assess and compare them.

5.3 Related research

Some of the studies presented in Section 2.2.4 have aims similar to those of this research, one of the differences is related to method: while others use mainly visual features and image analysis as a basis for classification, this research uses categorical and continuous vessels features to identify shape classes. Another difference is related to the type of artefact: complete or near complete vessels as in the case of this research, or ceramic shards in case of many others. This Section comments on three studies that have similar goals or apply methods that are more closely related to this thesis, or that raise relevant issues to it.

Methodology for typology development

Hörr et al. (2014) apply a combination of unsupervised, semi-supervised and supervised learning methods to create a methodology for a ML based typology development in archaeology. The method consists in the application of the different ML methods in a sequence, starting with an unsupervised phase (including the definition of features to estimate similarities between instances), continuing through a semi-supervised phase (when the relevance of the features is measured, the most relevant ones are selected and preliminary types are defined), and finishing with the supervised phase when several classifiers are trained with the labelled data and new instances are assigned to one of the defined types. If a misclassification is identified and the type assignment is proved incorrect, a new

label is given and a reclassification takes place. In Hörr et al. (2014) different ML methods are applied in a sequence of phases aiming to define a new typology for a pottery assemblage, while in this thesis the supervised and unsupervised methods are applied in a comparative way, aiming to analyse and assess an already established classification.

The dataset used by Hörr et al. (2014) consisted of 599 pottery vessels from the Lusatian Culture (1400-800 BC) cemetery of Kötitz, Germany. The vessels measurements were obtained automatically through 3D scanning (that way avoiding inaccuracies) and thirty-five features were initially selected, including categorical information, absolute and relative measurements. Among the features that were considered more important by the algorithms are the relative measurements, but also categorical ones like shape and number of attachments. Here the difference is significant, since the categorical feature additional_elem (additional elements) used in this thesis is not among the most important because these elements appear only in a few samples.

Among the several algorithms used by Hörr et al. (2014), some are the same or equivalent to the ones used in this thesis: Majority Voter, Logistic Regression, Random Forest, k-Nearest Neighbors, and C4.5 decision tree classifier. The typology development process after several refinements defined nineteen primary vessel types, and new instances were correctly classified with a probability of up to 95%. One interesting aspect is that the absolute size of the vessels was not taken into consideration for type definitions. The cultural uniformity and the quality of vessels in the assemblage contributed for the high level of prediction rate. In the same way that this thesis, some types were considered more difficult to classify and others had lower performance rates because of the low quantity of samples.

Classification based on ceramic chemical composition

In the second example study, Charalambous et al. (2016) apply ML algorithms to identify classes of utilitarian pottery from the Early/Middle Bronze Age Cyprus (c. 2400-1700 BC), but use chemical composition of ceramics as the basic data for classification. In this case the research focus is to identify degrees of similarity between types based on their chemical profiles, and address aspects of ceramic production and distribution.

The dataset used by Charalambous et al. (2016) includes 177 ceramic samples from eight different sites across the island, while the features are mineralogical and chemical characterisations (such as MgO and Al_2O_3), obtained from ED-XRF analysis. The methodology includes the application of supervised learning algorithms k-Nearest Neighbors, C4.5 decision tree classifier and LVQ (Learning Vector Quantisation). The reported results are the classification maximum accuracies (%) of 79.4 (KNN), 77.2 (C4.5) and 65.2 (LVQ). Some of the most useful results were obtained through the analysis of the confusion matrices, which indicated previously unidentified relationship between certain classes/fabrics of ceramic (Charalambous et al., 2016, p. 470).

One of the issues mentioned in the research is the small (for ML purposes) number of samples compared to the high number of classes (36). To overcome this limitation, the technique of bootstrapping with replacement was applied to generate the datasets of 177 samples (Charalambous et al., 2016, p. 467-8). This technique allows taking the samples as if it were a population and randomly selecting new samples from it several times (Drennan, 2009, p. 136). The dataset characteristics and the choice of parameters are possible causes for the lower performance of LVQ, which is considered a more complex algorithm, when compared to the other two (Charalambous et al., 2016, p. 470). Another issue in the research is related to the number of classes with only one member, which were included in the dataset nevertheless. It is not clear if the bootstrapping solved this issue or how the classification algorithms dealt with this limitation, since one of the principles of supervised learning is the division of the dataset in training and test parts, and samples of one class must be present in both datasets in order to train the model and assess the classification results.

Classification based on pottery decoration

This last example study, by Pawlowicz and Downum (2021), will be more briefly commented regarding their methods since it applies CNNs (Convolutional Neural Networks), which are a more complex category of ML, known also as Deep Learning, nevertheless there are other aspects in their study that are relevant for this thesis. Pawlowicz and Downum (2021) present an approach to typology using digital images of decorated pottery shards from the Tusayan White Ware tradition (c. AD 825-1300) from Arizona, USA.

According to Pawlowicz and Downum (2021, p. 1-2), one of the issues related to pottery classification is the ability of the analyst to apply consistent and accurate criteria. In some cases, specific classification systems are applied, for instance the 'ware-type-variety' system used in the archaeology of the North-American Southwest. 'Ware' is the broader category, which is based on both technology and decoration features, and it is further subdivided into types and varieties. Between seven and nine types of Tusayan White Ware are recognised, depending on the criteria used, but they broadly reflect different time periods (Pawlowicz & Downum, 2021).

Based on a dataset of 2,407 pottery shards, Pawlowicz and Downum (2021, p. 6-8) present the precision, recall and F_1-Score separated by types, where the F_1-Score varies from 0.394 to 0.899 (average of 0.825 among eight types). They compare these results to the ones provided by four pottery experts, which resulted in overall accuracies (all types included) between 0.736 and 0.869, based on a consensus dataset (the benchmark). Pawlowicz and Downum (2021, p. 6) point out that the accuracy achieved by a deep learning model might be limited by the accuracy of the type labels used to train the model.

The traditional Tusayan White Ware typology system still prevails despite its limitations and after a new attribute-based classification system was proposed. This alternative system was successfully used in correlation with tree-rings to predict site dates with promising results but, despite being more precise, it requires more effort to codify attributes and was not widely adopted by the archaeology community (Pawlowicz & Downum, 2021, p. 3-5). This case exemplifies an additional challenge for the application of ML in archaeology, since alternative systems based on more up-to-date concepts could result in higher performance when submitted to ML algorithms.

6 CONCLUSION

This research developed a ML model to classify archaeological pottery assemblages. Supervised and unsupervised learning methods and algorithms were integrated to concepts of quantitative classification of artefacts, and applied to a dataset of vessels of nine different shape classes from the Bronze Age Northeastern Syria. The importance of distinct types of features for the definition of artefact classes was identified, and the model performance in classification was evaluated through ML metrics. Alternatives classifications provided by clustering analysis were also provided and compared to the original dataset classification.

Based on the research results and the discussion presented in the previous chapters, it is possible to return to the research question and the three sub-questions defined in the introduction chapter. The main research question was defined as 'Which are the benefits and limitations of a machine learning classification model for pottery assemblages'? The following paragraphs first provide the answers to the sub-questions.

Which are the minimum features required to provide a basic classification, and which are additional features that could improve it?

The categorical features in the research dataset were represented by vessel characteristics such as rim orientations and profiles, base types and additional elements (handles, lugs and spouts). The continuous features were represented by absolute and relative vessel measurements. Based on the results provided by the DT classifier and the differences between the first and second training sessions (when some features were excluded), it is clear that the relative measurements, especially the H-Bd (total Height / Belly diameter ratio), which represents the overall vessel shape, and Bd-Nd (Belly diameter / Neck diameter ratio), which identifies the vessel shape between the belly and the neck, were more relevant for the assemblage classification. After these features come the absolute measurements, especially the base diameter and the neck diameter. Among the categorical features, the rim profile was the most relevant. Because of scikit-learn requirements, the categorical features had to be converted in a numerical codification and this resulted in the individual elements of rim profiles and other categorical features to be used, instead of the feature as a unit. That way, the

specific rim_profile[7] (round-folded outside), which is the third most common in the dataset, was considered the most relevant profile for shape identification.

The demonstrated importance of certain specific features is valid for the assemblage used in this research, other assemblages could result in different features having more or less importance. Nevertheless, the overall importance of the relative measurements for identification of vessel shapes certainly applies to other assemblages as well. All continuous features were used in the second training session, after the removal of less relevant features and, while all the categorical features (as a unit, not individually) were used, their importance score was lower than the continuous features scores. The importance of categorical features would likely increase if the additional elements were present in a larger number of samples. Other characteristics of the assemblage and scikit-learn restrictions also should be taken into consideration, as it is the case of features with NA (null) values in part of the samples. The base diameter and the relative measurement Bd-BaD (Belly diameter / Base diameter ratio) fall into this case, and the replacement of null values with zero or other calculated value has the potential to influence in the classification results since they impact on the vessel shape identification.

To conclude this question, an observation about the importance of data quality. An incorrect value for one feature in one sample causes the distortion in the interpretation of the vessel shape and, if this type of error occurs in a significant number of samples, the training of the ML model will be affected and consequently the performance of the model in the identification and classification of new samples.

To what extent can this model replicate classifications made by experts?

The ML model created using the supervised learning Voting Classifier algorithm provided the highest scores in accuracy (0.87) and F_1-Score (0.86) metrics, based on a dataset with 496 samples of pottery vessels. From this dataset, ¾ of the samples were used to train the model and ¼ to test and validate it. This means that the model correctly classified 87% of the samples in the validation dataset, consisting of 124 samples. Other algorithms (Logistic Regression and SVC) provided performances close to VC algorithm. The individual scores for each shape class vary from 0.50 to 0.96, with five out of nine classes returning a score

equal or higher than 0.80 (six out of nine if all algorithms are considered). For the C class, the lower score (0.50) is likely related to the low number of samples and the shape similarity with the E class, which has the largest number of samples. These are just some of the particularities that should be considered in the answer to this question, nevertheless it is reasonable to say that, based on the overall results, the ML model can replicate classifications made by experts with an accuracy of at least 80% in ⅔ of the cases.

There are other issues that could potentially influence these results, either in a positive or negative way. The first one is related to the quality of the dataset and the benchmark provided by the experts. The Arcane database proved to be an excellent source of information on pottery assemblages, providing most of the data required to creating the research dataset, which was complemented by some specific additional measurements. If the original classifications made by experts are consistent, the training part of the supervised learning process will be successful and the ML model performance will be satisfactory in the prediction of new samples' classes. Only a few minor problems were detected regarding the dataset, related to data input errors and potentially different criteria applied in the classification of assemblages of different archaeological sites. The second issue is related to the first: the more homogeneous the assemblage, the greater the probability of better performance in the classification. The choice of using more than one site was made with a trade-off in mind, the need of a minimal amount of good quality samples to make ML viable versus the homogeneity of the assemblage.

Finally, the choice of ML algorithms, their parameters and supporting methods (encoding, imputing of missing values) affect the likelihood of achieving better results. It was fundamental to test different algorithms, parameters variations and different dataset configurations in order to find the best possible combination.

Which other kinds or levels of classification that might be archaeologically relevant can the model suggest?

This is probably the most challenging of the sub-questions, as the answer cannot be compared or measured against existing information, but depends on defining new potential classification structures. The method that was initially associated with this question is unsupervised learning, however, the results of supervised

learning also contributed to it. The grouping of vessels in shape classes is the main classification system developed in the Arcane project and one of the foundations for developing the ML model, but there are also secondary systems such as the grouping in open, closed and miscellaneous shapes and, in some cases, the identification of subclasses in an unstructured way (e.g., cup/mugs and beakers in the G class). Some classes share more similarities with each other (e.g., jars, pots, and diverse classes of bowls). These secondary classification systems were useful in the answer to this question.

The first proposed classification structure (Figure 5.5) is based on supervised learning methods. The main criteria to group the shape classes were the results from the confusion matrices, classes were considered closer to each other when they were related through misclassifications, and also the results from the Decision Tree classifier, which seem in accordance with the concepts of open and closed shapes groups. The E class can be subdivided into two sub-classes, here they received generic names E1 and E2, but one of them is clearly composed of deeper bowls. Three classes, P, R and T were joined in one super class, named ‘Jars’. The ‘T – Flask/Bottle’ class can be considered a type of jar with the most restricted neck. As commented in the previous chapter, this illustration does not aim to show a complete taxonomic structure with all possible sub-classes and super classes; this could be a theme for future research.

The second proposed structure (Figure 5.6) is based on unsupervised learning methods. The first clear difference is the absence of the open and closed groups, since the clustering methods work based on the bottom-up principle, and there are no target classes (and consequently no super classes) that can be used to compare the results. In the case of the bowls and the P-T classes the coincidence between classification and clustering is stronger, while in the other cases there is no obvious relation between these two methods. The association of clusters with the shape classes was made in an exploratory way, using different algorithms and different number of clusters, and the observation in the supervised learning results regarding the completeness of the structure is also valid here. One of the main difficulties is to find meaningful names for the clusters (or super classes) that join the K-P-T and H-N-R classes, for that reason only a generic graphic representation is shown.

Based on the answers and comments to the sub-questions it is now possible to return to the main research question:

Which are the benefits and limitations of a machine learning classification model for pottery assemblages?

1) The **benefits** of a ML classification model for pottery assemblages include:

- The identification of features that have greater relevance for the definition of shape classes. This includes categorical (qualitative) and continuous (absolute and relative measurements) features.
- The identification of shape classes that have greater similarity to each other. This contributes to the elaboration of classification structures such as taxonomic trees and to the understanding of potential relationships among artefacts. A classification structure is fundamental to allow for the development of typologies and to address questions about the peoples and cultures that produced the artefacts.
- An increase in the quality of the artefact classification process carried out by experts, assisting in the revision of data input errors, the identification of potential misclassifications, and facilitating an agreement in the case of divergent opinions.
- The suggestion of new potential grouping of artefacts that were not previously considered through traditional classification criteria.

2) The **limitations** of a ML classification model for pottery assemblages include:

- The dataset has requirements regarding the relationship between the number of samples and the number of different target classes. Classes represented by few samples can make the training process difficult and consequently affect the prediction performance.
- The dataset should include samples that are homogeneous in their archaeological contexts, derived from culturally and chronologically related sites or assemblages.
- The samples, in the case of identification of pottery vessel shapes, should have a minimum level of completeness that allows basic measurements to be taken,

and the identification of certain categorical features (rim profiles, base types) and potential additional elements (handles, spouts).

- Vessels images are not mandatory but can be very useful in validating classification results.

To conclude, a note on future research, in addition to the possibilities already mentioned in the discussion chapter and in this conclusion.

The ML model based on supervised learning created for this research has the potential to be improved through the application of techniques such as feature extraction (to provide additional information about features importance), and through minor adjustments in the dataset (e.g., the checking and correction of certain samples measurements, and a possible revaluation of some samples' classes), and consequent retraining of the model. Such a task would require the participation of experts for validating the results provided by the model and suggesting the adjustments, but in turn it would allow completing a cycle of problem analysis, model training and validation, and increasing knowledge in archaeology. A further possibility would be to work on unclassified assemblages, using ML models as a complementary tool from the beginning of the classification process by experts.

ABSTRACT

Artefact classification is one of the main themes and an important practice since the beginnings of archaeology, while machine learning (ML) became one of the most efficient approaches to increase our knowledge in a number of disciplines. This thesis describes a ML model developed for the classification of pottery assemblages, identifying its benefits and limitations, focusing on the importance of artefacts features for the identification of vessel shape classes, and to what extent this kind of knowledge can be used to replicate classifications made by experts. The research also analyses different classes structures based on the ML model.

The research dataset was based on an assemblage of pottery vessels representing nine shape classes and four archaeological sites from the Bronze Age Northeastern Syria, made available by the Arcane project. The classification methodology was based on principles of quantitative archaeology, using vessel measurements and categorical features, implemented by supervised and unsupervised learning ML algorithms and supporting methods from the scikit-learn and SciPy libraries. The Anaconda platform, the Jupyter notebook environment and ImageJ for image processing complete the main software used through the research.

The research results indicate benefits and limitations in the application of ML models in the classification of pottery assemblages. The limitations are especially related to number of samples versus target classes, the homogeneity of the vessels context in the dataset, and the quality of data available for the samples. The results suggest that a ML model can be useful to experts, assisting in the identification of the most relevant artefact features and similarities among classes of artefacts, as well possible misclassifications, ultimately providing new insights into the classification of pottery assemblages in archaeology.

REFERENCE LIST

Adams, W. Y. & Adams, E. W. (1991). *Archaeological typology and practical reality: a dialectical approach to artifact classification and sorting*. Cambridge: Cambridge University Press.

Alpaydin, E. (2016). *Machine Learning: The New AI*. Cambridge, MA: The MIT Press (MIT Press Essential Knowledge Series).
https://mitpress.mit.edu/books/machine-learning

Anichini, F., Dershowitz, N., Dubbini, N., Gattiglia, G., Itkin, B. & Wolf, L. (2021). The automatic recognition of ceramics from only one photo: The ArchAIDE app. *Journal of Archaeological Science: Reports, 36*. DOI:10.1016/j.jasrep.2020.102788

Barberena, R., Cardillo, M., Lucero, G., le Roux, P. J., Tessone, A., Llano, C., Gasco, A., Marsh, E. J., Nuevo-Delaunay, A., Novellino, P., Frigolé, C., Winocur, D., Benítez, A., Cornejo, L., Falabella, F., Sanhueza, L., Santana Sagredo, F., Troncoso, A., Cortegoso, V., Durán, V. A. & Méndez, C. (2021). Bioavailable Strontium, Human Paleogeography, and Migrations in the Southern Andes: A Machine Learning and GIS Approach. *Frontiers in Ecology and Evolution, 9*(584325). DOI:10.3389/fevo.2021.584325

Barceló, J. A. (1995). Back-propagation algorithms to compute similarity relationships among archaeological artifacts. In J. Wilcock and K. Lockyear, (Eds.), *CAA 93. Computer Applications and Quantitative Methods in Archaeology (BAR International Series, 598)*, 165-176. Oxford: Tempvs Reparatvm. https://proceedings.caaconference.org/files/1993/24_Barcelo_CAA_1993.pdf

Barceló, J. A., Vila, A. & Gibaja, J. (2000). An Application of Neural Networks to Use-Wear Analysis. Some Preliminary Results. In K. Lockyear, T. J. T. Sly and V. Mihailescu-Birliba (Eds.), *CAA 96. Computer Applications and Quantitative Methods in Archaeology (BAR International Series, 845)*, 63-70. Oxford: Archaeopress.
https://proceedings.caaconference.org/paper/09_barcelo_et_al_caa_1996/

Bickler, S. H. (2021). Machine Learning Arrives in Archaeology. *Advances in Archaeological Practice, 9*(2), 186–191. DOI:10.1017/aap.2021.6

Bortolini, E. (2017). Typology and Classification. In A. M. W. Hunt (Ed.), *The Oxford Handbook of Archaeological Ceramic Analysis* (pp. 651-669). Oxford: Oxford University Press [Kindle edition].

Brandsen, A., Lambers, K., Verberne, S. & Wansleeben, M. (2019). User Requirement Solicitation for an Information Retrieval System Applied to Dutch Grey Literature in the Archaeology Domain. *Journal of Computer Applications in Archaeology, 2*(1), 21–30. DOI:10.5334/jcaa.33

Bruce, P., Bruce, A. & Gedeck, P. (2020). *Practical Statistics for Data Scientists* (2nd ed.). Sebastopol, CA: O'Reilly [Kindle edition].

Carbonell, J. G., Michalski, R. S. & Mitchell, T. M. (1983). An overview of Machine Learning. In R. S. Michalski, J. Carbonell and T. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach* (pp. 1-23). Berlin: Springer. DOI:10.1007/978-3-662-12405-5_1

Charalambous, E., Dikomitou-Eliadou, M., Milis, G. M., Mitsis, G. & Eliades, D. G. (2016). An experimental design for the classification of archaeological ceramic data from Cyprus, and the tracing of inter-class relationships. *Journal of Archaeological Science: Reports 7*, 465–471. DOI:10.1016/j.jasrep.2015.08.010

Clarke, D. L. (1978). *Analytical Archaeology*. Second Edition revised by Bob Chapman. New York: Columbia University Press.

Davis, D. S. (2020). Defining what we study: The contribution of machine automation in archaeological research. *Digital Applications in Archaeology and Cultural Heritage, 18*(e00152). DOI:10.1016/j.daach.2020.e00152

Drennan, R. D. (2009). *Statistics for Archaeologists: A Common Sense Approach* (2nd ed.). New York: Springer.

Dunnell, R. C. (1971). *Systematics in Prehistory*. Caldwell, NJ: The Blackburn Press.

Fiorucci, M., Khoroshiltseva, M., Pontil, M., Traviglia, A., Del Bue, A. & James, S. (2020). Machine Learning for Cultural Heritage: A Survey. *Pattern Recognition Letters, 133*, 102-108. DOI:10.1016/j.patrec.2020.02.017

Fletcher, M. & Lock, G. R. (2005). *Digging Numbers: Elementary Statistics for Archaeologists* (2nd ed.). Oxford: Oxford University School of Archaeology.

Géron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow. Concepts, Tools and Techniques to Build Intelligent Systems* (2nd ed.). Canada: O'Reilly [Kindle edition].

Gräslund, B. (2009). *The Birth of Prehistoric Chronology*. New York: Cambridge University Press.

Gualandi, M. L., Gattiglia, G. & Anichini, F. (2021). An Open System for Collection and Automatic Recognition of Pottery through Neural Network Algorithms. *Heritage, 4*, 140-159. DOI:10.3390/heritage4010008

Hörr, C., Lindinger, E. & Brunnett, G. (2014). Machine learning based typology development in archaeology. *ACM Journal on Computing and Cultural Heritage, 7*(1), Article 2. DOI:10.1145/2533988

Hunt, A. W. (2017). Introduction to the Oxford Handbook of Archaeological Ceramic Analysis. In A. M. W. Hunt (Ed.), *The Oxford Handbook of Archaeological Ceramic Analysis* (pp. 3-5). Oxford: Oxford University Press [Kindle edition].

Kipfer, B. A. 2000. *Encyclopedic dictionary of archaeology*. New York: Kluwer Academic/Plenum Publishers.

Klassen, S., Weed, J. & Evans, D. (2018). Semi-supervised machine learning approaches for predicting the chronology of archaeological sites: A case study of temples from medieval Angkor, Cambodia. *PLoS ONE 13*(11), e0205649. DOI:10.1371/journal.pone.0205649

Krieger, A. D. (1944). The Typological Concept. *American Antiquity*, *9*(3), 271-288. DOI:10.2307/275785

Kroeber, A. L. (1940). Statistical Classification. *American Antiquity*, *6*(1), 29-44. DOI:10.2307/275944

Lambers, K., Verschoof-van der Vaart, W. B. & Bourgeois, Q. P. J. (2019). Integrating Remote Sensing, Machine Learning, and Citizen Science in Dutch Archaeological Prospection. *Remote Sensing*, *11*(7), 794. DOI:10.3390/rs11070794

Lipo, C. P., Madsen, M. E. and Dunnell, R. C. (2015). A Theoretically-Sufficient and Computationally-Practical Technique for Deterministic Frequency Seriation. *PLoS ONE, 10*(4), e0124942. DOI:10.1371/journal.pone.0124942

Lyons, M. (2021). Ceramic Fabric Classification of Petrographic Thin Sections with Deep Learning. *Journal of Computer Applications in Archaeology, 4*(1), 188–201. DOI:10.5334/jcaa.75

MacLeod, N. (2018). The quantitative assessment of archaeological artifact groups: Beyond geometric morphometrics. *Quaternary Science Reviews, 201*, 319-348. DOI:10.1016/j.quascirev.2018.08.024

Makridis, M. & Daras, P. (2012). Automatic classification of archaeological pottery sherds. *ACM Journal on Computing and Cultural Heritage*, *5*(4), Article 15. DOI:10.1145/2399180.2399183

Midant-Reynes, B. (2000). The Naqada period. In I. Shaw (Ed.), *The Oxford History of Ancient Egypt* (pp. 44-60). Oxford: Oxford University Press.

Mikhailova, N., Mikhailova, E. & N. Grafeeva, N. (2019). The Application of Clustering Techniques to Group Archaeological Artifacts. In H. Adeli, L. P. Reis, Á. Rocha, and S. Costanzo (Eds.), *New Knowledge in Information Systems and Technologies - Volume 1* (pp. 50-57). (Advances in Intelligent Systems and Computing; Vol. 930). Springer Nature. DOI:10.1007/978-3-030-16181-1_5

Müller, A. C. & Guido, S. (2017). *Introduction to Machine Learning with Pyhton: a Guide to Data Scientists*. Sebastopol, CA: O'Reilly [Kindle edition].

Nelson, E., Hall, J., Randolph-Quinney, P. & Sinclair, A. (2017). Beyond size: The potential of a geometric morphometric analysis of shape and form for the assessment of sex in hand stencils in rock art. *Journal of Archaeological Science, 78*, 202-213. DOI:10.1016/j.jas.2016.11.001

Núñez Jareño, S. J., van Helden, D. P., Mirkes, E. M., Tyukin, I. Y. & Allison, P. M. (2021). Learning from Scarce Information: Using Synthetic Data to Classify Roman Fine Ware Pottery. *Entropy, 23*(1140). DOI:10.3390/e23091140

Oates, J. (2005). Digging Deeper at Tell Brak. *Proceedings of the British Academy, 131*, 1-39.
https://www.thebritishacademy.ac.uk/documents/2009/pba131p001.pdf

O'Brien, M. J. & Lyman, R. L. (1999). *Seriation, Stratigraphy and Index Fossils: The Backbone of Archaeological Dating*. New York: Kluwer Academic / Plenum Publishers.

O'Brien, M. J. & Lyman, R. L. (2003). Style, Function, Transmission: an Introduction. In M. J. O'Brien and R. L. Lyman (Eds.), *Style, Function, Transmission: Evolutionary Archaeological Perspectives* (pp. 1-32). Salt Lake City: The University of Utah Press.

Oonk, S. & Spijker, J. (2015). A supervised machine-learning approach towards geochemical predictive modelling in archaeology. *Journal of Archaeological Science, 59*, 80-88. DOI:10.1016/j.jas.2015.04.002

Orengo, H. A. & Garcia-Molsosa, A. (2019). A brave new world for archaeological survey: Automated machine learning based potsherd detection using high-resolution drone imagery. *Journal of Archaeological Science, 112*(105013). DOI:10.1016/j.jas.2019.105013

Orengo, H. A., Conesa, F. C., Garcia-Molsosa, A., Lobo, A., Green, A. S., Madella, M. & Petrie, C. A. (2020). Automated detection of archaeological mounds using machine-learning classification of multisensor and multitemporal satellite data. *PNAS, 117*(31). DOI:10.1073/pnas.2005583117

Orton, C., Tyers, P. & Vince, A. (1993). *Pottery in Archaeology*. Cambridge Manuals in Archaeology. Cambridge: Cambridge University Press.

Palermo, R. (2019). Imperial impact on a small scale: The site of Tell Barri between the 2nd and 4th c. CE. In *On the Edge of Empires: North Mesopotamia During the Roman Period (2nd–4th c. ce)* (1st ed., pp. 164–189). Routledge. DOI: 10.4324/9781315648255-6

Pawlowicz, L. M. & Downum, C. E. (2021). Applications of deep learning to decorated ceramic typology and classification: A case study using Tusayan White Ware from Northeast Arizona. *Journal of Archaeological Science, 130*(105375). DOI:10.1016/j.jas.2021.105375

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V. & Thirion, B. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830. http://jmlr.org/papers/v12/pedregosa11a.html

Peeples, M. A., & Schachner, G. (2012). Refining correspondence analysis-based ceramic seriation of regional data sets. *Journal of Archaeological Science, 39*(8), 2818-2827. DOI:10.1016/j.jas.2012.04.040

Petrie, W. M. F. (1899). Sequences in Prehistoric Remains. *The Journal of the Anthropological Institute of Great Britain and Ireland, 29*(3/4), 295-301. DOI:10.2307/2843012

Porcic, M. (2013). The goodness of fit and statistical significance of seriation solutions. *Journal of Archaeological Science, 40*(12), 4552-4559. DOI:10.1016/j.jas.2013.07.013

Pruss, A. (2013). A Synopsis of The Euro-Syrian Excavations at Tell Beydar. In D. Bonatz and L. Martin (Eds.), *100 Jahre archäologische Feldforschungen in Nordost-Syrien – eine Bilanz. Schriften der Max Freiherr von Oppenheim-Stiftung, 18* (pp. 133-148). Wiesbaden: Harassowitz. DOI:10.7892/boris.48565

Read, D. W. (2007). *Artifact Classification: A Conceptual and Methodological Approach.* Walnut Creek: Left Coast Press.

Rice, P. M. (1987). *Pottery Analysis: a sourcebook*. Chicago/London: The University of Chicago Press.

Rouse, I. (1960). The Classification of Artifacts in Archaeology. *American Antiquity*, *25*(3), 313-323. DOI:10.2307/277514

Santacreu, D. A., Trias, M. C. & Rosselló, J. G. (2017). Formal Analysis and Typological Classification in the Study of Ancient Pottery. In A. M. W. Hunt (Ed.), *The Oxford Handbook of Archaeological Ceramic Analysis* (pp. 181-199). Oxford: Oxford University Press [Kindle edition].

Spaulding, A. C. (1953). Statistical Techniques for the Discovery of Artifact Types. *American Antiquity, 18*(4), 305-313. DOI:10.2307/277099

van der Maaten, L., Boon, P., Lange, G., Paijmans, H. & Postma, E. (2007). Computer vision and machine learning for archaeology. In J. T. Clark and M. Hagemeister (Eds.), *Digital Discovery. Exploring New Frontiers in Human Heritage. CAA 2006. Computer Applications and Quantitative Methods in Archaeology. Proceedings of the 34th Conference, Fargo, United States, April 2006*, 476-482. Budapest: Archaeolingua.
https://proceedings.caaconference.org/paper/cd49_maaten_et_al_caa2006/

VanderPlas, J. (2017). *Python Data Science Handbook: Essential Tools for Working with Data*. Sebastopol, CA: O'Reilly [Kindle Edition].

VanPool, T. L. & Leonard, R. D. (2011). *Quantitative Analysis in Archaeology*. Chichester: Willey-Blackwell.

Verschoof-van der Vaart, W. B. & Lambers, K. (2021). Applying automated object detection in archaeological practice: A case study from the southern Netherlands. *Archaeological Prospection*, 1–17. DOI:10.1002/arp.1833

Weiss, H. (2013). Tell Leilan and the Dynamics of Social and Environmental Forces across the Mesopotamian Dry-Farming Landscape. In D. Bonatz and L. Martin (Eds.), *100 Jahre archäologische Feldforschungen in Nordost-Syrien - eine Bilanz. Schriften der Max Freiherr von Oppenheim-Stiftung, 18* (pp. 101-115). Wiesbaden: Harrassowitz. DOI:10.7892/boris.48565

Wilcock, J. D. (1999). Getting the Best Fit? 25 Years of Statistical Techniques in Archaeology. In L. Dingwall, S. Exon, V. Gaffney, S. Laflin and M. van Leusen (Eds.), *Archaeology in the Age of the Internet. CAA 97. Computer Applications and Quantitative Methods in Archaeology. Proceedings of the 25th Anniversary*

Conference, University of Birmingham, April 1997, 35-52. Oxford: Archaeopress. https://proceedings.caaconference.org/paper/07_wilcock_caa_1997/

INTERNET PAGES

Anaconda (2021). *Anaconda Individual Edition*. https://www.anaconda.com/products/individual/ Accessed on 31 January 2021.

Arcane (2016). *ARCANE Project*. http://www.arcane.uni tuebingen.de/index.html/ Accessed on 20 January 2021.

Arcane (2016b). *ARCANE Project*. *Published volumes*. http://www.arcane.uni-tuebingen.de/publication.html Accessed on 20 January 2021.

Getty (2004a). *form (general concept)*. Getty Art & Architecture Thesaurus. http://vocab.getty.edu/page/aat/300444970/ Accessed on 29 October 2021.

Getty (2004b). *shape (form attribute)*. Getty Art & Architecture Thesaurus. http://vocab.getty.edu/page/aat/300056273/ Accessed on 29 October 2021.

ImageJ (2021). *ImageJ: Image Processing and Analysis in Java*. https://imagej.nih.gov/ij/index.html Accessed on 4 August 2021.

Jupyter (2021). *The Jupyter Notebook*. https://jupyter.org/ Accessed on 31 January 2021.

OI (2021). *Ancient Near East Site Maps: Syria Site Map*. Oriental Institute, The University of Chicago. https://oi.uchicago.edu/research/computer-laboratory/ancient-near-east-site-maps/ Accessed on 24 February 2021.

Scikit (2021a). *scikit-learn: Machine Learning in Python.* https://scikit-learn.org/stable/ Accessed on 31 January 2021.

Scikit (2021b). *Image denoising using kernel PCA*. scikit-learn.org. https://scikit-learn.org/stable/auto_examples/applications/plot_digits_denoising.html#sphx-glr-auto-examples-applications-plot-digits-denoising-py Accessed on 3 November 2021.

Scikit (2021c). *A demo of K-Means clustering on the handwritten digits data.* scikit-learn.org. https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_digits.html Accessed on 4 November 2021.

Scikit (2021d). *Semi-supervised learning*. scikit-learn.org. https://scikit-learn.org/stable/modules/semi_supervised.html Accessed on 4 November 2021.

Scikit (2021e). *Imputation of missing values*. scikit-learn.org. https://scikit-learn.org/stable/modules/impute.html Accessed on 7 November 2021.

Scikit (2021f). *Encoding categorical features*. scikit-learn.org. https://scikit-learn.org/stable/modules/preprocessing.html#encoding-categorical-features Accessed on 7 November 2021.

Scikit (2021g). *Ensemble methods*. scikit-learn.org. https://scikit-learn.org/stable/modules/ensemble.html#ensemble Accessed on 7 November 2021.

Scikit (2021h). *Tuning the hyper-parameters of an estimator*. scikit-learn.org. https://scikit-learn.org/stable/modules/grid_search.html#grid-search Accessed on 8 December 2021.

Scikit (2021i). *Cross-validation: evaluating estimator performance*. scikit-learn.org. https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation Accessed on 9 December 2021.

Scikit (2021j). *Decision Trees*. scikit-learn.org. https://scikit-learn.org/stable/modules/tree.html Accessed on 20 February 2022.

Scipy (2021a). *SciPy: Fundamental algorithms for scientific computing in Python*. https://scipy.org/ Accessed on 16 November 2021.

Scipy (2021b). *Cluster hierarchy dendrogram*. Scipy.org. https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.dendrogram.html Accessed on 13 December 2021.

Tell Beydar (2016). *Tell Beydar/Nabada*. ECUMS - European Centre for Upper Mesopotamian Studies. http://www.beydar.org/ Accessed on 21 November 2021.

Tell Brak (2013). *Tell Brak: Occupational History*. Tell Brak project. McDonald Institute for Archaeological Research. https://www.tellbrak.mcdonald.cam.ac.uk/occupation.html Accessed on 18 November 2021.

Zenodo (2013). *Zenodo*. European Organization For Nuclear Research and OpenAIRE. https://www.zenodo.org/ Accessed on 15 March 2022.

APPENDICES

APPENDIX A.1 – K-MEANS RESULTS (2-3 CLUSTERS)

Original dataset

Shape	Clusters 0	1	Total
C	10		10
E	182		182
G	96		96
H	7	13	20
K	22		22
N	19	15	34
P	72	16	88
R	14	18	32
T	12		12
Total	434	62	496

Shape	Clusters 0	1	2	Total
C	10			10
E	176		6	182
G	96			96
H	3	4	13	20
K	20		2	22
N	10	7	17	34
P	57	9	22	88
R	8	5	19	32
T	12			12
Total	392	25	79	496

Original dataset reduced (10%)

Shape	Clusters 0	1	Total
C		1	1
E		18	18
G		10	10
H	2		2
K		2	2
N		4	4
P	1	8	9
R	1	2	3
T		1	1
Total	4	46	50

Shape	Clusters 0	1	2	Total
C	1			1
E	17		1	18
G	10			10
H		1	1	2
K	2			2
N	1		3	4
P	8	1		9
R	1		2	3
T	1			1
Total	41	2	7	50

5-6 samples each shape

Shape	Clusters 0	1	Total
C	5		5
E	6		6
G	6		6
H	1	4	5
K	5		5
N	3	3	6
P	5	1	6
R	2	4	6
T	5		5
Total	38	12	50

Shape	Clusters 0	1	2	Total
C	5			5
E	5	1		6
G	6			6
H	1	2	2	5
K	5			5
N	1	5		6
P	4	2		6
R	2	3	1	6
T	5			5
Total	34	13	3	50

APPENDIX A.2 – K-MEANS RESULTS (5 CLUSTERS)

Original dataset

Shape	Clusters 0	1	2	3	4	Total
C		3			7	10
E		56	1	1	124	182
G		72	3		21	96
H	4		4	11	1	20
K		13	8		1	22
N	5	8	7	13	1	34
P	9	40	31	8		88
R	5	7	8	12		32
T		11	1			12
Total	23	210	63	45	155	496

Original dataset reduced (10%)

Shape	Clusters 0	1	2	3	4	Total
C	1					1
E	16		1		1	18
G	8				2	10
H			1	1		2
K	1				1	2
N			3		1	4
P	2	1			6	9
R			2		1	3
T					1	1
Total	28	1	7	1	13	50

5-6 samples each shape

Shape	Clusters 0	1	2	3	4	Total
C		5				5
E	1	5				6
G		6				6
H	2	1	2			5
K		5				5
N	2	1		3		6
P	2	4				6
R	1	1		3	1	6
T		5				5
Total	8	33	2	6	1	50

APPENDIX A.3 – K-MEANS RESULTS (8 CLUSTERS)

Original dataset

	Clusters								
Shape	0	1	2	3	4	5	6	7	Total
C	2						1	7	10
E	40			1			22	119	182
G	59			7				30	96
H			3	3	8	4	2		20
K	10		2	10					22
N	6	5	9	4	10				34
P	28	9	16	29	5		1		88
R	5	4	8	4	11				32
T	10			2					12
Total	160	18	38	60	34	4	26	156	496

Original dataset reduced (10%)

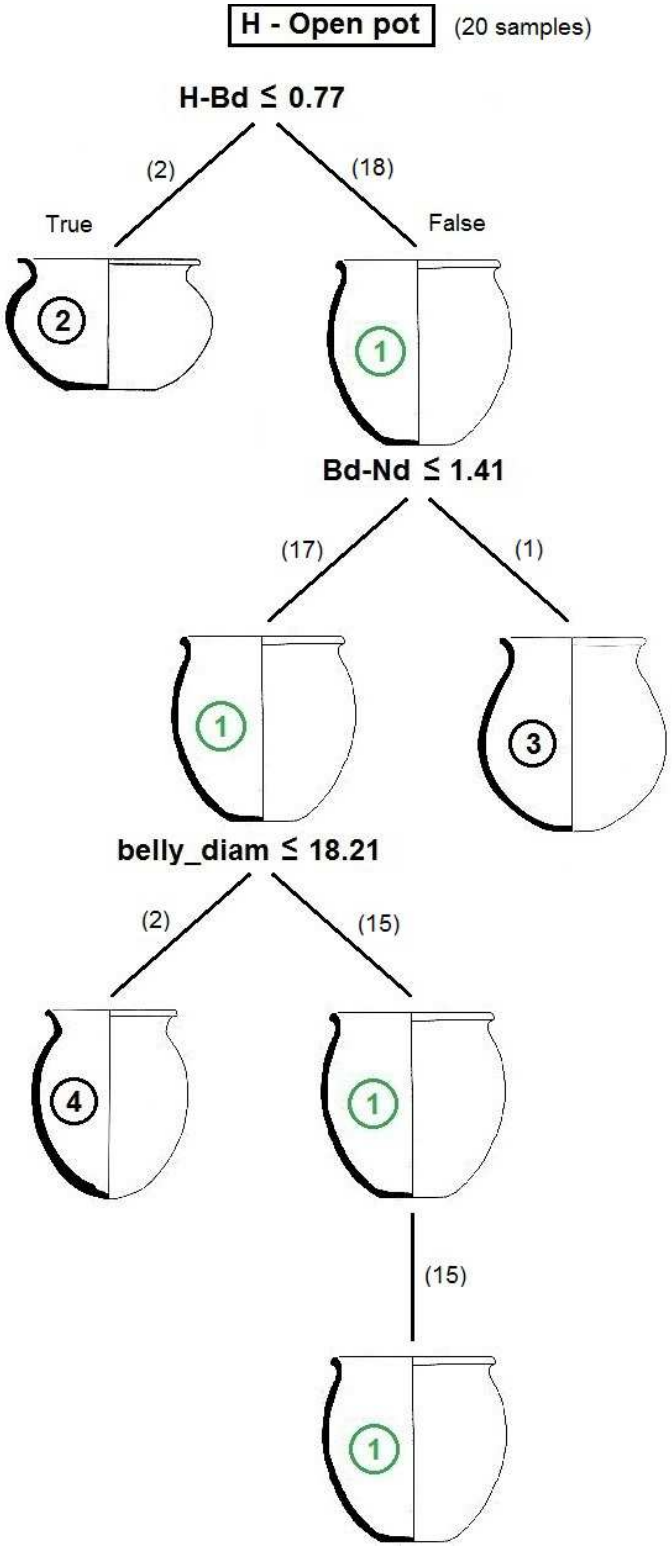
	Clusters								
Shape	0	1	2	3	4	5	6	7	Total
C	1								1
E	12			4	1		1		18
G	7			1	2				10
H		1					1		2
K				1	1				2
N			3		1				4
P	1			1	6	1			9
R			1		1			1	3
T					1				1
Total	21	1	4	7	13	1	2	1	50

5-6 samples each shape

	Clusters								
Shape	0	1	2	3	4	5	6	7	Total
C	1				4				5
E		1			5				6
G	3				3				6
H		1	2			1		1	5
K	2				1			2	5
N	1			3		2			6
P	2					2		2	6
R	1			2		1	1	1	6
T	3							2	5
Total	13	2	2	5	13	6	1	8	50

APPENDIX B.1 – VESSEL TYPES OF THE 'H – OPEN POT' SHAPE CLASS

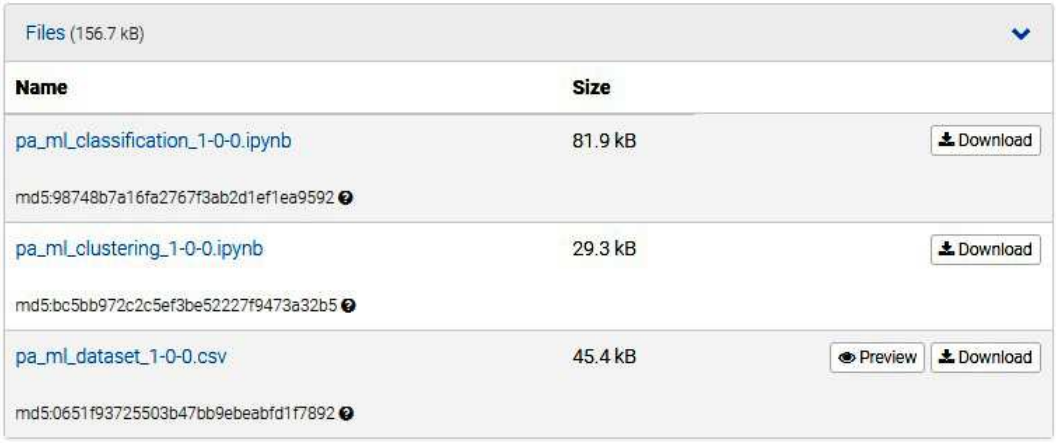
Different types of vessels of the 'H – Open pot' shape class separated according to the Decision Tree Classifier criteria. *Features*: H-Bd = total Height / Belly diameter ratio; Bd-Nd = Belly diameter / Neck diameter ratio; belly_diam = Belly diameter. *Vessels*: 1 = JZ001_P258; 2 = JZ004_P092; 3 = JZ001_P049; 4 = JZ001_P010. Images at different scales. After Arcane (2016).



APPENDIX C.1 – INSTRUCTIONS TO ACCESS THE ML SCRIPTS AND DATASET

The files available for download in the Zenodo repository are the Jupyter notebooks created to run one supervised learning session (Section 4.1.3) and the clustering analysis (Section 4.2.1), and the research dataset.

To access the scripts and the dataset:
https://doi.org/10.5281/zenodo.6368357



The dataset includes the features used in the second training session and the clustering analysis with k-Means, it does not include the features used in the first training session. The first nine features in the dataset were obtained/adapted from the ARCANE database, other features were created for this research. Dataset source: ARCANE Project (Arcane, 2016). Figure: files in Zenodo repository (Zenodo, 2013).