Comparing the Performance of Traditional Statistical Methods and Machine Learning Methods Across Different Complexity Levels in Simulated Data

Verlare, Lisa

Comparing the Performance of Traditional Statistical Methods and Machine Learning Methods Across Different Complexity Levels in Simulated Data

Lisa Verlare

Thesis advisor: Prof.dr. J.J. Goeman

Defended on August 1st, 2024

MASTER THESIS

STATISTICS AND DATA SCIENCE

UNIVERSITEIT LEIDEN

Contents

Abstract

This study explores the circumstances under which traditional statistical methods and machine learning methods perform best. The literature has provided no conclusive answer on when each approach performs best. Instead, many contradicting findings have been reported demonstrating situations from both perspectives. Often sample size plays a role in which approach is recommended as machine learning methods are said to perform better when the data sample is big.

We performed a simulation study, in which we varied several complexity parameters: number of covariates, interactions, interaction depth, regression coefficients, variance of $p(x)$, and formula complexity. Additionally, we reviewed whether sample size and continuous covariates had any bearing on results by reviewing results across different sample sizes and including continuous covariates in combination with binary covariates. To analyze the results, we made use of accuracy, sensitivity, and specificity.

From 138 models, we identified seven general patterns analyzed across different sample sizes: (a) a machine learning method performed best, (b) a traditional statistical method performed best, and (c) mixed performance. We extended the analysis to include more methods from both approaches. For each pattern and performance measure we selected models. This resulted in 20 median models in which not all patterns returned. In a similar analysis on three empirical data sets, similar behavior emerged, although the identification of patterns became more challenging.

Our findings indicate that the variety within each pattern is too great to conclusively identify which complexity parameters produce a particular pattern, although nuances do exist. Moreover, many similar models are spread out across multiple patterns. The identification of patterns has shown that the opposing views in the literature might be explained by the existence of these patterns. We find that traditional statistical methods outperformed complex machine learning methods in several patterns. Furthermore, we determine that sample size is not the sole determinant to select the best approach, as results demonstrate several instances in which traditional statistical methods perform better on larger sample size(s). This adds new insights into how sample size and methods are related.

Keywords: simulation study, complexity, sample size, machine learning, traditional statistics

Chapter 1

Introduction

1.1 Introduction

The ability to accurately predict whether a patient will have any kind of health issue is paramount. When a doctor finds a patient is likely to have dementia or suffer from heart failure, it does not breed confidence if the accuracy is only 50%. A model that supplies such accuracy does not help health professionals in diagnostics, as 50% accuracy is bordering on a coin toss. An accurate model is also very important in other domains besides medicine, such as scientific domains, finance, weather prediction, sales, and product recommendations (Jordan & Mitchell, 2015; Steyerberg, 2019). A company can lose revenue if bad product recommendations are made, as customers will not buy additional products that are recommended if they do not fit their preference. Therefore, a good predictive model is of the utmost importance.

Nowadays, researchers often use machine learning methods instead of traditional statistics, also known as conventional statistics (Shin et al., 2021), as increased data complexity leads them to seek methods that optimize performance and are able to handle complex data (Kokol et al., 2022; Ley et al., 2022; Rajula et al., 2020; Tollenaar & Van Der Heijden, 2013). Both approaches belong to the domain of predictive analytics, a domain within statistical analysis that can be used to make predictions about various items such as new patients, products, and stocks. By using previously provided data and applying methods such as logistic regression or machine learning approaches, a model can be built based on observed patterns in the data and is subsequently able to handle new data points to predict the outcome (Kumar & Garg, 2018; Ley et al., 2022). This means that based on past data, a doctor might be able to more accurately predict whether a patient suffers from a specific disease or a banker is able to predict whether someone will default on a loan. As both approaches are suitable for prediction, the method of choice seems to be more or less the choice of the researcher. Some researchers simply seem to prefer machine learning techniques over traditional statistical methods, as machine learning is known for its high predictive accuracy, or have found that it outperforms traditional statistical methods, whereas other researchers have found that machine learning methods do not perform better than traditional statistical methods. We will explore these different views in the Literature Review of section 1.2. Both approaches have been studied extensively, but there does not seem to be a consensus on which method is suited to which data. General statements are made regarding different aspects, such as complexity and non-linearity. For instance in their introduction to a new hybrid logistic regression model Levy and O'Malley (2020) described scenarios in which one should use logistic regression or machine learning methods (CART or random forest). They stated that when the underlying model is linear, logistic regression is more equipped to identify the true linear decision boundary in comparison to machine learning methods. In contrast, machine learning methods were recommended when the functional form is discontinuous which leads to a nonlinear decision boundary, or when interaction terms are included. They showed that as the strength of the interaction regression coefficient increased, the performance of logistic regression declined. The challenge for researchers is that usually one does not know the true model, which could hinder their ability to select the optimal model. No general guidelines have been established thus far on how to handle the obstacle of selecting the right approach, which might limit researchers in achieving well performing models and accurate predictions.

We want to find under what circumstances traditional statistical methods or machine learning methods perform best. To answer this research question, the objective of this study is to test different methods to explore whether there are differences in performance between the traditional statistical approach and machine learning approach. From both approaches several classification methods are tested on simulated data, to explore the circumstances under which the performance is highest. We will describe general patterns that are present in each performance measure to identify whether certain parameter or model choices always produce a particular pattern. The aim is to vary several of the parameters used for simulating the data, to have differing degrees of complexity within several data sets. The following model-building parameters will be adjusted:

- Number of covariates, defined as the number of main effects;
- Interaction depth and number of interactions, which refer to the depth of interactions included and the number of interactions per level;
- Number of unique regression coefficients, which indicates whether regression coefficients used in the underlying model should be identical, non-unique, or unique;
- The variation of the probability of success used to generate the binary outcome, and
- Formula complexity, which is the complexity of the formula traditional statistical methods require as input.

To further explore under which circumstances each approach performs best, the number of observations will be varied to assess how the number of observations might affect the performance of a method. Initially, analyses will be run on simulated binary covariates with a binary outcome because this is the simplest approach, along with the fact that medical data often contains information stored as binary covariates. To explore whether continuous covariates will influence which approach performs best, we will also run additional models where continuous covariates are included. To investigate whether our findings of the simulations translate to real-life data sets, we will look at several case studies and test whether their results fit findings of the simulation studies.

In the subsequent section, a literature review provides an overview of the current literature and its findings on the traditional statistics versus machine learning techniques, as well as a more detailed description of traditional statistics and machine learning. The set-up used for data simulations, analysis approach, model evaluation, as well as the methods that are considered are described in Chapter 2. In Chapter 3, the results from the simulation studies are provided and Chapter 4 is a case study where real data sets are analyzed to find whether theoretical findings also translate to real-life data. Chapter 5 brings this thesis to a close by providing a discussion of the results as well as possible limitations and implications for future research.

1.2 Literature Review

In this section, an overview of the current literature is provided. First, both traditional statistics and machine learning are briefly clarified. Second, an exploratory overview is given on the current findings regarding the performance of traditional statistics versus machine learning methods.

1.2.1 Traditional Statistical Methods

The term traditional statistical methods refers to regression methods, such as linear regression, logistic regression, penalized logistic regression (ridge, lasso, elastic net), and Cox regression models. In this study traditional statistical methods are used to describe the use of (penalized) logistic regression. These methods are rooted in mathematics, which means certain assumptions about the underlying relationship between predictors and the outcome are made, such as the distribution and linearity (Christodoulou et al., 2019; Frizzell et al., 2017; Ley et al., 2022; Lynam et al., 2020; Zhang et al., 2018). These assumptions are often very strong and require domain knowledge, which is why traditional statistical methods are often described as model driven (Ley et al., 2022). Often named as advantages of traditional statistical methods is the interpretability of the resulting model (Austin et al., 2022; Frizzell et al., 2017; Huang et al., 2022; Ley et al., 2022; Lynam et al., 2020; Rajula et al., 2020; Zhang et al., 2018) and how clear and user-friendly the techniques are (Frizzell et al., 2017; Ley et al., 2022; Lynam et al., 2020; Rajula et al., 2020). Additionally, the statistical approach allows the researcher to use statistical tests to determine whether covariates are statistically significant (Lynam et al., 2020). An important drawback of using traditional statistical methods is the fact that this approach requires human interpretation of which variables are important and which can be discarded during model building. If the researcher does not believe a covariate has a relationship with the outcome or the covariate was not measured, the resulting model could be missing important information about the underlying relationships. This in turn could lead to substandard accuracies (Ley et al., 2022; Rajula et al., 2020).

1.2.2 Machine Learning Methods

Machine learning methods can refer to many methods, well known are random forest and gradient boosting machines. In contrast to traditional statistical models, machine learning methods make no assumptions about underlying relationships between covariates and the outcome (Ley et al., 2022; Panaretos et al., 2018; Rajula et al., 2020; Shin et al., 2021; Zhang et al., 2018). Often this relationship is unknown (Akbilgic & Davis, 2019). They learn from the data provided to build an algorithm that fits as close as possible to the observations without any requirement of linearity or other distributional rules (Ley et al., 2022; Rajkomar et al., 2019; Rajula et al., 2020). Since the focus in machine learning methods lies on generating the most accurate predictions achievable (Ley et al., 2022; Rajula et al., 2020; Tollenaar & Van Der Heijden, 2013), without any consideration for underlying relationships, the methods are often described as data driven (Ley et al., 2022; Zhang et al., 2018).

Often named as advantages of machine learning methods are its flexibility (Rajula et al., 2020; Shin et al., 2021) and its ability to model complex relationships by being able to incorporate both non-linearity and interaction terms into the models without a need to pre-specify this (Akbilgic & Davis, 2019; Ley et al., 2022; Panaretos et al., 2018; Rajula et al., 2020; Tollenaar & Van Der Heijden, 2013), which would be required when modelling a traditional statistical model. Furthermore, machine learning methods are often the method of choice in high-dimensional data situations, as they are often found to perform better when the number of covariates is higher than the number of observations (Feng et al., 2019; Ley et al., 2022; Rajula et al., 2020; Shin et al., 2021; Tollenaar & Van Der Heijden, 2013).

However, studies have also found several drawbacks. First, many machine learning methods often forgo interpretability in favor of maximizing the accuracy of predictions. As this frequently results in models where the inner workings are unclear, the machine learning approach is often labeled to be a ‘black box’ (Austin et al., 2022; Churpek et al., 2016; Huang et al., 2022; Ley et al., 2022; Lynam et al., 2020; Rajula et al., 2020; Senders et al., 2018; Zhang et al., 2018). Second, many studies have noticed that sample size influences whether machine learning methods perform well. Only when enough data is available they can find the complex patterns in the data (Kokol et al., 2022; Ley et al., 2022; Rajkomar et al., 2019; Rajula et al., 2020). Furthermore, machine learning methods are prone to overfitting, which jeopardizes the generalization of the methods to new data (Feng et al., 2019; Rajula et al., 2020).

1.2.3 Traditional Statistics versus Machine Learning

The second point of interest is what current studies have found in regards to the dilemma whether the traditional statistics approach or machine learning approach will perform better.

Simulation Experiments

Some scholars have experimented to find when traditional statistics performs better than machine learning and vice versa. In a recent study, Bailly et al. (2022) have run a similar experiment to this research paper to explore how sample size and complexity might affect the performance of several methods. By using the Framingham study as a basis, they simulated datasets of varying sizes and varying complexity of interactions. They compared the performance of logistic regression, ridge regression, lasso regression, which they designated as machine learning methods but we regard as traditional statistical methods, and neural networks, designated as deep learning, in several experiments. These experiments varied the interaction order within the data as well as the interaction order specified in the regression formulas. They found that (penalized) logistic regression frequently outperformed neural networks when the models were accurately defined, i.e., when interactions that captured the underlying relationships in the data were included. This was true even for higher complexity and regardless of sample size. Moreover, penalized logistic regression methods lasso and ridge frequently outperformed unpenalized logistic regression as they reduced the chances of overfitting. They also concluded that the size of the data did not matter as much as the interactions introduced to the models. In another simulation study, Kirasich et al. (2018) compared the performance of logistic regression and random forest under different circumstances. They argued the existence of No Free Lunch Theorem, which theorizes there is not one algorithm that consistently outperforms other methods (Kuhn & Johnson, 2013). That is why it is important to compare the performance of several methods, as not one method will always result in the highest performance. By running several experiments Kirasich et al. (2018) found that both methods had a similar performance when the number of observations was below 1000, after which the methods deviated more from each other. In addition, they increased the number of covariates systematically and found that above 30 covariates, the accuracy of logistic regression kept improving but random forest no longer did.

Austin et al. (2021) utilized data-generating processes derived from statistical and machine learning methods to simulate binary outcomes, which were analyzed using the same methods as in the data-generating processes, to generate new predictions. They found that often (penalized) logistic regression and boosted trees, i.e., stochastic gradient boosting machines (*gbm),* outperformed other tested methods. A critical note by the authors is that these results could differ if more predictors had been considered and less observations available. A similar analysis is done by Austin et al. (2022) to predict the continuous outcome systolic blood pressure when a patient is released from the hospital. By using different data-generating processes based on statistical and machine learning methods to simulate outcomes for systolic blood pressure, with the help of sampled residuals, and using Monte Carlo simulations, they found that in all their analyses the neural network method underperformed in comparison to the other methods. Moreover, ordinary least squares regression either performed better or similar to the lasso and ridge regression, while boosted trees often performed best.

Applications on Data

Several studies have conducted systematic reviews of the literature. First, using 243 datasets, Couronné et al. (2018) compared the performance of logistic regression to that of the default random forest. They found that, in more than half of the datasets (69%), random forest did better than logistic regression. They also highlighted the fact that in tuned random forest models, the improvement in performance was most prominent in models which had previously, using default settings, performed badly relative to logistic regression. Second, in an extensive literature review, Shin et al. (2021) reviewed 20 studies about prediction models on heart failure patients, using outcomes mortality and readmission. As a whole, they found that the performance of traditional statistical methods lagged behind that of machine learning methods, although they advocated for more externally validated results. Similarly, Patel and Sengupta (2020) looked at the recent literature that used prediction models to predict cardiovascular events. They concluded that in many cases of predicting these events, machine learning methods performed better than traditional statistical methods. Finally, Senders et al. (2018) reviewed 30 studies that used machine learning methods for various neurosurgical purposes. Out of these studies, 7 compared their outcomes to a logistic regression model, i.e., traditional statistical model. In all cases, machine learning methods outperformed the logistic regression model. Using this information, the authors reported accuracy to have an overall median improvement of 15%. Based on these systematic reviews, the literature on comparisons between traditional statistics and machine learning methods is quite extensive. However, frequently the focus is on the application of methods within their own domain. To give a good overview, several examples are discussed.

Within the medical field, several comparisons have been made between traditional statistics and machine learning methods. Several lines of evidence suggest that machine learning methods outperform traditional statistical methods. This is illustrated by Lolak et al. (2023) who compared the model performance on the risk prediction of strokes for patients who are at high-risk. They showed that Extreme Gradient Boosting (XGBoost), a machine learning method, outperformed the traditional statistical methods that were used when comparing the model performance of risk prediction of strokes. Similarly, Panaretos et al. (2018) demonstrated that when predicting cardiometabolic risk, using food and nutrients as predictors, machine learning methods, KNN and random forest, outperformed the traditional statistical method linear regression. In a recent study, Zhang et al. (2018) investigated crash injury severity, which is a topic where traditional statistics is widely used, but machine learning methods are gaining more support. These kind of prediction models are often used to predict how serious a persons injuries might be following a crash, which could help guide hospitals in treatment decisions. The authors found that machine learning, in particular random forest, outperformed traditional statistics, ordered probit and multinomial logit model, that were used in their predictive analysis. This was also observed by Churpek et al. (2016), who sought to predict clinical deterioration in hospital wards by using data collected from multiple centers to predict the illness severity of a patient on the ward. They found that some of the applied machine learning methods, such as random forest, outperformed logistic regression. Other machine learning methods, aside from methods such as random forest, were frequently found by researchers to perform best. Feng et al. (2019) demonstrated that most machine learning methods, in particular several SVM approaches, resulted in better evaluation metrics than logistic regression when predicting the survival after a severe traumatic brain injury. They recommended the usage of machine learning methods in high-dimensional data cases, as the machine learning methods that were tested all performed on par or better than logistic regression. Desai et al. (2020) compared methods that predict several heart failure related outcomes, but found that compared to logistic regression, the improvement in performance of machine learning methods was minimal. When they added predictors from electronical medical records, which were mostly continuous, to the predominantly binary covariates from claims data, they observed improvements in GBM for some outcomes. Finally, in psychiatry predicting suicidal behavior is of grave importance. Conventionally, traditional statistics were used to predict the risk of committing suicide. Unfortunately, these methods yielded only mediocre results, which makes machine learning methods all the more enticing to researchers in this field (Grendas et al., 2022). In their study, Grendas et al. (2022) found that their selected machine learning method, a variable selection variant of random survival forest, made more accurate predictions than the traditional statistical model, Cox regression.

The studies presented thus far provide evidence that there are instances when researchers believe machine learning methods outperform traditional statistics in a meaningful way. However, this is contrasted by the fact there are just as much studies that find that traditional statistics perform similarly to machine learning methods. These are studies in which authors concluded that the performance was similar. In a systematic analysis of literature, Christodoulou et al. (2019) analyzed 71 studies on clinical prediction models in which logistic regression and machine learning methods were compared. They found that machine learning performance was comparable to that of logistic regression in scenarios in which they identified the risk of bias as small, if the risk was high machine learning methods did perform better. A few examples of other illustrations in the literature are studies such as that of Cao et al. (2022). They compared logistic regression and several machine learning methods to identify which approach performed best to predict renal function decline risk, using data spanning 10 years. They found that the gradient boosting model outperformed other methods. While the gradient boosting model also performed better than logistic regression, the difference in performance was not statistically significant. Moreover, Huang et al. (2022) investigated the performance of traditional statistics and machine learning in predicting noncardia gastric cancer. While they found machine learning method KNN had better accuracy and specificity compared to logistic regression, its ability to predict those who do have noncardia gastric cancer, i.e., sensitivity, was much lower. That led them to the conclusion that machine learning methods are comparable to logistic regression when considering all performance criteria. Furthermore, to accurately predict whether a patient had diabetes type 1, Lynam et al. (2020) compared several methods, utilizing only a small set of predictors. They found that logistic regression had a comparable performance to machine learning methods. Similarly, in their search to find a model that could better predict the heart failure readmission risk within 30 days after discharge, Frizzell et al. (2017) showed that none of the machine learning methods applied, such as random forest (C-statistic = 0.61) and GBM (C-statistic = 0.61), greatly improved upon the predictions made by traditional statistical methods, such as logistic regression (C-statistic = 0.62) and LASSO regression (C-statistic=0.62). These contrasting comparisons shows that often different results are found. There does not seem to be one conclusive method that always outperforms other methods in studies that have applied these approaches directly to their data.

From the literature discussed so far on applications on data, we have only illustrated situations in which machine learning methods outperformed traditional statistical methods or when machine learning methods performed similarly to traditional statistical methods according to the researchers. The last situation of interest we illustrate is when researchers state that traditional statistical methods performed better than machine learning methods. De Hond et al. (2022) found that logistic regression performed statistically significantly better (AUC = 0.88) compared to the XGBoost algorithm (AUC = 0.85) when they predicted severe exacerbations of asthma using data gathered through home monitoring. They note that logistic regression often underestimated the risks that were predicted, but attributed this to the small event-rate. Moreover, they suggest that the reason for logistic regression performing better could be due to absence of complexity in the data. Moreover, Hu et al. (2022) predicted delayed cerebral ischema for elderly patients above 60 in the hospital that had a subarachnoid hemorrhage. They found that in external validation LASSO (AUC = 0.894) outperformed machine learning methods such as random forest (AUC = 0.821) and XGBoost (AUC = 0.865). Finally, Sun et al. (2022) investigated whether logistic regression performed better than CART models when predicting community acquired pneumonia in individuals who had been to a doctor for a respiratory tract infection. After having used triangulation with logistic regression, penalized regression, and random forest to apply variable selection, they found that logistic regression (AUC = 0.80) overall performed better than the CART model (AUC = 0.68).

We discussed literature from all perspectives, without finding a definitive conclusion. Sometimes machine learning performed better, while in other cases traditional statistical methods performed better or similar. Note that statements on *similar* performance are subjective for each researcher. For some domains a 0.01 increase in AUC is significant, while other researchers see it as similar. Many studies reporting similar findings would probably fit under this last paragraph, in which traditional statistical methods slightly outperformed machine learning methods.

Chapter 2

Methods

In this chapter a detailed explanation is given on the methods that were used in this study. We clarify the design of the simulation study, our analysis approach and model evaluation. Moreover, we highlight the traditional statistical methods and machine learning methods we have used.

2.1 Simulation Study

In this section the design of the simulation study is explained to make it possible to reproduce. All analyses were performed using R (v4.3.2; R Core Team, 2023). Simulations were run on a laptop with an Intel Core i9 processor and NVIDIA GeForce RTX 4060 graphics card. Appendix B describes how the code that was used can be retrieved.

To identify the circumstances under which traditional statistical methods or machine learning methods perform better, a simulation study was used to investigate predictive performance. In a simulation study, data is generated from which results are obtained. According to Morris et al. (2019), data can be generated in two ways. Either by using a model where the underlying relationships are known and data is generated by drawing from a parametric distribution, or by generating synthetic data from a known data set, where repeated resampling is used to mirror the distributions in the original data. In this study we used the first approach, drawing from known parametric distributions, to create testable scenarios in which model-building parameters were varied, which we further elaborate on in section 2.2.1. Since we had first-hand knowledge of the underlying relationships, we were able to examine if there existed circumstances under which each approach performed better. We also systematically varied the sample size of the training data, which allowed us to observe whether training sample size had a major influence on the performance of the methods. In the remainder of this section, a general description of the underlying data-generating process is given.

2.1.1 Covariates

In this study we primarily generated binary covariate data, as dummy variables were straightforward to code and are frequently used in medical statistics. Examples include whether a person is receiving treatment or administering a specific drug. We specified the number of covariates.

Each binary covariate j was independently generated using a binomial distribution

$$X_j \sim B(N, p = 0.5).$$

N is defined as number of independent trials and p as probability of success for each independent trial (Bruce et al., 2020; Kuhn & Johnson, 2013). To ensure relatively balanced data when interactions were calculated, binary data was transformed to -1 and 1, instead of 0 and 1.

Medical data often contains both binary and continuous covariates. To consider both, we also incorporated the possibility to generate continuous data. The number of continuous covariates could vary between 0 and the maximum number of covariates to be included and was specified beforehand. Each continuous covariate j was independently generated using a standardized normal distribution

$$X_j \sim \mathcal{N}(\mu = 0, \sigma^2 = 1).$$

This distribution generates N random numbers for each independent covariate with mean μ and the variance σ^2. This distribution was selected because the variance and standard deviation are equal to that of the dummy variables used in the binary covariates, for which the specific calculation are described in Appendix A.

2.1.2 Binary Outcome

The outcome variable is often binary in medical statistics (Austin et al., 2021). Examples of possible outcomes include whether a person will be readmitted within a period, has a disease, or if death is likely.

The outcome variable Y was generated using a binomial distribution, for m replications

$$Y_m \sim B(N, p(x)),$$

where $p(x)$ is a vector containing the success probabilities for each individual i, which was created using binary logistic regression. By calculating the linear predictor

$$\eta = \mathbf{X} \cdot k\beta,$$

we were able to calculate the probabilities of success for each observation i using

$$p(x) = \frac{1}{1 + exp(-\eta)}.$$

An additional complexity parameter, k, was applied during this process to control the average variance in $p(x)$. We further elaborate on this in section 2.2.1.

By implementing Monte Carlo simulations to make $Y_1, \ldots, Y_m$ replications, using the same probability vector $p(x)$, m outcomes were repeatedly drawn from the binomial distribution. Using m replications takes into account the random nature of generated Y, as outcomes that were generated using a probability distribution would have different values in a new simulation (Harrison, 2010; Kroese et al., 2014).

2.1.3 Randomness of Simulation

To ensure reproducibility, we used random seeds which allows simulated models to be repeated using our provided code (James et al., 2021; Nunez et al., 2021; Wegmeth et al., 2023). We implemented random seeds because we randomly sampled from various distributions to generate the covariates, outcomes, and regression coefficients. Nunez et al. (2021) argued that methods depending on randomness demonstrate seed-dependent variability. Apart from aspects in methods, such as cross-validation and hyperparameter tuning, which are dependent upon randomness, other aspects in our study also depended on randomness, such as the aforementioned probability distributions, formula to generate data, and sampling of n. Running the experiment with only one random seed would not be enough to lessen randomness and subsequent variability in the estimates, because the random seed used might actually select outliers for some components. This could skew the results to either bad or good outlying performances which are misleading (Nunez et al., 2021; Wegmeth et al., 2023). This is also the reason why we generated m replications within one seed, to take into account the randomness of each seed. However, using only one seed does not ensure results are robust and generalizable. Results that are both robust and generalizable can be obtained by using multiple random seeds to give a more comprehensive picture of the results (Nunez et al., 2021; Wegmeth et al., 2023). Results were less likely to be affected by a random seed that produced outlying performance measures, which decreased variance due to randomness.

We have specified the number of seeds and replications for the analyses we have run. The different analyses, exploratory and in-depth, are further clarified in section 2.2.2. Different seeds were assigned to demonstrate that in separate analyses of the same model-building parameters, with different random conditions, results were robust. The exploratory analysis used seed 1 through 5 and 10 replications of the outcome per seed. We used seeds 6 through 8 for the in-depth analysis and 5 replications of the outcome per seed. Less random seeds were specified for the in-depth analysis as computational time increased due to the inclusion of more methods.

2.2 Analysis Approach

In this section the analysis approach is discussed. First, different complexity levels are explained. Second, the analyses and models are specified. Finally, we clarify how the observations are handled and how we use cross-validation.

2.2.1 Complexity Levels

To identify the circumstances under which traditional statistical methods or machine learning methods performed best, we looked at data with differing complexity levels. When data was generated, several parameters could be varied to decrease or increase the complexity level.

Number of Covariates

The number of covariates is the number of main effects that was used to generate the data. A higher number of covariates is often associated with higher complexity. Not only are there more possible interactions to evaluate (Ley et al., 2022; Li et al., 2022), by increasing the number of covariates the data dimensionality will also increase and data will become sparser (Altman & Krzywinski, 2018). The relationships within the data will become more difficult to find.

Interaction Depth and Number of Interactions

The interaction depth is the order that interactions are allowed to have (James et al., 2021; Kuhn & Johnson, 2013) and could easily be increased. Interaction depth 1 refers to main effects only and interaction depth 2 to both the main effect and first order interactions.

The number of interactions is sampled from all possible interaction combinations at each depth level. This is specified separately for each interaction depth level. Higher interactions increase the complexity of the data, which introduces a more intricate pattern and relationship between the covariates and outcome (Bailly et al., 2022), as we are no longer dealing with a linear decision boundary. Instead, the decision boundary that methods will need to identify becomes more non-linear and complex as more interactions are added (Levy & O'Malley, 2020). Thus, the larger the number of interactions, the higher the complexity of the data will be.

We clarify these parameters with the following illustration. Three main effects and two first-order interaction effects are specified, i.e., interaction depth 2. Three main effects are x_1, x_2, x_3. From possible interactions at this depth level: $x1 \times x2, x1 \times x3$, and $x2 \times x3$, only 2 interactions will be sampled to generate the data.

Number of Unique Regression Coefficients

To generate the data, the underlying model requires regression coefficients, β. The number of unique coefficients can be varied to influence the complexity level and are specified separately for each complexity level. We explored different situations, namely identical, unique, or non-unique, i.e., using only 2 or 5 different regression coefficients per depth level. The number of unique regression coefficients is considered a complexity parameter, as the influence coefficients have on the outcome plays a large role. Identical regression coefficients would mean covariates have an equal influence on the outcome, whereas unique (or non-unique) regression coefficients have varying effects on the outcome parameter. We found no specific literature on how this specific situation would affect complexity. We can imagine two different perspectives. First, unique regression coefficients could simplify the identification of large effects, as smaller effects have only a minor influence on the outcome. This implies a lower data complexity. Second, unique regression coefficients make relationships in the data more distinctive and create more intricate patterns that influence the outcome. This would suggest higher data complexity. Conversely, identical regression coefficients could affect data complexity in the same manner. The regression coefficients, β, were randomly generated for each interaction depth d using a Uniform distribution:

$$\beta_{b,v} = U(0,1),$$

where b is the number of unique regression coefficients to generate and v is the number of covariates at interaction depth d. Random generation of the regression coefficients was implemented during the data generation process, for each specified seed.

Continuing the example, for $\beta_{3,3}$, at $d = 1$, three unique regression coefficients need to be generated for the three main effects. Each covariate gets a unique regression coefficient. However, if we specify $\beta_{1,2}$ for $d = 2$, only one regression coefficient needs to be generated, while there are two interactions. The regression coefficients at $d = 2$ are identical.

Variance of the Probability of Success controlling Outcome Variable y

The outcome variable Y was generated using a binomial distribution, which required a vector $p(x)$, i.e., the probabilities of success for each individual. We used a complexity measure to control the average variance (δ) of $p(x)$, which could range between 0 and 0.25, as demonstrated in Appendix A. In this study δ could take values: $0.05, 0.10, 0.20$, where 0.10 was the default. We used a bisection algorithm to identify a constant, k, which scaled the regression coefficients, β, such that the resulting average variance in $p(x)$ matched the specified one. The specific process to find k is described in Algorithm 1. Upon finding k, steps 2 and 3 are repeated to definitively calculate $p(x)$. By varying the variance, we introduced different levels of uncertainty, i.e., noise, to the data. Garcia et al. (2015) described how the complexity of classification tasks can be amplified due to the inclusion of noise to outcomes. As this complexity increases, patterns become more challenging to differentiate, effectively increasing the data complexity. Classification methods would need to identify a decision boundary that was more complex than before additional noise was added.

Formula Complexity

Traditional statistical methods, such as logistic regression, require a formula specifying which covariates and interactions to use. Every effect needs to be manually included. In contrast, machine learning methods only require specification of the main effects. In this study, we used

Algorithm 1 Scaling the Probability of Success using a Bisection Algorithm

```
Input Parameters: Lower bound $a = 0$, upper bound $b = 1000$, target variance $\delta$, covariates $\mathbf{X}$, regression coefficients $\beta$
Starting values: Average variance $\Delta = \infty$, tolerance $\epsilon = 0.001$
while $\Delta > \epsilon$:
    1. Calculate the midpoint of the lower bound and upper bound:
           $c = \frac{a+b}{2}.$
    2. Calculate the linear predictor, using the calculated midpoint:
           $\eta = \mathbf{X} \cdot k\beta,$
       where $\mathbf{X}$ is the matrix containing the covariates, $k$ is the constant, and $\beta$ are the regression coefficients for each covariate.
    3. Calculate the logistic function
           $p(x) = \frac{1}{1 + exp(-\eta)}.$
       This will result in a vector $p(x)$ that contains the predicted probability of success for each observation, which for each individual $i$ can be denoted as $p_i$.
    4. Calculate the average variance:
           $\text{Average variance} = \Delta = \frac{1}{n}\sum_{i=1}^{n} p_i(1 - p_i).$
    5. Calculate the value, $f$, for the equation $f(x)$, this should equal 0 for the bisection algorithm to be a success:
           $f(x) = \Delta - \delta,$
       where $\delta$ represents the target variance for the complexity level.
    6. The following conditions are checked:
           if $f(x) = 0$: Return midpoint value $c$ as final constant $k$.
           else if $f(x) < 0$: Replace upper bound value, $b$, by the midpoint value $c$.
           else if $f(x) > 0$: Replace the lower bound value, $a$, by the midpoint value $c$.
```

formula complexity to denote the formula for traditional statistical methods. It used the same scale as interaction depth. We made no distinction between which covariates were added to the formula. All possible effects belonging to an interaction depth were included. The formula complexity ranged between 1 and 3. We included this parameter because it is the researcher's choice to include effects for traditional statistical methods. Bailly et al. (2022) showed that often regression models performed best if they had well-specified interactions. Hence, we would expect that by increasing the complexity of the formula by specifying higher interactions leads to a better performance as the (traditional statistical) models know to look for those effects as well.

Continuing the example, formula complexity 1 has formula: $x_1 + x_2 + x_3$, while the formula for formula complexity 2 is $x_1 + x_2 + x_3 + x1 \times x2 + x1 \times x3 + x2 \times x3$.

2.2.2 Analyses

Our analyses began with an exploratory analysis, as it was not possible to run all methods for every scenario, due to computational costs. In the exploratory analysis we ran many scenarios, using as few methods as possible. Based on measured computational times, we selected methods with the shortest computing time: logistic regression and LASSO regression for traditional statistical approaches, and k-nearest neighbors and random forest for machine learning approaches. All tested scenarios are described in Tables 1 – 6. Additionally, we reviewed median models in an in-depth analysis where all methods described in section 2.4 were used. They were selected from each pattern, identified in the exploratory analysis, for each performance measure by sorting all models within a pattern by the interaction depth as most important criteria. If the number of models was even, we chose one of the two options. Each tested model is described in Table 9.

2.2.3 Sample Sizes

To investigate whether the sample size could be a factor in selecting an approach under different circumstances, we included different sample sizes during the model training phase. We hoped to observe whether a small or large sample size determined the method to be selected. In a comparison between modern modelling methods (random forest, SVM, neural nets), also known as machine learning methods, and classical methods (logistic regression and CART), Van Der Ploeg et al. (2014) showed that logistic regression needed less events per variable than the machine learning methods to deliver a stable AUC value. They argued that machine learning methods should only be used when the sample size is large and has a high number of events per variable as that will result in (possible) higher performance and stable results. They favored using logistic regression when the sample size is small. Cui and Gong (2018), in their research of individualized cognition predictions, demonstrated an exponential increase in both accuracy and the stability of the accuracy estimates when sample size increased. This was true for both the machine learning methods, linear support vector regression and relevance vector regression, and the traditional statistical methods, which the authors considered machine learning methods but we categorized as traditional statistical methods: LASSO, ridge, and elastic net regression. Additionally, they discussed overfitting as a possibility if machine learning methods are applied on small data sets, which would impede the generalization ability of the model, as well as the fact that small data sets will only show a fraction of the patterns. These studies show that traditional statistical methods are often recommended for small sample sizes, while machine learning methods need a larger sample size. However, these studies used existing data, where the underlying structure was unknown to the researchers. By simulating the data, we expect to find that under certain circumstances these views are not always valid.

For both exploratory and in-depth analysis we used the same process. To evaluate each model, we independently generated a training and test data set according to the process described in section 2.1. Training and test data was generated once in every seed. Training data was

Table 1

Exploratory Simulation Studies: Number of Covariates, Interaction Terms and Depth

	Models																															
	E-1	E-2	E-3	E-4	E-5	E-6	E-7	E-8	E-9	E-10	E-11	E-12	E-13	E-14	E-15	E-16	E-17	E-18	E-19	E-20	E-21	E-22	E-23	E-24	E-25	E-26	E-27	E-28	E-29	E-30	E-31	E-32
Number of covariates																																
Main effects	2	10	20	30	40	2	10	20	30	40	30	30	30	30	30	10	10	10	10	10	10	30	30	30	50	5	5	10	15	50	20	20
First order interactions						1	1	1	1	1	2	3	4	5	30	3	5	10	20	40	5	1	5	100	1	5	2	8	15	15	1	10
Second order interactions																					3	1	3	100	1	5	10	6	10	10	1	1
Third order interactions																									1	5	2	4	10	10	1	20
Fourth order interactions																										1	1	2	10	10	1	1
Fifth order interactions																															1	1
Sixth order interactions																																1

Note. This table illustrates the different simulated models that were run, where only the number of covariates was varied. Complexity model-building parameters were fixed at: $\delta = 0.10$, formula complexity = 1, and unique regression coefficients β. For each interaction depth the number of main effects or number of interaction effects is specified in the table. Abbreviation E denotes models of the exploratory analysis.

Table 2

Exploratory Simulation Studies: Formula Complexity

	Models																																	
	E-33	E-34	E-35	E-36	E-37	E-38	E-39	E-40	E-41	E-42	E-43	E-44	E-45	E-46	E-47	E-48	E-49	E-50	E-51	E-52	E-53	E-54	E-55	E-56	E-57	E-58	E-59	E-60	E-61	E-62	E-63	E-64	E-65	E-66
Number of covariates																																		
Main effects	10	10	20	20	30	40	2	10	10	20	30	40	10	10	10	10	10	10	10	30	30	50	5	5	5	5	10	10	15	15	50	20	20	20
First order interactions							1	1	1	1	1	1	5	10	20	40	40	5	5	5	100	1	5	5	2	2	8	8	15	15	15	1	10	10
Second order interactions																		3	3	3	100	1	5	5	10	10	6	6	10	10	10	1	1	1
Third order interactions																						1	5	5	2	2	4	4	10	10	10	1	20	20
Fourth order interactions																							1	1	1	1	2	2	10	10	10	1	1	1
Fifth order interactions																																1	1	1
Sixth order interactions																																	1	1
Formula Complexity	2	3	2	3	2	2	2	2	3	2	2	2	2	2	2	2	3	2	3	2	2	2	2	3	2	3	2	3	2	3	2	2	2	3

Note. This table illustrates the different simulated models that were run, where only the formula complexity was varied. Complexity model-building parameters were fixed at: $\delta = 0.10$ and unique regression coefficients β. For each interaction depth the number of main effects or number of interaction effects is specified in the table. Abbreviation E denotes models of the exploratory analysis.

Table 3

Exploratory Simulation Studies: Unique Regression Coefficients

	Models																					
	E-67	E-68	E-69	E-70	E-71	E-72	E-73	E-74	E-75	E-76	E-77	E-78	E-79	E-80	E-81	E-82	E-83	E-84	E-85	E-86	E-87	E-88
Number of covariates																						
Main effects	10	10	20	20	2	10	10	20	20	30	10	10	10	10	10	10	10	10	10	10	30	15
First order interactions					1	1	1	1	1	1	5	5	5	10	10	10	5	5	5	5	5	15
Second order interactions																	3	3	3	3	3	10
Third order interactions																						10
Fourth order interactions																						10
Number of regression coefficients																						
Main effects	1	5	1	5	1	1	5	1	10	1	1	2	5	1	2	5	1	2	5	5	1	1
First order interactions					1	1	1	1	1	1	1	2	5	1	2	5	1	2	3	5	1	1
Second order interactions																	1	2	3	3	1	1
Third order interactions																						1
Fourth order interactions																						1

Note. This table illustrates the different simulated models that were run, where the number of unique regression coefficients was varied. Complexity model-building parameters are fixed at: $\delta = 0.10$ and formula complexity = 1. For each interaction depth the number of main effects or number of interaction effects is specified, as well as the number of unique regression coefficients. Abbreviation E denotes models of the exploratory analysis.

Table 4

Exploratory Simulation Studies: Average Variance of p

	E-89	E-90	E-91	E-92	E-93	E-94	E-95	E-96	E-97	E-98	E-99	E-100
Number of covariates												
Main effects	10	10	10	10	10	10	10	10	30	30	15	15
First order interactions			1	1	10	10	5	5	5	5	15	15
Second order interactions							3	3	3	3	10	10
Third order interactions											10	10
Fourth order interactions											10	10
Variance in the outcome	0.05	0.20	0.05	0.20	0.05	0.20	0.05	0.20	0.05	0.20	0.05	0.20

Note. This table illustrates the different simulated models that were run, where the average variance of the probability of success that determines the outcome y was varied. Complexity model-building parameters are fixed at: formula complexity = 1 and unique regression coefficients β. For each interaction depth the number of main effects or number of interaction effects is specified in the table. Abbreviation E denotes models of the exploratory analysis.

Table 5

Exploratory Simulation Studies: Combinations with Higher Formula Complexity

	Models														
	E-101	E-102	E-103	E-104	E-105	E-106	E-107	E-108	E-109	E-110	E-111	E-112	E-113	E-114	E-115
Number of covariates															
Main effects	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
First order interactions	10	10	10	10	10	10	10	5	5	5	5	5	5	5	5
Second order interactions								3	3	3	3	3	3	3	3
Third order interactions															
Fourth order interactions															
Number of regression coefficients															
Main effects	1	1	1	2	5	5	5	1	1	1	2	5	5	5	5
First order interactions	1	1	1	2	5	5	5	1	1	1	2	3	5	5	5
Second order interactions								1	1	1	2	3	3	3	3
Third order interactions															
Fourth order interactions															
Variance in the outcome	0.10	0.05	0.20	0.10	0.10	0.05	0.20	0.10	0.05	0.20	0.10	0.10	0.10	0.05	0.20

Note. This table illustrates the different simulated models that were run, using different combinations of complexity parameters. Complexity model-building parameter formula complexity is fixed at 2. For each interaction depth the number of main effects or number of interaction effects is specified, as well as the number of unique regression coefficients. Abbreviation E denotes models of the exploratory analysis.

Table 6

Exploratory Simulation Studies: Continuous Covariates

	Models																						
	E-116	E-117	E-118	E-119	E-120	E-121	E-122	E-123	E-124	E-125	E-126	E-127	E-128	E-129	E-130	E-131	E-132	E-133	E-134	E-135	E-136	E-137	E-138
Number of covariates																							
Main effects	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	5	5	5	5	15	15	15
First order interactions			5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	15	15	15
Second order interactions			3	3	3	3	3	3	3	3	3	3	3	3	3	3	5	5	5	5	10	10	10
Third order interactions																	5	5	5	5	10	10	10
Fourth order interactions																	1	1	1	1	10	10	10
Number of regression coefficients																							
Main effects	10	10	5	5	5	10	10	10	10	10	10	10	10	10	10	10	5	5	5	5	15	15	15
First order interactions			3	3	3	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	15	15	15
Second order interactions			3	3	3	3	3	3	3	3	3	3	3	3	3	3	5	5	5	5	10	10	10
Third order interactions																	5	5	5	5	10	10	10
Fourth order interactions																	1	1	1	1	10	10	10
Variance in the outcome	0.10	0.10	0.10	0.10	0.10	0.10	0.05	0.20	0.10	0.05	0.20	0.10	0.05	0.20	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
Number of continuous covariates	10	10	1	5	10	1	1	1	5	5	5	10	10	10	10	10	1	3	5	5	1	5	15
Formula Complexity	1	2	1	1	1	1	1	1	1	1	1	1	1	1	2	3	1	1	1	2	1	1	1

Note. This table illustrates the different simulated models that were run when continuous covariates were included. No complexity parameters were fixed. For each interaction depth the number of main effects or number of interaction effects is specified, as well as the number of unique regression coefficients. Abbreviation E denotes models of the exploratory analysis.

generated with $N = 50,000$ observations, from which during the training process observations were randomly sampled for each sample size n. The test data was generated with $N = 2,000$ observations, on which no random sampling took place. Regardless of the sample size n that was used during the model training phase, the test set remained the same size throughout the evaluation process. Using a test set matching the training sample size n would not accurately depict how models performed, because a small test set would only reflect a portion of the data. That is why we decided on a relatively large representative test data set. This allowed us to compare how models that were trained on sample size n, which gradually increased in size, performed on the same test data.

In the analyses we used observations on a x^2 scale, where x is an integer number. In the exploratory analysis, x consisted of integers in the interval: $x = [4, 6, \ldots, 42, 44]$ and in the in-depth analysis, the range reduced to $x = [4, 8, \ldots, 40, 44]$ to accommodate for the fact that including additional methods is computationally more expensive. While the simulated data within each seed contained N rows total, during the analysis n rows were randomly sampled according to each sample size, m times.

2.2.4 Hyperparameter Tuning and Cross-Validation

Once a random subsample n was selected within a seed, m models needed to be trained. Models often have default parameters, which might not lead to the most optimal results. Therefore, we used hyperparameters, which can influence the model complexity. The wrong choice of parameters can either overfit or underfit the data (Kuhn & Johnson, 2013). To find the best-fitting model for data generated in each seed, i.e., the optimal model, we have used hyperparameter tuning in combination with 10-fold cross-validation. Cross-validation allowed us to select the model and hyperparameter(s) that, on average, performed best across 10-folds (James et al., 2021). This model was used to predict outcomes for the test data. The hyperparameters to be tuned differed for each method and can be found in Table 7. In most methods we used a random search to specify the hyperparameter grids, as it made no difference whether we manually specified them or if they were randomly selected. The only exception was the gradient boosting model, where a manually defined grid was used, as a random search was not possible, and in the penalized regression models, which used a model-specified grid.

2.3 Model Evaluation

In this section we highlight each performance measure used. The measure of interest depends on the researcher's aim and whether it is costlier to misdiagnose someone or to fail to diagnose someone (Mallett et al., 2012; Naidu et al., 2023; Van Stralen et al., 2009). That is why, instead of looking at only one performance measure, multiple measures should be considered at the same time (Naidu et al., 2023; Van Stralen et al., 2009). As we used both traditional statistical methods and machine learning methods, we considered measures researchers from both approaches are familiar with: accuracy, sensitivity, and specificity.

A measure we considered and disregarded was the area under the ROC curve (AUC), which gives the classification performance over all possible probability thresholds which retains more information from the model outcomes (James et al., 2021). Often, the probability values that methods produce, such as logistic regression, are more informative than class labels, which are formed by assigning '1' when $p > 0.5$ and '0' when $p \leq 0.5$ (Bruce et al., 2020). However, not all methods provide these probabilities, such as SVM models. For this reason, all outcomes were predicted directly as class labels. In addition, as only one threshold was considered, only one point with a corresponding sensitivity and specificity value was provided. Theoretically, we could consider the (test) AUC value equal to the (test) accuracy estimate, which was possible, because the data was generated to be relatively balanced. In Appendix A we demonstrated this.

Accuracy

Accuracy is an overall performance measure, which considers all observations that were correctly classified (Bruce et al., 2020; Naidu et al., 2023).

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

A limitation of accuracy is that imbalanced data could distort the accuracy estimate, as the accuracy will be biased towards the majority class (Chawla, 2005; Naidu et al., 2023; Van Stralen et al., 2009).

Sensitivity

Sensitivity calculates the true positive rate, i.e., how many events are correctly predicted as positive (Bruce et al., 2020; Naidu et al., 2023; Van Stralen et al., 2009).

$$\text{Sensitivity} = \frac{TP}{TP + FN}.$$

Sensitivity is often the focus of studies, especially in medical statistics. For instance, we might be interested to know whether someone has a disease. It is important for medical statisticians to know the sensitivity, as a low value could indicate it might not even be worth the cost of investing resources into a model. However, often this class consists of few observations.

Specificity

Specificity is also known as the true negative rate, which calculates how many events are correctly predicted as negative (Bruce et al., 2020; Naidu et al., 2023).

$$\text{Specificity} = \frac{TN}{TN + FP}.$$

The negative event is often the majority class and of less interest in medical statistics. Nevertheless, the measure is still useful as it could show how well the model is performing.

2.3.1 General Evaluation of Performance Measures

To calculate the final performance estimate for either accuracy, sensitivity, or specificity a general approach was taken. For each sample size n all methods were evaluated. Within each method m replications were evaluated, which resulted in m performance measures for each seed s. To estimate the average of each performance measure in a single seed, we calculated:

$$\text{Performance}_s = \frac{1}{m}\sum_{i=1}^{m} Performance_i,$$

where $Performance_i$ can be any performance measure selected. This will produce, for each seed s, average performance measures for each method and sample size n.

To take into account the inherent randomness due to random seeds and improve generalizability, s performance measures were aggregated into one average result per sample size n and method:

$$\text{Overall Performance} = \frac{1}{s}\sum_{i=1}^{s} Performance_i.$$

2.4 Methods

In this section we describe each method that was used. These methods were chosen because they are often used in the literature and we are familiar with their concepts. Hyperparameters that were used to obtain the optimal model are described in Table 7. If we used a random search we specified the number of random hyperparameter combinations to be analyzed. The number of combinations that were explored was moderate, as many methods were computationally expensive to run and took more time as the training sample size increased. During the training process several methods occasionally issued warnings, such as prediction from a rank-deficient fit, missing values in the resampled performance measures, too few observations per class in a fold, lack of convergence or perfect separation, and no variance in a covariate. While we acknowledged these warnings, we continued with the analyses as they were just part of the process.

2.4.1 Null Model

The null model is the simplest model possible, i.e., the intercept-only model, which functioned as the baseline model to which all other methods were compared. The model always predicts the majority class in the training model to be the outcome. Other methods should perform better than or equal to this model (Peng et al., 2002). In R, the package *caret* (v6.0-94; Kuhn, 2008) was used to train the null model using 10-fold cross validation.

2.4.2 Traditional Statistical Methods

In this section we explain the different traditional statistical methods used in our analysis.

Logistic Regression

Logistic regression is the standard statistical model used when the outcome is binary and we are interested in modelling the probability that the outcome belongs to one of the classes (James et al., 2021). The class of interest in the data is 1, the positive class. The logistic regression uses the following function to model these probabilities:

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_j X_j}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_j X_j}},$$

where j is the number of covariates.

In R, the package *caret* (v6.0-94; Kuhn, 2008) was used to train the logistic model with 10-fold cross-validation. In this study logistic regression included all covariates belonging to the specified interaction depth level. This could lead to a more complex model than necessary, especially when interactions were involved. This could result in models that performed well on training data, but when predicting the outcome for the test data, performance was poor, indicating overfitting. In this situation, the variance of the model is quite high, which means a model will not generalize well on new data. Thus, the higher the complexity of a model, the higher the chances of overfitting are (James et al., 2021).

LASSO, Ridge, and Elastic Net Regression

Regularization methods are applied to reduce the chance of overfitting. These methods decrease the variance of the model, which will improve model performance. Consequently, the bias of estimates will increase (James et al., 2021; Kuhn & Johnson, 2013). In regularization methods a shrinkage penalty is added to the loss function of the logistic regression, with the goal to minimize this (Hastie et al., 2023; Kuhn & Johnson, 2013):

$$\min_{\beta_0,\beta\in\mathbb{R}^{j+1}} -\left[\frac{1}{N}\sum_{i=1}^{N} y_i\cdot(\beta_0+X_i\beta)-\ln(1+e^{\beta_0+X_i\beta})\right]+\lambda\left(\frac{(1-\alpha)}{2}\sum_{j=1}^{p}\beta_j^2+\alpha\sum_{j=1}^{p}|\beta_j|\right),$$

where α determines the weight given to the penalties and λ regulates how much effect the penalty has on the model coefficients (James et al., 2021; Kuhn & Johnson, 2013).

LASSO regression is specified by $\alpha = 1$. Coefficient estimates are shrunk to zero by the penalty parameter and some estimates are removed by equaling the coefficients to zero, if λ is large enough. This makes LASSO regression a feature selection method (James et al., 2021; Kuhn & Johnson, 2013). A drawback is that in a subgroup of covariates that are (highly) correlated, one will arbitrarily be picked, without considering whether that covariate is the best. Moreover, LASSO can pick no more than n covariates in a high-dimensional scenario (Zou & Hastie, 2005). Ridge regression is specified by $\alpha = 0$, where the coefficient estimates are shrunk by the penalty parameter λ to move in the direction of zero, without removing any from the model. It has a tendency to shrink the coefficients of (highly) correlated covariates to similar values (James et al., 2021). Both approaches attempt to improve the generalizability of the model, by using a shrinkage parameter.

Elastic net regression combines ridge and LASSO regression in one model. Kuhn and Johnson (2013) cite the regularization capabilities of the ridge penalty and variable selection approach of the LASSO penalty as a combination that will especially perform well if many variables have high correlations. In contrast to LASSO regression, elastic net will not choose one of the highly correlated covariates and discard others. Instead, it shrinks correlated covariates together and selects correlated covariates as a group (Zou & Hastie, 2005).

Several factors should be considered when choosing a method. James et al. (2021) recommended LASSO regression when only some covariates have large regression coefficients, while ridge regression should be used if regression coefficients are of similar size. Additionally, LASSO regression improves interpretability by producing a sparser model. Finally, the dimensionality of the data matters as James et al. (2021) stated ridge regression should be used when $p > n$, as logistic regression cannot produce a unique solution, while ridge regression can.

The package *glmnet* (v4.1-8; Friedman et al., 2010) was used to train LASSO and ridge regression models. Using the *cv.glmnet* function, 10-fold cross validation was used to tune the model over a function specified sequence of λ's to find the optimal λ. We used the optimal minimal λ, which produced the smallest average cross-validated error. This was preferred over the one standard error λ, as the minimal λ prioritizes minimization of the cross-validated error, which in predictive analysis is important, whereas the one standard error λ obtains sparser results, which makes them more interpretable (Hastie et al., 2023). Elastic net regression was trained using the package *caret* (v6.0-94; Kuhn, 2008). In these regularization methods, R occasionally produced errors, especially with very small sample sizes. Errors could occur when only one covariate is included, sampled data contains only a single observation of a class, or during cross validation a fold contains either none or one value of a class. These errors are considered acceptable. To allow the process to continue, if an error occurred, the performance measures were set to 0.50, representing a random guess.

2.4.3 Machine Learning Methods

A small selection of machine learning methods was used. They were selected partly because of familiarity as well as the fact that these methods are often mentioned in the literature. This means a large group of researchers will be familiar with these methods, making this study more accessible. All machine learning methods were trained using the *caret* package in R (v6.0-94; Kuhn, 2008) using 10-fold cross-validation.

K-Nearest Neighbors

K-Nearest Neighbors is a simple algorithm (Bruce et al., 2020). Hyperparameter K specifies the number of training points, i.e., *neighbors*, that need to be close to a point x and is described in Table 7. Using the Euclidean distance from point x to measure the K closest training data points, the proportion of points belonging to each class, i.e., the estimated probability, is calculated. Point x will belong to the class for which the probability is highest (James et al., 2021; Kuhn & Johnson, 2013). To ensure the scale of the data does not bias the estimation, the data is standardized (Bruce et al., 2020; Kuhn & Johnson, 2013). Nevertheless, KNN can still overfit on the data. When K is small, the algorithm will exhibit higher variance compared to larger K, as the flexibility of the model decreases. When K is too low, it will fit too closely on the data, but if K is too high it can also lead to underfitting as no clear pattern is found (Bruce et al., 2020; James et al., 2021; Kuhn & Johnson, 2013). Cross-validation finds K that will balance this best.

Random Forest

Random Forest is a tree-based ensemble classification method that uses bagging to generate B independent classification trees from the training data. In each decision tree b, only a random subsample of covariates is allowed to be used to make the split (James et al., 2021; Kuhn & Johnson, 2013). The default value is described in Table 7, which is also the default value from the *randomForest* package (v4.7-1.1; Liaw & Wiener, 2002). Allowing only a subset of covariates at each split leads to trees that include covariates that might not be included otherwise, due to the presence of influential covariates. This introduces randomness to the algorithm, which results in decorrelated trees. By adding this random component, a more diverse set of trees is introduced and the variance is decreased. The prediction for a new observation is based on a majority vote, where the prediction for an observation is based on what the majority of the B trees predicted (James et al., 2021; Kuhn & Johnson, 2013). A limitation of the *randomForest* package (v4.7-1.1; Liaw & Wiener, 2002) which the *caret* package used to train the model, is that the number of trees to grow is fixed at 500. However, multiple sources have stated that a higher number of trees will not result in overfitting (Breiman, 2001; James et al., 2021; Kuhn & Johnson, 2013). That is why we do not consider it a major issue to use this default value, as 500 trees is already quite large.

Gradient Boosting Machine

Gradient Boosting Machine (GBM) is a tree-based ensemble classification method which uses boosting. Boosting means trees are not grown independently, but in a sequential order. GBM starts by building the first model, $\hat{f}^1$, which can be considered the null model. The residuals are calculated for each observation. Subsequently, these residuals are used as the outcome on which a new tree, $\hat{f}^2$, is fitted to best predict these residuals. The algorithm learns by adding the new tree to the earlier tree: $\hat{f} = \hat{f}^1 + \lambda\hat{f}^2$, where λ represents the shrinkage parameter. This process keeps repeating itself for a number of B specified trees, which demonstrates the sequential nature of this model, as the same data keeps being used to build trees and correct its performance (James et al., 2021; Kuhn & Johnson, 2013). The GBM model requires four hyperparameters that need tuning in *caret*, which are described in Table 7.

XGBoost

XGBoost is a tree-based ensemble method, similar to GBM. The major difference between these two methods is the addition of a regularization parameter in XGBoost and its ability to scale up its operations, which also positively affects its speed (Chen & Guestrin, 2016). The method is also able to implement variable selection, similar to random forest. XGBoost optimizes the loss function that includes a regularization term. The model complexity is controlled by this regularization term, which will aid against the risk of overfitting (Bruce et al., 2020; Chen & Guestrin, 2016). Several hyperparameters need to be tuned to control overfitting, described in Table 7.

Support Vector Machine

Support Vector Machines (SVM) are useful in the case of binary classification. Several variations of the SVM algorithm are available, i.e., kernels, all tasked with finding a boundary that best separates the classes (James et al., 2021; Kuhn & Johnson, 2013). We used three different SVM kernels in this analysis: linear, polynomial, and radial.

The linear kernel assumes the data can be linearly separated by a hyperplane, but infinitely many variations of the hyperplane that separate the classes are possible. The kernel tries to find the hyperplane that maximizes the margin. First, the distance between each training point and the (possible) hyperplanes is calculated. Second, the shortest distance between each class and the (possible) hyperplanes is used to construct the margins, which results in many possible margins. Maximizing the margin is the goal, which is why the model chooses the hyperplane that belongs to the largest calculated margin. However, not all data can be entirely linearly separated. Therefore, a hyperparameter *cost (C)* is introduced to control bias and variance in the model (James et al., 2021), further explained in Table 7.

Data cannot always be linearly separated and using the linear SVM would result in poor results. The kernel would also be prone to overfitting, as each observation could have considerable influence on which hyperplane was selected (James et al., 2021). Non-linear kernels, such as polynomial and radial, can find boundaries that are considered more flexible (Kuhn & Johnson, 2013), but require more tuning. The hyperparameters are clarified in Table 7.

Table 7

Hyperparameters per Method

Method	Hyper-parameter	Default[1]	Meaning and justification	Hyperparameter Grid
LASSO regression	`lambda`	-	`lambda` regulates how much effect the penalty has on the model coefficients. The shrinkage becomes more extreme the larger λ becomes. As λ becomes larger, the variance becomes smaller and bias higher in the resulting model (James et al., 2021).	Model-specified default sequence
Ridge regression	`lambda`	-	See `lambda` of the LASSO regression.	Model-specified default sequence
Elastic Net Regression	`lambda`	-	See `lambda` of the LASSO regression.	Grid Search of 20 combinations
	`alpha`	0.1	`alpha` determines the weight given to the included penalties of LASSO and ridge regression.	
K-Nearest Neighbors	`k`	5	Number of neighbors. The default value could either lead to under- or overfitting (Bruce et al., 2020; James et al., 2021; Kuhn & Johnson, 2013), hence a grid is specified, with a relatively small maximum value due to computational time.	$[1, \min(\#\text{covariates}, 15)]$
Random Forest	`mtry`	$\sqrt{J}$	Random subsample of the number of covariates, J, allowed to make the split. To limit computational time we use the recommended default value as the maximum value.	$[1, \ldots, \min(\max(1, J), \sqrt{J})]$
Gradient Boosting Machine	`n.tree`	50	The number of trees that are built sequentially. James et al. (2021) warn for using too many trees, as they could fit too closely on the data, leading to overfitting.	50, 100, 200, 300, 400, 500, 1000
	`interaction.depth`	1	The interaction depth, also known as the number of splits. This directly influences the complexity that is possible within the tree. We included smaller numbers, as the sequential nature of boosting means lower interaction depth will work equally well (James et al., 2021).	1, 2, 3, 4
	`shrinkage`	0.1	The shrinkage controls the learning rate of the model. The lower the value, the slower the model will learn, which is a good approach to prevent overfitting, but often requires more trees to be built (James et al., 2021).	0.005, 0.001, 0.05, 0.01, 0.1
	`n.minobsinnode`	10	Minimum number of observations that should be present in the terminal nodes of trees. This was fixed at 1, because the actual number of observations in the terminal nodes can be larger. Moreover, as we vary the sample size, very small sample sizes are also included which might not work very well with the default.	1
XGBoost	`nrounds`	50	The maximum number of trees, also known as boosting iterations (Bruce et al., 2020).	Random Search of 15 combinations
	`max_depth`	1	The maximum depth a tree can have, i.e., the highest interaction order (Bruce et al., 2020).	
	`eta`	0.3; 0.4	The learning rate. A small η prevents overfitting, but does mean the model needs more trees to generate good predictions (Bruce et al., 2020).	
	`gamma`	0	The minimum loss reduction that is needed to make another split. The default of 0 indicates no loss is required.	
	`colsample_bytree`	0.6; 0.8	The subsample ratio of columns determines whether to use random forest-like sampling. A number less than 1 means only a fraction of covariates is chosen to construct the tree (Bruce et al., 2020).	
	`subsample`	0.5	The ratio that indicates how much of the training data is sampled (without replacement). Using this prevents overfitting (Bruce et al., 2020), as not all data is used in every tree, making the results more generalizable.	
	`min_child_weight`	1	The minimum sum of instance weight needed in a child controls the complexity of the tree. Higher values will make the tree less flexible, which prevents overfitting (Chen et al., 2024).	
Support Vector Machine: Linear Kernel	`C`	1	The cost of misclassification. If the costs are low, the margin can be quite wide and misclassification is more likely. If the costs are high, the margins will be tighter and it is costlier to misclassify an observation. This high cost can lead to a model that overfits, as it is too closely fit on the training data (Cerulli, 2023; James et al., 2021; Kuhn & Johnson, 2013).	Random Search of 15 combinations
Support Vector Machine: Polynomial Kernel	`degree`	1	The degree of the polynomial.	Random Search of 15 combinations
	`scale`	0.001	Scaling parameter.	
	`C`	0.25	See `C` of the linear kernel.	
Support Vector Machine: Radial Kernel	`sigma`	-	A scaling parameter which influences the behavior of the kernel. It determines how many training observations will influence the final prediction. If it is small, many training observations, also further away, will be used, but if sigma is large the behavior of the radial kernel can be considered more local. This means that training observations near the boundary will determine the final outcomes and increase the model flexibility (Cerulli, 2023; James et al., 2021). Also known as gamma.	Random Search of 15 combinations
	`C`	0.25	See `C` of the linear kernel.	

Note. J = total number of covariates. This table describes the hyperparameters that have been used during the training of models. For each hyperparameter a short description is provided. Occasionally an extra interpretation is given on how this parameter might affect the model fit.

[1] The default values were extracted from the *caret* package documentation, using `len = 1`. The default of the *caret* package when using function `train` is applying a grid search. Default values were pulled from the option `grid` with `len = 1`, i.e., `tuneLength = 1`.

- The default value is often a sequence that was generated, meaning it is data-driven and different for each data set.

Chapter 3

Results

To gain a better understanding of under which circumstances traditional statistical methods or machine learning methods performed better, we evaluated different simulated data scenarios using several complexity level measures. In this section we examine the patterns we observed and highlight several models of interest. Models were categorized in patterns according to which method had the highest performance estimate at each sample size. A general pattern was categorized as such if more than one model displayed this general behavior. A selection of results is included in this section to illustrate the patterns found in the exploratory analysis, which already required us to apply a level of interpretation to the produced results. The complete results are documented on the GitHub page, referred to in Appendix B without any interpretation. This is done because the number of results produced in this study was too extensive to include without initial interpretations. Additionally, an in-depth analysis is performed on the median model from each pattern and performance measure to explore whether methods, that are computationally more expensive, produce a similar pattern or have vastly different results. In this section we will refer to machine learning methods as ML and traditional statistical methods as ST to increase readability.

For each performance measure, we have separately analyzed the apparent general patterns. Performance measures are shown in a visualization using range $[0.50, 1]$, as results below 0.50 are worse than a random guess. Models that predict below this lower-bound value are not good models and not what researchers are looking for when building a good model. Due to this restricted range, the possibility exists that some results might not be visible. While it may not be visible, we did use this information for the formation of the patterns. An overview of all models can be found in Table 1 to 6 and Table 9. Due to the large range that is covered in each figure to compare patterns, they may not be entirely visible without zooming in. To ensure readability we used *pdf* versions, for which the reader can zoom in to view every detail.

3.1 Exploratory Analysis

In the exploratory analysis we have run 138 simulations using logistic regression, LASSO regression, KNN, and random forest. The observation scale was x^2, where $x = [4, 6, \ldots, 42, 44]$. We looked at each performance measure, of which accuracy was of main interest. Nevertheless, sensitivity and specificity were also important to examine, as research goals could call for a better sensitivity than accuracy. The division of models into patterns by performance measure can be found in Appendix C.

3.1.1 Accuracy

We identified six different general patterns that occurred repeatedly in several results. We discuss each pattern separately and provide representative visualizations of variations within patterns. A special case is model *E-1*, which was somewhat different from the other models. This model only included 2 main effects. We did not include this into a pattern, for any performance measure, as performance estimates became equal very fast. This is the only model with this behavior.

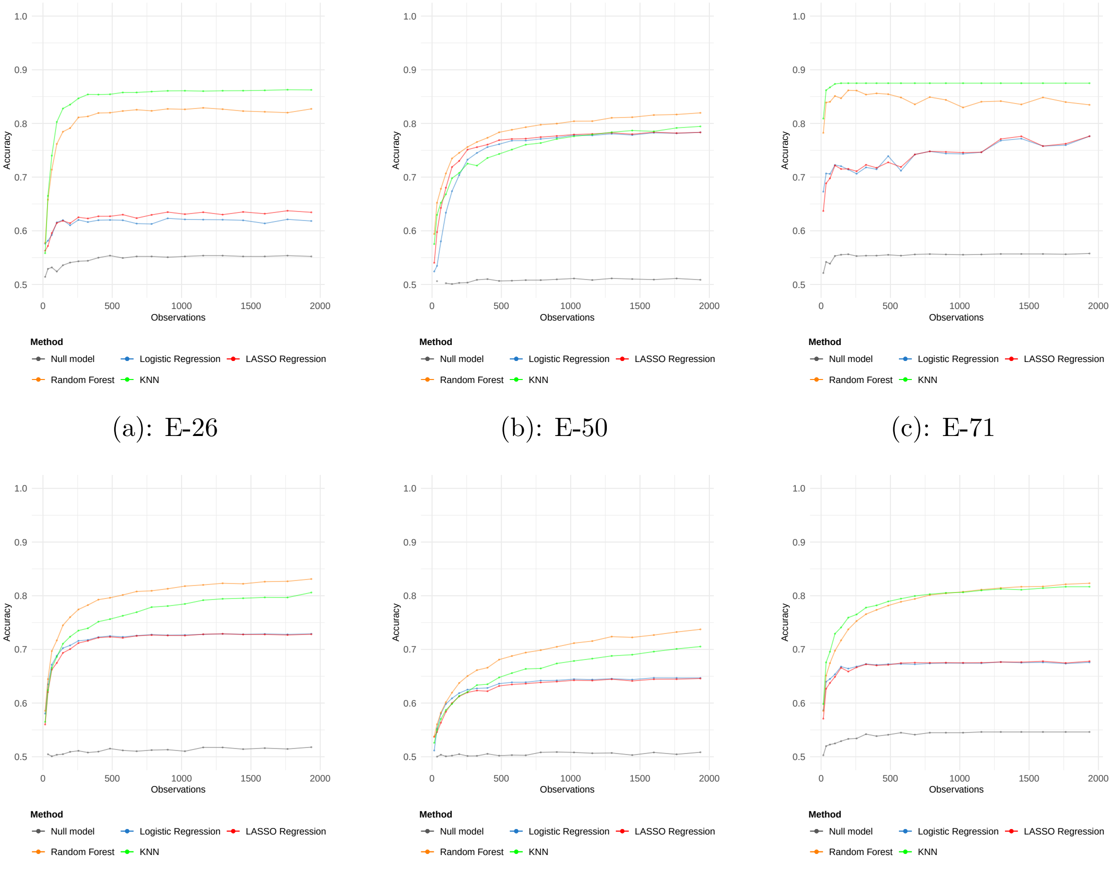
Pattern A

In this pattern both random forest and KNN, machine learning methods, generally outperformed traditional statistical methods. Models were included in this pattern when at least one machine learning method had superior performance on all points. The difference in accuracy between ML and ST became more distinct when the sample size was large, but also frequently when the sample size was smaller. For example, in Figure 1a, we observed a steep increase in accuracy for smaller sample sizes. Generally, for the smallest sample size, ML and ST methods started off close together. After which ML methods had a steep increase and reached a stable performance (Figure 1a) or a more gradual increase (Figures 1b, 1d, 1e, 1f). Almost all models included in Figure 1 showed that ST methods improved their performance when the sample size increased. They either kept increasing marginally (Figure 1e), reached a plateau around which they fluctuated around an invisible line (Figure 1a) or displayed a stable performance (Figure 1b, 1d, 1f). The exception is models like Figure 1c, which showcased model *E-71*, encompassing only 2 main covariates and 1 interaction effect. We observed a divide between both approaches, in which ST methods did not come close to the performance of ML. ML curves showcased both a stable performance for KNN, and a decreasing performance for random forest.

The models characterized as pattern A had several complexity measures in common. All models had at least interaction depth 2. Other settings varied, such as the number of covariates and interactions, which ranged from first-order interactions up to fourth-order interactions. The main covariates ranged between 2 to 30 covariates, while the first-order interactions ranged from 1 interaction to 100. As interaction depth increased, the ranges of number of interactions became smaller, i.e., between 2 and 10, except for model *E-24*, which showed similar behavior to Figure

Figure 1

Accuracy Performance of Pattern A



(a): E-26

(b): E-50

(c): E-71

(d): E-81

(e): E-99

(f): E-133

Note. This figure shows accuracy estimates on the test data measured across different training sample sizes for pattern A. A selection of exploratory models is displayed, for which the settings are available in Tables 1 to 6.

1e. The regression coefficients differed among the models. It ranged from unique regression coefficients (Figures 1a, 1b, 1e, 1f), identical regression coefficients (Figures 1c), to non-unique regression coefficients (Figures 1d). Generally, when models had a high interaction depth, they had unique regression coefficients (except for model *E-88*), while models with a smaller number of interactions more often displayed non-unique or identical regression coefficients.

All models had variance $\delta = 0.10$, except for models *E-93* and *E-99* with $\delta = 0.05$. Moreover, only two models had either 1 or 3 continuous covariates included, all with unique regression coefficients, interaction depth 5, and $\delta = 0.10$ (Figure 1f). We observed that as the number of continuous values increased, the ST curves slightly shifted upwards, while the ML curves shifted downwards. Moreover, the ML curves became steeper, while ST exhibited more stable curves close together. While most models included in this pattern had formula complexity 1, there were two models of formula complexity 2. In general, they followed the pattern showcased in Figure 1b, where ST curves remained closer to ML curves. Moreover, only a gradual increase is observed in ML performance, where KNN needs a very large sample size to overtake ST methods.

Pattern B

Pattern B consisted of models where traditional statistical methods outperformed machine learning methods. Models were included in this pattern when at least one traditional statistical method had superior performance on all points.

While the different models exhibited some variation in performance, in general the difference between the traditional statistical methods and machine learning methods was not considerably large. Moreover, logistic regression outperformed LASSO regression, especially for smaller sample sizes, while as sample size increased their performance became comparable.

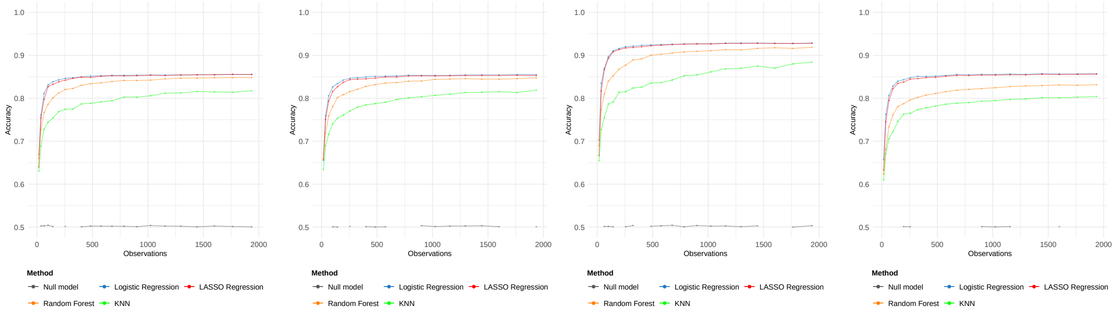
The model settings in this pattern consisted of models with formula complexity 1 and 10 main covariates. The number of regression coefficients was unique for all models, except model *E-68*, which had 5 non-unique regression coefficients. The variance parameter, δ, mainly consisted of $\delta = 0.10$, but also included a model with $\delta = 0.05$ (*E-89*). One model also included only continuous covariates, *E-116*, shown in Figure 2d. A few comparisons could be made, as models were very similar. Models *E-2* and *E-89* had variance $\delta = 0.10$ and $\delta = 0.05$, respectively. We observed an upward shift of both approaches when the variance was decreased to $\delta = 0.05$, while proportions between the approaches remained similar to model *E-2*. Models *E-2* and *E-68* had 10 and 5 unique regression coefficients, respectively. Figures 2a and 2b displayed almost identical curves. Comparing models *E-2* and *E-116* with 0 and 10 continuous covariates, respectively, we only noticed a downward shift of ML curves, while the ST curves remained almost identical.

Pattern C

The distinguishing feature of this pattern is the performance of logistic regression, which either lagged behind significantly or took a considerable amount of time to approach the LASSO curve. In this pattern traditional statistical methods overtook machine learning methods. Figure 3 also showed that random forest generally performed best of the two ML methods, as KNNs curves were not as steep. LASSO regression was always the first of the regression methods that caught up to the ML curves, whereas logistic regression either failed to do so or only caught up with either one or both ML curves very late. Logistic regression occasionally approached the LASSO curve (Figures 3a, 3e), but required a larger sample size to accomplish this. Variations in this pattern also showed logistic regression sometimes did not catch up (Figures 3b, 3c, 3d, 3f). Generally, logistic regression performed quite badly for smaller sample sizes. We even found

Figure 2

Accuracy Performance of Pattern B



(a): E-2 (b): E-68 (c): E-89 (d): E-116

Note. This figure shows accuracy estimates on the test data measured across different training sample sizes for pattern B. A selection of exploratory models is displayed, for which the settings are available in Tables 1 to 6.

extreme cases, Figures 3b and 3f, where logistic regression performed similar to the null model and only started improving once the sample size passed 1250.

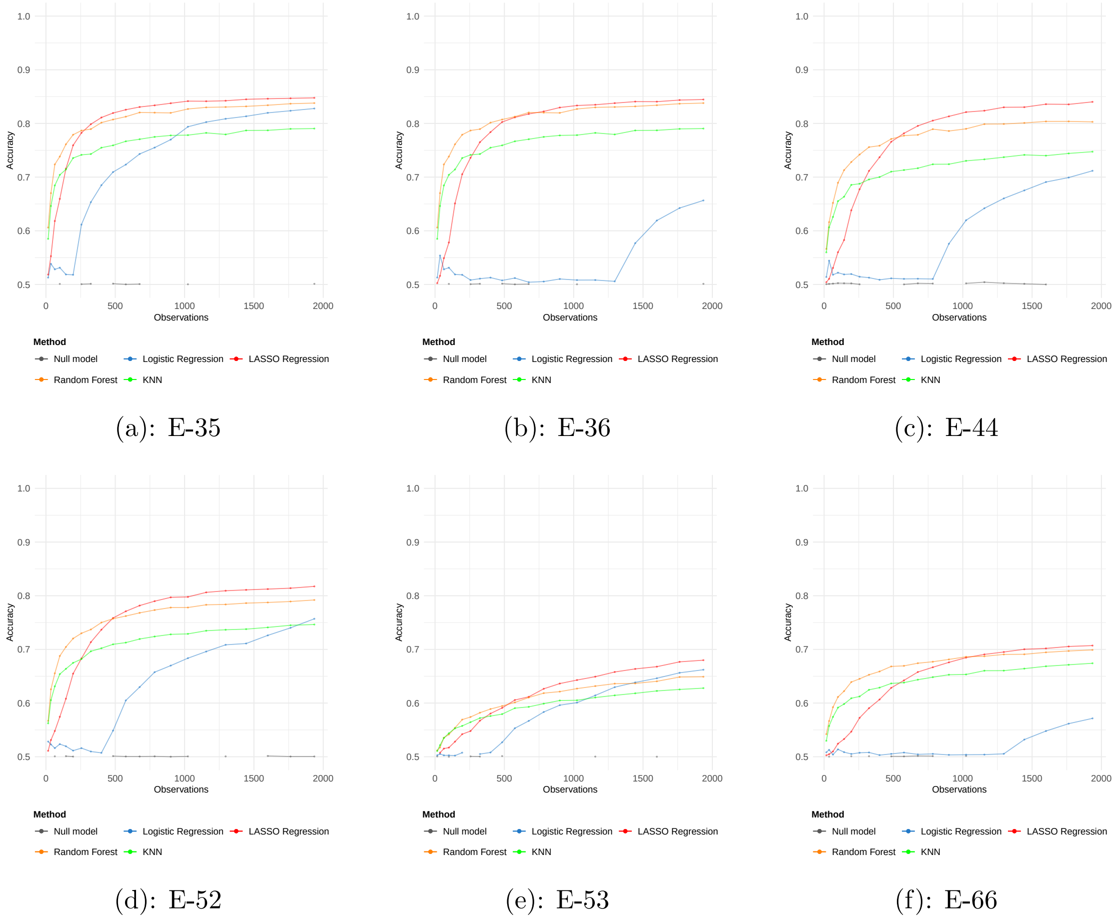
The models that we categorized as pattern C had several commonalities. All had either formula complexity 2 or 3 and target variance $\delta = 0.10$ and all, except one model (*E-131*), had no continuous covariates. Moreover, all models had unique regression coefficients. The only settings that truly varied were the number of covariates and interactions (and interaction depth). This ranged from only main effects up to sixth-order interactions. The main covariates ranged between 10 and 50 covariates, while first- and second-order interactions ranged between 1 and 100. Higher interaction-orders had a smaller number of interactions, ranging between 1 and 20.

Several comparisons were possible. Models *E-35* and *E-36* (Figures 3a, 3b) had the same settings, except for the formula complexity of 2 and 3, respectively. The behavior of logistic regression was extremely different, as in model *E-35* logistic regression began improving its performance before $n = 250$, while model *E-36* required a sample size past $n = 1250$ before any significant improvement was observed.

Generally, we observed that as the number of main effects increased, it took longer for LASSO regression to overtake random forest. This was apparent in models that only had main effects or main effects and one first-order interaction. When a high number of higher-order interactions was included, the effect was too difficult to observe. A few illustrations: model *E-53* (Figure 3e) had 100 first- and second-order interactions, while model *E-52* (Figure 3d) had only 5 first- and 3 second-order interactions. Both had 30 main effects. But the figures merely showed a small up- and downwards shift. A similar comparison is between models *E-35, E-42, E-64*, not all of which are displayed in the figure. What we did not see was a left- or right shift. This only occurred when we reviewed models where the main effects increased or when models such as *E-35*, increased their formula complexity to 3 (Figure 3b). This illustration also highlights the

Figure 3

Accuracy Performance of Pattern C



(a): E-35 (b): E-36 (c): E-44

(d): E-52 (e): E-53 (f): E-66

Note. This figure shows accuracy estimates on the test data measured across different training sample sizes for pattern C. A selection of exploratory models is displayed, for which the settings are available in Tables 1 to 6.

fact that logistic regression struggled when a higher number of combinations was possible due to higher formula complexity. In model *E-35*, logistic regression only had to identify effects for 210 possible combinations (20 + 190), whereas model *E-36* had 1350 possible effects (20 + 190 + 1140). While it is not logistic regression that ‘struggles’ to find effects, it is the limitation of logistic regression when data is high-dimensional as there exists not one particular solution for the effects (Cerulli, 2023; James et al., 2021; Van Wieringen, 2023). This also invoked several warnings during model training from the logistic model in R, as the model had trouble finding unique solutions that did not exist. All models that fall under the pattern have $p > n$ for small sample sizes. When more combinations are possible, due to more main covariates or formula complexity, the longer it takes until the model reaches a low-dimensional situation. This is partly due to the design of this study, because when we used formula complexity we added all possible interactions.

Pattern D

In this pattern, initially machine learning performed best for the smallest sample size(s). As the sample size increased, one of the traditional statistical methods managed to catch up and had superior or similar accuracy estimates. A distinguishing feature of this pattern is that a machine learning method eventually overtook the regression method(s) as sample size increased.

It varied when machine learning methods started outperforming regression methods after the regression methods initially overtook them. In some of the models it happened before $n = 250$ (Figure 4a), while in other models it happened after $n = 1000$ (Figure 4e). Regression methods were shown to either stabilize as the sample size increased (Figures 4a, 4b, 4d, 4e, 4f) or keep increasing (Figure 4c). It differed between models at what point ML methods started outperforming ST methods in smaller sample sizes. Model *E-113* (Figure 4e) showed random forest performed best in the first three sample sizes, while for model *E-21* (Figure 4b) it was only the first observation. Both figures 4c and 4e had formula complexity 2, but each produced different curves. Another variation is the relative difference between ML and ST methods. We observed quite a large gap between the two approaches in Figure 4b, while Figures 4a, 4c, 4d, 4e, 4f displayed the two approaches much closer together.

The models in which this pattern was observed differed in several of the complexity measures. All had at least interaction depth 2. The formula complexity ranged from 1 to 3, important to mention is that only one model had formula complexity 3. This illustrates that while pattern C contained many models with higher formula complexity, they were not confined to that pattern. Almost half of the models did not have unique regression coefficients, which also ranged from identical to unique (Figures 4a, 4b, 4c, 4d, 4f) and non-unique (Figure 4e). The variance parameter, δ, had all possible values, of which the large majority was $\delta = 0.10$. Six models had continuous covariates, ranging between 1 and 15.

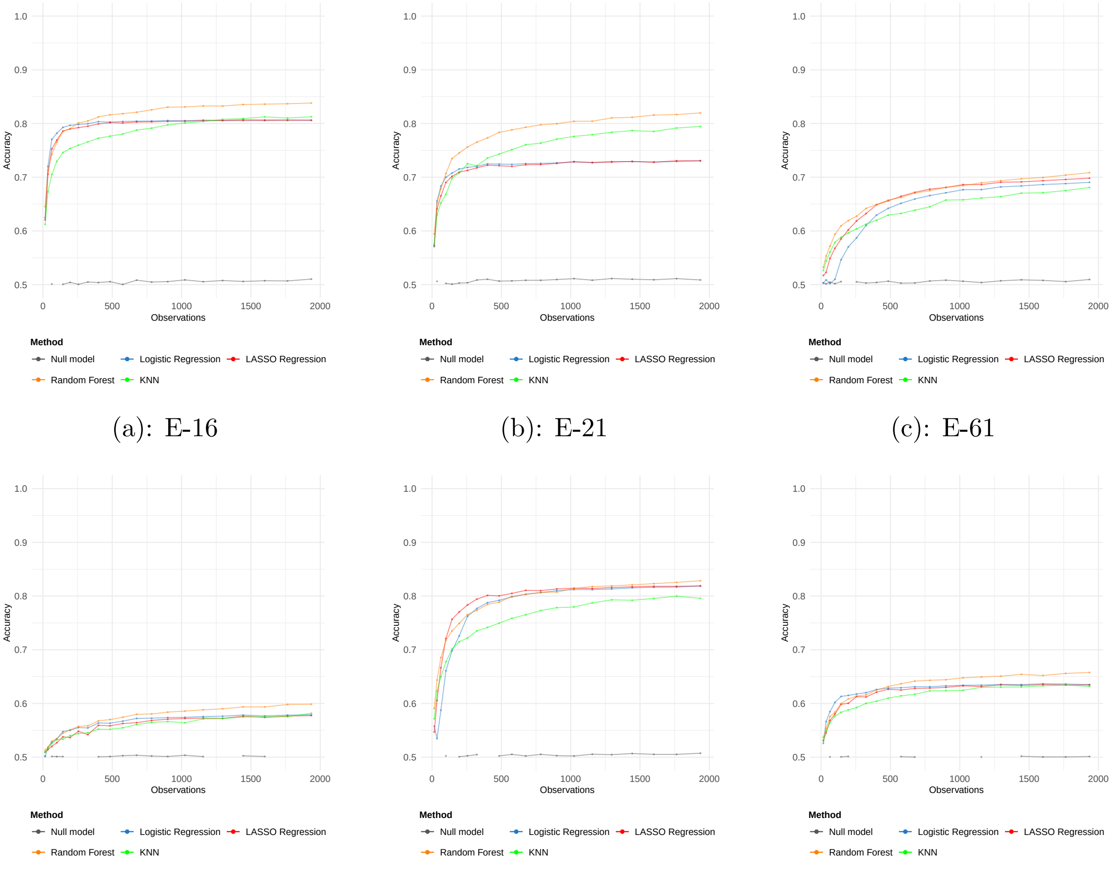
We list some possible comparisons. Figures 4b and 4f only differ in the variance, $\delta = 0.10$ to $\delta = 0.20$, respectively, as well as one continuous covariate. The addition of higher δ and one continuous covariate shifted down the curves of both approaches, but was more distinct for the ML curves. Additionally, models *E-123* and, while not displayed, *E-96* had an almost identical visualization. The only difference was the addition of 1 continuous covariate in *E-123*. Thus, it seems that the major difference in Figures 4b and 4f could be due to the increased variance.

Pattern E

This fifth pattern showcased the ability of traditional statistical methods to surpass machine learning methods. Either one or both machine learning methods had higher accuracy compared

Figure 4

Accuracy Performance of Pattern D



(a): E-16 (b): E-21 (c): E-61

(d): E-100 (e): E-113 (f): E-123

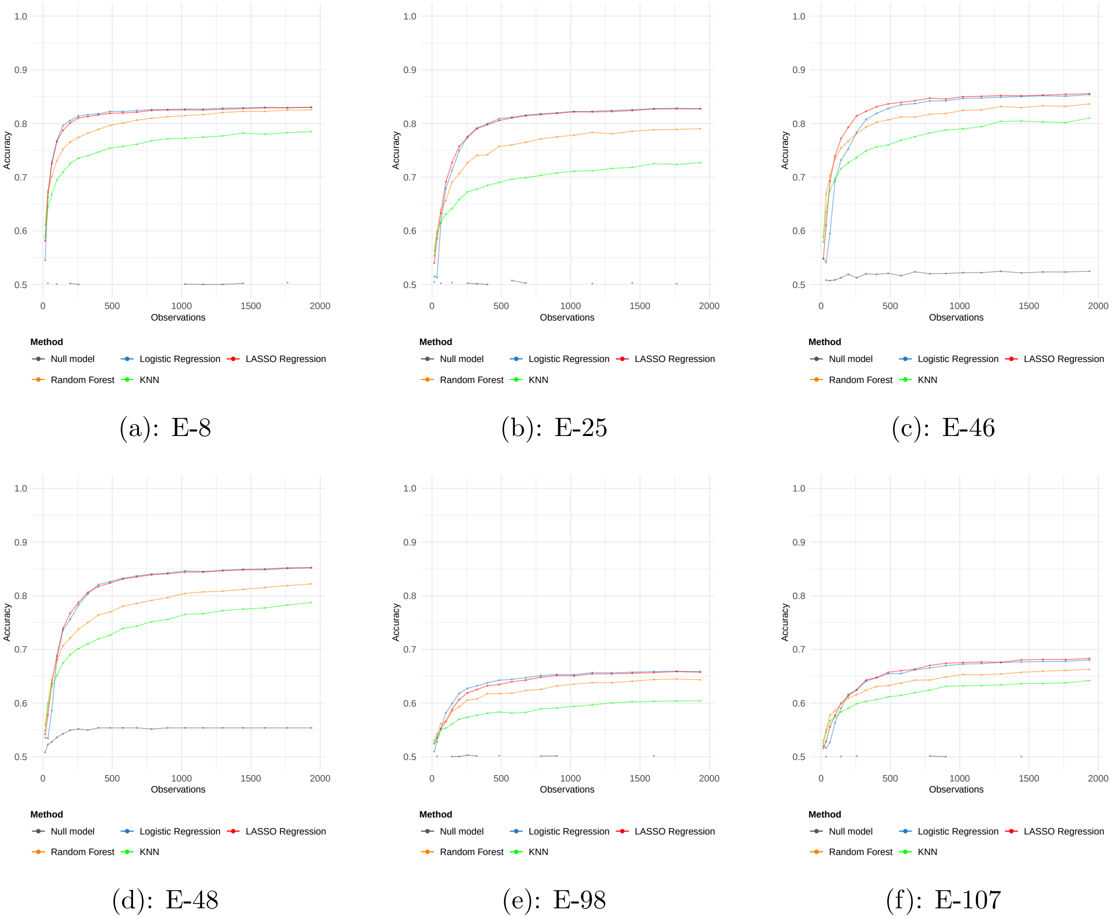
Note. This figure shows accuracy estimates on the test data measured across different training sample sizes for pattern D. A selection of exploratory models is displayed, for which the settings are available in Tables 1 to 6.

to traditional statistical methods for the first few sample sizes. The key identifier of this pattern is the fact that traditional statistical methods always overtook the machine learning methods as the sample size started to increase. Moreover, the performance of the regression methods often stabilized quickly, whereas the machine learning methods showed a gradual increase in accuracy, but not enough to catch up.

Several variations within this pattern were observed. First, ST methods oftentimes had a high performance that quickly stabilized, whereas the ML methods stayed behind and slowly kept increasing. ML curves either approached the ST curves closely (Figure 5a) or remained further away (Figure 5b). A variation on this is when instead of logistic regression, LASSO regression performed better. Noticeable is the fact that logistic regression did eventually catch

Figure 5

Accuracy Performance of Pattern E



(a): E-8

(b): E-25

(c): E-46

(d): E-48

(e): E-98

(f): E-107

Note. This figure shows accuracy estimates on the test data measured across different training sample sizes for pattern E. A selection of exploratory models is displayed, for which the settings are available in Tables 1 to 6.

up, while ML curves remained further away from both ST curves (Figures 5c, 5d, 5f). Models displayed this behavior when they had formula complexity 2. A different variation showed a lower overall performance for all approaches. ST performance kept stabilizing quickly and ML curves had a flatter slope than ST curves, resulting in a lower performance. ML curves either remained at a distance or came close to the ST curves as the sample size increased (Figures 5e, 5f). Models that exhibited this behavior varied in formula complexity, but all had variance $\delta = 0.20$, except model *E-30* which had $\delta = 0.10$. Important to add is that model *E-30* had 50 main covariates and interaction depth 5, while the models that were included with $\delta = 0.20$ had interaction depth 3 or lower.

Several complexity measures were the same among models. Of the 39 models all had zero continuous covariates, except for models *E-117*, *E-129* and *E-130* who each had 10 continuous covariates. Formula complexity was either 1 or 2. Other settings varied, where we saw that the majority had variance $\delta = 0.10$, but seven models were included with $\delta = 0.20$ and three had $\delta = 0.05$. Most models had only main effects or first-order interactions. Seven models were included with second-order interactions and one model each for third-, fourth-, and fifth-order interactions. The number of main effects varied between 10 up to 50, while the number of first-order effects ranged between 1 and 40. This drastically reduced for second-order interactions and higher-order interactions which ranged between 1 and 10. The number of unique regression coefficients varied. It seemed to depend on the number of covariates and interaction depth. Almost all models with interaction depth 3 and higher had unique regression coefficients (Figures 5a, 5b, 5c, 5d, 5e), as well as models with 30 main covariates or higher and interaction depth lower than 2. However, there were models included with 10 or 20 main effects, and any number of first-order interactions, with either identical or non-unique regression coefficients (Figure 5f).

We list some interesting comparisons. Figure 5a, with 20 main effects and 1 interaction effect, and Figure 5b, with 50 main effects and 1 first-, second-, and third-order interaction effect illustrated the increasing distance between ST and ML curves as both the main effects and interactions have increased. Moreover, the only difference between Figure 5c and 5d is a higher number of first-order interactions (10 vs. 40). It is clear that as the first-order interactions increased, the LASSO regression curve shifted down towards the logistic regression and the curves of the ML methods shifted down further.

Pattern F

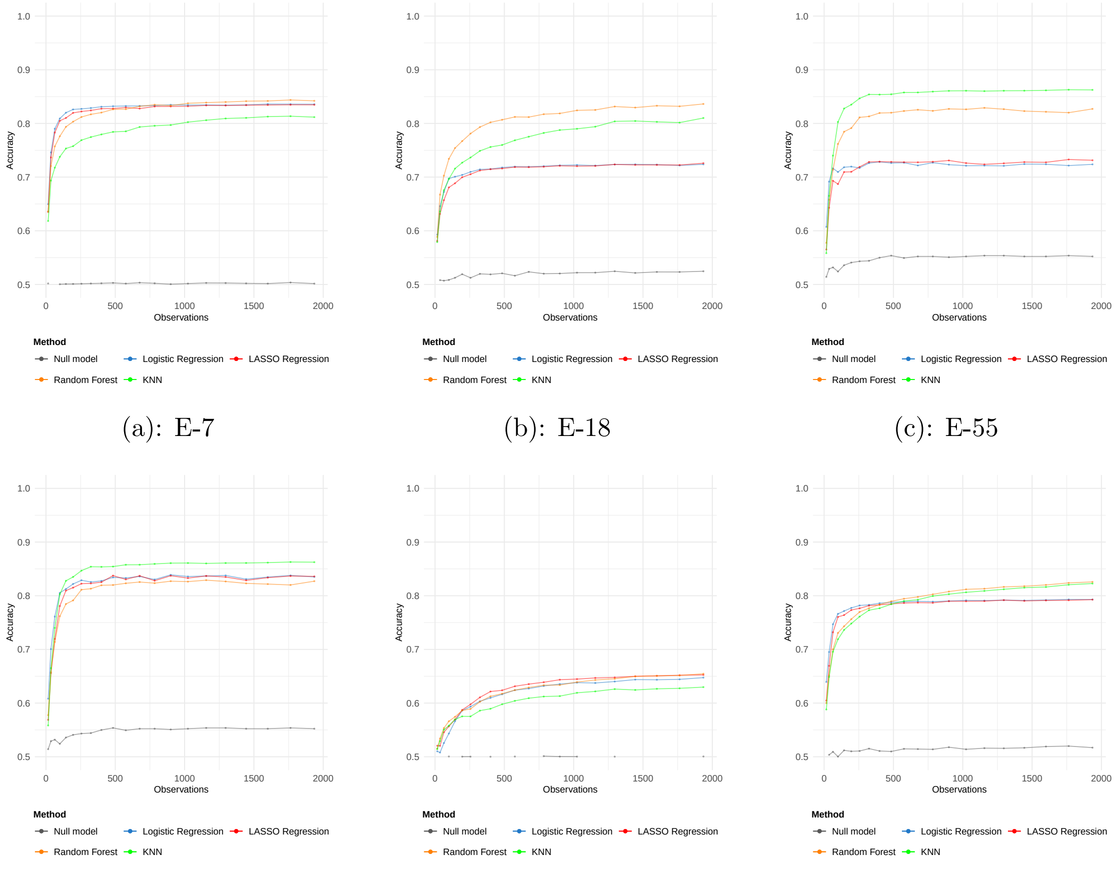
Pattern F showed that machine learning methods overtook traditional statistical methods. The models included in this pattern exhibited better performance from traditional statistical methods for smaller sample points than machine learning methods. Once the sample size increased, machine learning methods started outperforming traditional statistical methods.

When we looked at the general variations in the pattern, the accuracy curves of ML methods were sometimes steeper and quickly found a stable estimate (Figures 6c, 6d), while in other cases they increased steadily as the sample size increased (Figure 6a, 6b, 6e, 6f). Sometimes the ML curves remained close after having overtaken the ST curves (Figures 6a, 6d, 6e), while in other cases they moved away further (Figures 6b, 6c, 6f). In some cases, ST curves quickly increased and found a stable estimate (Figures 6a, 6b, 6f), while in other cases they kept fluctuating around an invisible line (Figures 6c, 6d), which meant accuracy estimates differed between sample sizes.

To go into more detail on the general pattern we found that in some cases, ST curves remained low and produced stable estimates (Figure 6b), while curves of ML methods kept increasing their performance. Models that showed this sub-pattern all had formula complexity 1 and almost all had unique regression coefficients. Similarly, other models demonstrated a stable performance for ST methods, but it took ML methods longer to outperform the ST methods (Figures 6a and

Figure 6

Accuracy Performance of Pattern F



(a): E-7 (b): E-18 (c): E-55

(d): E-56 (e): E-110 (f): E-128

Note. This figure shows accuracy estimates on the test data measured across different training sample sizes for pattern F. A selection of exploratory models is displayed, for which the settings are available in Tables 1 to 6.

6f). While in model *E-18* the superior performance of ML methods was already apparent in smaller sample sizes, other variations showed it took longer for ML curves to overtake the ST curves. Some showed that both ML methods were able to overtake the ST methods (Figure 6f), whereas sometimes we observed only random forest succeeding (Figure 6a). There was also the case in which neither approach produced stabilizing estimates but kept increasing, demonstrating similar low performance (Figure 6e). This model was generated using formula complexity 2 and it is apparent that the ML curve took until the last observation point to overtake the ST methods. The ST methods could also produce some fluctuation (Figures 6c and 6d). ST curves often fluctuated around an invisible line instead of the stable performance as seen in earlier figures, whereas ML curves had a very steep increase at the start and stable estimates as the sample size increased. A comparison can be made between these two figures, as models *E-55* and *E-56* are identical apart from the formula complexity, 2 or 3, respectively. In Figure 6c both ML methods were superior, whereas Figure 6d only showed KNN outperforming the regression methods as they had increased significantly.

The models that belonged to this pattern had quite some differences between them. The interaction depth was between 1 and 5. Three models had formula complexity 2, two models had formula complexity 3, and the rest had formula complexity 1. Half of the models included had continuous covariates, ranging between 1 and 10 and had an interaction depth of either 3 or 5. The number of unique regression coefficients varied between identical (Figure 6e), unique (Figure 6a, 6b, 6c, 6d, 6f) to non-unique combinations. When we reviewed the number of covariates, we found that the main covariates ranged between 5 and 15, first-order interactions ranged between 1 and 15, second-order interactions were between 3 and 10, third-order interactions ranged between 2 and 10, and fourth-order interactions ranged between 1 and 10. Most models had variance $\delta = 0.10$, four models with $\delta = 0.05$ were included (Figure 6f), and two models with $\delta = 0.20$ (Figure 6e).

3.1.2 Sensitivity and Specificity

We also review the results from the sensitivity and specificity measures. We identified seven patterns, of which six we previously discussed in the accuracy estimates (A-F). As our main focus was the accuracy measure, we looked at sensitivity and specificity results together. We either described variations in the pattern, similar to the previous section, or depending on the models included looked at variations in sensitivity and specificity together.

Pattern A

In pattern A machine learning methods consistently delivered superior performance, a trend evident in the sensitivity and specificity estimates across several models. However, the metrics differed in the number of models within this pattern: specificity had 40 models that exhibited this pattern, while sensitivity had only 2.

Sensitivity only included two models. One of these models displayed a similar pattern to the accuracy estimates in Figure 1c, with the key difference being that the regression methods showed less increase and more fluctuation around an invisible line at approximately 0.57. The other model, depicted in Figure 8a, showed ML curves consistently increasing. Both models had formula complexity 1, no continuous covariates, and variance $\delta = 0.10$. Their interaction depth was either 2 or 3, and they had either identical or non-unique regression coefficients. More models had superior machine learning performance for specificity estimates, with several variations possible. Often, ST curves demonstrated relatively stable behavior, with both regression methods performing similar (Figure 8e). However, in other models, the ST curves behaved more erratically with many fluctuations (Figure 8d). In both situations, ML performance was often considerably higher than that of ST methods. The stable behavior of the ST methods was mostly seen in models with a formula complexity of 1, (non-)unique regression coefficients, and models with variance $\delta = 0.10$ or $\delta = 0.05$. The erratic behavior of ST methods was mostly found in models with interaction depth 5 and unique regression coefficients, or in models with interaction depth 2 with mainly, but not all, identical regression coefficients. We also observed overall lower performance, where all curves fluctuated (Figure 8b), where ML curves kept increasing while ST curves remained lower and flatter. Models with this lower performance had unique or identical regression coefficients, formula complexity 1, 15 main and first-order effects, and 10 second-, third-, and fourth-order effects, and all but one had variance $\delta = 0.10$. The last variation was when logistic regression lagged behind LASSO regression in differing degrees, similar to pattern C, where the ST curves were no longer close together. However, random forest outperformed both methods. Models had either formula complexity 2 or 3 and no continuous covariates, with other settings varying greatly.

This pattern encompassed many variations. There was stable behavior by ST methods (Figure 8e), but also erratic behavior (Figure 8d). Instances occurred where ST methods had similar performance (Figures 8b and 8e), or diverged more (Figures 8c and 8d). ML methods sometimes performed considerably higher (Figures 8d and 8e), while in other cases, the approaches were close together (Figures 8b and 8c). Generally, the interaction depth ranged between 2 and 5 and formula complexity between 1 and 3, although not every sub-pattern included the full range. Almost all models had variance $\delta = 0.10$, with the exception of five models with $\delta = 0.05$ and one with $\delta = 0.20$. Almost every variation in this pattern contained at least one model with a continuous value, of which the values ranged between 1 and 5.

Pattern B

In pattern B traditional statistical methods consistently outperformed machine learning methods, which both sensitivity and specificity estimates displayed across several models. Sensitivity had 14 models that exhibited this pattern, while specificity had only one. Note we set the criteria that a pattern existed only if at least two models displayed the pattern. Nevertheless, we included the specificity model since pattern B was already found in the accuracy results.

There are two sub-patterns that kept returning in multiple models. First, models where ST curves started with a steep increase, after which their performance remained stable and high. ML curves either neared the regression methods (Figure 9a), or remained at a larger distance 9c). Models that displayed this variation had either formula complexity 1 or 2 and variance $\delta = 0.10$ or $\delta = 0.05$. Second, both approaches could also move closely together, especially as the sample size increased (Figure 9d). The performance often stabilized, with fluctuations around an invisible horizontal line. Models that displayed this behavior had variance $\delta = 0.20$, unique regression coefficients, and formula complexity 1. Model *E-39* also showed a distinct pattern, which no other models displayed. Figure 9b displays how LASSO regression immediately outperformed ML methods. However, the behavior fluctuated greatly for smaller sample sizes and stabilized more as sample size increased. Also of interest is the behavior of random forest, which stabilized rather quick and performed the worst. Settings that accompanied this model were 2 main and 1 interaction effect, $\delta = 0.10$, formula complexity 2, and unique regression coefficients. We believe the reason why LASSO regression outperformed ML methods, is because of the higher formula complexity and the simplicity of the model.

No common denominator is present between the models, as all settings varied. Main effects varied between 2 and 30, first-order interactions ranged between 1 and 40, and higher order interactions ranged between 1 and 3. The majority had interaction depth 1, 2, or 3, with the exception of one model with interaction depth 6. The regression coefficients varied between identical, unique (Figure 9a, 9b, 9c, 9d), and non-unique coefficients. Variance, δ, took on all values, but did show a distinction between sub-patterns. Three models with continuous values 5 or 10 were included (Figure 9d). Apart from three models with formula complexity 2, the formula complexity was often 1. Also of interest is the fact that model *E-2* overlapped in pattern B for both performance estimates (Figure 9a).

Pattern C

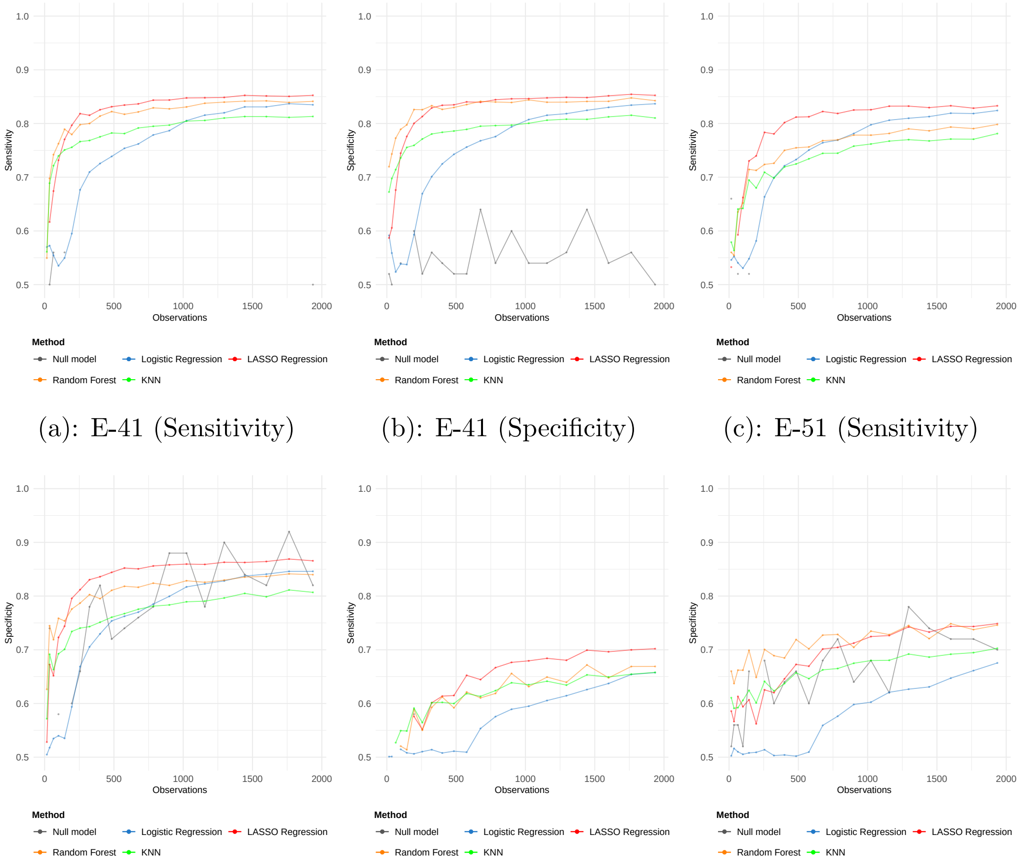
The key feature in pattern C was the poor performance of logistic regression, while LASSO regression outdid machine learning methods. In the accuracy results, models were included if machine learning performed better for smaller sample sizes but was overtaken by LASSO as the sample size increased. For sensitivity and specificity results we also observed that models were included if a traditional statistical method initially performed best for small sample sizes, was briefly overtaken by machine learning, and LASSO regression ultimately displayed superior performance. Both performance metrics had quite some models that fit pattern C, sensitivity had 18 and specificity had 14. All models in specificity were also present in sensitivity. Similar variations are present as discussed in pattern C of accuracy in section 3.1.1. To not reiterate this, we highlighted models in which the complementing or contrasting performance of sensitivity and specificity was visualized. This is the only pattern in which there is so much overlap of models.

As we know, pattern C was characterized by poor performance of logistic regression, which is due to the limitations of the logistic regression when $n \gg p$. This was also reflected in the performance of sensitivity and specificity. In general, two situations were observed. First, sensitivity and specificity produced a similar pattern in which only minor differences occurred (Figures 7a, 7b, 7c and 7d). Whereas there also existed a variation in which the curve of logistic regression shifted up and ML curves as well (Figures 7e and 7f). While the former figures look much more similar, the latter do not.

The models that were included in this pattern varied a great deal. In both sensitivity and specificity, models were included in which ST overtook ML as sample size increased either in

Figure 7

Pattern C: Sensitivity and Specificity Performance



(a): E-41 (Sensitivity)   (b): E-41 (Specificity)   (c): E-51 (Sensitivity)

(d): E-51 (Specificity)   (e): E-62 (Sensitivity)   (f): E-62 (Specificity)

Note. This figure shows sensitivity and specificity estimates on the test data measured across different training sample sizes for pattern C. A selection of exploratory models is displayed, for which the settings are available in Tables 1 to 6.

one move (Figure 7c and 7d) or in multiple alternating steps (Figure 7b and 7f). Models had interaction depth between 1 and 7, variance $\delta = 0.10$, and unique regression coefficients. We also saw, confined to sensitivity estimates, models in which ST methods performed best on the small sample size(s), but were overtaken by ML methods for a while. Distinguishing is the fact that eventually ST was able to outperform ML again (Figure 7a and 7e). Models in this scenario had interaction depth between 2 and 5, unique regression coefficients, and variance $\delta = 0.10$. Both scenarios had either formula complexity 2 or 3.

Pattern D

Both sensitivity and specificity are represented in this pattern. Sensitivity has 16 models and specificity 34. Only five models where present in both performance metrics. As this pattern contains many models, various sub patterns were possible. We highlight several, but nuances and variations still exist, such as only one method performing better later on, or more fluctuations.

The first variation demonstrated overall high performance of ST, whose performance stabilized as the sample size increased. ML methods started outperforming the ST methods early on and either kept increasing (Figure 10a) or remained quite close to the ST methods (Figure 10b). Models that demonstrated ML methods with these variations mainly had formula complexity 1 and some models with sensitivity estimates also had formula complexity 2. Variance varied between $\delta = 0.10$ and $\delta = 0.05$. There were also instances of low performance in general. In these figures at least random forest often performed well early on. Regularly only for one or two points ST had superior performance. All curves demonstrated more fluctuations in comparison to the other variation. Both approaches either gradually increased (Figure 10c) or remained on a similar performance level with small fluctuations (Figure 10d). Sometimes both ML curves had superior performance, whereas in other cases only one managed to outperform the ST methods. Two kinds of models kept gradually increasing. The first were smaller models with variance $\delta = 0.20$ and formula complexity 2, while the others were models with interaction depth 3 or 5, formula complexity 1, and variance $\delta = 0.10$. Models that demonstrated flatter curves were often smaller models of interaction depth 2 or 3, with variance $\delta = 0.20$, except for one model with interaction depth 7 and $\delta = 0.10$. The last sub-pattern is variation on pattern C, but here ML eventually outperforms LASSO regression. While we only show Figure 10e, the variation in this pattern is alike to what we saw in Pattern C. In some models, logistic regression did catch up with LASSO regression, while in others it was attempting to catch up. All models that had such a pattern had formula complexity 2 or 3, but varied in other settings.

Both sensitivity and specificity models that were characterized as pattern D had some version of these variations. Overall, a variety of models was included in this pattern for both sensitivity and specificity, as this pattern is very general. Summarizing the models from both performance measures together, interaction depth ranged between 1 and 7. Where main effects ranged between 10 and 50, first- and second-order effects ranged between 1 and 100, and higher-order interactions often ranged between 1 and 20. Identical and (non-)unique regression coefficients were all included, as well as all variance levels. Some models had continuous values, specificity models more than sensitivity models, and formula complexity had all possible values.

Pattern E

In this pattern both performance metrics had almost an equal number of models included, sensitivity had 24 models and specificity 21 models, of which some were overlapping models. In this pattern traditional statistical methods started outperforming machine learning methods.

Both sensitivity and specificity models produced a similar variation, in which ST curves started with a steep increase, after which their performance was high and stable. The ML curves were not as steep. Overall, they did not improve enough to have a similar performance (Figure 11a), or alternated with ST curves until the regression methods remained dominant (Figure 11b). Models that demonstrated this sub-pattern had formula complexity 1 and (non-)unique regression coefficients. A variation similar to pattern C is demonstrated in Figure 11c. This was not included as Pattern C, as the performance of logistic regression rapidly improved and as the sample size increased had mostly caught up with LASSO regression estimates. Models in both specificity and sensitivity had formula complexity 2, most had variance $\delta = 0.10$, with a few exceptions using $\delta = 0.05$, and all had 10 main effects. The interaction depth ranged between 1 and 3. The last variation, only present in sensitivity estimates, displayed both approaches generating lower estimates close to each other. While ST curves were superior, ML followed (very) close. An example is shown in Figure 11d, which is the extreme situation in which methods are very close together. Models that produced this lower performance with methods close together were characterized by variance $\delta = 0.20$ and either formula complexity 1 or 2.

If we consider sensitivity and specificity together, both had models with formula complexity 1 and 2 and included several models with 10 continuous covariates. However, there are also quite some differences. The interaction depth of specificity models ranged between 1 and 3, whereas sensitivity models also had a model with interaction depth 4. Models belonging to the sensitivity estimates had identical, non-unique, and unique regression coefficients, while specificity models did not have identical regression coefficients. The variance also differed between metrics, as specificity only had $\delta = 0.05$ and $\delta = 0.10$, while sensitivity also had models with $\delta = 0.20$.

Pattern F

In this pattern machine learning methods outperformed traditional statistical methods as the sample size increased, either in one move or by alternating until they performed best. Sensitivity had 48 models that displayed this behavior, whereas specificity only had 23.

Both sensitivity and specificity estimates displayed several common sub-patterns. Figure 12a is such a variation, in which the performance of ST did not improve much, instead it fluctuated around an invisible line. Moreover, the approach often only performed best for the first sample size. The performance of ML, however, had a steep increase. It either kept increasing or stabilized, similar to the figure, as the sample size increased. Models that displayed this behavior were either models with interaction depth 5 and variance $\delta = 0.10$ and varying formula complexity or models with either interaction depth 2 or 3 where the regression coefficients were either identical or non-unique for sensitivity estimates or all options for specificity estimates. We also saw several models, specific to sensitivity, in which it took ML curves quite some time to overtake the ST curves (Figure 12d). Similar patterns were observed in specificity, but in these models it either took a lot of alternating between approaches to overtake the ST curves (Figure 12b) or rather quickly (Figure 12e). Other variations in these sub-patterns are lower overall performance and less fluctuations. Models that produced these results varied in all of their settings. A sub-pattern which we observed in sensitivity estimates was when both approaches moved very close together. Shown in Figure 12c, we observed several models for which the performance was quite low and all approaches remained close together. Some models also kept increasing their performance, while keeping the proportions similar. The models that displayed this performance had either interaction depth 3 or 5 and varied the variance between $\delta = 0.10$ and $\delta = 0.20$. Their regression coefficients were unique, but formula complexity varied.

Pattern G

A pattern that we did not clearly observe in the accuracy estimates, but did in sensitivity and specificity was pattern G. In this pattern traditional statistical methods performed best for the smallest sample size(s) and for larger sample size(s). In-between machine learning methods managed to outperform for a time. An interesting observation is the fact that the pattern that a single model, *E-39*, displayed in its accuracy estimates, is now represented in pattern G. While a model with this pattern would not have been recognized as a pattern in accuracy, it was a general pattern in sensitivity and specificity. A variation spotted in both sensitivity and specificity is when the ST curves performed relatively high, somewhere between 0.70 and 0.90. ML curves either remained a way below (Figure 13a) or increased to a close vicinity. Models with this sub-pattern had interaction depth ranging from 1 to 5 and all had variance $\delta = 0.10$. Another variation, specific to specificity, in which high performance was visible is Figure 13b. In this variation ST performed best on the first observation, after which random forest did better for one point. In subsequent points LASSO regression either kept outperforming or alternating with random forest, while logistic regression took longer to improve its performance. Both models that displayed this behavior had 10 main and first-order effects, formula complexity 2, and identical regression coefficients. Specific to sensitivity are Figures 13c and 13d. Both displayed lower performances, where either the estimates stabilized as the sample size increased (Figure 13d) or ST curves separated from ML curves and kept increasing more (Figure 13c) These models had interaction depth 1 to 3, but varied between other settings.

3.1.3 Comparison Between Performance Measures

Table 8 represents confusion matrices between the different performance measures, e.g., if a model was classified as pattern A for both accuracy and sensitivity this was counted in the corresponding cell. A few notable observations. Across all matrices, pattern C consistently had many models where both measures classified them into pattern C, which indicates this pattern is strongly reflected in all measures. Only few models had classifications that were different between patterns. In both Table 8a and 8b accuracy classified many models into pattern E,

Table 8

Confusion Matrix Exploratory Analysis

		Accuracy						
	Pattern	A	B	C	D	E	F	G
Sensitivity	A	2	0	0	0	0	0	0
	B	0	4	0	1	8	0	0
	C	0	0	18	0	0	0	0
	D	4	0	0	9	0	3	0
	E	1	0	0	1	22	0	0
	F	19	0	0	12	0	17	0
	G	0	0	2	2	9	2	0

(a) *Accuracy versus Sensitivity*

		Accuracy						
	Pattern	A	B	C	D	E	F	G
Specificity	A	21	0	2	10	0	6	0
	B	0	1	0	0	0	0	0
	C	0	0	14	0	0	0	0
	D	2	0	4	11	12	5	0
	E	0	3	0	0	18	0	0
	F	3	0	0	4	5	11	0
	G	0	0	0	0	4	0	0

(b) *Accuracy versus Specificity*

		Sensitivity						
	Pattern	A	B	C	D	E	F	G
Specificity	A	1	1	0	6	1	28	3
	B	0	1	0	0	0	0	0
	C	0	0	14	0	0	0	0
	D	1	7	4	5	5	8	4
	E	0	5	0	0	10	0	6
	F	0	0	0	5	4	12	2
	G	0	0	0	0	4	0	0

(c) *Sensitivity versus Specificity*

Note. The confusion matrices demonstrate how 138 models from the exploratory analysis are characterized into specific patterns based on accuracy, sensitivity, and specificity. The distribution of models assigned to each pattern highlights both the overlap and differences in patterns between the measures.

whereas both specificity and sensitivity dispersed these models more into pattern B, D, F, and G. Both sensitivity and specificity classified many models as pattern F, whereas accuracy divided them mostly among patterns A, D, and F. As a final observation, the contrasting behavior in classification of accuracy pattern A between Tables 8a and 8b: in Table 8b almost all models were classified as pattern A in both measures, whereas in Table 8a almost all models classified as A under accuracy shifted towards pattern F in sensitivity.

3.2 In-Depth Analysis

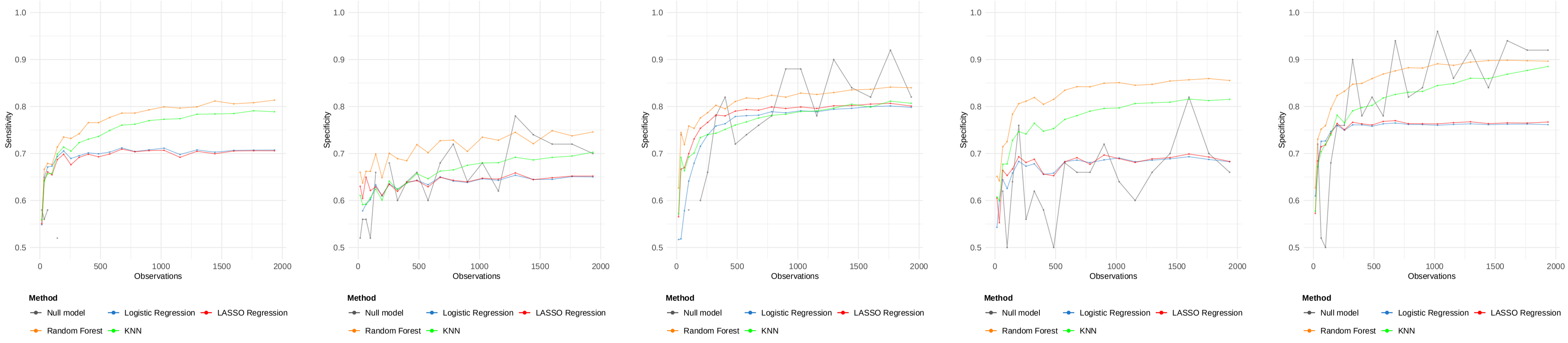
For the in-depth analysis we selected 20 median models, with observation scale, x^2, where $x = [4, 8, \ldots, 40, 44]$. Median models were selected from each pattern and performance measure, where interaction depth was used as the primary criterion to sort the models. This meant that six median models were selected for accuracy and seven each for sensitivity and specificity. The scope of this analysis is limited, as we only reviewed a few models as described in Table 9. The patterns found in the exploratory analysis are linked to those identified during this in-depth analysis, listed in Table 10. We will review whether more computationally expensive methods demonstrated similar behavior to the methods from the exploratory analysis or if, by using more methods, some patterns disappear. A selection of the results are shown in Figure 14, while the remaining can be found on the GitHub page referred to in Appendix B.

3.2.1 Accuracy

The exploratory analysis identified six different patterns apparent in the accuracy. Of the 20 models, seven retained the same pattern in both analyses, while the others changed. We will discuss what changes were observed in each pattern, summarized in Table 10.

Figure 8

Pattern A: Sensitivity and Specificity Performance

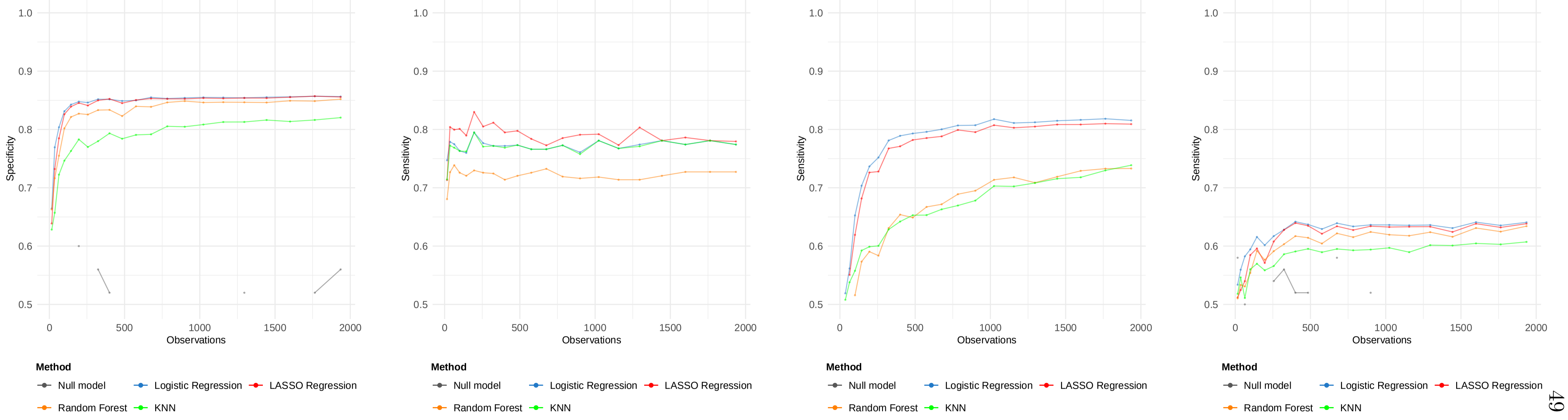


(a): E-86 (Sensitivity) (b): E-29 (Specificity) (c): E-50 (Specificity) (d): E-77 (Specificity) (e): E-95 (Specificity)

Note. This figure shows sensitivity and specificity estimates on the test data measured across different training sample sizes for pattern A. A selection of exploratory models is displayed, for which the settings are available in Tables 1 to 6.

Figure 9

Pattern B: Sensitivity and Specificity Performance

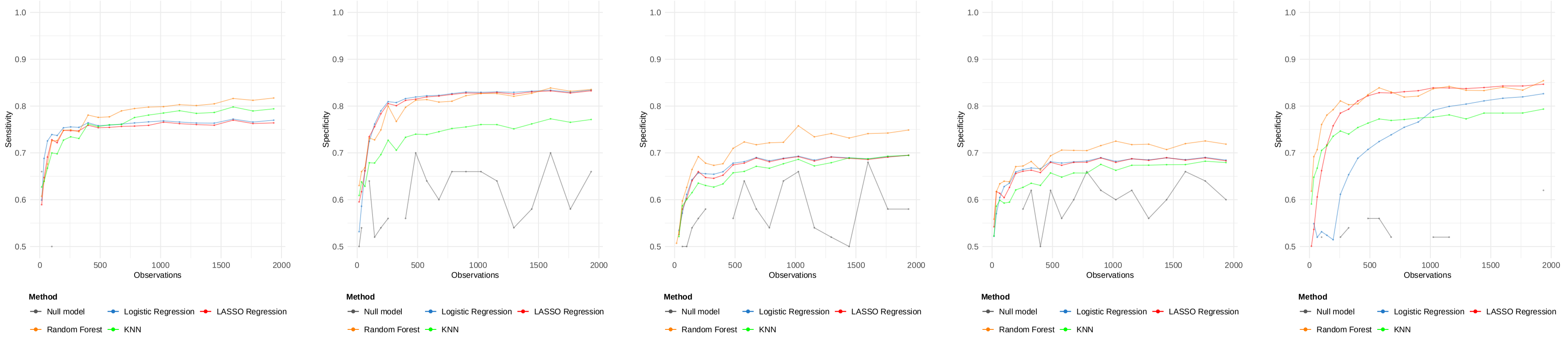


(a): E-2 (Specificity) (b): E-39 (Sensitivity) (c): E-48 (Sensitivity) (d): E-129 (Sensitivity)

Note. This figure shows sensitivity and specificity estimates on the test data measured across different training sample sizes for pattern B. A selection of exploratory models is displayed, for which the settings are available in Tables 1 to 6.

Figure 10

Pattern D: Sensitivity and Specificity Performance

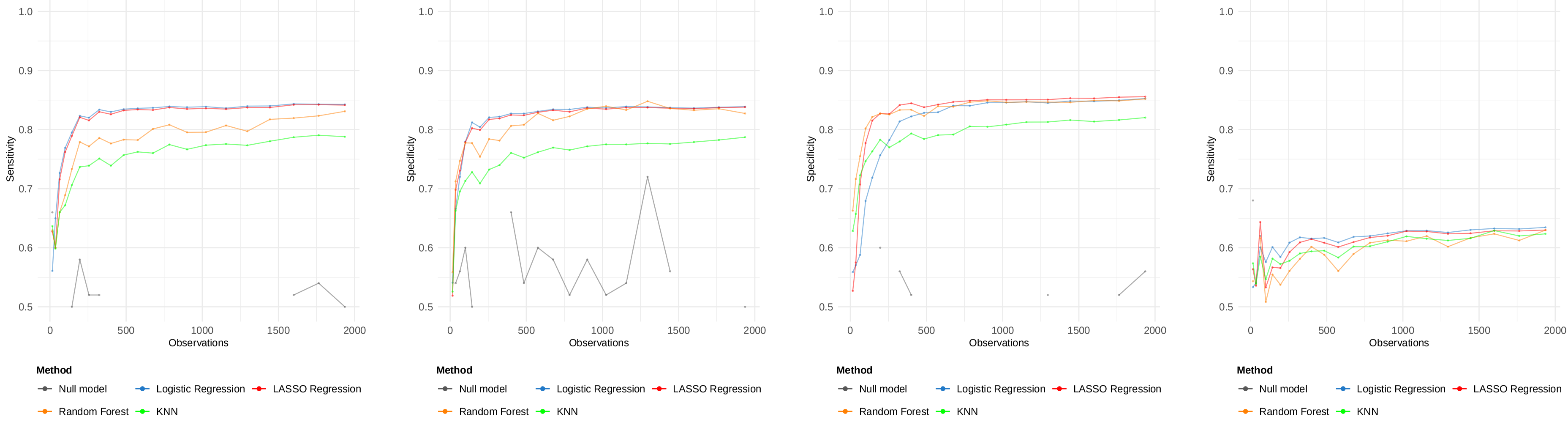


(a): E-17 (Sensitivity) (b): E-13 (Specificity) (c): E-87 (Specificity) (d): E-32 (Specificity) (e): E-35 (Specificity)

Note. This figure shows sensitivity and specificity estimates on the test data measured across different training sample sizes for pattern D. A selection of exploratory models is displayed, for which the settings are available in Tables 1 to 6.

Figure 11

Pattern E: Sensitivity and Specificity Performance

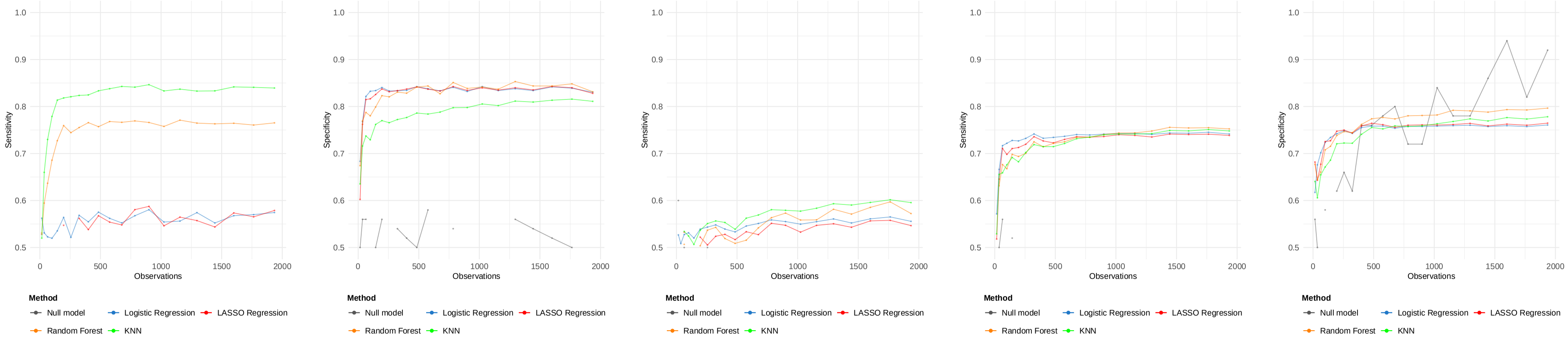


(a): E-75 (Sensitivity) (b): E-75 (Specificity) (c): E-33 (Specificity) (d): E-96 (Sensitivity)

Note. This figure shows sensitivity and specificity estimates on the test data measured across different training sample sizes for pattern E. A selection of exploratory models is displayed, for which the settings are available in Tables 1 to 6.

Figure 12

Pattern F: Sensitivity and Specificity Performance



(a): E-27 (Sensitivity)

(b): E-67 (Specificity)

(c): E-24 (Sensitivity)

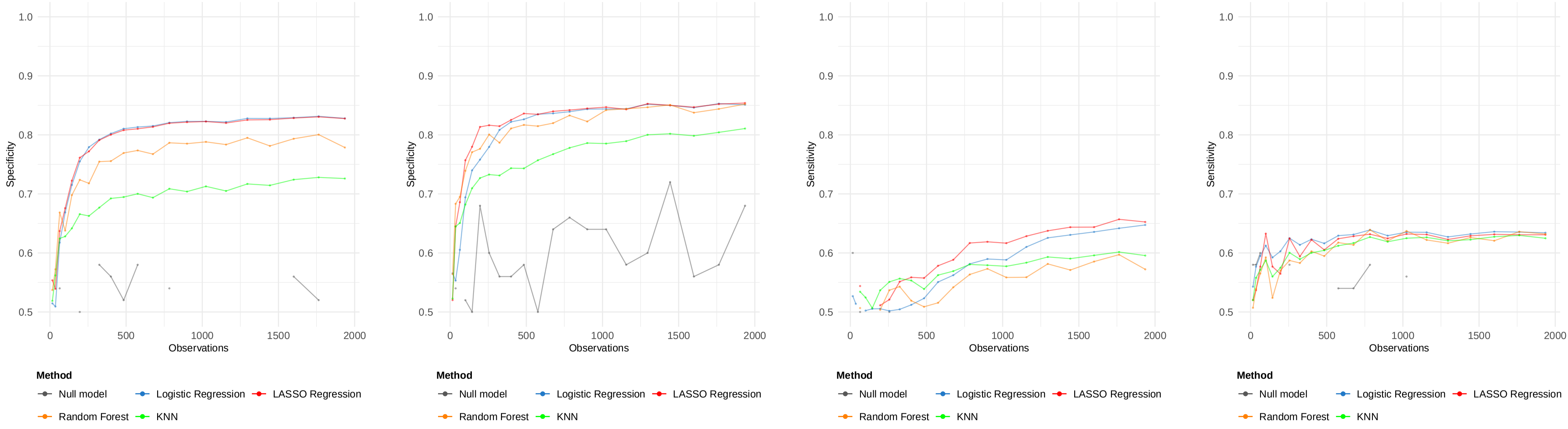
(d): E-127 (Sensitivity)

(e): E-127 (Specificity)

Note. This figure shows sensitivity and specificity estimates on the test data measured across different training sample sizes for pattern F. A selection of exploratory models is displayed, for which the settings are available in Tables 1 to 6.

Figure 13

Pattern G: Sensitivity and Specificity Performance



(a): E-25 (Specificity)

(b): E-101 (Specificity)

(c): E-53 (Sensitivity)

(d): E-123 (Sensitivity)

Note. This figure shows sensitivity and specificity estimates on the test data measured across different training sample sizes for pattern G. A selection of exploratory models is displayed, for which the settings are available in Tables 1 to 6.

Table 9

In-Depth Simulation Studies

	Accuracy						Sensitivity							Specificity						
	D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9	D-10	D-11	D-12	D-13	D-14	D-15	D-16	D-17	D-18	D-19	D-20
Corresponding Pattern	A	B	C	D	E	F	A	B	C	D	E	F	G	A	B	C	D	E	F	G
Corresponding Exploratory Model	84	89	49	111	11	128	86	13	41	95	12	125	46	118	2	52	114	104	120	9
Number of covariates																				
Main effects	10	10	10	10	30	10	10	30	10	10	30	10	10	10	10	30	10	10	10	30
First order interactions	5		40	5	2	5	5	4	1	5	3	5	10	5		5	5	10	5	1
Second order interactions	3			3		3	3			3		3		3		3	3		3	
Number of regression coefficients																				
Main effects	2	10	10	2	30	10	5	30	10	10	30	10	10	5	10	30	5	2	5	30
First order interactions	2		40	2	2	5	5	4	1	5	3	5	10	3		5	5	2	3	1
Second order interactions	2			2		3	3			3		3		3		3	3		3	
Variation in the outcome	0.10	0.05	0.10	0.10	0.10	0.05	0.10	0.10	0.10	0.05	0.10	0.05	0.10	0.10	0.10	0.10	0.05	0.10	0.10	0.10
Number of continuous covariates	0	0	0	0	0	10	0	0	0	0	0	5	0	1	0	0	0	0	10	0
Formula Complexity	1	1	3	2	1	1	1	1	3	1	1	1	2	1	1	2	2	2	1	1

Note. This table illustrates the different simulated models that were run for the in-depth analysis. No complexity parameters were fixed. For each interaction depth the number of main effects or number of interaction effects is specified, as well as the number of unique regression coefficients. Abbreviation *D* denotes models of the in-depth analysis. The *Corresponding Pattern* reference denotes patterns identified in the exploratory simulation study in section 3.1 and *Corresponding Exploratory Model* denotes models in Tables 1 - 6.

Pattern A Both models that were found to be part of pattern A in the exploratory analysis, remained pattern A in the in-depth analysis, as can be seen in Figure 14a.

Pattern B While models *D-2* and *D-15* (Figure 14b) were part of pattern B in the exploratory analysis, this pattern was no longer observed in the in-depth analysis. Instead, both now displayed a pattern similar to pattern E. In both models we observed that the first sample size ($n = 16$) was the point which now had changed, as now a machine learning method (GBM or linear SVM) outperformed the traditional statistical methods.

Pattern C Three models were initially classified in this pattern in the exploratory analysis. In the subsequent in-depth analysis one model remained in the same pattern, whereas two models changed to pattern A. Both models that changed from pattern C to A between analyses were models with formula complexity 3 and 10 main effects. The number of first-order interactions was either 1 or 40. Model *D-3* is shown in Figure 14c. The model that stayed in pattern C had formula complexity 2, 30 main effects, 5 first- and 3 second-order interactions. This also reflected what we saw in the exploratory analysis, as the number of main order effects increased that curves often shifted more towards the right. Apparent is the fact that machine learning methods, performed well, also for smallest sample sizes. The model still classified as pattern C (Figure 14d) showed that LASSO regression took until a sample size higher than 1250 to outperform GBM. Afterwards, both approaches remained close together. We also observed that regression methods ridge and logistic regression often lagged behind on elastic net and LASSO regression. This is reasonable, as the latter have the ability to set coefficients to zero, whereas the former can only shrink effects without removing them from the model.

Pattern D Initially four models were characterized as pattern D in the exploratory analysis. No were left in the in-depth analysis, as all models had changed to pattern A, as can be seen in Figures 14e and 14f. This meant that instead of traditional statistical methods overtaking machine learning methods for a limited time, somewhere at the beginning or in the middle, machine learning methods were now superior.

Pattern E Six models were classified as pattern E in the exploratory analysis. Only two remained in pattern E in the in-depth analysis. The four other models changed to pattern D. Models *D-5* and *D-20* were both in the same pattern in the exploratory analysis, but now found themselves in different patterns for the in-depth analysis. The only difference between the two models is one extra first-order interaction term. While in model *D-20* (Figure 14h), machine learning methods did better for the first observation, especially linear SVM, traditional statistical methods, especially logistic regression, soon took over. In contrast to model *D-5* (Figure 14g), where machine learning performed best for the first two and last sample sizes. All other points had better performance from traditional statistical methods. Although the differences between approaches, as sample size increased, were very small.

Pattern F Three models were categorized as pattern F in the exploratory analysis. Two remained in this pattern, but one transformed into pattern A. For example, in model *D-6* (Figure 14i) a similar pattern to Figure 6f of the exploratory analysis is seen. The major difference was the addition of more methods, while in the exploratory analysis the traditional statistical methods outperformed machine learning methods for sample sizes below 500, in model *D-6* logistic regression outperformed machine learning in the first sample size. In model *D-12* (Figure 14j) pattern A persisted, where radial SVM outperformed all others. In the original model *E-125*, machine learning methods had surpassed traditional statistical methods well before a sample size of 250. Now, they had the best estimates from the start.

The newly classified in-depth models in each pattern often did not have one distinctive identifier. In all patterns, except C and F, several formula complexity options were included. Pattern F only included two models with formula complexity 1, whereas pattern C included one model with formula complexity 2. The variance was constant at $\delta = 0.10$ for pattern D, whereas in other patterns with more than one model both $\delta = 0.10$ and $\delta = 0.05$ were included. As this concerned median models, the interaction depth ranged only between 1 and 3. Pattern A, consisted of models with interaction depth 2 or 3, D only included models with interaction depth 2, E consisted of models with interaction depth 1 or 2, and F had two models with interaction depth 3. Observed was also that in pattern F, only two models were included, in which the number of main effects was equal to the number of continuous covariates. Pattern A had models with no or several continuous covariates, whereas pattern C, D, and E did not include any continuous covariates. The number of unique regression coefficients varied in patterns A, D, and F. The exploratory analysis identified six general patterns in which we could categorize almost all models. After using additional methods from both approaches, often more computationally expensive, we found that only five patterns remained. The majority of the models (9) could be classified as pattern A. Other patterns that still had models included were pattern C (1), D (4), E (4), and F (2). Pattern B was no longer observed among the in-depth models.

3.2.2 Sensitivity and Specificity

The exploratory analysis identified seven patterns apparent in the sensitivity and specificity results. Sensitivity retained eight models in the original pattern, whereas sensitivity kept five.

Pattern A In the exploratory analysis sensitivity included one model with pattern A and specificity included six models. The model belonging to sensitivity, *D-7*, was still in pattern A in the in-depth analysis. This was not the case for specificity, as only two out of six models were still in pattern A when more methods were added. The remaining four either transformed into pattern D (2), such as model *D-3* in Figure 14k, or F (2).

Pattern B Of the four sensitivity models in pattern B in the exploratory analysis, none remained in pattern B for the in-depth analysis. Instead, two were designated to pattern E and two to pattern D. For the specificity model estimates, we can see that there was only one model with pattern B in the exploratory analysis, which changed to pattern D in the in-depth analysis.

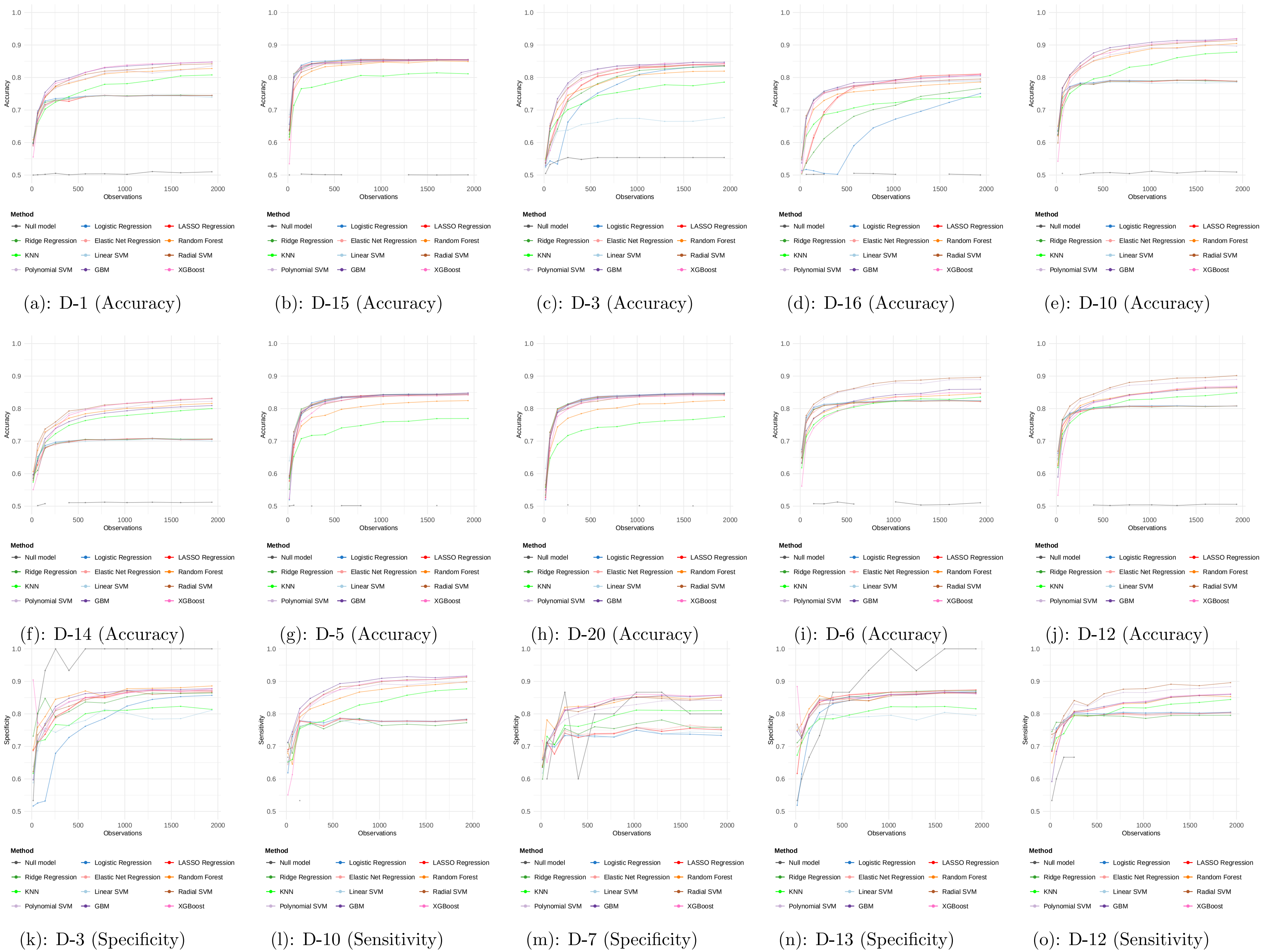
Pattern C Of the models we used for the in-depth analysis two models for both sensitivity and specificity belonged to pattern C during the exploratory analysis. None remained in this pattern. Instead, for the sensitivity estimates one model was now categorized in pattern A and the other in pattern D. For the specificity we observed that both models were now characterized as pattern D.

Pattern D In the exploratory analysis we categorized one model in pattern D for sensitivity, while specificity had three models. For sensitivity this model was no longer pattern D, but had changed into pattern F as seen in Figure 14l, where ridge regression did better for the first sample size, after which GBM had superior performance. For the specificity two models were still included in pattern D, while the other model was now pattern A (Figure 14m).

Pattern E Initially two models were classified as pattern E for sensitivity and five for specificity. For the sensitivity one model retained the same pattern in the in-depth analysis, whereas the other model was now pattern D. For specificity four models changed their pattern to D and only one remained pattern E (Figure 14n).

Figure 14

In-Depth Analysis: Performance



(a): D-1 (Accuracy)

(b): D-15 (Accuracy)

(c): D-3 (Accuracy)

(d): D-16 (Accuracy)

(e): D-10 (Accuracy)

(f): D-14 (Accuracy)

(g): D-5 (Accuracy)

(h): D-20 (Accuracy)

(i): D-6 (Accuracy)

(j): D-12 (Accuracy)

(k): D-3 (Specificity)

(l): D-10 (Sensitivity)

(m): D-7 (Specificity)

(n): D-13 (Specificity)

(o): D-12 (Sensitivity)

Note. This figure shows the estimates for accuracy (a-j), sensitivity (l, o), and specificity (k, m, n) on the test data measured across different training sample sizes for a selection of in-depth models. The settings are available in Table 9.

Pattern F In the exploratory analysis seven models were included in this pattern for sensitivity, for specificity only two. Four models' sensitivity patterns remained F, while the remaining shifted to pattern A (2) and D (1) (Figure 14o). For specificity one model remained pattern F, while the other changed to pattern A when more methods were included.

Pattern G Sensitivity had three models belonging to pattern G in the exploratory analysis, specificity only had one. For the in-depth analysis, two models remained pattern G and one changed to pattern F for the sensitivity. The specificity model changed to pattern D.

The settings for the in-depth models, in their new patterns, still varied a great deal. Some observations: the interaction depth was either 2 or 3 for patterns A, D, and F, whereas pattern E had interaction depth 1 or 2. The only pattern with a single interaction depth, namely 2, was pattern G. Patterns D and E had unique regression coefficients, whereas the others also had non-unique coefficients. The variation was $\delta = 0.10$ for pattern G, whereas other patterns also had $\delta = 0.05$. No continuous covariates were included in patterns E and G. The formula complexity varied for most patterns, except for pattern E who had formula complexity 1, and pattern G who had formula complexity 2. For specificity both pattern A and F had interaction depth 3, whereas the other models had more options. Except for pattern D and E, all patterns contained continuous covariates. Most other settings varied. The exploratory analysis identified seven general patterns in total. For the sensitivity results we found that five patterns were still included and only four patterns remained for specificity. For sensitivity the patterns are relatively evenly spread pattern A (4), D (5), E (3), F(6), and G (2), whereas for specificity the majority of the models (12) was pattern D. The remaining models for specificity were A (4), E (1), and F (3). Both pattern B and C were no longer present for both metrics. Moreover, in specificity pattern G had disappeared as well.

3.2.3 Comparison Between Performance Measures

From Table 11, several observations can be highlighted. Models classified as Pattern D under accuracy consistently classified all or almost all models the same in both measures. Matrices 11a and 11b also showed a similar trend of models that were classified as pattern A by accuracy to either pattern D or F. In Table 11a the majority was classified under pattern F by sensitivity, whereas 11b classified most in specificity pattern D. Furthermore, the dispersion of models classified into pattern D under specificity in Table 11b shows that under accuracy, the models were dispersed between patterns A, C, D, and E. This was also observed in Table 11c, including pattern G as well.

We have seen that not all patterns identified in the exploratory analysis withstood the increase in computationally expensive methods. In all performance measures, pattern B was no

Table 10

Side-by-side Comparison of Original and In-Depth Patterns

	Accuracy		Sensitivity		Specificity	
Model	Exploratory	In-Depth	Exploratory	In-Depth	Exploratory	In-Depth
D-1	A	A	F	F	A	D
D-2	B	E	B	E	E	D
D-3	C	A	G	F	A	D
D-4	D	A	F	F	A	A
D-5	E	D	B	D	E	D
D-6	F	F	F	F	F	A
D-7	A	A	A	A	D	A
D-8	E	D	B	D	D	D
D-9	C	A	C	A	C	D
D-10	D	A	D	F	A	F
D-11	E	D	E	D	E	D
D-12	F	A	F	D	A	F
D-13	E	E	G	G	E	E
D-14	D	A	F	F	A	A
D-15	B	E	B	E	B	D
D-16	C	C	C	D	C	D
D-17	D	A	F	A	D	D
D-18	E	D	G	G	E	D
D-19	F	F	F	A	F	F
D-20	E	E	E	E	G	D

Note. This table highlights the patterns into which the 20 in-depth models were classified, compared to the patterns assigned to their corresponding exploratory models.

longer present. As we only inspected 20 in-depth models, this analysis was quite limited. The disappearance of patterns, such as B, C, and G, does not mean other future models will never exhibit these patterns again.

Table 11

Confusion Matrix In-Depth Analysis

		Accuracy						
	Pattern	A	B	C	D	E	F	G
Sensitivity	A	3	0	0	0	0	1	0
	B	0	0	0	0	0	0	0
	C	0	0	0	0	0	0	0
	D	1	0	1	3	0	0	0
	E	0	0	0	0	3	0	0
	F	5	0	0	0	0	1	0
	G	0	0	0	1	1	0	0

(a) *Accuracy versus Sensitivity*

		Accuracy						
	Pattern	A	B	C	D	E	F	G
Specificity	A	3	0	0	0	0	1	0
	B	0	0	0	0	0	0	0
	C	0	0	0	0	0	0	0
	D	4	0	1	4	3	0	0
	E	0	0	0	0	1	0	0
	F	2	0	0	0	0	1	0
	G	0	0	0	0	0	0	0

(b) *Accuracy versus Specificity*

		Sensitivity						
	Pattern	A	B	C	D	E	F	G
Specificity	A	1	0	0	0	0	3	0
	B	0	0	0	0	0	0	0
	C	0	0	0	0	0	0	0
	D	2	0	0	4	3	2	1
	E	0	0	0	0	0	0	1
	F	1	0	0	1	0	1	0
	G	0	0	0	0	0	0	0

(c) *Sensitivity versus Specificity*

Note. The confusion matrices demonstrate how 20 models from the in-depth analysis are characterized into specific patterns based on accuracy, sensitivity, and specificity. The distribution of models assigned to each pattern highlights both the overlap and differences in patterns between the measures.

Chapter 4

Case Studies

In this chapter we look at several case studies to investigate whether the patterns we uncovered in Chapter 3 are also reflected in data that was not simulated. Three data sets were selected and briefly discussed. Next, a general description is given on how the data was analyzed. As the general approach was the same as in Chapter 2 we focus on aspects that were changed. Finally, we present the results and examine how they relate to the results obtained in Chapter 3.

4.1 The Framingham Study

The Framingham Study is considered to be one of the longest-lasting longitudinal studies in the cardiovascular domain. The study started in 1948 and has since recorded data on multiple generations. From each participant medical data is gathered, through questionnaires and physical exams. Researchers also gather non-medical data in regard to lifestyle, to create a complete picture of the patient. Over the course of the longitudinal study, researchers were able to identify key risk factors for several heart-related illnesses, such as heart failure, as well as other diseases, i.e., neurological. Many more developments came from the Framingham heart study (Andersson et al., 2021).

In this study we have used the Framingham data set from the R-package *riskCommunicator* (v1.0.1; Grembi, 2022). It should be noted that the data set is provided as a teaching dataset and was altered to make sure data remained anonymous. Therefore, we cannot make any claims about actual findings in this data. Our only goal is to find whether non-simulated data provides similar patterns as simulated data. The data from this package is a subset of data on 4434 participants collected at three time points, 1956 through 1968, resulting in a total of 11627 observations. In this study we used only data points from the first period, to ensure the independence of the observations. We selected 18 covariates of interest to predict the outcome variable, cardiovascular disease (CVD). Due to the nature of some of the methods used in this study, we were unable to keep missing values. The percentage of missing values was only 0.80%, which meant we used a complete case analysis. The final sample consisted of 3826 participants, of which 1011 (26.42%) experienced cardiovascular disease during follow-up. A total of 1731 (45.24%) participants were male and 2095 (54.76%) were female. Their ages ranged between 32 and 70 years (M = 49.90, SD = 8.68). Numerical values included were Serum Total Cholesterol, which ranged between 113 and 600 mg/dL (M = 237.11, SD = 44.23); systolic blood pressure, which ranged between 83.5 to 295 mmHg (M = 132.96, SD = 22.46); diastolic blood pressure, which ranged between 48 and 142.5 mmHg (M = 83.11, SD = 12.11); heart rate, which ranged between 44 and 143 beats per minute (M = 75.75, SD = 12.08); BMI, which ranged between 15.54 and 56.8 (M = 25.82, SD = 4.08); the number of cigarettes smoked per day, which ranged from 0 to 70 (M = 8.99, SD = 11.94); and glucose, which ranged between 40 and 394 mg/dL (M = 82.09, SD = 24.42).

Other variables indicated that 48.72% (n = 1864) of participants currently smoked, 2.85% (n = 109) were diabetic, and 3.40% (n = 130) currently used anti-hypertensive medication. The sample also included information on disease prevalence: 4.39% (n = 168) had coronary heart disease, 3.27% (n = 125) had angina pectoris, 2.04% (n = 78) had myocardial infarction, 0.68% (n = 26) had had a stroke, and 32.44% (n = 1241) were hypertensive. Education was measured on a 1-4 scale, using the following categories: 0-11 years (n = 1619), high school or GED (n = 1127), some college (n = 631), and college graduate or higher (n = 449). The information on the meaning of education was pulled from the *cvdd* dataset, a subset of the data we used, also present in the *riskCommunicator* package.

4.2 Diabetes in 130 US hospitals (1999-2008)

We use data supplied by the UCI Machine Learning Repository (Clore et al., 2014), which was originally used in the study by Strack et al. (2014). Strack et al. (2014) used data that encompassed demographic and medical information of patients at 130 hospitals in the US from 1999 to 2008. Their goal was to find factors that are important in predicting whether someone is readmitted to the hospital within 30 days. The data from the repository consisted of 101766 observations and 50 covariates. We partially followed the preprocessing steps of the authors. Several variables (medical specialty, payer code, and weight) had quite some missing values, leading to removal of these variables. While the authors kept medical specialty, we did not. Strack et al. (2014) explain the absence of weight data by the lack of legislation to capture data in an organized format. The percentage of missing values after this removal was only 0.09%, which led to a complete case analysis. Similar to the authors, we recoded the outcome variable. It initially consisted of three values (within 30 days, after 30 days, no readmission), which was recoded to within 30 days and otherwise. Moreover, since data spanned so many years, patients visited the hospital multiple times. To remain statistically independent, only the first patient visit was kept. Other preprocessing steps we took were recoding variables containing diagnoses, to shrink the number of categories, originally around 700, to only nine. Other variables without any variation were also removed, as they would only increase computational time without contributing information.

This resulted in processed data with 68630 observations and 41 covariates. This sample consisted of 6126 (8.93%) participants to be readmitted within 30 days. A total of 32047 (46.70%) participants were male, 36582 (53.30%) were female, and one was unknown. The study included the race: Caucasian (n = 52842), African American (n = 12665), Hispanic (n = 1477), Asian (n = 485), and other (n = 1161). Age was discretized, where those in $[70, 80)$ (n = 17643) and $[60, 70)$ (n = 15414) occurred most often, followed by $[50 - 60)$ (n = 11999) and $[80 - 90)$ (n = 11247). Due to the extensiveness of all categories, we only highlight the largest categories and the covariates that were of interest. We do not discuss the admission type, the reason for discharge, or the admission source. Continuous covariates that were included are the time in hospital, which ranged between 1 and 14 days (M = 4.32, SD = 2.96); number of lab tests done, which ranged between 1 and 132 (M = 43.13, SD = 20.01); number of procedures (not lab), which ranged between 0 and 6 (M = 1.44, SD = 1.76); number of medications administered, which ranged between 1 and 81 (M = 15.81, SD = 8.29); number of outpatient visits in the previous year, which ranged between 0 and 42 (M = 0.29, SD = 1.08); number of emergency visits in the previous year, which ranged between 0 and 42 (M = 0.11, SD = 0.52); number of inpatient visits in the previous year, which ranged between 0 and 12 (M = 0.18, SD = 0.61); and number of diagnoses in the system, which ranged between 3 and 16 (M = 7.35, SD = 1.89).

Discrete covariates included were many with the same categories. We will not discuss them individually, but they indicated whether the dosage of the drug was increased or decreased, if the dosage did not change, and 'no' if no drug was prescribed. Other variables included whether the glucose serum test was taken, if so, the categorical range was recorded. For 65271 participants this test was not taken, 1686 had normal results, 939 participants had results higher than 200 mg/dL, and 734 had results higher than 300 mg/dL. Similar is the A1c test, where 56349 participants had not taken the test, 3683 had normal results, 2805 had results higher than 7% and lower than 8%, and 5793 had results greater than 8%. Additionally, for 52003 participants (75.77%) diabetes medication was prescribed. For 30707 (44.74%) there was some change in diabetes medication (dosage or name). The last covariates were three (recoded) variables that indicated the primary (1), secondary (2), and additional secondary diagnosis (3). The categories were circulatory, diabetes, digestive, genitourinary, injury, musculoskeletal, neoplasms, respiratory, and other.

4.3 Census Income

The third case study is the census income dataset (the adult dataset), provided by UCI Machine Learning Repository (Becker & Kohavi, 1996). This dataset was made from the US Census Bureau Database in 1994. The goal of this data set was to accurately predict whether a participant earns more than 50,000 dollars (50K) a year. Included information were elements of demographic and socio-economic information. The original dataset was provided in separate train and test files. We merged these files together to create one data set, because we used our own train-test split instead of the pre-specified split. The raw data consisted of 48842 subjects. The percentage of missing values was 0.88%, which was low enough such that complete case analysis could be used. Two covariates were removed as they did not add any information for prediction, i.e. *fnlwgt* represented a weight and *education-num* corresponded to *education*. The final data had 45222 observations and 13 covariates. The outcome variable predicts whether the income is higher than 50K a year. In the data 11208 subjects (24.78%) earned an income higher than 50K.

A total of 30527 (67.50%) participants were male and 14695 (32.50%) were female. Their ages ranged between 17 and 90 years (M = 38.55, SD = 13.22). Numerical values included were the capital gain, which ranged between 0 and 99999 dollars (M = 1101.43, SD = 7506.43); capital loss, which ranged between 0 and 4356 (M = 88.60, SD = 404.96); and hours worked per week, which ranged between 1 and 99 hours (M = 40.94, SD = 12.01). Discrete variables included the working class, with several categories to represent it: private (n = 33307), federal government (n = 1406), local government (n = 3100), incorporated self-employment (n = 1646), unincorporated self-employment (n = 3796), state government (n = 1946), and without pay (n = 21). The marital status was also included, using the following categories: divorced (n = 6297), married with spouse in the Armed Forces (n = 32), married to a civilian spouse (n = 21055), married with spouse absent (n = 552), never married (n = 14598), separated (n = 1411), and widowed (n = 1277). Occupation was specified using 14 categories, of which the largest five were crafting or repairing (n = 6020), professional specialty occupations (n = 6008), executive or managerial position (n = 5984), administrative and clerical work (n = 5540), and sales (n = 5408). The main relationship of the subject to the head of the household was indicated: husband (n = 18666), wife (n = 2091), unmarried (n = 4788), not in family (11702), other relative (n = 1349), and own child (n = 6626). Race was included, using the following categories: American-Indian-Eskimo (n = 435), Asian-Pacific-Islander (n = 1303), Black (n = 4228), White (n = 38903), and other (n = 353). The final discrete variable included was the native country of the subject, which consisted of 41 categories, the largest two being the Unites States (n = 41292) and Mexico (n = 903).

4.4 Analysis Approach

In this section we describe the changes to the analysis approach to analyze the case studies. No data simulation was performed, which meant a majority of the complexity parameters could not be used because their usage was for data generation. The only remaining complexity parameter that could be used was the *formula complexity*, which we have varied between 1 and 2.

Each case study was highly imbalanced, i.e. Framingham only had 26.42% of CVD cases, the diabetes data consisted of only 8.93% participants that were readmitted within 30 days, and in the census income data only 24.78% earned more than 50K. To ensure good predictions could

Table 12

Case Studies: Balanced Observations and Train-Test Split

	Balanced Data Set			Train-Test Data	
	Positives	Negatives[a]	Total Observations	Training (N)	Testing
Framingham Study	1011	1011	2022	1618	404
Diabetes Study	6126	6126	12252	9802	2450
Income Census	11208	11208	22416	17934	4482

Note: This table described the number of observations in each balanced data set and the number of observations that was used to train and test the data. The data is partitioned in 80% training and 20% test. [a] The original negative cases for each case study were as follows: Framingham ($n = 2815$), Diabetes Study ($n = 62504$), and Income Census ($n = 34014$).

be made, we made a similar analysis as in our simulated data. We created a balanced data set for each case study. This meant we selected all positive cases and randomly sampled an equal number from the negative cases. This guaranteed a data set with 50% positive and 50% negative cases. Table 12 describes how many observations were included in the final data sets, as well as how many observations were included in the training and test datasets. Similar to continuous covariates in the simulations, in the case study datasets continuous covariates were standardized in order to get $\mu = 0$ and $\sigma = 1$.

The balanced data set was divided into training and test data, using an 80-20 split, which was also balanced. The difference with our simulation study is that we did not simulate any data. Instead, we randomly sampled n observations from the training data set with N observations, m times. We did not take additional measures to ensure a balanced sample for each sample size n, as random sampling from the balanced training data should create representative distributions.

To limit computing time, we mainly used the exploratory analysis. We expected this to take relatively long, as the case studies often had many more covariates than we previously considered. The sample size scale and methods used are the same, except for the Framingham study. Due to the sample size restrictions, we could not sample n observations for all sample sizes. As the Framingham training data consisted of $N = 1618$, we restricted the maximum sample size to $n = 1600$, i.e., $x = [4, 6, \ldots, 38, 40]$. Moreover, we used seed 9 through 13 and 10 replications for exploratory analyses and seed 14 through 16 and 5 replications for in-depth analyses.

4.5 Results

We describe how our case study accuracy results related to the simulation study results. As accuracy was our main focus, we ignored sensitivity and specificity which are documented on the GitHub page (Appendix B). Moreover, we solely describe in which pattern they would fall, as we cannot make any claims about the underlying relationships in the data since they are unknown.

We first discuss the Framingham data set. Figures 15a and 16a would both be characterized as pattern D. Both analyses demonstrated a situation in which machine learning methods performed best for the smallest sample size(s) and when sample size became larger. At a certain point. traditional statistical methods performed best for a limited time. In Figure 15a this pattern repeated itself only once, whereas in Figure 16a machine learning and traditional statistical methods alternated more than once. Figures 15b and 16b, in which formula complexity 2 was used, were not in agreement on the pattern. Figure 15b demonstrated the exploratory analysis which reflected pattern D. Random forest and LASSO regression alternated for the top spot. This in contrast to Figure 16b, where machine learning methods performed best all throughout the analysis, matching pattern A.

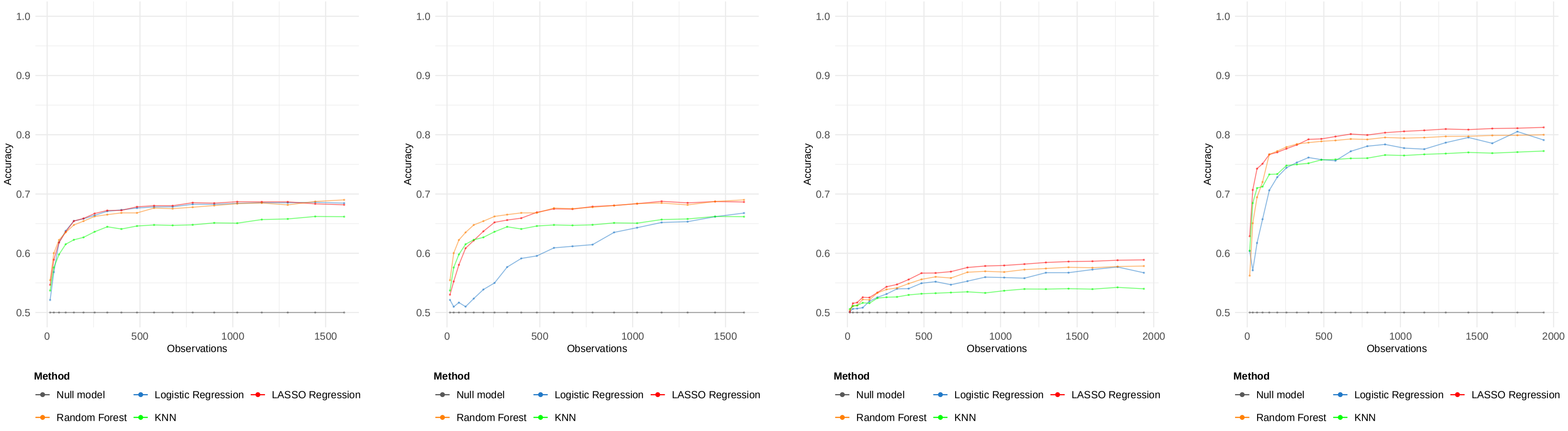
Apparent in both the exploratory and in-depth analysis is the difference between which methods performed best. Random forest and LASSO regression were no longer the highest-performing methods. In the in-depth analysis random forest only performed best on the smallest sample size(s). When formula complexity was 1, ridge regression was the traditional statistical method of choice. Moreover, the radial and polynomial kernel of support vector machines were the machine learning methods that mainly performed best, in both analyses. As the sample size increased, at one point linear support vector machine performed best in formula complexity 2.

Second, the diabetes case study is shown in Figures 15c and 16c. The exploratory analysis figure showcased very low performance, similar to what we had observed in some other models (Figures 4d or 3e). KNN did best at the first point, after which LASSO regression started outperforming. Hence, the most likely pattern would be pattern E. We could argue for pattern C, as logistic regression did not catch up to LASSO regression and we did not use formula complexity 2. However, the curve was not as drastic as we found in pattern C, as well as the fact that it did catch up with random forest. Therefore, it is inconclusive which pattern this model belonged to, either pattern C or E. The in-depth analysis (Figure 16c) with formula complexity 1, demonstrated similarly low performance, but showed a different pattern, namely pattern G. Methods alternated between traditional statistical methods elastic net and LASSO regression and machine learning method linear SVM. Machine learning methods such as GBM and radial SVM did come close to the best performing curves. We should note that logistic regression is still on the lower end of performance estimates. Based on the low performance, also displayed by machine learning methods, the data could be too complex for the logistic regression model.

Finally, Figures 15d and 16d demonstrate exploratory and in-depth analyses on the census income data set, with formula complexity 1. The exploratory analysis would fall either under pattern C, as logistic regression never fully caught up with LASSO regression, or pattern G. We are undecided, as logistic regression did lag more, than what we had previously observed in pattern G of the simulation study. However, pattern C included models with formula complexity 2, whereas we specified formula complexity 1. The in-depth analysis showed a similar logistic regression pattern, but would be classified as pattern D, as it was no longer a regression method that outperformed machine learning methods. Instead, elastic net was briefly able to overtake linear SVM at the second point, after which GBM permanently outperformed all other methods.

Figure 15

Exploratory Analysis



(a): Framingham: Formula Complexity 1

(b): Framingham: Formula Complexity 2

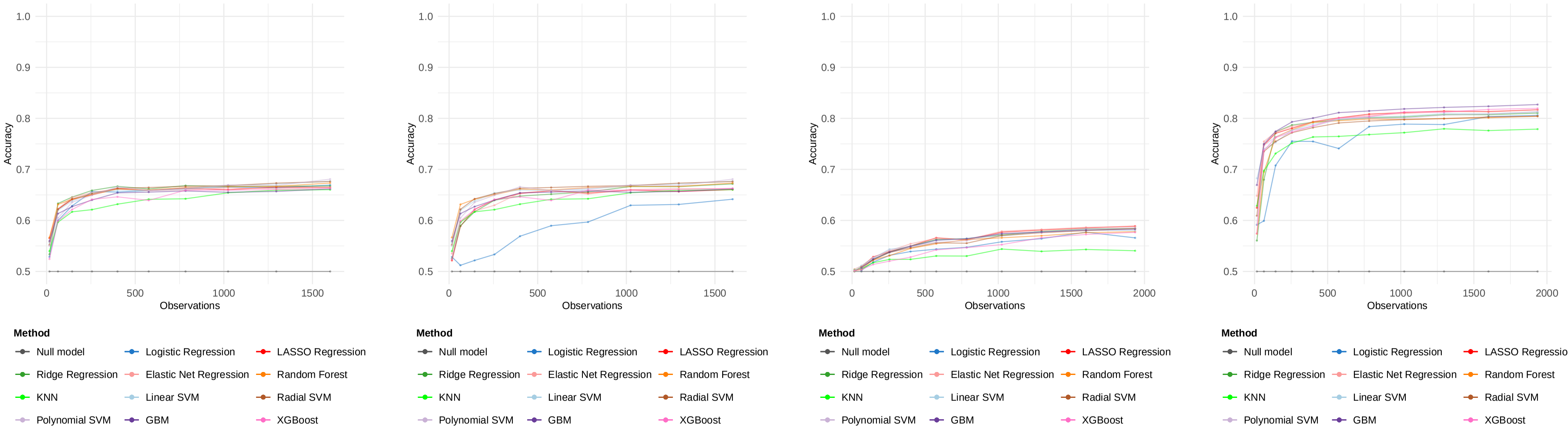
(c): Diabetes: Formula Complexity 1

(d): Census Income: Formula Complexity 1

Note. This figure shows accuracy estimates on the test data measured across different training sample sizes for three different case studies: Framingham, Diabetes, and Census Income using the exploratory analysis.

Figure 16

In-Depth Analysis



(a): Framingham: Formula Complexity 1

(b): Framingham: Formula Complexity 2

(c): Diabetes: Formula Complexity 1

(d): Census Income: Formula Complexity 1

Note. This figure shows accuracy estimates on the test data measured across different training sample sizes for three different case studies: Framingham, Diabetes, and Census Income using the in-depth analysis.

Chapter 5

Discussion

5.1 Discussion

In this thesis, we conducted simulations in order to find under which circumstances traditional statistical methods and machine learning methods perform best. We generated binary data through various simulations, varying model-building parameters to introduce different complexity levels within the data and models. The additional objective was to assess whether sample size and the inclusion of continuous covariates also influenced performance, apart from the model-building parameters that influenced complexity.

We performed an exploratory analysis using a limited set of methods, logistic regression, LASSO regression, KNN, and random forest, on 138 different models. We identified recurring patterns in the model's performance estimates over the whole range of samples that reflected the general behavior of machine learning and traditional statistics approaches. We found patterns where either the machine learning approach or traditional statistics approach exclusively had superior performance, as well as several mixed patterns. We reviewed whether models that were included in a specific pattern had similar settings to identify the circumstances under which this pattern was generated. Median models identified from patterns in the exploratory analysis underwent an in-depth analysis using all methods described in Chapter 2 to ascertain whether patterns persisted in more complex methods. We extracted several performance measures to visualize performance of methods across different sample sizes, focusing mainly on accuracy estimates and to a smaller extent sensitivity and specificity. Additionally, we performed a case study, in which three different data sets underwent a similar analysis in order to see whether patterns we identified in the simulation phase were also present in real-life data.

The results indicate that each pattern contains a variety of models with different settings. No single setting consistently dominates within a pattern. This suggests that, within each pattern, there is not one specific factor or circumstance that influences which approach performs best. Models incorporating continuous covariates are spread across all identified patterns, suggesting that the inclusion of continuous covariates does not favor one particular approach or pattern. Additionally, our study demonstrates that sample size does affect the performance of both approaches, with no clear consistent trend indicating which approach is best for small or large sample sizes. The patterns we observed in our simulation studies persist in real-life data, with no other new patterns appearing. However, identifying the actual pattern seems more challenging. Moreover, when we extended the case study analysis to include the in-depth analysis we observe a similar shift in patterns as in our simulation study.

The results of the exploratory analysis suggest that while several patterns exist in the data, there was not one specific kind of model that consistently matched to one pattern. Instead, many similar models were spread across several patterns. Naturally, each pattern contained nuances. For instance, pattern C contained models with higher formula complexity, but this was not exclusive to pattern C. Each pattern contained several variations, some which matched in settings, other did not. An example is pattern E, in which models with higher variance had a different visualization than other variations within that pattern. Some consistency within patterns did exist, such as pattern B. However, when considering which patterns re-occurred in the in-depth analysis, we cannot be sure that this pattern actually exists. The fact that some patterns disappeared once we used a more complex analysis, makes it reasonable to assume that once more complex models are added, some patterns are no longer viable, as more complex models could perform better than computationally inexpensive models that were used in the explanatory analysis. However, this is contradicted by the fact that in the in-depth analysis several patterns remained in which traditional statistical methods outperformed, often more complex, machine learning methods. The fact that we did not find particular settings in patterns aligns with the often-varying reported results in the literature. Even within a single study, Couronné et al. (2018) found that in 69% of the datasets random forest outperformed logistic regression. Consequently, this also means that in 31% of studies this was not the case.

Our results also contribute a clearer understanding as to why there were so many contradicting claims in the debate about traditional statistical methods versus machine learning, described in the literature review. While we were unable to conclusively identify specific circumstances under which each approach performed best, we did identify patterns that do not support the theory that machine learning methods achieve good performance only when the sample size is substantial (Kokol et al., 2022; Ley et al., 2022; Rajkomar et al., 2019; Rajula et al., 2020). Some patterns exhibited superior performance of one approach regardless of sample size, whereas sample size certainly mattered in other patterns to find the best performing approach. This is supported by our case studies of real-life data, where similar patterns to those found in the simulated data, in which data relationships were known, persisted. Our results provide new insight into the relationship between sample size and best performing approach, which contradicts Van Der Ploeg et al. (2014) who recommended using logistic regression when the sample size is small and machine learning methods when the sample size is large to gain stable results. We observed instances where the opposite was evident, which aligns with our hypothesis that there do exist circumstances in data that invalidate this statement. This could be explained by the fact that researchers often make use of one data set with a fixed sample size to obtain results, without insight into the actual underlying data relationships. If, like the researchers discussed in our literature review, they report that machine learning methods outperformed a traditional statistical method, they might have found themselves in a pattern which looked favorably to that approach at that specific sample size in combination with the relationships inherent in the data. In contrast, we reviewed results across a large range of sample sizes with knowledge of underlying relationships.

The results of this study should be interpreted with caution due to some limitations. First, we included several versions of each model to observe the different variations when we changed complexity levels. This may have impacted the identification of patterns. If less variations had been present, we might have successfully identified similar settings in patterns. Second, the generalizability of the results is limited by the fact that outcomes were generated based on logistic regression, meaning covariates had a linear relationship with the log-odds. By limiting ourselves to only generating linear based data, we have excluded a large portion of data structures. In relation to this, we also generated data based on the assumptions that every covariate had a relationship to the outcome and no correlation among covariates existed. This is unlike real-life data, which often contains irrelevant predictors, as well as covariates that have some sort of relationship with each other. By not adding these aspects to our data simulation, we may have lost an opportunity to shape the data as real-life as possible. A more thorough study should consider the addition of both irrelevant and correlated predictors, and other data structures. Third, our results in both the simulation study and case studies are based on balanced (generated) data, which might not represent real-life data. Our randomly sampled data from a relatively large balanced data set often had 30-60% positive cases, while medical data is often plagued by only a fraction of cases. Further research is needed to assess how both approaches perform when there is a high imbalance, such as only 5% cases. Other limitations include methodological choices. In hyperparameter tuning we set the number of combinations in the (random) search quite low due to computational constraints. This limited methods in finding the optimal model. Moreover, we were restricted by the usage of class labels as some methods were unable to output probabilities, which would have provided more information. Beyond the scope of this study was generating and assessing high-dimensional data due to computational limitations. Several results exhibited signs of high-dimensionality when the sample size was small, but no data was generated where over the whole sample size range $p \gg n$.

Other avenues for future research include expanding the study to encompass a more diverse selection of models and highly imbalanced data. This would allow researchers to more thoroughly review whether the found patterns persist and under which circumstances. We recommend that researchers should only use unique regression coefficients and implement formula complexity no higher than three. This will keep analyses realistic, as generally regression coefficients are unique and more complex interactions are rare as well as computationally expensive. As the evolution of data continues, researchers should consider implementing deep learning models. For instance, Jeong et al. (2020) demonstrated that in predicting the five stages of chronic kidney disease for a highly imbalanced data set, autoencoders had the best overall performance. Apart from deep learning, researchers could use unsupervised clustering methods to investigate whether patterns observed in this study also appear when data sets, either per sample size or all data, are clustered. Additionally, hybrid approaches, which incorporate both approaches, might provide further insights. Levy and O'Malley (2020) have developed a novel hybrid approach in which both machine learning and logistic regression are incorporated. By using a machine learning method, for instance random forest, as a variable selection mechanism for meaningful interactions, logistic regression could incorporate these as predictors and possibly produce higher estimates and at the same time benefit from the advantage of interpretability. The authors use Shapley values to identify meaningful interactions which are then added to data used for the logistic regression model. Finally, it is imperative that researchers keep in mind the no free lunch theorem and compare multiple methods to find which performs best for their specific situation. As we have seen, for very similar data the results can differ greatly between and within patterns.

5.2 Conclusion

We found no conclusive evidence that one type of model consistently ensures that a traditional statistical approach or machine learning approach has superior performance. Our study identified general patterns, showing there was a myriad of model settings that exist within one pattern. The often-made assumption that machine learning performs better when the sample size is big is often a misconception.

Reference list

Akbilgic, O., & Davis, R. L. (2019). The promise of machine learning: When will it be delivered? *Journal of Cardiac Failure*, *25*(6), 484–485. https://doi.org/10.1016/j.cardfail.2019.04.006

Altman, N., & Krzywinski, M. (2018). The curse(s) of dimensionality. *Nature Methods*, *15*(6), 399–400. https://doi.org/10.1038/s41592-018-0019-x

Andersson, C., Nayor, M., Tsao, C. W., Levy, D., & Vasan, R. S. (2021). Framingham heart study: JACC focus seminar, 1/8. *Journal of the American College of Cardiology*, *77*(21), 2680–2692. https://doi.org/10.1016/j.jacc.2021.01.059

Austin, P. C., Harrell, F. E., Jr., Lee, D. S., & Steyerberg, E. W. (2022). Empirical analyses and simulations showed that different machine and statistical learning methods had differing performance for predicting blood pressure. *Scientific Reports*, *12*(1). https://doi.org/10.1038/s41598-022-13015-5

Austin, P. C., Harrell, F. E., Jr., & Steyerberg, E. W. (2021). Predictive performance of machine and statistical learning methods: Impact of data-generating processes on external validity in the "large N, small p" setting. *Statistical Methods in Medical Research*, *30*(6), 1465–1483. https://doi.org/10.1177/09622802211002867

Bailly, A., Blanc, C., Francis, É., Guillotin, T., Jamal, F., Wakim, B., & Roy, P. (2022). Effects of dataset size and interactions on the prediction performance of logistic regression and deep learning models. *Computer Methods and Programs in Biomedicine*, *213*, Article 106504. https://doi.org/10.1016/j.cmpb.2021.106504

Becker, B., & Kohavi, R. (1996). *Adult* [Data set]. UCI Machine Learning Repository. https://doi.org/10.24432/C5XW2

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32. https://doi.org/10.1023/a:1010933404324

Bruce, P., Bruce, A., & Gedeck, P. (2020). *Practical statistics for data scientists: 50+ essential concepts using R and Python* (2nd ed.). O'Reilly Media, Incorporated.

Cao, X., Lin, Y., Yang, B., Li, Y., & Zhou, J. (2022). Comparison between statistical model and machine learning methods for predicting the risk of renal function decline using routine clinical data in health screening. *Risk Management and Healthcare Policy*, *15*, 817–826. https://doi.org/10.2147/rmhp.s346856

Cerulli, G. (2023). *Fundamentals of supervised machine learning: With applications in Python, R, and Stata* (1st ed.). https://doi.org/10.1007/978-3-031-41337-7

Chawla, N. V. (2005). Data mining for imbalanced datasets: An overview. In O. Maimon & L. Rokach (Eds.), *Data mining and knowledge discovery handbook* (pp. 853–867). Springer US. https://doi.org/10.1007/0-387-25465-X_40

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., Li, M., Xie, J., Lin, M., Geng, Y., Li, Y., & Yuan, J. (2024). *Xgboost: Extreme gradient boosting* [R package version 1.7.7.1]. https://CRAN.R-project.org/package=xgboost

Christodoulou, E., Ma, J., Collins, G. S., Steyerberg, E. W., Verbakel, J. Y., & Van Calster, B. (2019). A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models. *Journal of Clinical Epidemiology*, *110*, 12–22. https://doi.org/10.1016/j.jclinepi.2019.02.004

Churpek, M. M., Yuen, T. C., Winslow, C., Meltzer, D. O., Kattan, M. W., & Edelson, D. P. (2016). Multicenter comparison of machine learning methods and conventional regression for predicting clinical deterioration on the wards. *Critical Care Medicine*, *44*(2), 368–374. https://doi.org/10.1097/ccm.0000000000001571

Clore, J., Cios, K., DeShazo, J., & Strack, B. (2014). *Diabetes 130-US hospitals for years 1999-2008* [Data set]. UCI Machine Learning Repository. https://doi.org/10.24432/C5230J

Couronné, R., Probst, P., & Boulesteix, A. (2018). Random forest versus logistic regression: A large-scale benchmark experiment. *BMC Bioinformatics*, *19*, Article 270. https://doi.org/10.1186/s12859-018-2264-5

Cui, Z., & Gong, G. (2018). The effect of machine learning regression algorithms and sample size on individualized behavioral prediction with functional connectivity features. *NeuroImage*, *178*, 622–637. https://doi.org/10.1016/j.neuroimage.2018.06.001

De Hond, A. A. H., Kant, I. M. J., Honkoop, P. J., Smith, A. D., Steyerberg, E. W., & Sont, J. K. (2022). Machine learning did not beat logistic regression in time series prediction for severe asthma exacerbations. *Scientific Reports*, *12*, Article 20363. https://doi.org/10.1038/s41598-022-24909-9

Desai, R. J., Wang, S. V., Vaduganathan, M., Evers, T., & Schneeweiss, S. (2020). Comparison of machine learning methods with traditional models for use of administrative claims with electronic medical records to predict heart failure outcomes. *JAMA Network Open*, *3*(1), e1918962. https://doi.org/10.1001/jamanetworkopen.2019.18962

Feng, J., Wang, Y., Peng, J., Sun, M., Zeng, J., & Jiang, H. (2019). Comparison between logistic regression and machine learning algorithms on survival prediction of traumatic brain injuries. *Journal of Critical Care*, *54*, 110–116. https://doi.org/https://doi.org/10.1016/j.jcrc.2019.08.010

Friedman, J. H., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, *33*(1), 1–22. https://doi.org/10.18637/jss.v033.i01

Frizzell, J. D., Liang, L., Schulte, P. J., Yancy, C. W., Heidenreich, P. A., Hernandez, A. F., Bhatt, D. L., Fonarow, G. C., & Laskey, W. K. (2017). Prediction of 30-day all-cause readmissions in patients hospitalized for heart failure. *JAMA Cardiology*, *2*(2), 204–209. https://doi.org/10.1001/jamacardio.2016.3956

Garcia, L. P. F., De Carvalho, A. C. P. L. F., & Lorena, A. C. (2015). Effect of label noise in the complexity of classification problems. *Neurocomputing*, *160*, 108–119. https://doi.org/10.1016/j.neucom.2014.10.085

Grembi, J. (2022). *Riskcommunicator: G-computation to estimate interpretable epidemiological effects* [R package version 1.0.1]. https://CRAN.R-project.org/package=riskCommunicator

Grendas, L. N., Chiapella, L., Rodante, D. E., & Daray, F. M. (2022). Comparison of traditional model-based statistical methods with machine learning for the prediction of suicide behaviour. *Journal of Psychiatric Research*, *145*, 85–91. https://doi.org/10.1016/j.jpsychires.2021.11.029

Harrison, R. L. (2010). Introduction to monte carlo simulation. *AIP Conference Proceedings*, *1204*(1), 17–21. https://doi.org/10.1063/1.3295638

Hastie, T., Qian, J., & Tay, K. (2023, August 19). *An introduction to glmnet*. Retrieved May 23, 2024, from https://cloud.r-project.org/web/packages/glmnet/vignettes/glmnet.pdf

Hu, P., Liu, Y., Li, Y., Guo, G., Su, Z., Gao, X., Chen, J., Qi, Y., Xu, Y., Yan, T., Ye, L., Sun, Q., Deng, G., Zhang, H., & Chen, Q. (2022). A comparison of LASSO regression and tree-based models for delayed cerebral ischemia in elderly patients with subarachnoid hemorrhage. *Frontiers in Neurology*, *13*, Article 791547. https://doi.org/10.3389/fneur.2022.791547

Huang, R. J., Kwon, N. S., Tomizawa, Y., Choi, A. Y., Hernandez-Boussard, T., & Hwang, J. H. (2022). A comparison of logistic regression against machine learning algorithms for gastric cancer risk prediction within real-world clinical data streams. *JCO Clinical Cancer Informatics*, *6*, e2200039. https://doi.org/10.1200/cci.22.00039

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning: With applications in R* (2nd ed.). Springer US. https://doi.org/10.1007/978-1-0716-1418-1

Jeong, B., Cho, H., Kim, J., Kwon, S. K., Hong, S., Lee, C., Kim, T., Park, M. S., Hong, S., & Heo, T.-Y. (2020). Comparison between statistical models and machine learning methods on classification for highly imbalanced multiclass kidney data. *Diagnostics*, *10*(6), 415. https://doi.org/10.3390/diagnostics10060415

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255–260. https://doi.org/10.1126/science.aaa8415

Kirasich, K., Smith, T., & Sadler, B. (2018). Random forest vs logistic regression: Binary classification for heterogeneous datasets. *SMU Data Science Review*, *1*(3), Article 9. https://scholar.smu.edu/datasciencereview/vol1/iss3/9

Kokol, P., Kokol, M., & Zagoranski, S. (2022). Machine learning on small size samples: A synthetic knowledge synthesis. *Science Progress*, *105*(1), 003685042110297. https://doi.org/10.1177/00368504211029777

Kroese, D. P., Brereton, T., Taimre, T., & Botev, Z. I. (2014). Why the Monte Carlo method is so important today. *WIREs Computational Statistics*, *6*(6), 386–392. https://doi.org/10.1002/wics.1314

Kuhn, M. (2008). Building predictive models in R using the caret package. *Journal of Statistical Software*, *28*(5), 1–26. https://doi.org/10.18637/jss.v028.i05

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling*. Springer US. https://doi.org/10.1007/978-1-4614-6849-3

Kumar, V., & Garg, M. L. (2018). Predictive analytics: A review of trends and techniques. *International Journal of Computer Applications*, *182*(1), 31–37. https://doi.org/10.5120/ijca2018917434

Levy, J. J., & O'Malley, A. J. (2020). Don't dismiss logistic regression: The case for sensible extraction of interactions in the era of machine learning. *BMC Medical Research Methodology*, *20*, Article 171. https://doi.org/10.1186/s12874-020-01046-3

Ley, C., Martin, R. K., Pareek, A., Groll, A., Seil, R., & Tischer, T. (2022). Machine learning and conventional statistics: Making sense of the differences. *Knee Surgery, Sports Traumatology, Arthroscopy*, *30*(3), 753–757. https://doi.org/10.1007/s00167-022-06896-6

Li, D., Kong, Y., Fan, Y., & Lv, J. (2022). High-dimensional interaction detection with false sign rate control. *Journal of Business & Economic Statistics*, *40*(3), 1234–1245. https://doi.org/10.1080/07350015.2021.1917419

Liaw, A., & Wiener, M. (2002). Classification and regression by randomforest. *R News*, *2*(3), 18–22. https://CRAN.R-project.org/doc/Rnews/

Lolak, S., Attia, J., McKay, G. J., & Thakkinstian, A. (2023). Comparing explainable machine learning approaches with traditional statistical methods for evaluating stroke risk models: Retrospective cohort study. *JMIR Cardio*, *7*, e47736. https://doi.org/10.2196/47736

Lynam, A. L., Dennis, J. M., Owen, K. R., Oram, R. A., Jones, A. G., Shields, B. M., & Ferrat, L. A. (2020). Logistic regression has similar performance to optimised machine learning algorithms in a clinical setting: Application to the discrimination between type 1 and type 2 diabetes in young adults. *Diagnostic and Prognostic Research*, *4*, Article 6. https://doi.org/10.1186/s41512-020-00075-2

Mallett, S., Halligan, S., Thompson, M., Collins, G. S., & Altman, D. G. (2012). Interpreting diagnostic accuracy studies for patient care. *BMJ*, *345*, e3999. https://doi.org/10.1136/bmj.e3999

Morris, T. P., White, I. R., & Crowther, M. J. (2019). Using simulation studies to evaluate statistical methods. *Statistics in Medicine*, *38*(11), 2074–2102. https://doi.org/10.1002/sim.8086

Naidu, G., Zuva, T., & Sibanda, E. M. (2023). A review of evaluation metrics in machine learning algorithms. In R. Silhavy & P. Silhavy (Eds.), *Artificial intelligence application in networks and systems* (pp. 15–25). Springer International Publishing. https://doi.org/10.1007/978-3-031-35314-7_2

Nunez, Y., Gibson, E. A., Tanner, E. M., Gennings, C., Coull, B. A., Goldsmith, J., & Kioumourtzoglou, M.-A. (2021). Reflection on modern methods: Good practices for applied statistical learning in epidemiology. *International Journal of Epidemiology*, *50*(2), 685–693. https://doi.org/10.1093/ije/dyaa259

Panaretos, D., Koloverou, E., Dimopoulos, A. C., Kouli, G.-M., Vamvakari, M., Tzavelas, G., Pitsavos, C., & Panagiotakos, D. B. (2018). A comparison of statistical and machine-learning techniques in evaluating the association between dietary patterns and 10-year cardiometabolic risk (2002–2012): The ATTICA study. *British Journal of Nutrition*, *120*(3), 326–334. https://doi.org/10.1017/s0007114518001150

Patel, B., & Sengupta, P. (2020). Machine learning for predicting cardiac events: What does the future hold? *Expert Review of Cardiovascular Therapy*, *18*(2), 77–84. https://doi.org/10.1080/14779072.2020.1732208

Peng, C.-Y. J., Lee, K. L., & Ingersoll, G. M. (2002). An introduction to logistic regression analysis and reporting. *The Journal of Educational Research*, *96*(1), 3–14. https://doi.org/10.1080/00220670209598786

R Core Team. (2023). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria. https://www.R-project.org/

Rajkomar, A., Dean, J., & Kohane, I. (2019). Machine learning in medicine. *New England Journal of Medicine*, *380*(14), 1347–1358. https://doi.org/10.1056/nejmra1814259

Rajula, H. S. R., Verlato, G., Manchia, M., Antonucci, N., & Fanos, V. (2020). Comparison of conventional statistical methods with machine learning in medicine: Diagnosis, drug development, and treatment. *Medicina*, *56*(9), 455. https://doi.org/10.3390/medicina56090455

Senders, J. T., Staples, P. C., Karhade, A. V., Zaki, M. M., Gormley, W. B., Broekman, M. L. D., Smith, T. R., & Arnaout, O. (2018). Machine learning and neurosurgical outcome prediction: A systematic review. *World Neurosurgery*, *109*, 476–486.e1. https://doi.org/10.1016/j.wneu.2017.09.149

Shin, S., Austin, P. C., Ross, H. J., Abdel-Qadir, H., Freitas, C., Tomlinson, G., Chicco, D., Mahendiran, M., Lawler, P. R., Billia, F., Gramolini, A., Epelman, S., Wang, B., & Lee, D. S. (2021). Machine learning vs. conventional statistical models for predicting heart failure readmission and mortality. *ESC Heart Failure*, *8*(1), 106–115. https://doi.org/10.1002/ehf2.13073

Steyerberg, E. W. (2019). *Clinical prediction models: A practical approach to development, validation, and updating* (2nd ed.). Springer Cham. https://doi.org/https://doi.org/10.1007/978-3-030-16399-0

Strack, B., DeShazo, J. P., Gennings, C., Olmo, J. L., Ventura, S., Cios, K. J., & Clore, J. N. (2014). Impact of HbA1c measurement on hospital readmission rates: Analysis of 70,000 clinical database patient records. *BioMed Research International*, *2014*, Article 781670. https://doi.org/10.1155/2014/781670

Sun, X., Douiri, A., & Gulliford, M. (2022). Applying machine learning algorithms to electronic health records to predict pneumonia after respiratory tract infection. *Journal of Clinical Epidemiology*, *145*, 154–163. https://doi.org/10.1016/j.jclinepi.2022.01.009

Tollenaar, N., & Van Der Heijden, P. G. M. (2013). Which method predicts recidivism best?: A comparison of statistical, machine learning and data mining predictive models. *Journal of the Royal Statistical Society Series A: Statistics in Society*, *176*(2), 565–584. https://doi.org/10.1111/j.1467-985x.2012.01056.x

Van Der Ploeg, T., Austin, P. C., & Steyerberg, E. W. (2014). Modern modelling techniques are data hungry: A simulation study for predicting dichotomous endpoints. *BMC Medical Research Methodology*, *14*, Article 137. https://doi.org/10.1186/1471-2288-14-137

Van Stralen, K. J., Stel, V. S., Reitsma, J. B., Dekker, F. W., Zoccali, C., & Jager, K. J. (2009). Diagnostic methods I: Sensitivity, specificity, and other measures of accuracy. *Kidney International*, *75*(12), 1257–1263. https://doi.org/10.1038/ki.2009.92

Van Wieringen, W. N. (2023). Lecture notes on ridge regression. https://doi.org/10.48550/arXiv.1509.09169

Wegmeth, L., Vente, T., Purucker, L., & Beel, J. (2023). The effect of random seeds for data splitting on recommendation accuracy. In A. Said, E. Zangerle, & C. Bauer (Eds.), *Perspectives on the evaluation of recommender systems workshop (PERSPECTIVES 2023), co-located with the 17th ACM conference on recommender systems* (Vol. 3476). CEUR-WS.org. https://ceur-ws.org/Vol-3476/paper4.pdf

Zhang, J., Li, Z., Pu, Z., & Xu, C. (2018). Comparing prediction performance for crash injury severity among various machine learning and statistical methods. *IEEE Access*, *6*, 60079–60087. https://doi.org/10.1109/access.2018.2874979

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, *67*(2), 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x

Appendix A

Theoretical Calculations

Distribution Continuous Covariates

To determine which distribution to use for the continuous covariates, we calculated theoretical values for the mean and variance.

Based on the values we have used during the simulation of binary covariates (-1 and 1), we calculated the following mean:

$$\mathbb{E}[X] = \sum_i x_i p_i = -1 \times 0.50 + 1 \times 0.50 = 0.$$

Using this same logic, we calculated the variance:

$$Var(X) = \mathbb{E}[(X-\mu)^2] = \sum_{i=1}^{n} p_i (x_i - \mu)^2 = 0.50(-1-0)^2 + 0.50(1-0)^2 = 1.$$

The standard deviation is equal to the variance, as $\sigma = \sqrt{\mathbb{E}[(X-\mu)^2]} = \sqrt{1} = 1$.

Complexity Level Variance

To determine the maximum variance between which δ can range we needed to calculate this. As every observation is independent the variance can be calculated using the variance from a single Bernoulli trial.

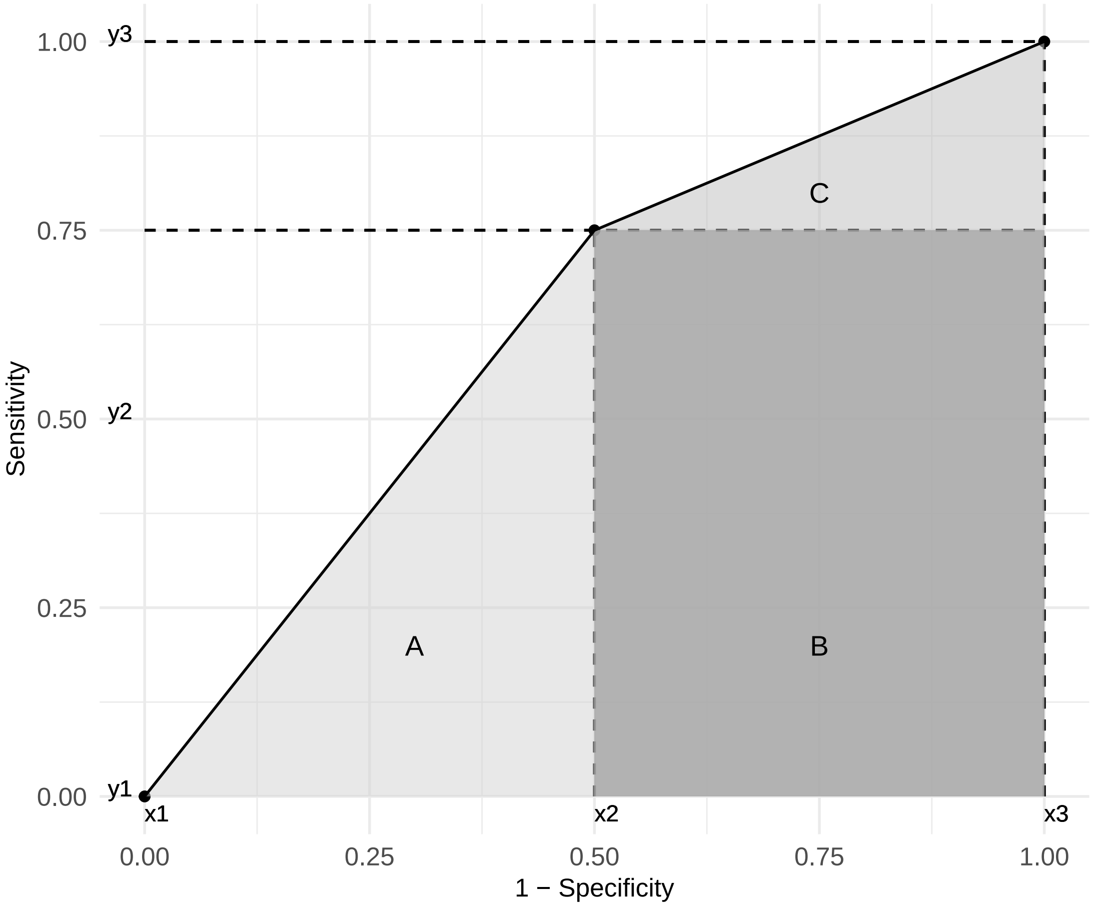
We calculated the following maximum:

$$\sigma^2 = p(1-p) = 0.50(1-0.50) = 0.25.$$

Equality of AUC Value and Accuracy

Figure A.1

Visualization of an AUC curve with a single point



First, we show that the Balanced Accuracy can be considered equal to the AUC value if only one point is available. A visualization of this is shown in Figure A.1, which we use as an example.

To calculate the AUC for Figure A.1, we can calculate the area under the curve. This is calculated as follows for areas A, B, and C:

$$A = \frac{1}{2} y_2 x_2$$
$$B = (x_3 - x_2)(y_2 - y_1)$$
$$C = \frac{1}{2}(y_3 - y_2)(x_3 - x_2)$$

These variables and accompanying values, according to the figure, are defined as follows:

$$y_1 = 0, y_2 = \text{Sensitivity} = y, y_3 = 1$$
$$x_1 = 0, x_2 = 1 - \text{Specificity} = 1 - x, x_3 = 1$$

We can calculate the AUC value as follows:

$$\begin{aligned}
&AUC = A + B + C = \frac{1}{2}y_2x_2 + (x_3 - x_2)(y_2 - y_1) + \frac{1}{2}(y_3 - y_2)(x_3 - x_2) \\
&= \frac{1}{2}(y(1-x)) + (1-(1-x))y + \frac{1}{2}\left((1-y)(1-(1-x))\right) \\
&= \frac{1}{2}y - \frac{1}{2}xy + xy + \frac{1}{2}x - \frac{1}{2}xy \\
&= \frac{1}{2}y + \frac{1}{2}x + xy - \frac{1}{2}xy - \frac{1}{2}xy = \frac{1}{2}y + \frac{1}{2}x \\
&= \frac{x+y}{2} = \frac{\text{Specificity} + \text{Sensitivity}}{2}
\end{aligned}$$

This outcome equals the Balanced Accuracy equation. We can go further, and show that Balanced Accuracy is equal to Accuracy when the data is assumed to be balanced. If we assume

$$\text{Prevalence} = \frac{TP + FN}{TP + FN + TN + FP} = 0.50,$$

this describes the proportion of the data that belongs to the class that is classified as 1. Prevalence is balanced when:

$$\begin{aligned}
&TP + FN = 0.50(TP + FN + TN + FP) \\
&TP + FN = 0.50TP + 0.50FN + 0.50TN + 0.50FP \\
&0.50TP + 0.50FN = 0.50TN + 0.50FP \\
&TP + FN = TN + FP
\end{aligned}$$

We defined Accuracy as follows:

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

Using the balanced prevalence, accuracy can be written as:

$$\begin{aligned}
&\text{Accuracy} = \frac{TP + TN}{TP + FN + TP + FN} = \frac{TP + TN}{2TP + 2FN} \\
&= \frac{TP + TN}{2(TP + FN)}
\end{aligned}$$

We can rewrite Balanced Accuracy in a similar manner:

$$
\begin{aligned}
&\text{Balanced Accuracy} = \frac{\text{Specificity} + \text{Sensitivity}}{2} \\
&= \frac{\frac{TN}{TN+FP} + \frac{TP}{TP+FN}}{2} \\
&= \frac{\frac{TN}{TP+FN} + \frac{TP}{TP+FN}}{2} \\
&= \frac{TN}{2(TP+FN)} + \frac{TP}{2(TP+FN)} \\
&= \frac{TN+TP}{2(TP+FN)}
\end{aligned}
$$

This shows that, when considering a balanced data set,

$$\text{AUC} = \text{Balanced Accuracy} = \text{Accuracy}$$

Appendix B

Code

The code used in this study is available on Github (https://github.com/l-verl/Master-Thesis). Additionally, the complete results, i.e., figures, can also be found here.

Appendix C

Pattern Classification

In this Appendix we show how each model was classified per performance measure. As these only concern the exploratory models, the abbreviation *E* is dropped to increase readability. Some models are not included, as sometimes a model did not specifically fit into a pattern.

Table C.1

Accuracy Patterns

Pattern	Models
A	6, 19, 20, 24, 26, 27, 29, 50, 59, 71, 72, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 88, 93, 99, 132, 133
B	2, 68, 89, 116
C	34, 35, 36, 37, 38, 41, 42, 43, 44, 49, 51, 52, 53, 54, 62, 63, 64, 65, 66, 131
D	15, 16, 21, 32, 60, 61, 73, 74, 76, 87, 95, 96, 100, 108, 109, 111, 112, 113, 114, 118, 123, 126, 135, 136, 138
E	3, 4, 5, 8, 9, 10, 11, 12, 13, 14, 22, 23, 25, 30, 31, 33, 40, 45, 46, 47, 48, 69, 70, 75, 90, 92, 97, 98, 101, 102, 103, 104, 105, 106, 107, 115, 117, 129, 130
F	7, 17, 18, 28, 55, 56, 57, 58, 67, 91, 94, 110, 119, 120, 121, 122, 124, 125, 127, 128, 134, 137

Note. This table demonstrates into which pattern each model was classified according to its accuracy estimates. The model numbers correspond to models of the exploratory analysis, which are denoted with abbreviation *E*- in this paper. Model E-39 was not included as it was the only model to display a different behavior in the accuracy estimates. Model E-1 was not included as it did not fit into any of the general patterns.

Table C.2

Sensitivity Patterns

Pattern	Models
A	71, 86
B	2, 11, 13, 31, 39, 48, 68, 69, 89, 92, 106, 116, 126, 129
C	34, 35, 36, 37, 38, 41, 42, 43, 44, 51, 52, 54, 62, 63, 64, 65, 66, 131
D	16, 17, 21, 32, 50, 60, 72, 74, 76, 87, 88, 94, 95, 99, 112, 122
E	3, 4, 6, 8, 9, 10, 12, 14, 22, 23, 25, 33, 40, 45, 70, 75, 96, 97, 98, 101, 102, 103, 115, 117
F	7, 18, 19, 20, 24, 26, 27, 28, 29, 55, 56, 57, 58, 59, 61, 73, 77, 78, 79, 80, 81, 82, 83, 84, 85, 91, 93, 100, 108, 109, 111, 113, 114, 118, 119, 120, 121, 124, 125, 127, 128, 132, 133, 134, 135, 136, 137, 138
G	5, 15, 30, 46, 47, 49, 53, 67, 90, 104, 105, 107, 110, 123, 130

Note. This table demonstrates into which pattern each model was classified according to its sensitivity estimates. The model numbers correspond to models of the exploratory analysis, which are denoted with abbreviation *E-*. Model E-1 was not included as it did not fit into any of the general patterns.

Table C.3

Specificity Patterns

Pattern	Models
A	6, 15, 18, 19, 20, 21, 27, 28, 29, 39, 49, 50, 53, 55, 56, 59, 61, 71, 72, 73, 77, 78, 79, 80, 81, 84, 85, 88, 93, 95, 99, 100, 109, 111, 118, 121, 125, 132, 133, 136
B*	2
C	34, 38, 41, 42, 43, 44, 51, 52, 54, 62, 63, 64, 65, 131
D	3, 7, 8, 13, 23, 24, 30, 31, 32, 35, 36, 37, 48, 60, 66, 69, 74, 86, 87, 91, 92, 96, 103, 107, 110, 112, 113, 114, 123, 124, 126, 129, 137, 138
E	4, 5, 10, 11, 12, 22, 33, 40, 45, 46, 47, 68, 70, 75, 89, 104, 105, 106, 116, 117, 130
F	14, 16, 17, 26, 57, 58, 67, 76, 82, 83, 90, 94, 97, 98, 108, 115, 119, 120, 122, 127, 128, 134, 135
G	9, 25, 101, 102

Note. This table demonstrates into which pattern each model was classified according to its specificity estimates. The model numbers correspond to models of the exploratory analysis, which are denoted with abbreviation *E-*. Model E-1 was not included as it did not fit into any of the general patterns.
* Although we stated that a pattern did not exist if only one model displayed a certain pattern, the pattern was already created in an earlier performance measure. Hence, we included this.