

Evaluation of tree based statistical learning supporting the bespoke manufacturing of low volume, super luxury cars

J. Pooley

A thesis submitted for the degree of
Doctor of Philosophy

School of Mathematics, Statistics and Actuarial Science

University of Essex

Date of submission for examination: March 2026

Dedicated to my late father, Simon.

Acknowledgements

Completing this thesis would not have been possible without the unwavering support from my many supervisors: Berthold Lausen, Osama Mahmoud, Ivan Schuch and Henrik Nordmark. They had the belief in me and the project even when I was unsure if I could do it. I cannot thank them enough. All the additional support from everyone I worked with cannot be understated too; giving me ideas to work with, answering my questions as best they could, and for generally giving me a chance to do something new and exciting.

Abstract

Manufacturing low-volume bespoke, super luxury cars is challenging. Many difficulties arise; the relatively low volume of data, the variation of the vehicles hiding patterns, the human builders lowering the consistency, for example. Yet all of these problems must be managed to create perfection. Solving these problems requires careful data selection, choosing features that make an impact; the specification of each vehicle, the production associates who built it, amongst others. Furthermore, to predict how a vehicle will perform, given the information known about it, a suitable machine learning method needs selecting.

In this thesis, the process of choosing the most appropriate problem is discussed: predicting how vehicles will perform in their final inspection. Selecting the best features and engineering them into a suitable form. Established machine learning methods are discussed: random forest, XGBoost, optimal tree ensembles, neural networks and semi-supervised generative adversarial networks, as well as some novel methods, utilising P-Value adjusted decision trees in the optimal tree ensemble framework, residual boosting and conditional inference trees. These methods are compared to determine the most effective approach for predicting final inspection outcomes.

The results demonstrate that the random forest model achieved the highest predictive performance with a precision of 66%, outperforming a novel method of PASTE-boost with a precision of 52%. The findings show that the final inspection result is predictable but improvements may require more observations or more influential predictors to allow for a more reliable prediction. This thesis demonstrates the potential and limitations of machine learning approaches for supporting quality assessment for low-volume manufacturing processes.

Contents

Abstract	ii
List of Tables	vi
List of Figures	ix
1 Introduction	1
1.1 Motivation	1
1.2 Research Objective	2
1.3 Contributions to Industry Knowledge	2
1.4 Contributions to Academic Knowledge	6
1.5 Thesis Outline	7
2 Literature Review	9
2.1 Introduction	9
2.1.1 What is Data Science?	9
2.1.2 What is Trustworthy Data Science?	10
2.1.3 History	12
2.2 Accuracy	14
2.3 Robustness and Fairness	15
2.4 Transparency and Accountability	15
2.5 Privacy and Security	16
2.6 Ethics	17
2.7 Communication and Explainability	18

2.8	Further Solutions	19
2.8.1	Data Trusts	19
2.8.2	EU Guidelines	19
2.8.3	Formal Methods	20
2.9	Research Groups	20
2.9.1	Research Alliance Ruhr	20
2.9.2	The Alan Turing Institute	21
2.10	Conclusions and Research Ideas	21
3	Data Descriptions	23
3.1	Data Introduction	23
3.2	Logistics Data	24
3.3	Quality Data	25
3.4	Audit Data	26
4	Data Engineering & Initial Feature Selection	27
4.1	Station-wise Features	27
4.1.1	Number of visits to a station	28
4.1.2	Time after beginning of shift	30
4.1.3	Time into station	32
4.1.4	Quality index score found in each station	32
4.1.5	Checked OK and Checked Not OK percentage	34
4.1.6	Time spent in station	38
4.1.7	Rework Minutes	41
4.1.8	Additional features	43
4.2	Other Considered Features	44
4.2.1	Defect Type and Place Embeddings	44
4.2.2	Reworker Embeddings	45
4.2.3	Station Sequence Embeddings	46
4.2.4	Station Undersensitivity	47

5	Related Methods	49
5.1	Introduction	49
5.2	Random Forests	49
5.2.1	Decision Trees	49
5.2.2	Ensemble method	52
5.3	Optimal Tree Ensembles	53
5.4	Gradient Boosting	54
5.4.1	XGBoost	55
5.4.2	LightGBM	57
5.5	Logistic Regression	58
5.6	Artificial Neural Networks	59
5.7	Semi-Supervised Generative Adversarial Networks	63
6	Proposed Methods	68
6.1	Introduction	68
6.2	Method	69
6.2.1	P-Value Adjusted Selected Tree Ensembles (PASTE)	70
6.2.2	Conditional Inference Trees	74
6.2.3	Residual Boosting	79
6.3	Testing & Results	81
6.4	Discussion	87
6.5	Conclusions about new methods	89
7	Final Methodological Pipeline	90
7.1	Introduction	90
7.2	Data	91
7.3	Methodological Definition	92
7.3.1	Extreme Thresholds	96
7.4	Results	97
7.4.1	No Feature Selection Results	97

7.4.2	Feature Selection Results	99
7.4.3	Extreme Thresholds	100
7.5	Conclusion	101
8	Final Conclusions	103
8.1	Contributions	103
8.2	Prompts for Future Work	106
A	Appendix	120

List of Tables

3.1	The weighting value for every quality severity score.	26
6.1	Hyperparameter search spaces used for tuning the machine learning methods, using the R implementation of each of the methods. Here d denotes the number of predictor variables.	82
6.2	Description of classification datasets used in the tests, showing the number of instances n , number of features d , feature type (R : Real, I : Integer, C : Categorical), class distribution, classification type, and source of each dataset. .	83
7.1	Hyperparameters and their ranges for each of the tested machine learning methods.	95
7.2	Results for Random Forest tests with various thresholds of the data. The precision, recall and F1 score are calculated, and the total number of positive classes (failures in inspection) are shown. [REDACTED] [REDACTED]	96
A.1	Results for the data without feature selection.	120
A.2	Results for the data with feature selection.	120

List of Figures

1.1	A view of the configurator, as seen by a customer, to select between various clock and headliner options.	4
4.1	Boxplots showing the number of visits to each station. Stations with the same value for $> 98\%$ of observations have been removed. The pink line represents the end of the production line where the vehicle moves into the off-line section of production.	29
4.2	Boxplots showing the number of minutes into a shift when every vehicle entered each station. The pink line represents the end of the production line where the vehicle moves into the off-line section of production.	31
4.3	Boxplots for the hour the vehicles entered each station. The pink line represents the end of the production line where the vehicle moves into the off-line section of production.	33
4.4	Boxplots for the quality index score captured at each station. Only values between are shown; any values outside this range are outliers. Stations with the same value for $> 98\%$ of observations have been removed. The pink line represents the end of the production line where the vehicle moves into the off-line section of production.	35
4.5	The Checked OK percentage at each station. Stations with the same value for $> 98\%$ of observations have been removed. The pink line represents the end of the production line where the vehicle moves into the off-line section of production.	36

4.6	The Checked Not OK percentage at each station. Stations with the same value for $> 98\%$ of observations have been removed. The pink line represents the end of the production line where the vehicle moves into the off-line section of production.	37
4.7	The number of minutes vehicles have been in each station. This is a compressed view, showing only [REDACTED]; an extended view is shown in Figure 4.8. The pink line represents the end of the production line where the vehicle moves into the off-line section of production.	39
4.8	The number of minutes vehicles have been in each station. This is the full view, showing the maximum values up to [REDACTED] for a very extreme case. The pink line represents the end of the production line where the vehicle moves into the off-line section of production.	40
4.9	A boxplot showing the total rework minutes per vehicle at each station. Only values between [REDACTED] are shown, with [REDACTED]. The pink line represents the end of the production line where the vehicle moves into the off-line section of production.	42
4.10	A bar plot showing the distribution of the exterior colour for light and dark colours.	43
4.11	A bar plot showing the distribution of the upholstery colour for light and dark colours.	44
5.1	The change between visible surface detail and no visible surface detail on Mars during a dust storm as picture by NASA [72].	51
6.1	Boxplots for the accuracy over 100 with-replacement bootstrap sample predictions for the 8 tested algorithms (CISTE, CISTE-Boost, OTE-Boost, OTE, PASTE, PASTE-Boost, Random Forest, and XGBoost) across the 7 datasets. . .	85
6.2	Boxplots for F1 score over 100 with-replacement bootstrap sample predictions for the 8 tested algorithms and across all 7 datasets.	86
6.3	Boxplots for the kappa value over 100 with-replacement bootstrap sample predictions for the 8 tested algorithms and across all 7 datasets.	86

- 7.1 The ROC-AUC score across 5 split point thresholds; [REDACTED]. These results are across a 5-fold cross validation with no feature selection. The baseline ROC-AUC score is shown in pink, representing the ROC-AUC score if the trained model made random predictions. The dot represents the mean value across the 5 folds. The whiskers represent the range of values from all 5 folds. . 92
- 7.2 A boxplot showing the ranges of precision for each of the tested methods, when no feature selection has taken place. 98
- 7.3 A boxplot showing the ranges of precision for each of the tested methods, when feature selection has taken place. 100
- 7.4 A grid of confusion matrices showing the prediction results for different cuts of the data. Each matrix is a combination of the confusion matrices for all 10 folds tested on. This is for all of the extreme thresholds mentioned in Section 7.3.1, including a confusion matrix only predicting values [REDACTED], the closest values to the threshold. 102

Introduction

1.1 Motivation

In vehicle manufacturing there is an overwhelming volume of data available. These data come from many places within the production cycle; the logistics of parts deliveries, the specification of the vehicles, the quality of the build, the distribution of the vehicles to customers, amongst others. As with all data sets, being able to utilise it is the challenge. At [REDACTED], these data are used thoroughly around the production plant to maintain the efficacy of the production process, and further improve it over time. These improvements are created by using dashboards and expert knowledge to analyse the data and calculating the best factors to change. However, improvements are often small scale because finding a clear pattern and a cause for said pattern is challenging. A key area for further, larger scale improvements is extracting more from the data, finding causes of issues, and understanding why a particular data point gives a particular result. Machine learning can help provide the answer to some of the more complex questions that cannot be answered with basic analysis.

The main goal of this thesis is to explore various established and new machine learning methods to find how to best support the production of vehicles at the plant. During the project, many methods and datasets were tested to find where machine learning could be best applied to give the most valuable insights, and provide the most impactful feedback.

1.2 Research Objective

Over the course of the research, various research questions were considered and researched. As knowledge of the data and the plant increased, these questions were gradually answered and new questions emerged. The final research objective was:

Is it possible to predict if a vehicle will pass or fail its Audit test using factors from the production process?

At the end of the production cycle, [REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED] Using only domain knowledge, no reason could be given as to why a vehicle would fail, therefore machine learning methods were tested to try and find a solution. This question will be discussed in more detail in Chapter 7.

1.3 Contributions to Industry Knowledge

As mentioned in Section 1.2, many questions were asked which had been thought of before the project, but without any means of answering them. While the main focus of this thesis is surrounding one research question, the data selection specifically was guided by previous investigations and results, as discussed in more depth in Chapter 3. Nonetheless, these questions are outlined below and provided valuable information for the associates at the plant, alongside a couple of additional tools to use.

One such tool was created for the main research question itself. After creating the best pipeline available for the data using the methods outlined in Section 7, a Python script for predicting whether a vehicle would pass or fail Audit was created. This maintained the preprocessing steps, used the highest performance method, and output a prediction result for each of the vehicles in question. This pipeline will be placed into a system where data can be automatically pulled when a vehicle enters a specific station, as specified in the script. Therefore, a prediction can

be made before Audit, [REDACTED]

[REDACTED]

Two of the other influential research questions that impacted the feature selection are outlined below:

Do individual customer selected options make an impact to the build quality?

One of the key features to the brand is allowing customers to customise their vehicle to whatever their preferences are, even outside of the standard range. Shown in Figure 1.1, the customer can choose one of four different clock styles and one of three headliner options, already giving twelve possibilities in only two options. Having such a tremendously large option set for each vehicle means that a vehicle on the production line is likely to have never been made exactly before. However, for many of these options, they will not have a detrimental effect on the build of the vehicle, the clock being a perfect example, as whatever it may look like, it can still be placed in the dashboard in the same manner. Therefore, it is important to only find the options that will affect the build of the vehicle, such as the difference between the black cloth headlining and the panoramic sunroof, also shown in Figure 1.1. Additionally, certain options on the vehicle are one of the factors that is likely to determine the ability to find defects on it. Different vehicle colours, interior and exterior and different upholstery types used in the car can obfuscate defects. A scratch on a dark coloured material is more likely to be noticed than a scratch on a light coloured vehicle. [REDACTED]

[REDACTED]

What associate and environmental factors make an impact on the quality of the vehicles?

[REDACTED] prides itself off of making hand-built vehicles, and the associates are the most important part of the process. Without them, the sheer variety and customisability of the vehicles would not be possible. Naturally, the associates are far more inconsistent than a robot would be, which means that they are likely the largest source of variability within the manufacturing plant. Inconsistencies in performance for a person from day to day can cause problems, there are many different factors at play that cannot be measured nor controlled, least of all by an employer. For example, the mental state of any given associate will affect their work in both good and bad ways,

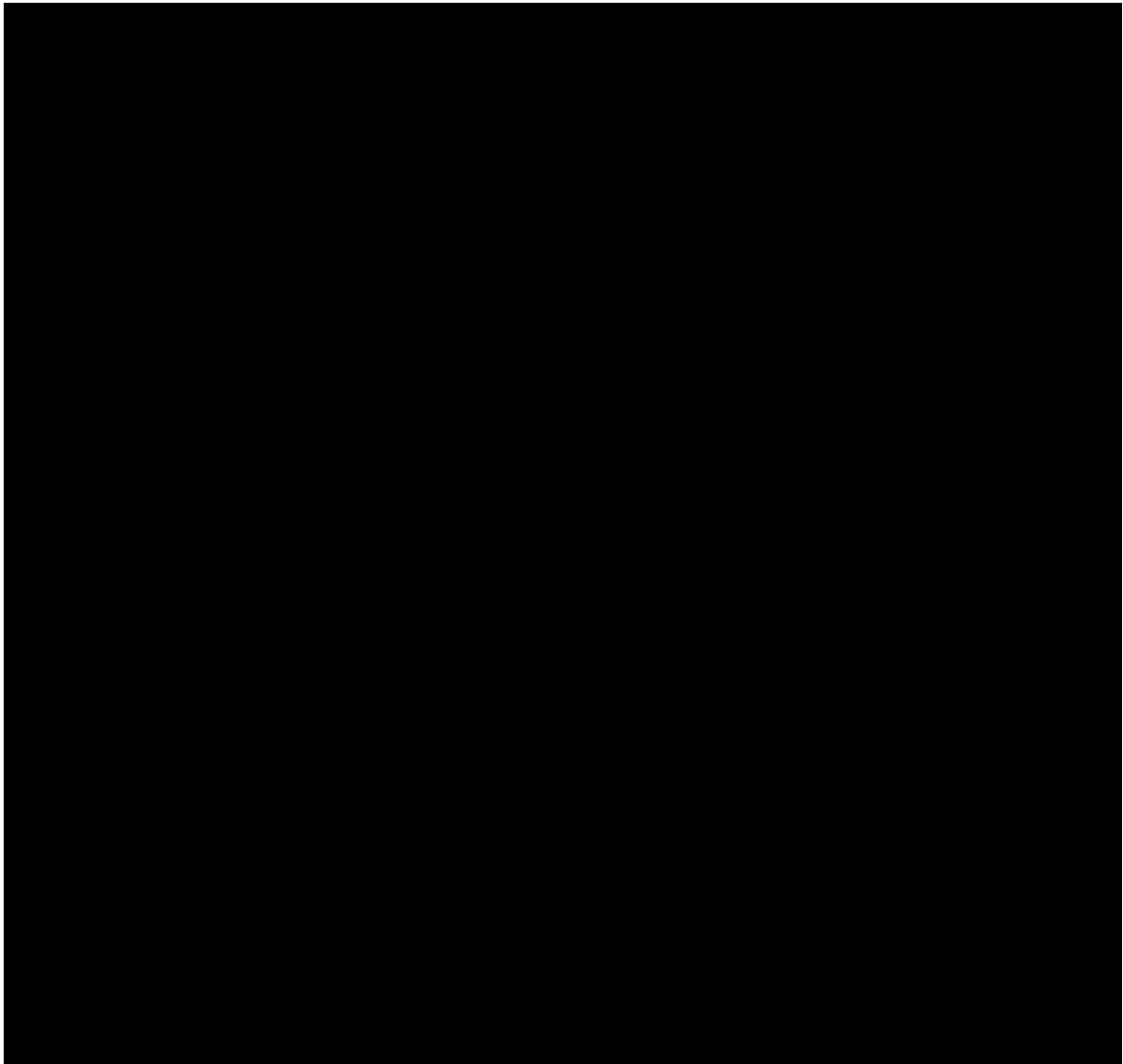


Figure 1.1: A view of the configurator, as seen by a customer, to select between various clock and headliner options.

such as, if there are problems in an associate's home life they may be distracted from the job, decreasing their performance. It is not reasonable to expect to know this information. However, there are some metrics that can be measured explicitly and implicitly about the associate and their work environment that may have an effect.

One such metric is the associates' experience. Many associates have been working at the plant for a very long time, but with a flow of new apprentices and associates changing roles, it is still a dynamic system. It may be expected that an associate with higher experience will do a better job than a less experienced member, but perhaps a newer associate has been more

recently trained and has not got any bad habits. Therefore, the effect of experience will need investigating.

Environmental effects will have indirect effects on the associate performance. An example of this is the weather. If the weather is cold and gloomy, as is common in the Winter, associate morale may be lower. Alternatively, when the weather is warm and sunny, it may feel like a waste to spend the day inside doing work, rather than being outside and enjoying the weather, negatively impacting performance. Additionally, temperature itself can affect the materials on the vehicles, such as the metal on the body panels. In hot conditions, metal will expand, and since the margin for error on body panel gap faults is so low, a small expansion could impact the error. This also may cause defects to be missed as the vehicle could have been built and inspected during hot conditions where the panels looked closer together, but it was a cooler day when it was inspected in Audit, which meant the panels compressed slightly, but enough to be out of the tolerance range.

On a related note, time effects can affect associate performance, and Audit performance. The day of the week can vary morale. For example, morale on Fridays could be higher as the weekend is the following day, but this may also cause associates to be more lax in their approach. This is another factor worth exploring. Additionally, the month of the year would likely have similar effects, relating to the climate and the proximity to holidays. One extra potentially influential factor could be sunrise and sunset time. Days with less daylight can have an impact on the moods of people. This would also take into account daylight savings inherently, where there is a large change. It will be valuable to find if there is a noticeable impact around these dates.

Another important factor in relation to the associates is their sensitivity to defects. This specifically applies to the associates in the inspection areas. [REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

However, undersensitivity is significantly more difficult to calculate than oversensitivity, as discussed in Chapter 4.

1.4 Contributions to Academic Knowledge

For the academic side, research towards new tree based methods was undertaken, as shown in detail in Chapter 6. The premise behind this research was to find a novel way of using P-Value adjusted decision trees, originally proposed by Lausen et al. [57], a method that generates trees by selecting and splitting on the variables with the lowest P-Value based on a given statistical test. The proposal combines this method with the method described in Khan 2020, Optimal Tree Ensembles (OTE), which creates a forest of decision trees and selects only the trees that contribute positively towards the ensemble [54]. This new method gives a P-Value Adjusted Selected Tree Ensemble, also known as PASTE, a selection of trees that are statistically significant in regards to the response variable and only contribute positively to the ensemble.

In addition to this, an adaptation of the OTE method with conditional inference trees, as proposed by Hothorn et al. [47], was created in order to test how well other tree growing methods work within the OTE framework. The method was called Conditional Inference Selected Tree Ensembles or CISTE.

Furthermore, a pragmatic approach to using prediction residuals and bootstrap samples was created to attempt to “boost” the predictive abilities of a trained model. This framework trains a model on the bootstrap samples, then uses this model to predict on the out-of-bag (OOB) sample, the probabilities are output and the residuals found. A gradient booster is trained on these residuals and the OOB sample (which has the same features as the original bootstrap sample). This can then be used on the original bootstrap sample prediction probability to correct for mistakes.

These methods provide a new way of using multiple established methods, alongside a new

theoretically simple idea.

1.5 Thesis Outline

This thesis steps through the process that built up the research problem. Starting with a literature review showcasing the cutting edge of trustworthy data science, answering questions about what it is, various areas of research within the topic and how some potential issues are avoided. Two trustworthy data science research groups are discussed, highlighting other ongoing research. The chapter is rounded off by giving conclusions and areas of trustworthy data science that could be further researched. Throughout the rest of the thesis, trustworthiness is always paramount and discussed when considering methods to use.

The following chapter outlines the original data at [REDACTED]. This is broken down into three categories: logistics, quality and Audit data. Discussing the original structure of the data is important to give context to problem as a whole, highlighting why a large study like this has not been undertaken on the data previously. Additionally, it adds further context to the following chapter, data engineering and feature selection. Here, all of the input features for the final methodological pipeline are created. The methods used to create each feature and a range of plots showing distributions, challenges and giving further context behind decisions about each feature. Furthermore, additional features that were considered and, in some cases, tested are discussed. These were all discarded from the final feature selection as they were too difficult to create, could not be implemented effectively or did not appear to give any value to the problem.

The next two chapters discuss the methods used within the final pipeline. This includes a detailed description of the established methods, covering the basics about each method and an in-depth walk-through of how some of the intricacies work. The novel methods chapter covers the newly tested and implemented methods: PASTE, CISTE and residual boosted variants. These utilise previously established methods: P-Value adjusted decision trees, conditional inference trees and optimal tree ensembles.

The penultimate chapter covers the final methodological pipeline. A brief discussion of the data and problem before setting up the final feature selection and method for testing the problem. This discusses whether the further feature selection is necessary, and whether the method should

only use extreme values that are more influential on the Audit results. Finally, the thesis is wrapped up with the conclusions about the benefits of the project, what was learned, directions for future research and an overview of the project as a whole.

Literature Review

2.1 Introduction

This literature review will give an insight into information surrounding the topic of trustworthy data science within a range of research areas, from the problems caused by untrustworthy methods, to good practices that minimise the risks of these problems.

2.1.1 What is Data Science?

“Data science is evolving as a new discipline from a family of intellectual disciplines exploiting the interplay of data and knowledge, building on rich data and domain knowledge to produce value in a variety of forms” [34]. This value can help provide solutions in any domain with sufficient data. In the current information era where big data is important, data scientists play crucial roles in many industries, earning it the name of the “sexiest job of the 21st century” [32]. Specialists can complete sections of the process, for example, a data engineer could process the data before the analysis, however, a data scientist may have knowledge of the methodological aspects of a data science project [79], for example:

1. Advising on planning projects and experiments,
2. Sourcing, collecting, and generating data.
3. Processing the data, making it usable.

4. Analysing and understanding the data, often using artificial intelligence (AI) or statistical methods.
5. Explaining and communicating the insights.

These insights can be found using specially designed tools, programs adapted from other methods, or bespoke programs designed to solve the problems faced, such as machine learning (ML) algorithms.

Potential problems arise, especially as algorithms get more complicated, and data sizes increase. Is the process ethical, will they use biases to further accentuate stereotypes? Can we trust the analysis, and the results that follow?

2.1.2 What is Trustworthy Data Science?

Data science methods are getting increasingly complex and data sizes are always increasing, making the tasks of understanding, explaining, and correctly completing a project even more difficult. Naturally, this could lead to more scepticism about the conclusions drawn. The research field of trustworthy data science looks at trust issues and aims to help understand and minimise their effects.

The Oxford English Dictionary defines trustworthiness as “able to be relied on as honest or truthful”. However, in data science, this can be applied in various manners. The data and analysis need to be complete and correct, otherwise, incorrect conclusions can be found.

Traditional Computing

Ensuring trust is not trivial, but there are certain properties that can be followed to improve the trustworthiness of data science. Bill Gates laid the foundations of trustworthy properties in his email to employees about trustworthy computing: availability, security, and privacy [37]. This was far simpler with deterministic computing, however with the introduction of advanced algorithms and AI, the lines were blurred and more properties were required to maintain trust.

Accuracy

In addition to the foundations, the accuracy of the data itself and the analysis became paramount. It can be a difficult task to guarantee accuracy with big data and probabilistic models, such as the data used for analysis on violence against women in Indonesia being inaccurate, leading to an incorrect crime rate [74]. A user should not trust inaccurate data or analysis, as the results will be incorrect. Improving accuracy will lead to greater trust in the results. However, it is important to note that accuracy and other metrics derived from data and machine learning techniques can only be used by acknowledging uncertainty, especially if these data or methods will be used for predictions. If we are to use these, we need to provide confidence intervals with averages and the predictions, meaning the interpretation of the metrics are not misleading.

Robustness and Fairness

The methodology needs to be robust and fair. Any new data needs to be processed and analysed correctly despite differences from previous sets. In the case of AI, the training data will need to be diverse enough to handle less common or new inputs. As shown when Google Photos automatically labelled black people as “gorillas” due to a lack of diversity within the dataset [71]. Problems are caused when there is a lack of robustness, leading to unforeseen biases. Fairness within data science is important since biases and other issues can naturally arise. For trustworthiness, these biases need to be removed to ensure there is no discrimination.

Transparency and Accountability

An assured method of maintaining trust is to be transparent and have accountability for any problems that may occur. Maintaining transparency in all areas is crucial as this can give clients a better understanding of how the data has been used and manipulated into the final result. This is a challenge for AI particularly as interpreting and understanding how an AI tool came to its conclusion is not always straightforward. The worst offenders are ‘black boxes’ where information is entered and a result is generated, without knowledge of the intermediate steps [105]. Accountability is important as it gives the creators a sense of responsibility, theoretically disincentivising wrongdoings.

Ethics

Furthermore, the process needs to be ethical. The data needs to be collected and adapted ethically and in accordance with any data laws in place. One of the difficulties with a number of popular AI tools, such as ChatGPT and DALL·E, is the ethics in their use cases [92]. There have been countless news articles about ChatGPT and further iterations scoring top marks on school and university work, and even successfully coding, potentially having vast impacts on how educative establishments test their students and how a wide range of jobs function [35, 51]. DALL·E and other image-generating tools have come under fire for stealing art and using it in the training data with the outputs heavily resembling that of the original, without any credit, as discussed in section 2.6 [81, 1]. These examples can be used in the context of data science as they cover unethical usage of data and potentially unethical deployment of analysis tools.

Finally, to create and maintain trust in the data science being presented, it is beneficial for any client to trust the data scientist and their motivations behind the analysis [33].

These are some properties that underpin trustworthy data science, and as such, the review will be structured to reflect this. Some hold more importance than others in various contexts. They will be discussed throughout, including examples of when the properties are not considered and the consequences that follow.

2.1.3 History

The idea of using datasets to calculate trends has existed for a long time. One of the first cases of using a clean dataset was in the 16th century as Tycho Brahe took astronomical observations by eye, but was surprisingly accurate. Johannes Kepler then used this data to develop his famous laws of planetary motion [26].

Modern data science grew out of statistics, with the term first coined in the 1960s. John Tukey saw the potential of using computers to do data analysis in 1962 as described in the paper "The Future of Data Analysis" [97]. Later, in 1977, he then wrote a paper suggesting to use the data to select hypotheses, using data to direct the analysis rather than just as observations [98].

By 1994, companies had begun collecting vast quantities of personal information, looking

for one of the first iterations of data-driven targeted marketing. However, these companies were overwhelmed by the sheer quantity of data, and could not use it to its fullest extent. Equally, they knew the potential of the data and kept striving to reach it [15]. This was an early look at data use trust issues. What personal information were these companies collecting? How were they using it?

The very first data protection law was passed by Sweden in 1973, requiring anyone who wanted to collect personal data to have a license [11]. Other regulations were put in place around the world to limit what could be stored and used, and in 1995 the European Union (EU) introduced the Data Protection Directive [31]. Two decades later, in 2016, the General Data Protection Regulation (GDPR) was introduced, further enhancing the trust of the general population relating to their own data privacy and security [30], with minor alterations made by the UK in 2018 for the UK's Data Protection Act [45]. Data consent is a crucial factor of GDPR. The user must explicitly be asked whether they consent to their data being processed and how it will be used. In turn, this improves data trust as the security and transparency of the data are maximised [13].

Data science appeared early on as a topic within the IFCS conference in 1996, that took place in Japan [43], and further continued as a research field in 2001 in William S. Cleveland's 'Data Science: an Action Plan for Expanding the Technical Areas of the Field of Statistics' [27], with this cemented in the launch of the Data Science Journal a year later. Soon followed by the Technical University of Dortmund offering a Master's course in Data Science in 2002. Nathan Yau gave an overview of what he believes a data scientist should be able to do: 'computer science, mathematics, statistics, data mining, graphic design, and human-computer interaction' [73], notably before the term data scientist was a common phrase.

Neural networks attempt to replicate how neurons in the brain work by making connections within a network. Deep learning neural networks create connections with many layers, allowing them to learn complex patterns across large data sets, and throughout this time have been improving. In the 2010s, the power of graphics processing units (GPUs) had become enough to train neural networks with significantly improved accuracy [106]. Generative neural networks skyrocketed in popularity in the late 2010s and early 2020s, generating pictures, text, and audio,

amongst others, with a simple prompt, leading to the present day.

Various aspects of data science are currently experiencing trust issues; users' data being collected and potentially processed by large corporations and ever-improving analysis methods, with the general public and even its creators not always understanding it [25]. These trust issues need to be dissected, understood, and solved.

2.2 Accuracy

Data is currently more available than ever before. Anyone can collect and store it, and make it available online. This works very well for crowd-sourcing efforts, where groups will collect data and amalgamate it into a dataset. However, this can cause potential variations of the quality within the dataset [41].

For some applications, automated generation of data is a necessity. One way to automate this is by using sensors, however, they cannot always be trusted. A famous example of sensors failing is in Google Maps. An artist successfully changed a green, traffic-free road, into a red, congested road by rolling 99 smartphones along the road in a cart with Google Maps open [100]. Alternatively, if a sensor fails, it can output dirty data, containing missing data, biases, and general noise [93]. Any inaccuracies within the data need to be dealt with, else they could cause issues later in the process.

AI and other non-deterministic methods are far more complex than traditional computing. They are probabilistic, making it far more complicated to predict the outcome. Two identical sets of input variables may lead to two different outcomes since certain algorithms will have randomness baked into them, such as Random Forest ensembles [20]. This aspect of data science needs to be understood to ensure the results are accurate and therefore trustable, despite slight variations in results. There are certain methods that can be used to verify the accuracy of probabilistic methods, such as formal methods [103] as discussed in section 2.8.3.

2.3 Robustness and Fairness

While a dataset may not have any errors, its distribution may not be sufficient to cover every necessary outcome and cause biases within the analysis, leading to unfair analysis. Various old datasets have been found to have biases within them, naturally leading to newer datasets attempting to remove the biases, but not always to great effect [95]. Understanding the biases is necessary in order to remove them. In extreme cases, biases can perpetuate and cause discrimination. As an example, Amazon created an AI tool to help with their hiring process, in turn, saving significantly on time needed to be spent on recruiting. However, for a number of job types, the AI would rate male candidates higher than female candidates due to the majority of the input data being CVs from the male employees [55]. There are tools available to help improve the fairness of a model, such as Microsoft's Fairlearn software [101].

For methods to be robust, they need to be able to adapt to any new data entered. If there is noise within the data, this can severely affect the performance of a model. The robustness and fairness of a model often depends on the data the model is trained on. However, this can be dealt with within the model itself or after the fact [59]. An example of a non-robust model failing in its analysis was when a self-driving car killed a pedestrian as it did not detect them [56]. The official reason for this was said to be the poor visibility but was found to not likely be the case. This AI was unreliable and not trustworthy, due to its lack of robustness and not adapting to the situation.

2.4 Transparency and Accountability

Transparency is a key factor in trustworthy data science. Sharing what data is used in a study and how it is processed enhances trust, as the results should be replicable [14]. For research influencing the public, such as medical trials, trust is paramount. An example of poor transparency is the testing of a drug for treating influenza. The company involved did not use published data, nor did it exaggerate the negative effects of the drug, leading to the drug being rolled out without full knowledge of the impacts [48]. This was using data in an anti-trust way and needs to be avoided to ensure trustworthiness.

The accountability of any issues within data science is a difficult subject. Who is responsible for the results and maintenance of trust and ethics? Many people will likely have different responsibilities within a project, and different interactions with the data, therefore it can be difficult to understand who is responsible for each output. One suggestion to improve accountability and responsibility is for data scientists to ethically examine their work and the outputs regularly but also being trained and working in an ethical environment should lower the concern about this [60].

Furthermore, as ML models become more capable, leading to increased public interest, the profit potential for the developing group increases. Previously, input parameters and data sources would be shared to increase trust and reproducibility, however, as more powerful models are created by large corporations, this is no longer the norm, as shown in Figure 2 in Solaiman [87]. Consequently, the trust in these "black box" models are reduced as it is unknown exactly what they are doing and cannot be reproduced. For improved transparency, there has been more research on white box models [99]. However, both black boxes and white boxes have their advantages and disadvantages [62].

2.5 Privacy and Security

Data privacy is more important than ever, given data is the most valuable asset in modern-day society [12]. Individuals and companies both have sensitive data that they may not want to share with corporations or the general public. An example of this is in automotive manufacturing. Naturally, these companies do not want to release their trade secrets to their competitors. More importantly, the employees manufacturing the vehicles have sensitive and identifiable information which all needs to remain private too. Without measures in place, such as anonymising the data or using data trusts (section 2.8.1), this information cannot be used by external data users. Furthermore, for corporations with distributed databases, maintaining security can be even more challenging. Data needs to remain secure and private when connecting across other systems. Only systems with the correct privileges should be able to access the data, which is more complex on large, distributed systems [107].

Maintaining privacy while analysing the data is not ensured, the data can be anonymised

and methodologies can be unbiased. However, ML models have the potential to find intrusive correlations [96]. For example, data could be collected from clicks and activity on a website that shows items from different cultures or areas of a country. While the data being collected is not against privacy laws, the data will be able to show a correlation that will be able to identify a user's ethnicity or culture. This is a privacy breach and would make results from using these insights untrustworthy.

2.6 Ethics

Another problem with the collection of data is related to the use of it, best described using an example. AI image generation is a hot topic currently, with many questioning the ethics surrounding it. Text-to-image generation is shrouded in controversy as the dataset [85] used in the popular tool Stable Diffusion [81] contains copyrighted artwork created by independent artists and corporations. Scraping this data from the internet may not be illegal, however, using it to generate work in a similar style is a legal and moral grey area. Currently, there is no precedent regarding its legality as it could fall under 'fair use' in copyright law. There are lawsuits against Stability AI (the creators of Stable Diffusion) in process with respect to using copyrighted material, one such example is from Getty Images [1]. To remain ethical as a data scientist, any data collected should avoid this grey area.

Another problem that has had plenty of media coverage is in relation to the ethics of the algorithms and the monetary benefits of rapidly deploying an AI system. The ethics of AI is a relatively new and diverse research field. Ethical problems can be incredibly varied, from the lack of data security to the sentience of AI [90] and even potentially plagiarising artistic works [44], as discussed previously. Using and deploying an ethical AI is no trivial task, but in order to maintain trust, it must be done.

There is not much money in ensuring the ethics of AI, disincentivising large corporations from focusing on the ethics before rushing their new technology out of the door, allowing for problems caused earlier, be it biased data or a problematic algorithm, to come to the fore and cause further damages, as addressed by OpenAI prior to releasing GPT-2 [88]. Recently, however, there appear to have been fewer precautions taken when releasing new technologies. GPT-2 was

released over nine months during 2019, to lower the risk of misuse and allow the public to “adapt to a world in which it is prudent to mistrust everything they read a little more” [88]. GPT-4, the most powerful generative language model created thus far, on the other hand, was released in full, while the world was still adapting to their previous iteration, ChatGPT.

2.7 Communication and Explainability

Effective communication is vital to trustworthiness, related to data science or not. In data science, communicating results and methods to those without background knowledge can be a challenge. During the COVID-19 pandemic, trust in the Center for Disease Control in the United States dropped by 10% [67], and this was likely due to the lack of effective communication [86]. Without communication, there is less trust in the results being released and the methods to find them.

A key area in communication and trust is explainability. If the methods used can be explained in simple terms, it is far easier to gain trust. The biggest unknown in many data science projects is the analysis, especially if it uses AI. Explainable AI (XAI) is more important than ever before as AI is given more control over the decisions being made in the world [17]. However, understanding complex algorithms initially or how they reached the result can be challenging.

Data visualisation is an important factor for the explainability and trustworthiness of data analysis. Being able to observe trends in the data directly allows us to come to our own conclusions, without the need to trust others’ judgements. Visualisations can be used for most types of analysis, there are even methods for the visualisation of how machine learning makes decisions. Packages such as SHAP [63] and LIME [80] explain how features and values within a dataset affect the response variables.

2.8 Further Solutions

2.8.1 Data Trusts

A method of mitigating many of the data science trust issues is by using data trusts. A data trust is a “proven and trusted framework to ensure exchanges are secure and mutually beneficial” [42], this means any group within the data science field can be within a trust, such as data processors or data subjects. It will not inherently increase trust itself, however, it can provide a way to make it more likely to trust trustworthy groups. These frameworks can cover levels of trustworthiness, ethics and any architecture that may be used [77]. A level of trustworthiness must be defined, and ethical rules must be put in place to ensure any use of the data is done in a morally correct way. For this to work, data trusts will have members that will have to adhere to the rules and maintain a high level of trustworthiness. The ethical rules should cover any ethical ambiguity as discussed in section 2.6.

Alternatively, instead of data trusts remaining as frameworks, they could be made into architectures. While data would not necessarily be stored within an architecture, a catalogue of where it can be found would be needed. This allows for easy accessibility between organisations within the trust, while maintaining security (i.e. data does not have to move between members) and the rules of the trust itself. Accountability is also accounted for as each member of the trust will hold each other accountable to maintain trustworthiness [77].

Being part of a data trust implies that the data being processed or used is trustworthy. Not increasing members’ trust, but inclining others to perceive them as trustworthy, leading to other benefits of trustworthy organisations.

2.8.2 EU Guidelines

The EU introduced some guidelines for trustworthy AI, ensuring AI is lawful, ethical, and robust [29]. One key idea was for human involvement in AI: human-before-the-loop, human-in-the-loop, and human-over-the-loop. This allows for humans to design the algorithms, maintain the AI, correct wrong choices, and continue improving the AI’s performance, as described in more detail

in Davinder et al. [50]. This has two main benefits; the AI is more accurate as any wrong outputs can be manually corrected, and this by definition means that AI will not replace humans in the workforce, but instead will complement them. The method of ‘before, during, and over the loop’ also allows for a better understanding of the cause of any problems that may occur, enhancing the accountability and transparency of the model [50].

2.8.3 Formal Methods

Trust is subjective, however, there are methods to verify the code will work as intended in regards to attributes of data science and AI, formal methods [103]. Verification tests are taken to see if the attributes can be proven, or alternatively proven to not always be true, i.e. a counterexample. If proven to always be true, the program will always work as intended and trust can be easier gain. In the cases of counterexamples, it can show that improvements are needed. This is far simpler for deterministic computing than for algorithms or ML tools since they can have randomness or hidden layers and machine-generated code. In addition, not all properties of trustworthy data science will be verifiable. There are methods to detect and un-bias data as well as methods to maintain fairness such as Microsoft’s FairLearn project [101].

2.9 Research Groups

Around the world, there are data science research groups, all looking at various areas of the field. Trustworthy data science, especially AI usage, is now under the spotlight particularly thanks to the recent, well-known advances in AI, as people are concerned about the potential of AI and their own futures. There are groups dedicated to looking at trustworthy data science and how trust can be maintained in the current age, including trustworthy data, analysis, and communication.

2.9.1 Research Alliance Ruhr

The Research Alliance Ruhr is a research collaboration with universities and other institutions with the aim to strengthen the Ruhr region in Germany [2]. Within this alliance is the Research

Center Trustworthy Data Science and Security [3], which focuses on a human-centred research approach, meaning humans are always in control of the intelligent systems. Any algorithms made need human verification and understanding in order to be researched and used. In addition to their trustworthy principles, they have experienced researchers working in collaboration with industries, as well as graduate programs, allowing the next generation of data scientists to have trust as a basis for their research.

2.9.2 The Alan Turing Institute

The Alan Turing Institute is the UK's "national institute for data science and AI" [5]. Due to the impacts of data use and AI, they believe it is paramount that research on these topics is developed ethically. It is open to collaboration with every university within the UK in order to allow them to collaborate and further enhance research. There are many forums that take place, such as the Trustworthy AI forum [9] where discussions about whether AI with humans in-the-loop or on-the-loop were trustworthy and ethical. Additionally, there have been further forums looking into topics such as keeping synthetic data trustworthy [8] and enhancing privacy and security of national identity systems [7].

2.10 Conclusions and Research Ideas

To conclude, trustworthy data science is currently an important topic in data science. Concerns about data privacy and the emergence of intelligent systems have never been higher. As a data scientist, it is important to be able to maintain trustworthiness in your work. It is important for programs to be accurate, robust, fair, transparent, accountable, private, secure and ethical, and communicated effectively once the results have been determined.

Trustworthiness in data science and AI is an evolving topic, many changes are likely to be seen over the coming years as algorithms become more advanced and the lines between human thought and computer generated ideas become blurrier. As seen throughout the research for this review, trustworthy AI has been predominant in regard to the current literature, and many resources are being put into understanding how to keep AI safe and trustworthy. However, AI

is just one part of data science, and data science is just one part of AI, therefore more research needs to be done in the context of trustworthy data science. Here are a few suggestions:

- How is privacy and security maintained on a distributed, production critical database?

Distributed databases have problems, especially in regards to security and privacy, as they store data in many tables across the database. Relying on connecting keys between tables also has the potential to cause trust issues. Research into how distributed, production critical databases is vital for companies to ensure their databases do not go offline through any activity, yet there does not appear to be a vast amount of research into it.

Data Descriptions

3.1 Data Introduction

[REDACTED]

[REDACTED]

[REDACTED] Nonetheless, there is software

available allowing quick and relatively easy access, despite the complex structure.

There are two main databases in use throughout this study; the logistics database and the quality database. The logistics database contains all the information about what each vehicle is, where it has been, when it will get to other areas and everything else relevant to the movement and description of the vehicle and work process surrounding it. The quality database contains all the information about captured information, including defects, necessary checks to ensure processes have taken place, and other necessary scans that can help track what has happened to the vehicle during its production. The information below is not necessarily the data used in the final prediction model, but contains the features that were considered for it, or what was required to engineer other features.

3.2 Logistics Data

The logistics database consists of many standalone tables, and there are two used in this study. The first of which contains information related to the order information and specification of the vehicle: this will be referred to as the orders table. The other contains information related to the location of the vehicle at different times, and will be referred to as the trackpoint table.

The orders table has an entry for each vehicle [REDACTED]

[REDACTED]. The model of the vehicle is also selected as it will be necessary to filter for only one model at a time to decrease the complexity of the problem. Since the problem is to find the factors of vehicle manufacturing that cause the Audit score to be worse, the specification of the vehicle is important. Therefore, the upholstery colour, exterior colour, and the specific options for each vehicle can be taken from this table.

The trackpoint table contains [REDACTED]

[REDACTED] This table gives a view of how each vehicle moved through the production line during its time in the plant. In terms of features, this can be used to find when a vehicle was in a particular station.

3.3 Quality Data

The quality database consists of [REDACTED]. A capture report was made to put all the useful information in one place.

This capture report contains information about different types of captures made during the vehicles' journey through the plant. Some of these captures will be made on all vehicles, such as required checks that an associate completes when they add a specific part. Some captures can be as simple as an associate scanning themselves into a station, giving a record of who worked on any given vehicle at any given time. However, the captures that will be used for the study are relevant to when an associate finds a defect with a part of the vehicle.

When a defect is captured, multiple fields are filled in: [REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED] This information provides a detailed report of any defect.

If a defect is reworked further along in production, this will also be reported on in the same entry, along with [REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

3.4 Audit Data

The data for the Audit inspection and results is available within the quality database, however, there are edge cases that show why this version of the Audit score should not be used. The quality data stores all of the Audit results, however, [REDACTED]

[REDACTED]

[REDACTED]

[REDACTED] Therefore, a separate Audit database is used, maintaining an up-to-date record of the valid Audit captures. The dataset used from this database simply contains the VIN and the Audit Score. The Audit Score for a vehicle is defined by the Equation 3.1:

$$[REDACTED] \quad (3.1)$$

[REDACTED] is defined in Table 3.1. [REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

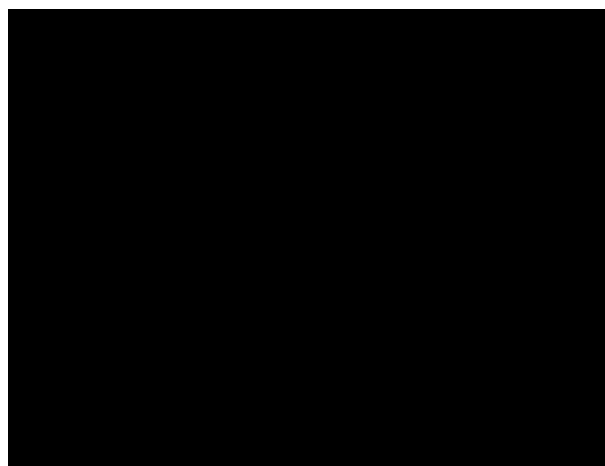


Table 3.1: The weighting value for every quality severity score.

Data Engineering & Initial Feature Selection

It was logical to use features that contained the full detail of the production process as previous tests had proven unsuccessful. This idea was originally not used as the "curse of dimensionality" was a consideration, however using all features as a starting point and reducing that number is a straightforward and logical route to take. Therefore, for every station on the production line, certain features were chosen and recorded, producing a station-wise landscape of production for each vehicle. This gives a well defined problem: **Can the information gained from each station regarding the quality and movement of vehicles determine whether a vehicle will pass or fail its final audit inspection?** Originally, the two most frequently built vehicle models were going to be used, but after tests using some of these features below, one of the models performed very poorly, most likely due to its larger variation in build specification. This is something to be considered for future work, but only one model, the ██████, is considered hereafter.

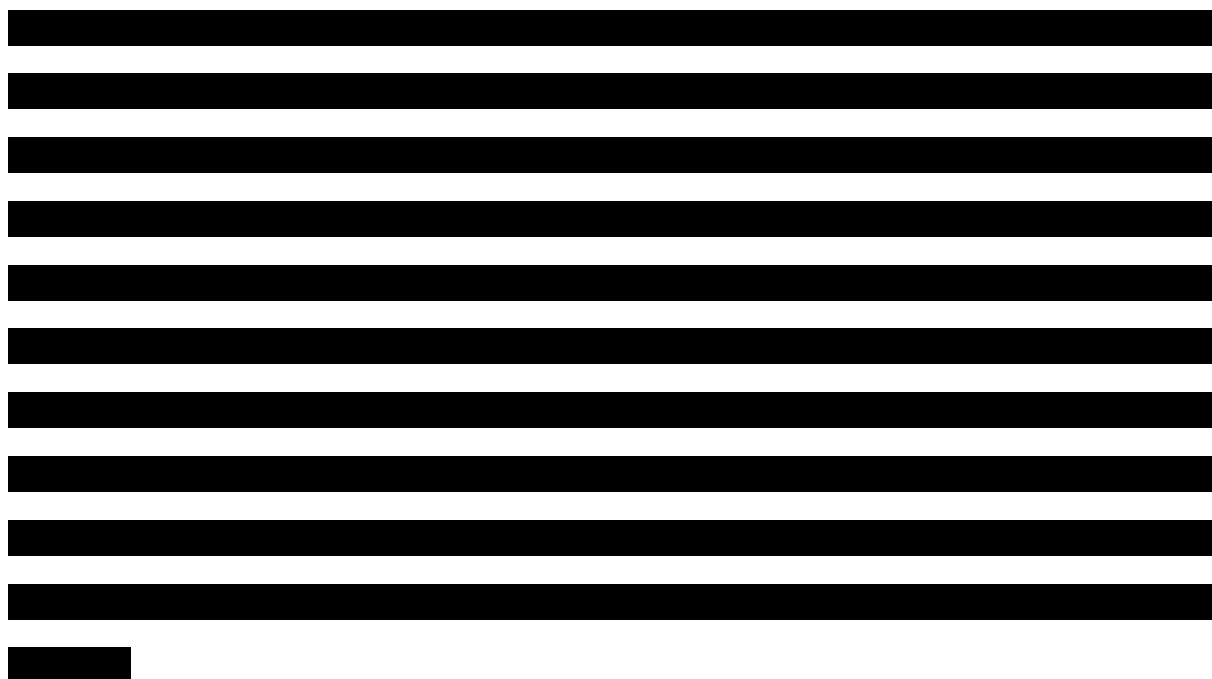
4.1 Station-wise Features

One of the difficulties with such a question is the complexity. Within each station there can be hundreds of minor differences between each vehicle, for example, how far a screw is tightened, changes of personnel, the temperature within the plant, et cetera. However, the influence of many of these is unlikely to be significant.

Combining information gained from domain experts and the data available provides a set of features that can be used and will likely determine the Audit inspection outcome. Earlier considerations for the features used summary values, [REDACTED]. This removed the granularity needed for such a complex system. Therefore, any feature that was easy to interpret, to maintain explainability, and could be extracted from the main databases, to make an application of the method simpler, were all considered and input into a further feature selection process, discussed in Chapter 7. For the initial feature selection, any features with $> 98\%$ identical observations before feature selection have been removed in order to quickly reduce the feature space without losing important information. This reduces the number of features from 808 to 251.

4.1.1 Number of visits to a station

As the vehicles make their way through the production process they will flow through a range of stations. [REDACTED]



The boxplots in Figure 4.1 shows many stations have no variance in the number of times a vehicle visits it. [REDACTED]



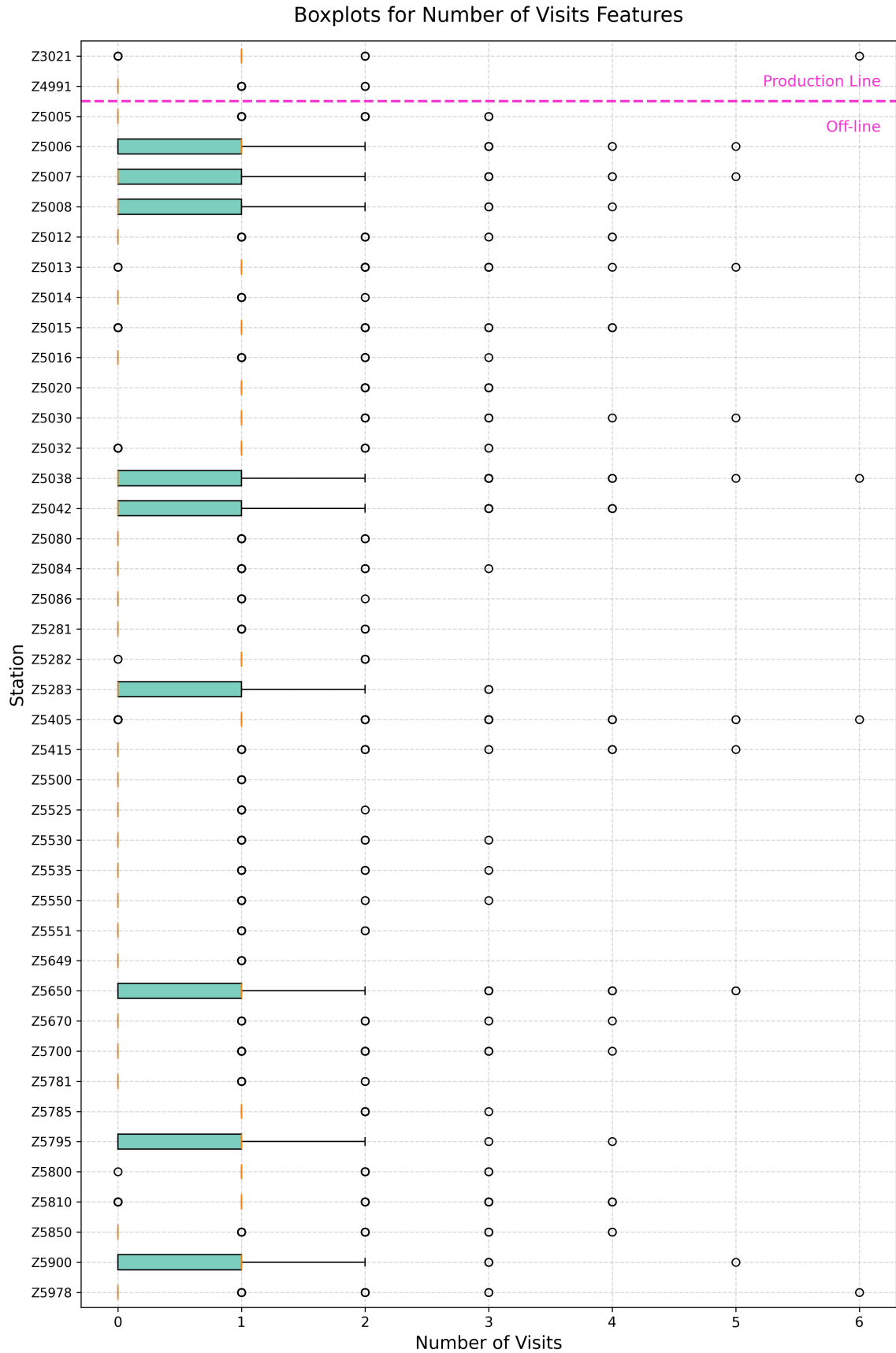


Figure 4.1: Boxplots showing the number of visits to each station. Stations with the same value for $> 98\%$ of observations have been removed. The pink line represents the end of the production line where the vehicle moves into the off-line section of production.

[REDACTED]

4.1.2 Time after beginning of shift

[REDACTED]

[REDACTED] Throughout each shift, the associates will naturally vary in performance due to fatigue and other factors. Although, these are subjective and very difficult to define. One way to define an effect in a similar vein is by taking the time after the beginning of the shift, measured in minutes. Since each observation in the data is an individual vehicle, the time the vehicle enters each station after the beginning of the shift can be taken.

As shown in Figure 4.2, the range is between [REDACTED]

[REDACTED]

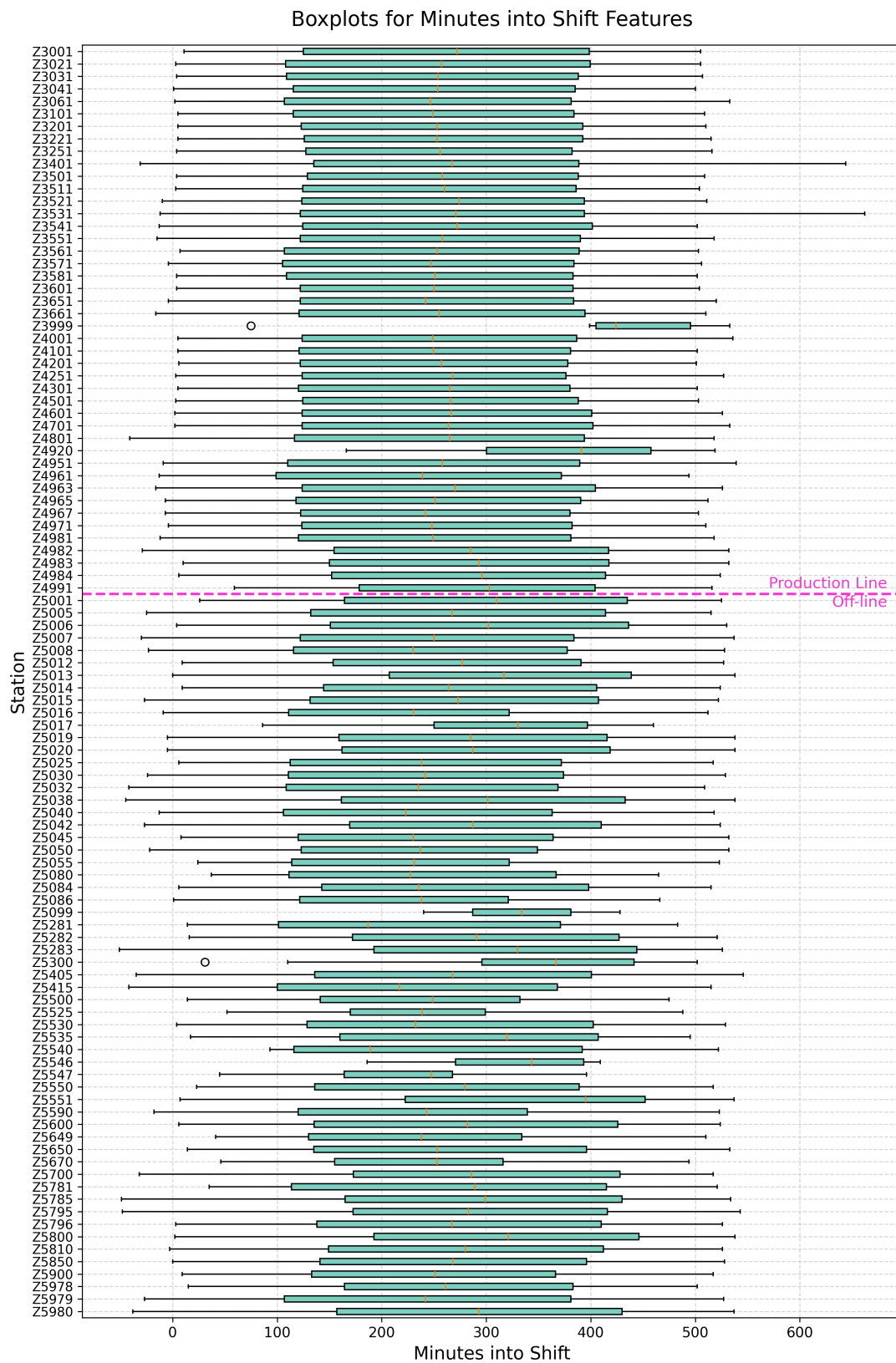


Figure 4.2: Boxplots showing the number of minutes into a shift when every vehicle entered each station. The pink line represents the end of the production line where the vehicle moves into the off-line section of production.

4.1.3 Time into station

The time at which a vehicle enters the station is similar to the time after the start of the shift, except it provides some further information about whether it was in the [REDACTED]

[REDACTED] The time unit is in hours. Since there is a large gap [REDACTED], there is no need to make it into a cyclical variable since there will be no continuity between days. [REDACTED]

Similarly to the time after the beginning of shift, [REDACTED]

4.1.4 Quality index score found in each station

[REDACTED] as found in Equation 3.1, [REDACTED]

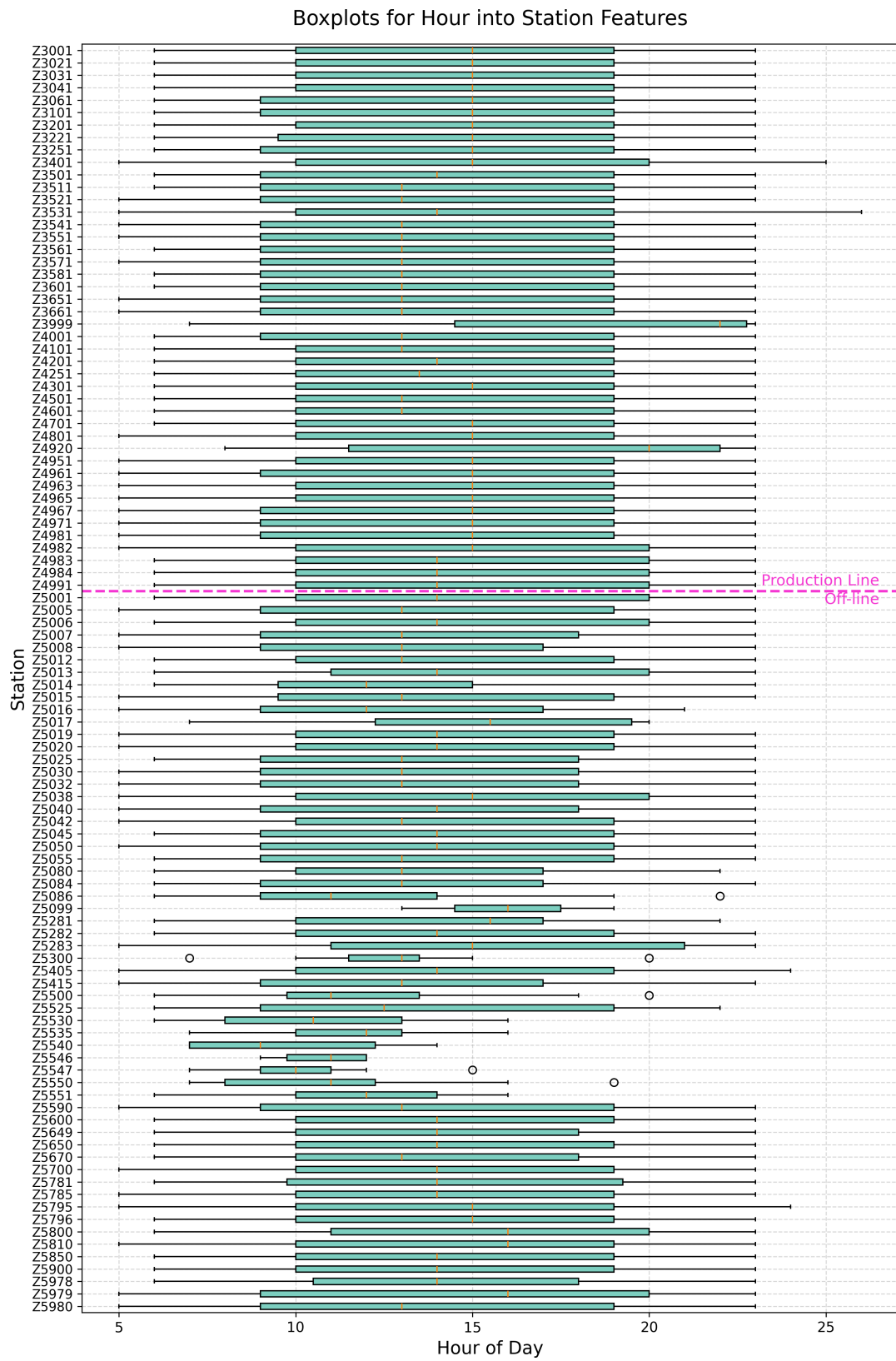


Figure 4.3: Boxplots for the hour the vehicles entered each station. The pink line represents the end of the production line where the vehicle moves into the off-line section of production.

[REDACTED]

[REDACTED]

[REDACTED]

As seen in Figure 4.4,

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

4.1.5 Checked OK and Checked Not OK percentage

As briefly outlined in Chapter 3,

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

$$[REDACTED] \quad (4.1)$$

[REDACTED]

[REDACTED]

[REDACTED]



Figure 4.4: Boxplots for the quality index score captured at each station. Only values between [redacted] are shown; any values outside this range are outliers. Stations with the same value for > 98% of observations have been removed. The pink line represents the end of the production line where the vehicle moves into the off-line section of production.

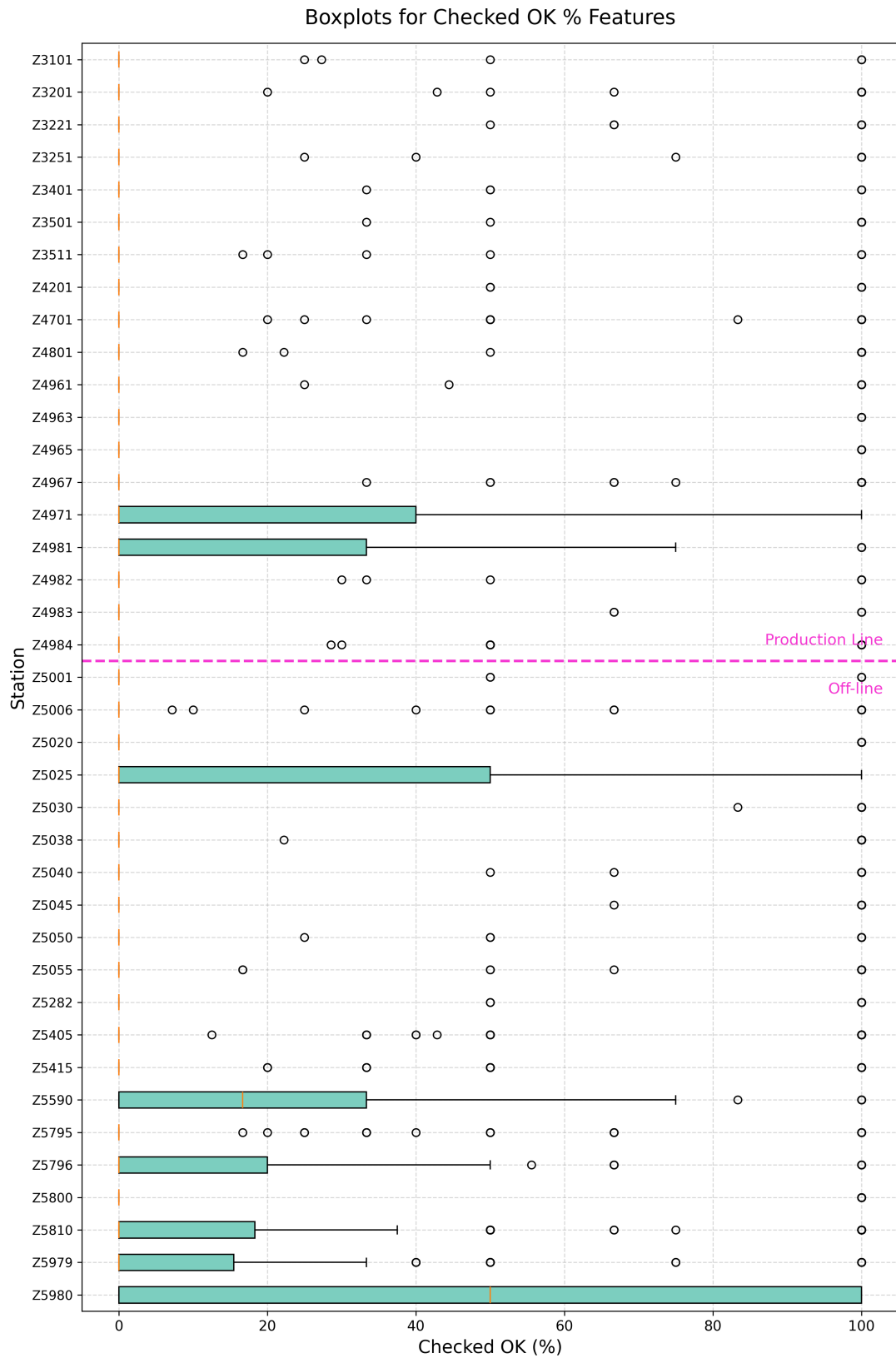


Figure 4.5: The Checked OK percentage at each station. Stations with the same value for $> 98\%$ of observations have been removed. The pink line represents the end of the production line where the vehicle moves into the off-line section of production.

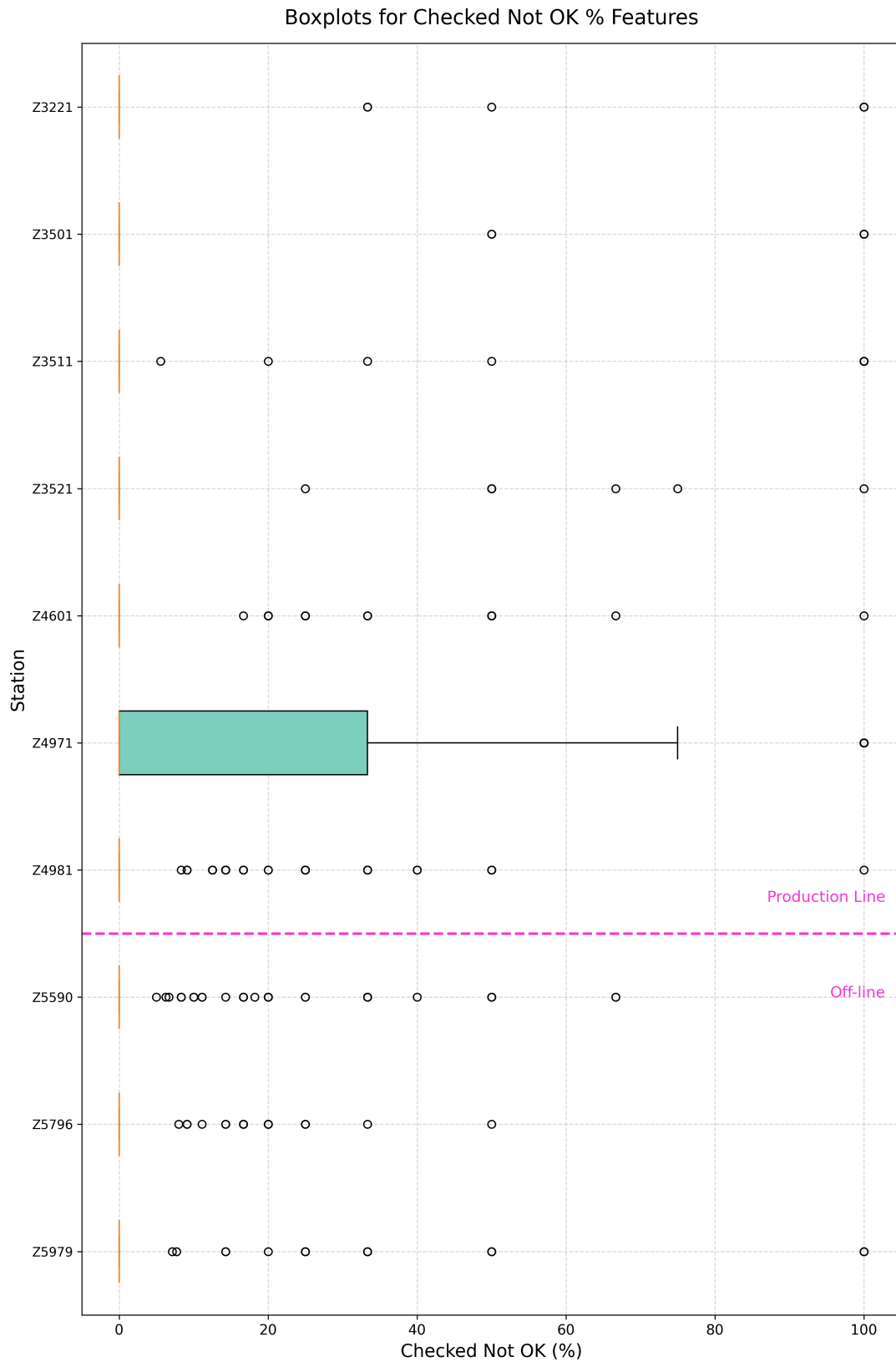


Figure 4.6: The Checked Not OK percentage at each station. Stations with the same value for $> 98\%$ of observations have been removed. The pink line represents the end of the production line where the vehicle moves into the off-line section of production.

As seen in Figure 4.5,



4.1.6 Time spent in station

The total time a vehicle spends in each station is a basic measure. This will show if a vehicle has spent an extraordinary amount of time in any given station. There could be a number of causes for this,



As can be seen in Figure 4.7,



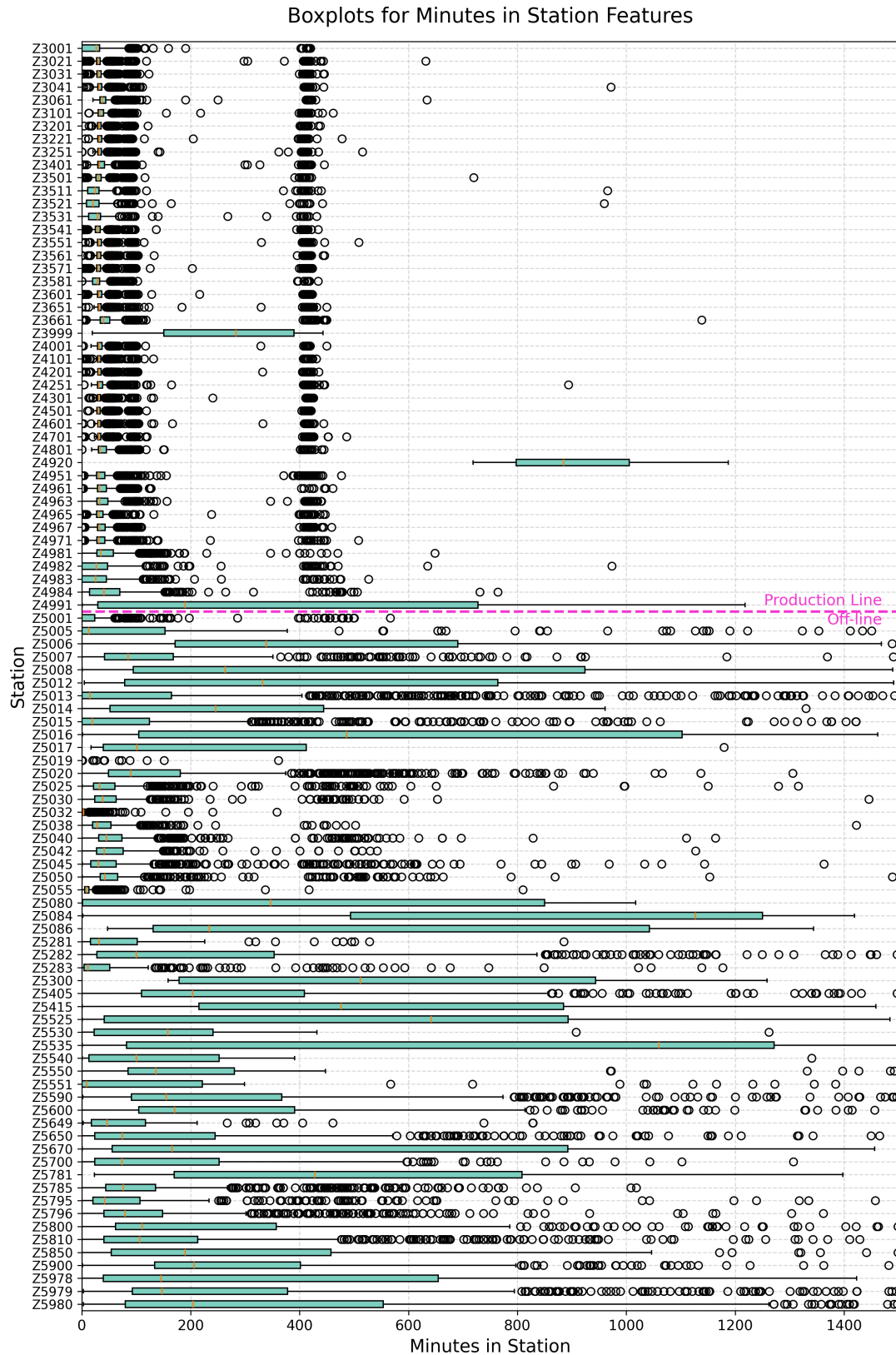


Figure 4.7: The number of minutes vehicles have been in each station. This is a compressed view, showing only [redacted]; an extended view is shown in Figure 4.8. The pink line represents the end of the production line where the vehicle moves into the off-line section of production.

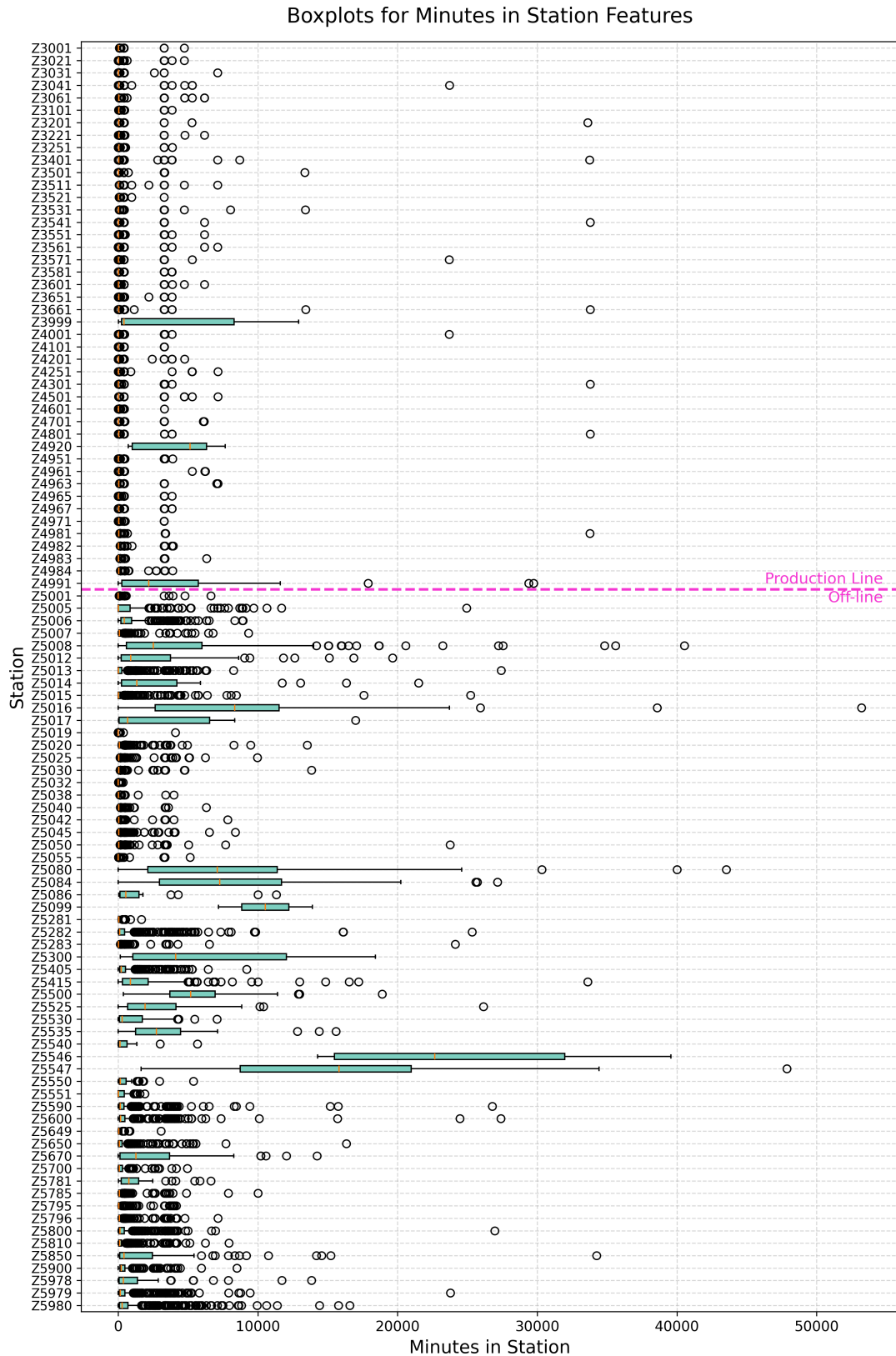
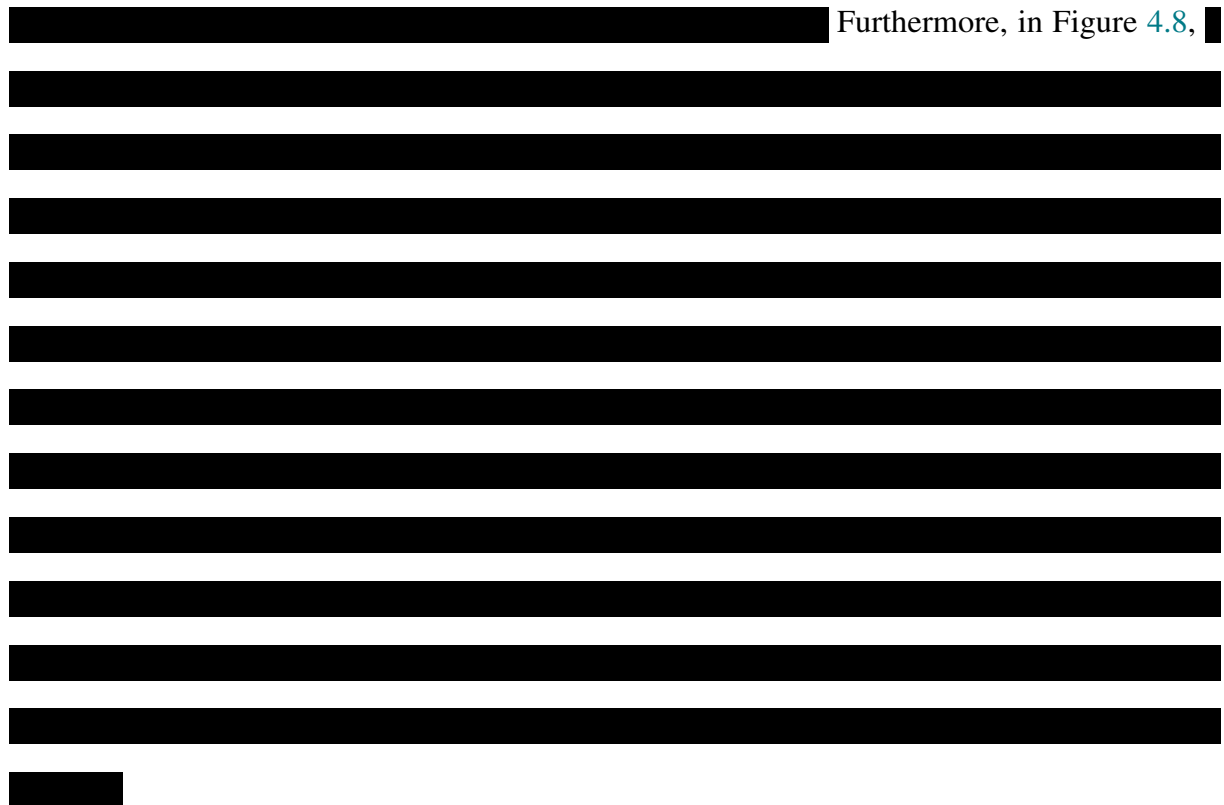
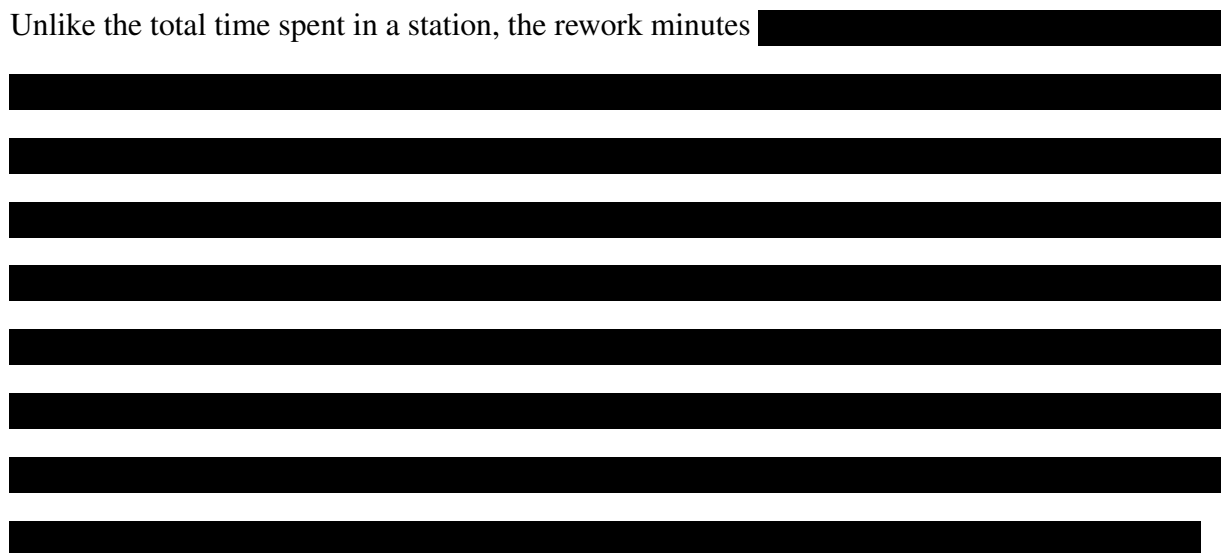


Figure 4.8: The number of minutes vehicles have been in each station. This is the full view, showing the maximum values up to [REDACTED] for a very extreme case. The pink line represents the end of the production line where the vehicle moves into the off-line section of production.

Furthermore, in Figure 4.8, 

4.1.7 Rework Minutes

Unlike the total time spent in a station, the rework minutes 

As seen in Figure 4.9, 

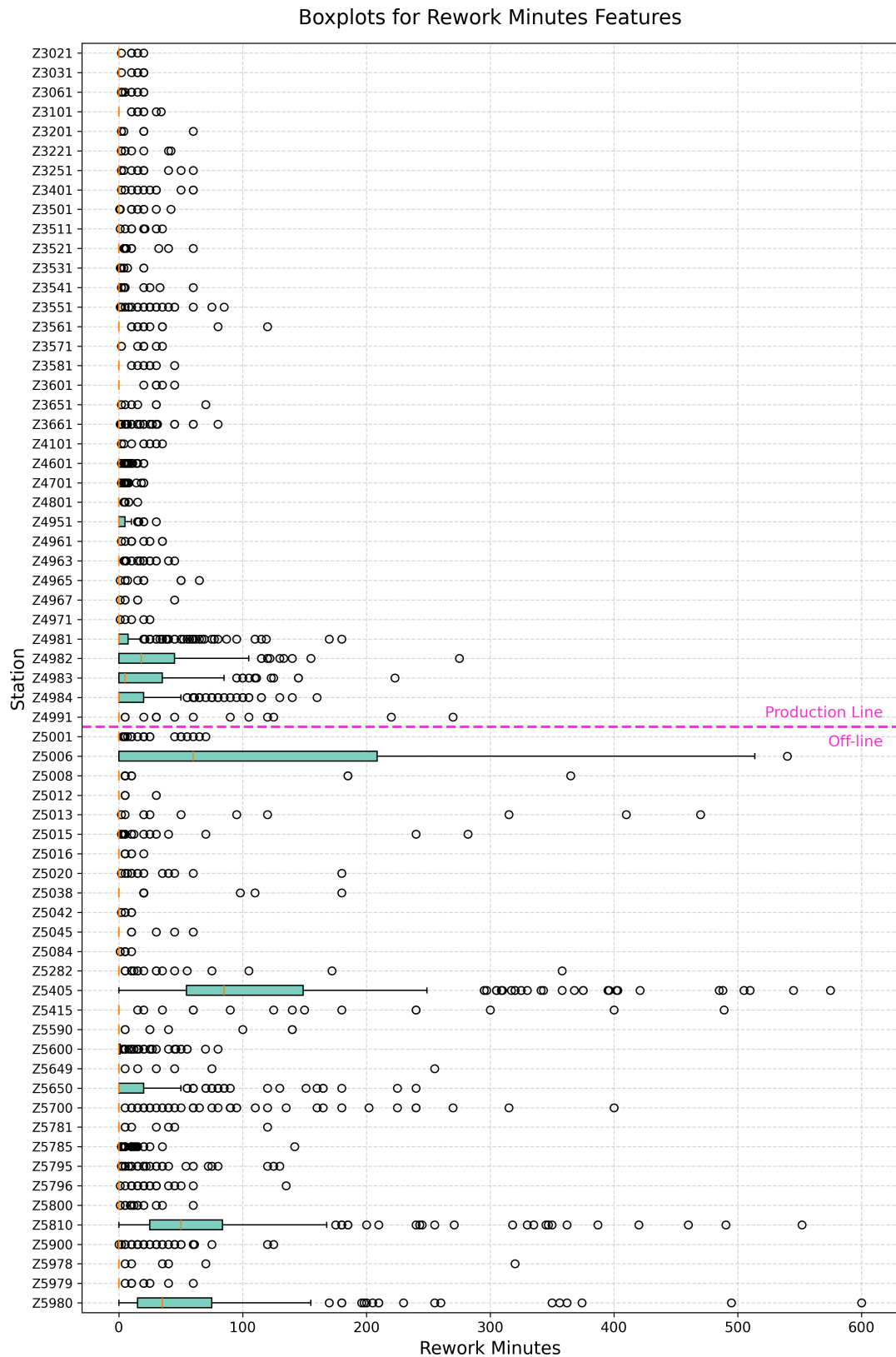


Figure 4.9: A boxplot showing the total rework minutes per vehicle at each station. Only values between [redacted] are shown, with [redacted]. The pink line represents the end of the production line where the vehicle moves into the off-line section of production.

[REDACTED] This effect is seen both here and in Figure 4.7. [REDACTED]

4.1.8 Additional features

There are two other features included before feature selection. As discussed previously, there are two main sets of colours for each vehicle: the exterior colour and the upholstery colour. [REDACTED]

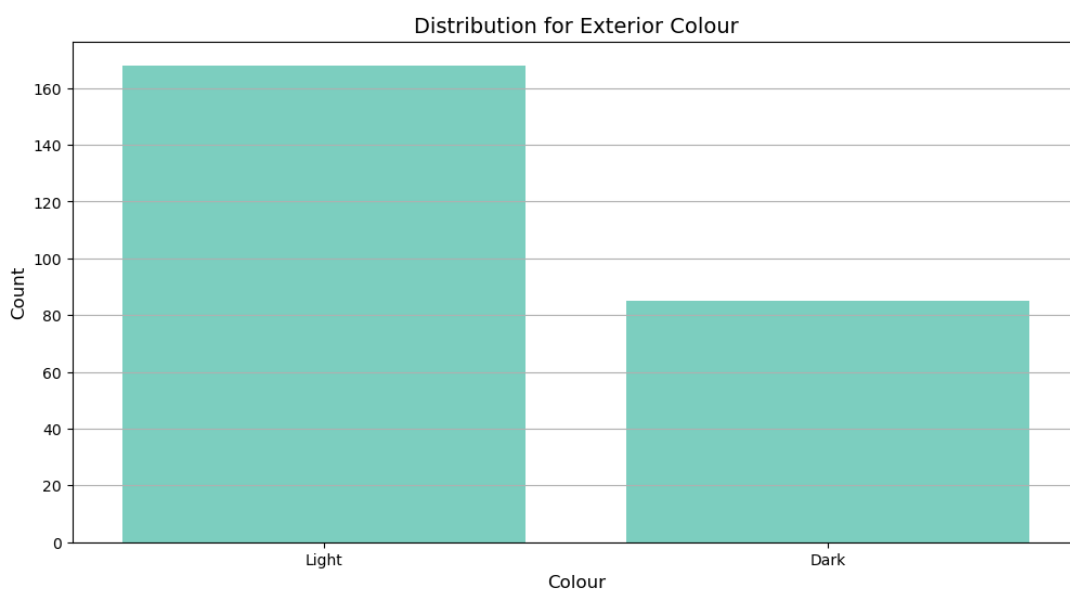


Figure 4.10: A bar plot showing the distribution of the exterior colour for light and dark colours.

As shown in Figures 4.10 and 4.11,

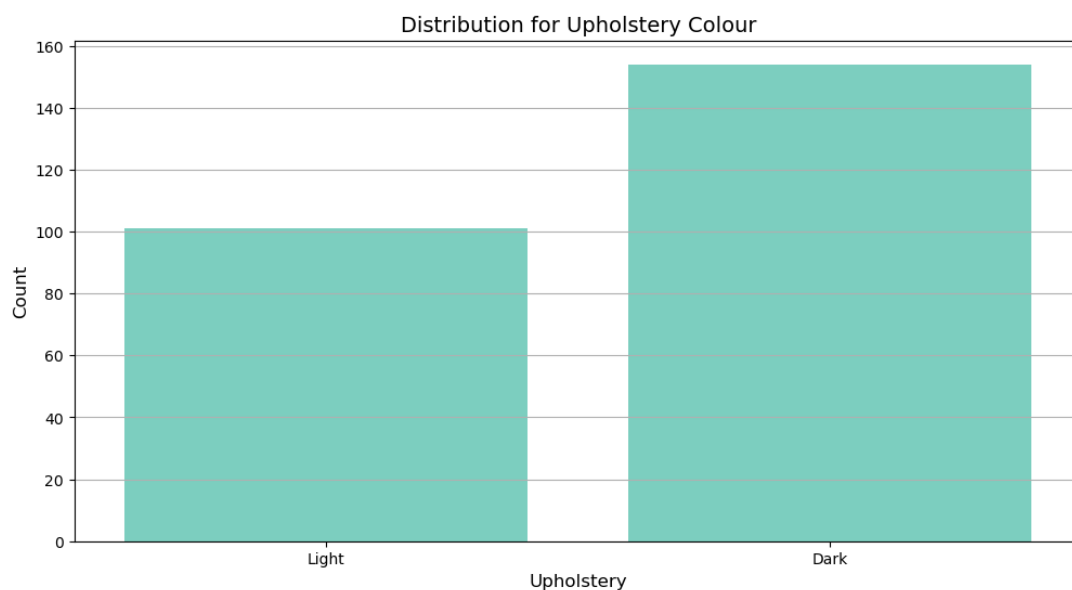


Figure 4.11: A bar plot showing the distribution of the upholstery colour for light and dark colours.

4.2 Other Considered Features

Although these are the final set of features, there were other features that were created and tested. These used other methods than all other features, specifically looking at using embeddings, but were not used in the final model as they added complexity and lowered interpretability without any extra performance.

4.2.1 Defect Type and Place Embeddings

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

Initially, all fields are processed separately, removing punctuation, capital letters, uninfor-

mative words and strings which contain non-alphabetical characters and then reducing each of the words to its root via lemmatisation. The total set of words are then combined and are trained using FastText [16]. This method captures semantic meanings and relationships of every word and creates a vector representation for each word. FastText can consider sub-words too, where parts of words can also get semantic meanings, which can be useful for any non-standard words, which may be found in real-world, free-text data. For this example, [REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

4.2.2 Reworker Embeddings

[illegible]

[illegible][illegible]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED] [46]. [REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

4.2.4 Station Undersensitivity

This image consists of ten solid black horizontal bars stacked vertically, completely obscuring any text or graphics that might have been present. The bars are uniform in width and height, spanning most of the page's width.

[REDACTED]

[REDACTED]

[REDACTED]

Related Methods

5.1 Introduction

To find the highest performance method for predicting the Audit outcome, many well established methods were tested. These methods are described below in detail. While they may not all be well suited to the problem at hand, it was nonetheless beneficial to test how they performed to further understand the data. Since none of these methods had been used on this data before, the success or failure of an established method is furthering the knowledge, therefore providing a benefit. Additionally, since the data the methods will be predicting on is binary, and not numerical, the numerical versions of the methods below will not be discussed.

5.2 Random Forests

Random forests were first proposed by Breiman [20]. In this paper a methodology was described that uses many tree predictors to form an ensemble, a predictor that is far more robust than an individual tree when given unseen data.

5.2.1 Decision Trees

In their simplest form, decision trees are a series of questions with binary answers that eventually lead to a result, a class prediction [21]. They start at the root node and branch out to the final

nodes, the leaves. For example, a dataset could be the celestial bodies in the Solar System, this includes planets, dwarf planets, asteroids, the Sun, comets, et cetera. Starting with all of the objects, they can be split in a binary manner in many ways, for example, two groups, planets and not planets, giving two branches. From here, planets can be further split into smaller categories, gaseous planets or not, and so on. Splitting a small, well defined category like this will end after only a few split points, where one label dominates the rest. For example, if it is a planet, not a gaseous planet, and has no visible surface detail; the resultant label is most probably Venus, thanks to its thick atmosphere.

However, for more poorly defined examples, it is not as simple. For a prediction, a dataset will contain many entries for each label, where entries could be from observations at given time intervals. For Mars, if an entry was made during a dust storm, there would be no visible surface detail, as shown in Figure 5.1. This means the leaf contains multiple labels and therefore would give class probabilities.

The training data would determine the probability in this leaf, Venus would have all of its entries with no visible surface detail, while Mars would only have a few. For example, over 100 entries, the ratio of Mars to Venus could be 3 : 97, for example. Therefore, for a simple tree, the probability of predicting Mars from this set of features in this specific leaf would be $\frac{3}{100}$ while for Venus, it would be $\frac{97}{100}$.

That is the general idea behind a decision tree, and how they split data. However, they can also deal with continuous data just as easily [21]. In a continuous variable, the data can be split into two subsets, just as in categorical variables, by choosing a split point. Data on one side will make up one branch and data on the other side will create the second branch. The maximum number of split points is $n - 1$ for n unique entries for a given variable.

One of the challenges is finding the best split point amongst all of the possibilities in all types of variables. There are multiple methods to this, the two most common are using the Gini impurity and Cross-Entropy [19], as defined below:

$$\text{Gini impurity} = \sum_{i=1}^k p_i(1 - p_i), \quad (5.1)$$

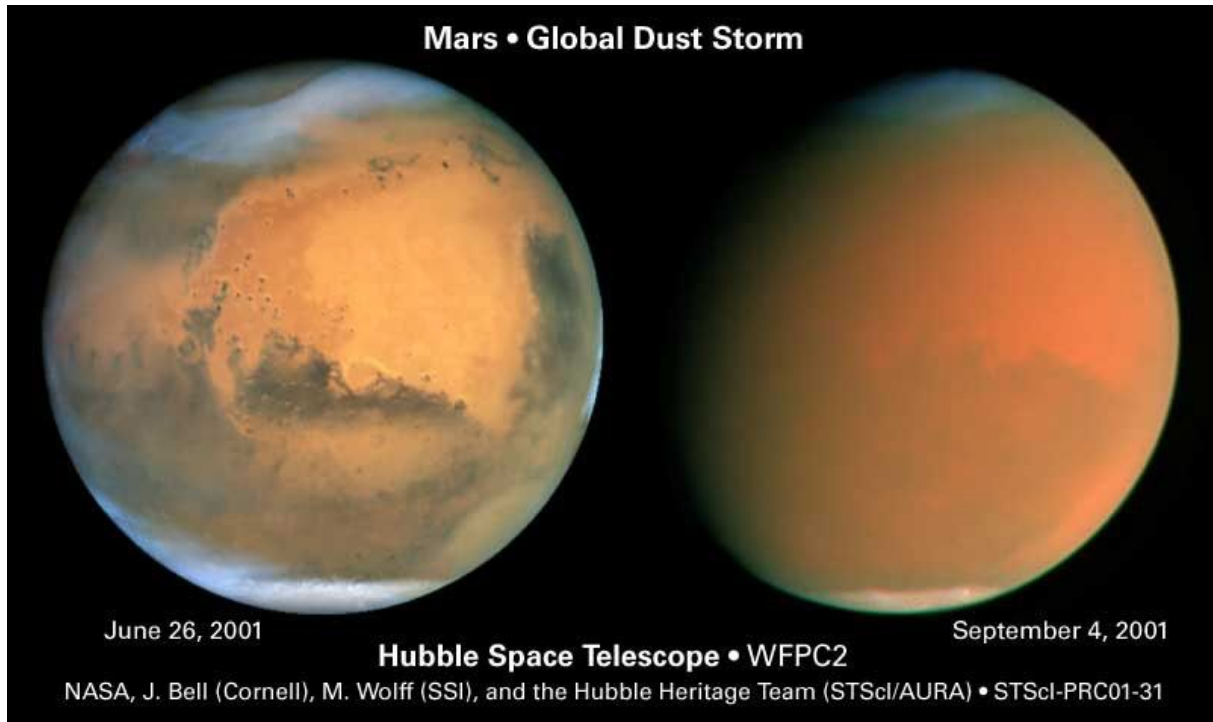


Figure 5.1: The change between visible surface detail and no visible surface detail on Mars during a dust storm as picture by NASA [72].

$$\text{Cross-entropy} = - \sum_{i=1}^k p_i \log(p_i), \quad (5.2)$$

where k is the number of classes, and p_i is the proportion of cases that belong to class i . The Gini impurity is a measure defining how likely it is that a new, entirely random data point is misclassified when given a random class label at a given node according to the node's class distribution. Cross-entropy measures the uncertainty of a class distribution in a given node. Both give very similar results, and minimising the value is the objective. Gini impurity is less computationally expensive, therefore is used as the default for many Pythonic tree based methods. During the training of a decision tree, the best split for a node will be determined by the split point that gives the lowest Gini impurity or cross-entropy score [21].

Decision trees are prone to overfitting [21]. For example, if a tree continues to split until the leaf nodes are pure, regardless of how few samples there are in the leaves, it will likely be overfitted. This is because the tree has been tuned too finely when there is no information to

gain in doing so. Often this means that the tree has been tuned to noise. In turn, this makes predictions unstable and unreliable. Therefore, there are methods to reduce overfitting. Reducing the maximum depth of the tree and requiring a minimum number of samples in order to split again are two methods that can prevent the model from tuning too finely.

Additionally, there are two methods discussed in this thesis on how to grow a decision tree: level-wise and leaf-wise [24, 52]. Level-wise will grow the tree from the top-down, layer by layer, balancing the loss reduction from each side of the tree at the same time. Leaf-wise builds the tree one branch at a time, maximising the reduction of loss at each data split. Given no restraints on depth or splitting, both methods will result in very similar trees. However, since depth is often restricted, using level-wise or leaf-wise growth will make a difference. A leaf-wise growth is likely to learn quicker with fewer splits than level-wise, but is also more prone overfitting because of this, and therefore more regularisation is necessary.

5.2.2 Ensemble method

A single tree can be used as a successful predictor, however they are sensitive to changes in training data. Even one small change can give a completely different tree, which will lead to different prediction results. This high variance makes them unstable and unreliable in many use cases. This tree diversity can be exploited, however, when combining the abilities of many trees into an ensemble [21].

For Random Forest, many trees are grown, each on a random subset of observations, a bootstrap sample. This randomly selects observations, with replacement, thereby allowing for repeated observations in the same sample [20]. The remaining data not selected can be used as an out-of-bag (OOB) sample. Alongside the sampling, a random selection of the available features is selected for each split in the tree. This creates the necessary diversity for the forest. Every individual tree makes its own prediction. The final prediction of the ensemble is simply the prediction that most of the trees agree on, although this is only true for classification tasks. In effect, this reduces the variance of the individual trees by averaging out the effect of them all. Naturally, overfitting can still occur, especially if the data is noisy. However Random Forests are far less prone to this than individual trees [20].

There is, once again, a range of tuneable parameters to improve the training speed, robustness and other properties the forest. The parameters mentioned in Section 5.2.1 are still applicable. In addition, the number of trees, and the number of features are parameters that can affect the computational performance drastically, especially given larger datasets.

As expected, Random Forests do have some drawbacks compared to simpler models [20]. As there are many trees involved with an average being taken for the final result, the interpretability is somewhat limited, unlike for a single decision tree where the route taken to find the prediction is clear. Nonetheless, this is somewhat counteracted by the improved stability of the feature importance values compared to a single tree, which can help provide more reliable insight towards the cause of the problem.

5.3 Optimal Tree Ensembles

Optimal Tree Ensemble (OTEs) is a method developed by Khan et al. [54], and further refined in a subsequent paper [53]. This uses an ensemble of trees and reduces the complexity of the forest by selecting only the trees that improve the performance. A detailed method is shown in Section 6.2 using the same framework as the original OTE method. However, in its original state single decision trees were grown instead of the condition inference trees and the p-value adjusted trees shown in the later chapter.

As an overview of this method, the data set is split into a range of bootstrap samples with replacement, and corresponding out-of-bag (OOB) samples. For each bootstrap sample, one tree is grown and error value $\hat{err}_t$ is calculated using this formula, as found in [53]:

$$\hat{err}_t = \frac{1}{|\bar{S}_t|} \sum_{\mathbf{x}_i \in \bar{S}_t} I(y \neq \hat{y}), \quad (5.3)$$

where $|\bar{S}_t|$ is the size of the OOB sample, y is the true class label for the bootstrap sample, $\hat{y}$ is the predicted class label for the given tree and $I(y \neq \hat{y})$ is an indicator function that states if y matches $\hat{y}$ with 1 if they do not match and 0 if they do.

From here, the trees can be ranked in order of lowest error to highest error [53]. In the

implementation of the method into code, there is a parameter that can remove the highest-error percentage of trees, such as the top 10%, to further increase the training speed by removing trees that are unlikely to be of benefit. Starting with the tree with the lowest error, the Brier Score is calculated using the corresponding OOB sample to that tree [22]. The Brier Score is a measure of accuracy for probabilistic predictions. It is defined by the equation:

$$BS = \frac{1}{N} \sum_{t=1}^N (p_t - o_t)^2 \quad (5.4)$$

where p is the predicted probability, o is the observed value across N observations. Thereafter, the 2nd tree is added to the first as an ensemble. Once again, the Brier Score for this new ensemble is calculated. If the Brier Score for the 2nd calculation is lower than the 1st, then the 2nd tree remains in the ensemble and the process continues with the 3rd lowest error tree. However, if the 2nd calculation has a higher Brier Score than the first, then the 2nd tree is removed from the ensemble and the process continues, as before, with the 3rd tree, but not the 2nd. Once every tree in the selection has been tested, the final ensemble is available and consists only of trees that have added value to the ensemble.

There are benefits of this model over a random forest. The final ensemble is theoretically less noisy than a random forest, reducing the variability of predictions. As shown in Khan et al. [53], fewer trees are necessary to achieve similar results to a much larger random forest, therefore also reducing the prediction time. However, training time increases compared to a Random Forest since calculating the errors and Brier Scores after growing the trees takes additional time, especially since the Brier Score calculations need to be completed in sequence.

5.4 Gradient Boosting

Gradient boosting is a type of machine learning method that uses an ensemble of weak learners in order to create a powerful predictive model. There are various specific algorithms that utilise gradient boosting. In this thesis, three gradient boosting algorithms are discussed: XGBoost, LightGBM and CatBoost [24, 52, 78]. The latter of the three is discussed in Section 6.2.3, with

XGBoost and LightGBM discussed below.

5.4.1 XGBoost

XGBoost is a popular method that uses gradient descent for boosting [24], starting with weak learning decision trees to end up with a powerful ensemble. Initially, the first tree is trained on a training set of the data as a level-wise decision tree, the base learner. The residuals of this base learner are calculated, by the predicted values subtracted from the actual values. Further trees are then grown using this new information, trying to minimise the loss via a functional gradient descent. Each new tree calculates a negative gradient of the loss function based on the prediction results [24]. This is different to gradient descent, since XGBoost uses prediction results rather than adjusting model parameters, as would be seen in a Neural Network.

To start the process, a set of default probability predictions need to be chosen, with the same length as the number of observations. The simplest in the case of a binary classification is an array of 0.5 values, sat exactly in the middle of both classes. These are then compared to, and subtracted from, the real values which will be either 0 or 1, resulting in an array of 0.5 and -0.5 values. For future trees, the values will diverge from ± 0.5 . These residuals are fundamental to the process, and are the gradients referred to within gradient boosting. The root node of the first tree consists of these gradients. A similarity score of the root node is found using the formula below:

$$\text{Similarity Score} = \frac{(\sum_i g_i)^2}{\sum_i h_i + \lambda}, \quad (5.5)$$

where λ is the regularisation factor, g_i is the gradient values and h_i is given as $p_i(1 - p_i)$. However, the equations for both g_i and h_i will vary depending on the loss function being used. Here, the binary cross-entropy loss is used:

$$L(y_i, p) = -[y_i \log(p) + (1 - y_i) \log(1 - p)]. \quad (5.6)$$

It is also worth noting that the original paper from Chen and Guestrin [24] does not mention a similarity score, but it is introduced as the optimised value of the regularised objective function. Nonetheless, the similarity score plays as an easy concept comparatively, and will continue to be used.

Once the score is found for the root node, the best split for the node needs to be calculated. Every feature is tested and every unique value is tested within each feature. Each value is sorted in ascending order, then with the first two values, the mean is taken and the node is split depending on if the gradients are higher or lower than that value, giving two child nodes. For this split, the gain of each split is calculated using the similarity score of each of the nodes [24]:

$$\text{Gain} = \text{Score}_{\text{left}} + \text{Score}_{\text{right}} - \text{Score}_{\text{root}}. \quad (5.7)$$

Then this process is repeated for all values and features in the data. The split with the largest gain is selected as the split point. This whole process is repeated until the tree is fully grown, which is determined by a specified maximum depth or a minimum number of samples in a leaf. Once the tree is produced it is pruned from the bottom up based on another given parameter, γ . If γ is more than the gain of a given node, then the node is removed from the tree, and this is repeated for the parent node. If γ is less than the gain, then the node remains, even if parent nodes had lower gain. The process of split point selection and pruning is determined by a generalised formula in Chen and Guestrin [24]:

$$\mathcal{L}_{\text{split}} = \frac{1}{2} \left(\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda} \right) - \gamma, \quad (5.8)$$

where I is an instance set on a leaf, I_L is the left leaf and I_R is the right leaf [24]. The first tree is complete, and the output values of the tree are needed in order to influence the second tree growth. The outputs, or weights w , are given by the generalised formula [24]:

$$w = -\frac{\sum_{i \in I} g_i}{\sum_{i \in I} h_i + \lambda}. \quad (5.9)$$

Now the results can be taken from the first tree and added to the default predictions for the second tree. In the case of classification, log odds need to be used to maintain the probability between 0 and 1. It is also important to note that the new corrections are multiplied by a learning rate to avoid overfitting. The next tree is grown and the cycle is repeated until every tree is created and the final result is found using predictions from all grown trees, as in an ensemble [24].

XGBoost has the benefit of using past results to improve future trees in place of a random selection of trees that Random Forest has. However, this will take longer to train, particularly in the greedy form of the algorithm where every split point is considered. There are optimisation that can be made, and are discussed in the Chen and Guestrin paper [24]. Selecting optimal values for the regularisation parameters, λ and γ can be a challenge too, and selecting incorrectly will vary the ability of the model and could lead to overfitting or underfitting.

5.4.2 LightGBM

LightGBM is yet another gradient boosting machine [52], using similar principals to XGBoost but executed in a different manner. There are two main differences; tree growing and histogram based learning.

As mentioned previously, LightGBM uses leaf-wise tree growth rather than level wise that XGBoost uses. Instead of finding a split that maximises the gain on all child leaves, it selects the leaf with the largest gain, and continues to grow leaves from that side, resulting in a deeper and narrower tree. This method of growing trees can produce good predictions with fewer splits, but is more prone to overfitting [52].

Histogram-based learning changes the search for the split points. In XGBoost, it is a greedy search, meaning every possible split point is considered, but histogram-based learning optimises this by using bins instead. Each feature has a histogram created for it with a chosen maximum

number of bins [52]. For example, a feature could have values between 0 and 10 and a maximum number of bins at 2, which would result in bins with values 0 to 4 and 5 to 10. Once the bins have been created, each one will have the sum of gradients and Hessians (as represented by h_i in Equation 5.5) from every data point within it at the given node. The algorithm will then consider splitting the data at each bin, giving values that are less than or within the bin and greater than the bin. The gain of each split is calculated as the sum of the gradients and Hessians between the splits. This can massively reduce the complexity of the problem compared to XGBoost, particularly for data sets with many potential split points. Once the splitting takes place, the algorithm can further use the histograms on the child nodes too, but more optimally than starting from scratch again. Since the parent node has all the data from both of the child nodes and the histograms were known, calculating the histograms for one child node is all that is necessary, as the difference between the parent and child nodes can be found to create the histograms for the other child [52].

Naturally, using binning comes at a price, each split is no longer guaranteed to be the best possible split, resulting in a potential loss of accuracy [52]. This effect is less influential on smaller data sets as there are fewer split points, however, the faster training time from the binning method will not be as useful on a smaller data set too. This benefit and drawback can be balanced by changing the maximum number of bins, but there is no “one size fits all”.

5.5 Logistic Regression

Logistic regression is one of the simpler methods of predicting using machine learning as the problem can be presented a single equation [94]:

$$z = \mathbf{w} \cdot \mathbf{x} + b \quad (5.10)$$

$$= w_1x_1 + w_2x_2 + w_3x_3 + \dots, \quad (5.11)$$

where $\mathbf{w}$ is a vector of weights, $\mathbf{X}$ is the matrix of observations and b is a constant. This equation shows that each observation has its own value of z , while $\mathbf{w}$ and b remain constant for

all observations. z is then followed by a sigmoid transformation of z :

$$\sigma(z) = \frac{1}{1 + e^{-z}}. \quad (5.12)$$

This gives a sigmoid function in terms of z . For each observation, z varies and results in a different corresponding value for $\sigma(z)$ in the range $[0, 1]$, giving a probability for the observation. Since this is a binary prediction, the usual method to determine whether the prediction is 0 or 1 is whether the value of $\sigma(z)$ is less than or greater than 0.5.

To create a model for predictions, the weights $\mathbf{w}$ and the constant b need to be optimised for the dataset. This can be done by minimising a loss function, such as the log loss function:

$$\mathcal{L}(\mathbf{w}, b) = -\frac{1}{m} \sum_{i=1}^m \left(y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right), \quad (5.13)$$

where y_i is the true value for the observation, $\hat{y}_i$ is the predicted probability and m is the number of observations. Gradient descent is used to find the optimal values for $\mathbf{w}$ and b , and subsequently achieves the trained logistic regression model [94].

An issue with logistic regression is that it assumes there is a linear relationship between the features and the log odds of the response variable [94], therefore strong correlations between features can cause issues. While this may have limited impact for many examples, such as age and eye colour, within a production line where many outcomes are somewhat dependent on the sequence of events that led up to it, there will be some neutral correlation between features. Therefore, it is not likely to be the most powerful method for the data within this thesis, but due to its simplicity and explainability it is necessary to test how it performs.

5.6 Artificial Neural Networks

Artificial neural networks (ANNs) cover a vast array of methods and ideas within modern machine learning. ChatGPT is a very complex example of a transformer which is a specific

type of ANN. Nonetheless, the ANN discussed within this section is far simpler, a feed-forward network with backpropagation. The specifics of the used ANN will be described later on within this section.

ANNs differ from other machine learning methods as they can learn the features and structure of the data reducing the need for feature engineering, making it more appropriate for many use cases of large data. ANNs were originally inspired by the neurological systems in the brain by using a set of connected neurons to communicate with each other to give an output [82]. Every feed-forward network is organised into a set of layers, the simplest design is an input layer, a hidden layer, and the output layer. Each layer contains a set of nodes, and for a feed-forward network, each node in one layer connects to every node in the next layer [39].

The input layer contains as many nodes as there are features. These are all connected to the nodes in the first hidden layer. For each connection, a weighting is applied to the original value. The new value is then input into hidden layer where a non-linear activation function and bias is applied and outputs a new value. The process is repeated for all hidden layers until the output layer is reached. For the use case in this thesis, the output layer has one node with a potential value in the range $[0, 1]$. Similarly to logistic regression, an output probability will be given and can be classified based on a given threshold value. This is a basic description of a feed-forward network. Various changes can be made to this network to improve predictive performance; changing the number of hidden layers, changing the size of the hidden layers, changing the activation function and making use of backpropagation.

Once the output layer gives an output, the performance can be determined using a loss function. As with previous methods, the loss function needs to be minimised to get the best results for the network. To give feedback to the network from the loss function, backpropagation is used [83]. This starts at the output layer and works backwards through the network by calculating the gradient of the loss function with respect to every connection weight, and by using gradient descent, updates the connection weights to best optimise the loss function.

There are a number of popular activation functions: ReLU, Leaky ReLU, sigmoid and softmax. Each of these have their own uses, but the first goal of each of these is to create non-linearity in the model, allowing the network to make non-linear relationships between the

inputs and outputs. ReLU (Rectified Linear Unit) has a function as defined by [70]:

$$f(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}. \quad (5.14)$$

If the input value into the node, including the weights, is less than or equal to 0, the ReLU function will clamp it to 0, while anything above 0 will remain as it was. Any negative values before the activation function are set to 0, effectively turning off the node. However, this can also lead to a perpetual "off state", a phenomenon called "dying ReLU" where some nodes are permanently stuck at 0. When the input for a ReLU node is constantly negative or 0, the gradient of the activation function remains at 0, meaning the gradient descent cannot take place, therefore the node will never be updated and remain at 0, a dead node. To remove this issue, a "leaky" version is used, Leaky ReLU [64]. A small but significant change where negative values are mapped to a slope, rather than a flat 0 value, as shown in the definition below:

$$f(x) = \begin{cases} x & x > 0 \\ \alpha x & x \leq 0 \end{cases}, \quad (5.15)$$

where α is the gradient of the slope for negative values. This reduces the issue with dying ReLU and maintains the effects of the ReLU function.

Softmax is another non-linear activation function, particularly useful for networks with multiple output nodes [39]. In effect, it collapses the output values to the range $[0, 1]$ by normalising exponentials of the input values, but importantly the values of all the outputs sums to 1, and can be interpreted as a probability. Unlike other activation functions, such as ReLU, this is almost exclusively used immediately prior to the output layer, and will take inputs from all previous nodes to output the total probability distribution across all classes. The definition for the output vector derived from softmax function is as defined below:

$$\sigma(\mathbf{z}_i) = \frac{e^{\beta z_i}}{\sum_{j=1}^K e^{\beta z_j}}, \quad (5.16)$$

where β is a constant, usually set to 1 and K is the number of output nodes.

Similarly to XGBoost and logistic regression, gradient descent is used to train the model and improve the results. In this model, the weights of the connections are changed to decrease the loss. These weights are initialised at the beginning of the training process, but can have an impact on the time to optimise the result and the performance. Unlike methods such as XGBoost that uses a flat constant value for the initialisation, this should not be used for an ANN as all the nodes within a layer will learn the same features, heavily restricting its ability. Instead, a random set of values is used in order to allow nodes to learn different features [39].

There are many specific methods to this randomness, but the method used in the thesis is using a form of the Glorot initialisation [38]. This is defined as the uniform distribution:

$$\mathcal{U}\left(-\frac{1}{\sqrt{n_{in}}}, \frac{1}{\sqrt{n_{in}}}\right), \quad (5.17)$$

where n_{in} is the number of feature input into the connection. This initialisation method creates a stable variance of initial parameters across the connections, reducing the chance of weights becoming too large or too small.

ANNs come with their own set of challenges too. The complexity is generally higher than many traditional machine learning methods, reducing the explainability. Setting up the best network for the data is virtually impossible, as there are an unlimited number of arrangements of layers and nodes and inherent randomness. Arguably worse for explainability are the hidden layers that are not explicitly defined or easily interpretable by the developer, creating potential trust issues, especially in sensitive applications. These issues can be indirectly improved by using model analysis methods, such as SHAP values, giving insight into feature contributions and model behaviour [63]. Nonetheless, the performance of a well-tuned ANN can outperform other powerful methods due to their ability to learn the structure of features and data implicitly.

For the final tests as found in Chapter 7, a specific structure of neural network was chosen before hyperparameter tuning. The chosen neural network is a basic feed-forward network, consisting of the input layer, two fully connected hidden layers and the output layer. The first hidden layer has 64 nodes, and the second has 32 nodes. Each hidden layer has dropout, randomly disabling a percentage of the nodes at each layer to further reduce overfitting [89]. The percentage is left as a hyperparameter to be tuned. The activation function of each hidden layer is ReLU and a sigmoid function converts the values into probabilities between 0 and 1 for the output node. In addition, the sampling weight of the positive class is reduced to 0.75 to readjust for the imbalance of the data. The Adam Optimiser is used to optimise the loss function, which is binary cross-entropy.

5.7 Semi-Supervised Generative Adversarial Networks

Generative Adversarial Networks (GANs) have become popular during the early 2020s and are particularly known for the ability to generate images similar to a training data set, originally proposed by Goodfellow et al. [40]. This method uses two neural networks that work against each other in a balancing act to find the most successful overall model. These two models are the generator and the discriminator. The generator creates fake data, with the goal to create a data set that matches the distribution of the training data set. While this may sound beneficial on its own, a distribution that matches the training set may produce results that are impossible to have. For example, if the original dataset contained images of both cats and dogs, a generator could create an image of an animal with unique features from both species. This output would match the distribution as it is visually plausible. However, in reality it is impossible to have such a creature. To counter the generator, the discriminator is trained simultaneously against it, in a game-like scenario. The goal of the discriminator in the game is to be able to identify whether the data it is fed is real data or generated data. If a discriminator determines that data is fake, its output node will give a 0, and if it believes it is real, the output node will give a 1. A discriminator that makes no mistakes is also problematic as it will not give any useful information for the generator to train on and the learning stops. This can lead to the generator producing a limited variety of samples. Therefore, finding an optimal balance between the two models is key to creating a

successful GAN [40].

For a more detailed description: the generator starts with random noise and its network generates data somewhat resembling the real data. This is initially completed in the same way as the neural networks above, using a feed-forward network with backpropagation. The discriminator takes two inputs, real samples and the generated samples. Here is where it determines whether each sample is real or fake. Since the generator output is directly connected to the input of discriminator, any feedback to the gradient from the discriminator's loss function L_D is fed back into the generator. The generator then starts again with random noise and improves its samples to try and fool the discriminator, and this process continues [40]. During the training, the generator is trying to minimise its own loss function:

$$L_G = -\frac{1}{m} \sum_{i=1}^m \log(D(G(z_i))), \quad (5.18)$$

where m is the size of each minibatch of samples (as explained in more detail after Equation 5.20, z_i is the random noise, $G(z_i)$ is the generated sample from the noise and $D(G(z_i))$ is the estimated probability that the sample is real from the discriminator. This shows that the generator wants to fool the discriminator as much as possible [40]. Meanwhile, the discriminator has its own loss function to minimise:

$$L_D = -\frac{1}{m} \sum_{i=1}^m \log(D(x_i)) - \frac{1}{m} \sum_{i=1}^m \log(1 - D(G(z_i))), \quad (5.19)$$

where x_i is the real data samples, $D(x_i)$ is the probability that x_i is real according to the discriminator and the other variables as they are in L_G . This shows the punishment and reward the discriminator receives on incorrect and correct predictions respectively. It also shows that the discriminator wants to be able to guess no fake samples as real to minimise its loss [40]. Therefore, together, the generator and discriminator use the minmax loss:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log (1 - D(G(z)))], \quad (5.20)$$

where G is the generator network, D is the discriminator network, $p_{\text{data}}(x)$ is the real data distribution, with the rest as defined in the L_G and L_D [40]. The $\mathbb{E}$ indicates the estimated value across the real data (for $x \sim p_{\text{data}}$) and the fake data (for $z \sim p_z(z)$). In effect, this is taking the average value across the calculation of $\log(D(x))$ and $\log(D(G(z)))$ across a minibatch of samples. Therefore, each time the minmax loss is calculated, the average value across all samples in the minibatch is output.

If the generator successfully fools the discriminator by creating samples that are predicted as real samples, but these samples have low diversity, the generator loss will remain low and the discriminator will struggle to learn from it, a mode collapse [40]. Based on the loss functions L_G and L_D , neither of these have a requirement for any diversity, therefore it is not encouraged. The problem lies in the calculation of the discriminator where each sample is considered individually. No information from one sample is present in the calculation of another sample, consequently, multiple identical samples could be within one batch without the discriminator loss being negatively affected. An improvement for this is called "minibatch discrimination" [84], which takes the distribution of each minibatch sample into account during the discriminator training, thereby adding batch-aware features into the discriminator. $D(x_i)$ and $D(G(z_i))$ will thus depend on other samples within the minibatch. If a non-diverse minibatch is generated with minibatch discrimination, the discriminator will negatively feed back to the generator, effectively forcing it to generate more diversity.

Semi-Supervised GANs (SSGANs) still maintain this same structure as a GAN, with an additional output from the discriminator predicting the class of a given set of features, as discussed in Odena [75]. For images, this could be identifying what is present within the picture and for the data in this study, it would be whether the vehicle will pass or fail. This can be accomplished in various ways; an extra output node that will classify the sample into $K + 1$ classes, where K is the number of classes and the extra one is for when the sample is deemed to be fake. Alternatively, the SSGAN could have two output heads where the discriminator

has two outputs: one predicts if a sample is fake and the other makes a class prediction (i.e. it will always predict a class, even for fake samples). The second SSGAN is more akin to an Auxiliary Classifier GAN (ACGAN), as discussed in Odena et al. [76]. The advantage of using the ACGAN over the traditional style of SSGAN is every sample has a class prediction given to it, even if it is fake, providing additional information about a decision boundary, even if the data point does not exist within real data but is close to reality [76]. The SSGAN in this thesis is an ACGAN, and is referred to as a semi-supervised GAN hereafter.

An important distinction between SSGANs and GANs is the labelling of the data. GANs do not require labelled data at all as they only create new data. SSGANs, on the other hand, can utilise labelled and unlabelled data together [75]. Considering the production data set, this is ideal as only ██████ of the total data is labelled: the Audit vehicles. The earlier methods discussed only utilise this small proportion of the data, potentially missing out on larger scale patterns. The unlabelled data can be used to train the GAN to learn the structure of the data, where the classification does not matter and with far more data, this could result in capturing the structure more accurately. The labelled data is then used for the classification only creating decision boundaries around the known already known structure.

One of the difficulties with GANs and SSGANs, along with other networks, is the difficulty in tuning them effectively. Balancing the generator and discriminator loss against each other is challenging. Perhaps spending more time on developing the SSGAN would have been beneficial and given a stronger result. An inherent benefit of using a SSGAN in the context of this thesis is the relatively low volume of original data. Since only ██████ of the total data is labelled, an accurate SSGAN could label real unlabelled samples, providing extra data for other methods to learn from. Although, any inaccuracies would be problematic, meaning there would have to be thorough validation. Not only that, but a successful GAN could generate plausible rare cases that would otherwise appear very infrequently within the data and plausible data for new models or innovations could allow for problems to be expected and prepared for. Naturally, this would require careful validation, however, the possibility is there.

For the SSGAN used in Chapter 7, a specific network architecture was chosen. The generator is a fully connected feed-forward network with two hidden layers, with dimensions of 512 and

1024 respectively, each with a dropout rate of 0.5 and leaky ReLU activation. The initialisation chosen is the Glorot initialisation, as defined by Equation 5.17. The generated samples are then passed through a feature transformer to ensure the same form as the original data set, improving the stability of the model. The discriminator also had two hidden hidden layers, with dimensions 512 and 256, a dropout rate of 0.1 and minibatch discrimination [84], with the batch size determined by hyperparameter tuning. Additionally, noise is injected into features within the network to reduce overfitting more. Two heads give the outputs, one for real or fake predictions and another for the class prediction. The loss defined here is centre loss that encourages the model to learn well separated classes [102]. The Adam Optimiser is also used for the SSGAN, with a learning rate of 0.002, with the centre loss optimiser using a standard gradient descent optimisation with a learning rate of 0.004.

Proposed Methods

6.1 Introduction

In data science, there is an endless search for robust, powerful and explainable machine learning algorithms. There are many methods that provide solutions to these problems. Random Forest, established by Breiman [20] is a tree ensemble method that produces high performance results on many datasets. Gradient boosting, as originally created by Friedman [36], can use multiple weak learners to create a robust and powerful model. Meanwhile, methods such as Optimal Tree Ensembles, hereafter referred to as Selected Tree Ensembles (STEs), proposed by Khan et al. [54], use only the strong learners to create a less complex ensemble of trees that retains the high performance of a random forest. Additionally, the conditional inference method proposed by Hothorn et al. [47] uses association criteria to identify features used when building a decision tree.

The method proposed in this thesis takes multiple of these approaches and adapts them into a single solution. Combining the statistical tests in the tree growing phase and the selection of strong learners provide the basis of our proposal, P-Value Adjusted Selected Tree Ensemble (PASTE), allowing for an interpretable and rapid prediction once a model has been trained. However, while this is an innovative and powerful approach, more can be done to further improve the results. A method of leveraging residuals and a gradient boosting method to enhance

the results of any model was created and applied to the PASTE model, giving PASTE-boost. Predicting the residuals of the trained algorithm using the original predictors gives a pragmatic method to correct for the mistakes made by it.

Furthermore, it is beneficial to consider how other tree growing methods work alongside the STE framework [54]. Therefore, conditional inference trees were chosen due to their statistical test basis. This results in the creation of the CISTE method (Conditional Inference Selected Tree Ensemble), an STE that uses conditional inference trees in place of the random forest decision trees described in [54].

Additionally, the residual-boosting method can be applied to any machine learning method where confidence levels for each prediction can be calculated. In order to find if the method improves results noticeably, it was applied to CISTE and the original OTE method, giving CISTE-boost and OTE-boost respectively.

6.2 Method

The method used in PASTE uses ideas from a combination of methods from both Lausen [57] and Khan et al. [53]. However, the PASTE method distinguishes itself by combining the two referenced methods. PASTE grows decision trees using statistical tests to determine the selected feature and split point within the data, in place of reducing the Gini impurity. This comes with some potential benefits; reduced overfitting, robustness on imbalanced data and an overall low chance of randomness, due to the statistical tests giving meaningful relationships between the predictors and response variables. Nonetheless, the method follows the same steps after growing the trees.

After the method has been completed, the predicted probabilities can be improved by using a machine learning method to predict the residuals, in this case, the gradient boosting method Catboost was chosen [78].

For both the P-Value Adjusted Tree method and the Conditional Inference Tree method, the same notation can be used. Both methods follow the same steps, however they split the data within each node in different ways. Using the notation as found in [53], the training data is represented by

$$\mathcal{L}_n = (\mathbf{X}, Y) = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}, \quad (6.1)$$

where $\mathbf{X}$ is an $n_{\text{obs}} \times d$ matrix, and Y is a vector of length n_{obs} . For each instance of $\mathbf{x}_i$, d is the number of features, and y_i is the value representing one of $n_{\text{class}} \geq 2$ classes.

Before any trees are grown, the data needs to be separated into bootstrap samples and out-of-bag (OOB) samples. A random selection of observations are included in a bootstrap sample, $\mathcal{L}_{\mathcal{B}} = (\mathbf{X}_B, Y_B)$, and the remaining observations create the OOB sample, $\mathcal{L}_{\mathcal{O}} = (\mathbf{X}_O, Y_O)$. There will be one bootstrap sample for each tree. This is necessary for the selected tree ensemble (STE) creation once the trees have been grown. Without these bootstrap samples, the P-value adjusted trees would be highly correlated, leading to a reduced diversity that is crucial for ensemble performance.

6.2.1 P-Value Adjusted Selected Tree Ensembles (PASTE)

When a dataset contains features on different scales or with differing quantities of potential cutpoints, there is a large potential for biases within model training as shown in other works by Breiman [21]. Features with larger scales and more potential cutpoints are more likely to be selected, specifically when working with decision tree models, therefore the method shown in this thesis uses P-values to account for different scales.

In order to find the optimal cut point, the maximally selected test statistic is found, M_j , as shown below, and defined in Lausen 1992 and Lausen 1994 [58, 57]:

$$M_j(\varepsilon) = \max_{\rho \in [x_j(\varepsilon), x_j(1-\varepsilon)]} T_{j\rho}, \quad (6.2)$$

where ε is the ratio of the minimum subset size to the number of total observations, n , therefore εn is the minimum subset size. When growing a tree, this can be used as a stopping criterion. $x_j(\bullet)$ is the quantile of the factor X_j in the considered subset. $T_{j\rho}$ is the standardised two sample statistic chosen, which compares the subsets on either side of the cutpoint ρ .

In this thesis, the selected test statistic used is a regularised version of the G-test [66], which is defined by:

$$T = 2 \sum_i O_i \ln \left(\frac{O_i}{E_i} \right) \times \exp(-\lambda n_{cutpoints}), \quad (6.3)$$

where O is the observed count, E is the expected count under the null hypothesis, λ is a tunable regularisation factor between 0 and 1, and $n_{cutpoints}$ is the number of possible cutpoints in a given feature. This allows the value of the G-test to be adjusted relative to the number of given cutpoints for a feature, reducing the significance of features with many cutpoints. In order to use the G-test, a contingency table of the response variables from both subsets of a split needs to be created. Including the tunable regularisation factor allows for flexibility in how much penalty is applied depending on the dataset being tested. Since this is being used to test whether the split gains information about the response variable, the null hypothesis is stated as: the proportions of classes within the response variable are the same in both subsets.

In order to measure factors on different scales, as is the original idea behind P-value adjustments, a P-value must be calculated for each maximally selected test statistic, creating a comparable value no matter the scale. Two P-value approximations will be considered (P_1 and P_2), as shown in equations 6.4 and 6.5. P_1 is defined in Miller and Siegmund [69], and is an approximation formula for the P-value derived by the asymptotic distribution, based on b approaching infinity. P_2 is an upper inequality using the asymptotic distribution for the number of observations, n , approaching infinity, as defined in Worsley [104].

$$P_1(\varepsilon) = \frac{4\phi(b)}{b} + \phi(b)\left(b - \frac{1}{b}\right) \ln \left(\frac{\varepsilon_2(1 - \varepsilon_1)}{\varepsilon_1(1 - \varepsilon_2)} \right), \quad (6.4)$$

where $\phi(b) = \frac{\exp\left(\frac{-b^2}{2}\right)}{\sqrt{2\pi}}$, $\varepsilon = \varepsilon_1 = 1 - \varepsilon_2$. b^2 is the observed value of the test statistic.

$$P_2(\varepsilon) = 1 - F_1(b^2) + \sum_{i=1}^{k-1} D(l_i, l_{i+1}), \quad (6.5)$$

where k is the number of splits, l_i is the size of the subset with factor values less than or equal to $\rho_1 < \dots < \rho_k$, where ρ_i is the cutpoint at for the considered split. $D(i, j) = \left(\frac{2}{\pi}\right) f_2(b^2) \left[t_{ij} - \frac{\left(\frac{b^2}{4} - 1\right)(t_{ij})^3}{6} \right]$, $t_{ij} = \sqrt{1 - \frac{i(n-j)}{j(n-i)}}$, $f_2(b^2) = 0.5 \exp\left(\frac{-b^2}{2}\right)$ and F_1 is the chi-squared distribution function.

Once P_1 and P_2 have been calculated, the minimum is selected as the final adjusted P-value to reduce the effect of the conservative nature of P_1 and P_2 individually:

$$P(M_j(\varepsilon \geq b^2) \approx \min(P_1(\varepsilon), P_2(\varepsilon)). \quad (6.6)$$

This is repeated across the bootstrap sample using different split points to find the minimum P-value across the dataset, P^* .

Once the best split point has been found, two branches stem from the root node of the tree about the split point. The process is then repeated until the tree has grown to the fullest extent as defined by the parameters. A maximum depth, $d_{\max}$, can be defined as is common in other tree building methods. Additionally, considering the split points are defined by their P-values, if the best P-value found is not below the significance threshold, α , the branch will not continue since the data on either side of the split point is not notably statistically different.

After the first tree has been grown from one bootstrap sample, many others are grown using other bootstrap samples, creating a set of trees that can be entered into the Selected Tree Ensemble methodology, as described in Section 5.3. In short, the trees are ordered depending on their error, each tree then has its Brier Score calculated. The highest ranked tree is added to an ensemble and then each subsequent tree is compared to the ensemble. If its Brier Score is higher than the ensemble, it is removed, but if the score is lower, the tree is added to the ensemble. This process repeats until all trees have been tested and the P-value selected tree ensemble is created.

In summary, the method follows algorithm 1 followed by algorithm 2:

Algorithm 1: P-Value Adjusted Tree Creation

Data: A collection of pairs $\{(\mathcal{L}_{\mathcal{B}}^{(z)}, \mathcal{L}_{\mathcal{O}}^{(z)})\}_{z=1}^Z$ **Result:** A set of P-Value Adjusted Trees, $\{S_k\}$.**for** $z = 1$ *to* Z **do** Initialise root node with $\mathcal{L}_{\mathcal{B}}^{(z)}$; **while** $P^* < \alpha$ *and* $depth < d_{\max}$ **do**

Select a node to split;

for *each possible split point in that node* **do**

Calculate the test statistic using equation 6.3;

 Calculate P_1 and P_2 using equations 6.4 and 6.5; Let $P = \min(P_1, P_2)$; **end** Select the split point with the minimum P-value, P^* ;

Split the node into two child nodes about the selected split point;

end**end****return** $\{S_k\}$

Algorithm 2: Selected Tree Ensemble

Data: A set of trees $\{S_k\}$ with corresponding $\{\mathcal{L}_B, \mathcal{L}_O\}$

Result: A selected tree ensemble, $\mathcal{E}$

Rank $\{S_k\}$ in order of increasing error as defined by equation 5.3;

Calculate the Brier Score of each tree using corresponding $\mathcal{L}_O$;

Initialise ensemble $\mathcal{E} \leftarrow \{S_1\}$;

Calculate ensemble Brier Score $BS(\mathcal{E})$;

for $k = 2$ **to** K **do**

 Create temporary ensemble $\mathcal{E}' = \mathcal{E} \cup \{S_k\}$;

 Calculate Brier Score $BS(\mathcal{E}')$;

if $BS(\mathcal{E}') < BS(\mathcal{E})$ **then**

$\mathcal{E} \leftarrow \mathcal{E}'$;

end

end

return $\mathcal{E}$;

6.2.2 Conditional Inference Trees

The below method is as laid out in Hothorn et al. [47]. Conditional inference trees also use P-values to determine the feature and split point within each node, however, they are considered consecutively, rather than simultaneously. In order to find P-values, an hypothesis needs to be specified, in this case, a global hypothesis [47]. There are two main steps to be repeated in order to generate conditional inference trees:

1. Test if the global hypothesis can be rejected. If it cannot be rejected, the tree has been fully grown, and the recursion can stop. Otherwise, select the feature, X_{j^*} with the strongest effect on the response variable $\mathbf{Y}$
2. Choose a subset of the data, A^* , to split the dataset into two. This choice is determined by a further statistical test.

The null hypothesis, H_0 , states that there is no dependence between any of the individual

features, X_j , and the response variable, simultaneously for all m partial hypotheses:

$$H_0 = \bigcap_{j=1}^m H_0^j, \quad (6.7)$$

where $H_0^j : D(\mathbf{Y}|X_j) = D(\mathbf{Y})$. H_0^j states that the distribution of the response variable $\mathbf{Y}$ given a feature X_j is identical to the distribution of $\mathbf{Y}$, i.e. X_j has no effect on the $\mathbf{Y}$.

Feature Selection

In order to test the null hypothesis, a statistical test needs to be completed for all values of $\mathbf{X}_j$. Non-integer value case weights need to be specified in order to define a generic algorithm $\mathbf{w} = (w_1, \dots, w_n)$. In each node of the tree, each observation from $\mathcal{L}_{\mathcal{B}}$ has a corresponding w . If the observation is an element of the node, $w > 0$, if not $w = 0$. A linear test statistic, $\mathbf{T}_j$, can be used to measure the relationship between $\mathbf{Y}$ and X_j for all $j = 1, \dots, d$:

$$\begin{aligned} \mathbf{T}_j(\mathcal{L}_{\mathcal{B}}, \mathbf{w}) = \text{vec} \left(\sum_{i=1}^n w_i g_j(X_{ji}) \right. \\ \left. \times h(\mathbf{Y}_i, (\mathbf{Y}_1, \dots, \mathbf{Y}_n))^T \right) \in \mathbb{R}^{p_j q}, \end{aligned} \quad (6.8)$$

where g_j is a non-random transformation of feature X_j into $\mathbb{R}^{p_j}$ space. p_j is the dimension space of the transformation of X_j . h is an influence function that is dependent on the response variables, and is permutation symmetric, meaning the order of the responses is irrelevant. It maps a response variable $\mathbf{Y}_i$ and the full set of response variables $(\mathbf{Y}_1, \dots, \mathbf{Y}_n)$ to $\mathbb{R}^q$ space. Hothorn et al. further describes g_j and h [47]. As shown in Equation 6.8, $\mathbf{T}_j$ is dependent on both X_j and $\mathbf{Y}$, but this dependency is not known in most cases. Since the null hypothesis states there is no dependence, the features can be fixed and the response variables are permuted. All possible permutations are considered and compared to the original data in order to find if an association is present between X_j and $\mathbf{Y}$, which constitute permutation tests. Both the conditional expectation μ_j and covariance Σ_j of $\mathbf{T}_j$ under the null hypothesis and all permutations can then be found, as

stated in Hothorn et al. [47] and derived in Strasser 1999 [91]. These can be used to create a univariate test statistic, c , that maps a standardised linear statistic from $\mathbf{T} \in \mathbb{R}^{pq}$ to $\mathbf{t} \in \mathbb{R}^{pq}$ into a single real number. In this case, c takes the form:

$$c_{\max}(\mathbf{t}, \mu, \Sigma) = \max_{k=1, \dots, pq} \left| \frac{(\mathbf{t} - \mu)_k}{\sqrt{(\Sigma)_{kk}}} \right|, \quad (6.9)$$

where Σ is the covariance matrix. However, multiple c values cannot be fairly compared unless their covariances are on the same scale. This scaling issue can be circumvented by using the corresponding P-values. In order to select the most statistically significant feature, the feature X_{j^*} , with the lowest P-value is selected, where the P-value, P_j , for each feature is defined as:

$$\begin{aligned} P_j &= \mathbb{P}_{H_0^j}(c(\mathbf{T}_j(\mathcal{L}_{\mathcal{B}}, \mathbf{w}), \mu_j, \Sigma_j) \\ &\geq c(\mathbf{t}_j, \mu_j, \Sigma_j) | S(\mathcal{L}_{\mathcal{B}}, \mathbf{w})), \end{aligned} \quad (6.10)$$

where $S(\mathcal{L}_{\mathcal{B}}, \mathbf{w})$ is the symmetric group of every permutation over all elements $(1, \dots, n)$ where the case weights are $w_i = 1$. An important practical consideration is when d , the number of features, is large. Other methods often test a default number of features at each node, either $\sqrt{d}$ or d [61, 47]. Growing a large set of trees while using recursive partitioning on large d will be computationally intensive, leading to a long training time. This is true for all partial hypotheses and will give an unbiased variable selection, but the global hypothesis needs to be constructed. There are multiple methods to complete this, however the chosen method is to use the Bonferroni correction. Every P_j is adjusted according to how many statistical tests have been completed, m , through the simple equation:

$$P_{\text{adj},j} = P_j m. \quad (6.11)$$

If the P -value is used as a stop criterion and none of the adjusted P-values meet the specified confidence level, α , the algorithm halts. If not, X_{j^*} is selected as the feature to split the data on.

The Bonferroni correction provides a conservative estimate for the adjusted P-values, leading to fewer false positives, but potentially more false-negatives. This can be useful for anomaly detection, and similar problems where the false-positive rate should be minimised.

Binary Split Point Selection

In order to get two subsets of data, two case weight vectors are defined; $\mathbf{w}_{\text{left}}$ and $\mathbf{w}_{\text{right}}$ where $w_{\text{left},i} = w_i I(X_{j^*i} \in A)$ and $w_{\text{right},i} = w_i I(X_{j^*i} \notin A)$ where I is the indicator function, and A is a subset of data. This splits the data into instances within A and not. Once again, the permutation method described above is used to find the optimal split point. The quality of the split is then measured using a two-sample statistic, similar to Equation 6.8, as shown below:

$$\begin{aligned} \mathbf{T}_{j^*}^A(\mathcal{L}_B, \mathbf{w}) = \text{vec} \left(\sum_{i=1}^n w_i I(X_{j^*i} \in A) \right. \\ \left. \times h(\mathbf{Y}_i, (\mathbf{Y}_1, \dots, \mathbf{Y}_n))^T \right) \in \mathbb{R}^{p^j q}, \end{aligned} \quad (6.12)$$

for p response variables and q values for predictor X_j . Identically to before, the conditional expectation $\mu_{j^*}^A$ and covariance $\Sigma_{j^*}^A$ are computed, allowing the optimal split, A^* , to be found via maximising the test statistic:

$$A^* = \underset{A}{\operatorname{argmax}} c(\mathbf{t}_{j^*}^A, \mu_{j^*}^A, \Sigma_{j^*}^A). \quad (6.13)$$

Both steps are repeated until a stopping criterion is met. Within this study, the stopping criteria were when either $P_{j^*} > \alpha$ or the predefined maximum depth of a tree was reached.

Once again, after many trees are grown, the selected tree ensemble methodology begins, as discussed in Section 5.3, resulting in the final conditional inference selected tree ensemble.

In summary, the method is defined as below in algorithm 3, once again followed by algorithm 2:

Algorithm 3: Conditional Inference Tree Creation

Data: A collection of pairs $\{(\mathcal{L}_{\mathcal{B}}^{(z)}, \mathcal{L}_{\mathcal{O}}^{(z)})\}_{z=1}^Z$

Result: A set of conditional inference trees $\{S_k\}$

for $z = 1$ *to* Z **do**

 Initialise root node with $\mathcal{L}_{\mathcal{B}}^{(z)}$;

while $P^* < \alpha$ *and* $depth < d_{\max}$ **do**

for each feature, X_j *in the node* **do**

 Calculate linear statistic using equation 6.8;

 Calculate permutation-based P-value P_j for feature X_j using equation 6.10;

 Adjust P_j using Bonferroni correction, as found in equation 6.11;

end

 Select feature X_{j^*} with smallest $P_{\text{adj},j}$;

if $P_{\text{adj},j^*} > \alpha$ *or* $node\ depth = d_{\max}$ **then**

 Mark node as terminal;

continue to next node;

end

 For candidate splits A of X_{j^*} : Compute two-sample statistic $\mathbf{T}_{j^*}^A$ using equation 6.12;

 Calculate optimal split using equation 6.13;

 Split node into left and right children about the optimal split point;

end

end

return $\{S_k\}$

6.2.3 Residual Boosting

Using the predicted probability for each instance, $\hat{\mathbf{P}}_i$, a residual for each instance and class can be found using the following formula:

$$r(C_k|\mathbf{x}_i) = I(y_i = C_k) - \hat{P}(C_k|\mathbf{x}_i), \quad (6.14)$$

where y_i is the observed response value for that instance and $I(y_i = C_k)$ is the indicator function:

$$I(y_i = C_k) = \begin{cases} 1 & \text{if } y_i = C_k \\ 0 & \text{otherwise} \end{cases} \quad (6.15)$$

In other words, since this is a classification problem, the actual probabilities of the observed values are inherently binary, therefore, each residual will be a continuous value in the range $[-1, 1]$. Subsequently, a residual matrix, $\mathbf{R}$, is found, containing the residuals for every class, $1 \dots n_{\text{class}}$, and instance, $1 \dots n$;

$$\mathbf{R} = \begin{bmatrix} r(C_1|\mathbf{x}_1) & r(C_2|\mathbf{x}_1) & \cdots & r(C_{n_{\text{class}}}|\mathbf{x}_1) \\ r(C_1|\mathbf{x}_2) & r(C_2|\mathbf{x}_2) & \cdots & r(C_{n_{\text{class}}}|\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ r(C_1|\mathbf{x}_n) & r(C_2|\mathbf{x}_n) & \cdots & r(C_{n_{\text{class}}}|\mathbf{x}_n) \end{bmatrix}. \quad (6.16)$$

In any classification problem, each response class will have its own residual for each observation. In a binary classification problem, $n_{\text{class}} = 2$, the residual of one class would be entirely dependent on the other. However, for a multi-class classification problem, $n_{\text{class}} > 2$, there is no longer such a relationship. Therefore, for such problems, each class will have its own independent residual field. This independence necessitates the creation of $\mathbf{R}$, containing the full set of residuals. This study focuses on binary classification problems, however the boosting method is suitable for any number of classes.

To set up the problem to improve the results of the STE, each column in $\mathbf{R}$ are used as the response variables for a gradient boosting algorithm, using the same predictors as before, giving a new dataset:

$$\mathcal{N} = (\mathbf{X}, \mathbf{R}) = (\mathbf{x}_i, \mathbf{r}_i) \mid (\mathbf{x}_i, y_i) \in \mathcal{T}, \quad (6.17)$$

where $\mathbf{r}_i$ is a vector within $\mathbf{R}$ for one instance. This allows the problem to be set up as a multi-target regression problem.

Since there are many powerful gradient boosting algorithms available, it was logical to use one of these to boost the $\mathcal{N}$ rather than creating a new variation. Therefore the method as described by Prokhorenkova et al. [78] is followed, using $\mathbf{R}$ as the response variables. The CatBoost algorithm was chosen as it can deal with categorical data seamlessly, allowing for a fast and powerful implementation. This gradient boosting method allows for multi-target regression, therefore, for each instance predictions for the residual of each class can be made simultaneously, giving a matrix of predicted residuals, $\hat{\mathbf{R}}$:

$$\hat{\mathbf{R}} = \begin{bmatrix} \hat{r}(C_1|\mathbf{x}_1) & \hat{r}(C_2|\mathbf{x}_1) & \cdots & \hat{r}(C_{n_{\text{class}}}|\mathbf{x}_1) \\ \hat{r}(C_1|\mathbf{x}_2) & \hat{r}(C_2|\mathbf{x}_2) & \cdots & \hat{r}(C_{n_{\text{class}}}|\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{r}(C_1|\mathbf{x}_n) & \hat{r}(C_2|\mathbf{x}_n) & \cdots & \hat{r}(C_{n_{\text{class}}}|\mathbf{x}_n) \end{bmatrix}, \quad (6.18)$$

where $\hat{r}(C_k|\mathbf{x}_i)$ is the predicted residual for the class C_k on instance $\mathbf{x}_i$. Since the original residuals can be considered as a correction to the predicted probability, as shown in Equation 6.14, the new residuals can be considered as corrections. The boosting model now contains the information about the correction, meaning it can be used to adjust predictions made on new data, using an adapted version of Equation 6.14. This can be improved upon to remove residuals that are likely to be exaggerated. For many machine learning algorithms, a confidence level, $Conf$, can be calculated for every prediction, this gives direct feedback as to how likely it is that a prediction is correct without test data. This can be calculated in a few different ways, but specifically for forest algorithms it is calculated by taking the ratio of trees that agree on the predicted class. $Conf$

will always be larger than $\frac{1}{n_{class}}$, therefore it is scaled down to range between 0 and 1. $Conf$ can then be used as a scaling factor for the residuals, as shown in the following equation, removing the need to adjust predictions that are predicted with 100% confidence, and making the largest adjustments to the values closest to $\frac{1}{n_{class}}$, the baseline probability:

$$\hat{\mathcal{P}}(C_k|\mathbf{x}_i) = Conf \times \hat{R}(C_k|\mathbf{x}_i) + \hat{P}(C_k|\mathbf{x}_i), \quad (6.19)$$

where $\hat{\mathcal{P}}(C_k|\mathbf{x}_i)$ represents the adjusted probability for every individual element of $\hat{\mathbf{P}}$. Additionally, this provides a regularisation effect where the corrected probabilities do not exceed the bounds of $[0, 1]$, providing a suitable probability. The predicted class is once again chosen by $\hat{y}_i = \arg \max_k \hat{\mathcal{P}}_i(C_k|x_i)$.

Additionally, the boosting implemented within PASTE-boost can be used from the results of any algorithm, provided the predicted probabilities can be returned from the model and that the confidence can be calculated. In Section 6.3, the results of this being applied to both the CISTE and OTE-boost method, mentioned in Section 6.1, are discussed.

6.3 Testing & Results

The performances of PASTE, PASTE-boost, CISTE and CISTE-boost were compared to a range of other established and powerful algorithms: Random Forest, XGBoost and Optimal Tree Ensemble [20, 24, 53]. These algorithms are good benchmarks to compare against since PASTE-boost contains similar methods to these, and importantly, are considered to be very good predictors. Additionally, the aforementioned OTE-boost was included in order to gain an understanding of how the gradient boosting of residuals can perform on a previously established algorithm.

In order to split the data into train and test datasets, a 70%/15%/15% (training/validation/testing) stratified split was completed to guarantee low frequency classes would appear in both subsets. Each algorithm was tested on every dataset, showing how they perform with a range of input properties. They then underwent 10-fold hyperparameter tuning for every dataset using the validation dataset, with the Bayesian Optimisation process defined by Joy et. al [49], to maximise

the performance. Bayesian Optimisation was selected due to its efficiency and effectiveness in finding the best possible parameters within a given range. This optimisation was carried out of 15 iterations with 10 initial points. Other methods, such as grid search, were tested, however these methods were more computationally expensive and can not explore values outside of a pre-specified grid. For a method such as PASTE, that will take a while to train considering the recursive statistical tests, efficiency of hyperparameter tuning is very important. The hyperparameter search space is defined in Table 6.1. After the hyperparameter tuning was complete, the models were tested over the 10-folds of test data corresponding to each learning set, minimising any sampling biases, and values for accuracy, precision, recall, F1 Score and Cohen's Kappa [28] were found. The datasets are described in Table 6.2.

Method	Hyperparameter	Range
Random Forest	ntree	100 to 500
	mtry	1 to $\lfloor \sqrt{d} \rfloor$
	nodesize	1 to 50
XGBoost	nrounds	100 to 500
	eta	0.01 to 0.10
	max_depth	3 to 9
	subsample	0.5 to 1.0
	colsample_bytree	0.5 to 1.0
	min_child_weight	1 to 5
OTE/OTE_boost	t.initial	100 to 500
	p	0.1 to 0.8
	tree_no	100 to 500
	mtry	1 to $\lfloor \sqrt{d} \rfloor$
	cb_iterations	100 to 2000
COTE/COTE_boost	tree_no	5 to 15
	ct_mtry	1 to 2
	cb_iterations	50 to 2000
PASTE/PASTE_boost	tree_no	5 to 20
	max_depth	3 to 20
	cb_iterations	50 to 2000
	lambda	0.00 to 0.99

Table 6.1: Hyperparameter search spaces used for tuning the machine learning methods, using the R implementation of each of the methods. Here d denotes the number of predictor variables.

Data Set	n	d	Feature Type ($R/I/C$)	Class Distribution	Source
Mammography	569	30	30/0/0	357:212	[10]
Monk 3	554	6	0/0/6	288:266	[10]
Parkinson's	754	754	754/4/0	654:192	[10]
SA Heart	462	9	5/3/1	302:160	[4]
Tic-Tac-Toe	958	9	0/0/9	626:332	[10]
Titanic	891	5	2/0/3	549:342	[6]
Voting	435	16	0/0/16	267:168	[10]

Table 6.2: Description of classification datasets used in the tests, showing the number of instances n , number of features d , feature type (R : Real, I : Integer, C : Categorical), class distribution, classification type, and source of each dataset.

The values shown below in Figures 6.1, 6.3 and 6.2 are the accuracy, kappa and F1 scores for each algorithm across the datasets. For clarity, the accuracy for classification is defined as:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \quad (6.20)$$

$$= \frac{TP + TN}{TP + TN + FP + FN}, \quad (6.21)$$

where TP is true positive, TN is true negative, FP is false positive and FN is false negative. This can be a good performance metric for balanced datasets, however, unbalanced datasets could provide a higher than expected value. A specific issue that could arise from unbalanced datasets is when the trained model only predicts the majority value. This would give an accuracy $> 50\%$ for a binary classification, but there would be no information within the prediction. Other metrics are used in combination with accuracy to give a far greater view of the performance.

The F1 score is a commonly used measure that balances both the precision and recall of a model, which are defined as:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (6.22)$$

and

$$\text{Recall} = \frac{TP}{TP + FN}. \quad (6.23)$$

Precision is the ratio of true positive values to the total number of predicted positives, and recall is the ratio of true positive values to the number of observations in the class. For both precision and recall, a macro-average is found by finding the average precision and recall for every class. This value is used to find the F1 score:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (6.24)$$

This gives a good balance between both the precision and recall, which tackles class imbalance scenarios as well as balanced scenarios.

The kappa value is defined as defined by Cohen [28], which states that:

$$\kappa = \frac{p_o - p_e}{1 - p_e}. \quad (6.25)$$

p_o is the “relative observed agreement among raters”, i.e. how many times the prediction agrees with the true values. p_e is the “hypothetical probability of chance agreement”, i.e. the probability that the prediction and true value agree purely by chance. For binary classifications, these can be defined as:

$$p_o = \text{Accuracy}, \quad (6.26)$$

as defined earlier, and

$$p_e = \frac{(TP + FP)(TP + FN)}{(TP + TN + FP + FN)^2} + \frac{(FN + TN)(FP + TN)}{(TP + TN + FP + FN)^2}.$$

The kappa value is therefore a measure of how well a model performs when chance is removed, removing any issues caused by the imbalance of classes. A kappa value of 1 means that the model is predicting the data 100% correctly without random chance. A value of 0 means the performance of the model is as likely due to random chance and learning from the data, i.e. the model is not sufficiently learning. A negative kappa value indicates that the model is performing worse than random chance.

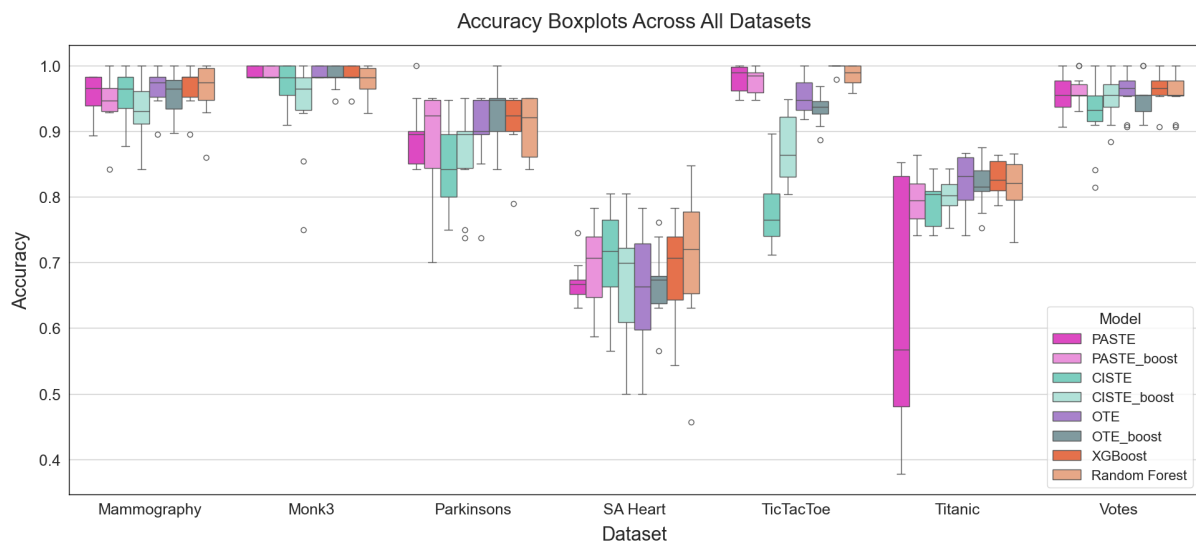


Figure 6.1: Boxplots for the accuracy over 100 with-replacement bootstrap sample predictions for the 8 tested algorithms (CISTE, CISTE-Boost, OTE-Boost, OTE, PASTE, PASTE-Boost, Random Forest, and XGBoost) across the 7 datasets.

Starting with CISTE and CISTE-boost, it can be seen from Figures 6.1, 6.2 and 6.3, that they perform similarly to the established methods (OTE, XGBoost and Random Forest) overall. However, they appear to struggle on the categorical-only dataset, TicTacToe, with an accuracy of 75%, compared to Random Forest with 98%, but performed well on the Voting dataset, the other categorical-only dataset, with an accuracy of 93%, compared to XGBoost with 95%. CISTE-boost performs worse than CISTE on three datasets; Mammography, Monk and SA

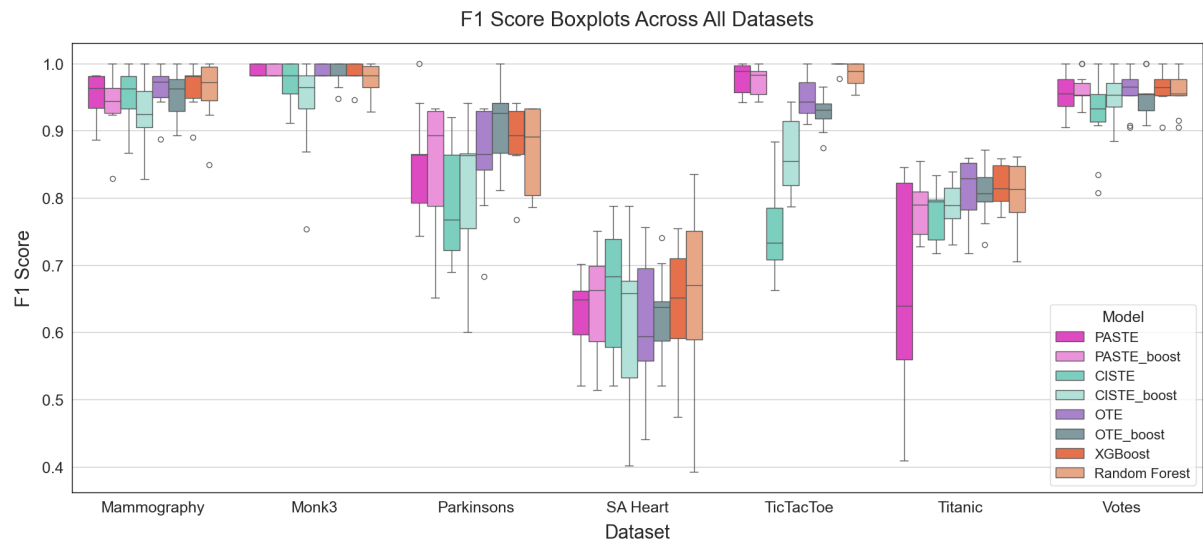


Figure 6.2: Boxplots for F1 score over 100 with-replacement bootstrap sample predictions for the 8 tested algorithms and across all 7 datasets.

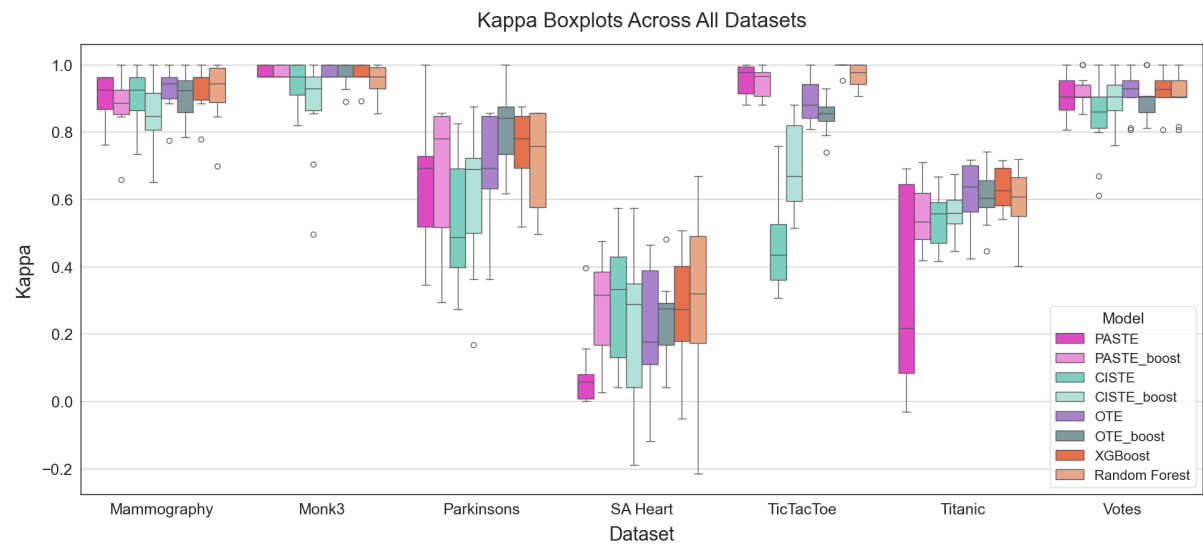


Figure 6.3: Boxplots for the kappa value over 100 with-replacement bootstrap sample predictions for the 8 tested algorithms and across all 7 datasets.

Heart, outperforming CISTE on Parkinsons, TicTacToe and Voting marginally, with a similar median but smaller variability on Titanic. The largest improvement for the boosting method is on TicTacToe, with a 10% improvement in accuracy, from 75% to 85%, and 20% improvement in kappa from 45% to 65%.

PASTE performs comparatively well on five datasets, and poorly on two; SA Heart and Titanic. For SA Heart, the accuracy is comparable with other poorly performing methods, with an accuracy around 65%, however, its Kappa value is significantly worse at 10%, whereas the next lowest is for OTE at 20%. Its F1 score median is approximately equal to the other methods. For Titanic, it is significantly worse than all methods, with an accuracy and F1 score 20% lower than the second lowest, a score of 55%, and a kappa score 35% lower, at only 20%. Its score variance is also enormous, with an inter-quartile range of 30, 35 and 60% for accuracy, F1 score and kappa.

PASTE-boost performs equally to PASTE on four of the datasets, and outperforms it on the other three, Parkinsons, SA Heart and Titanic. For the Parkinsons and SA Heart datasets, PASTE-boost performs around 5% better in both accuracy and F1 score, and 10% in kappa. For the Titanic dataset, it outperforms PASTE by 20, 15 and 30% for accuracy, F1 score and kappa, and has significantly less variability. Generally, PASTE-boost performs equally to the established methods, and only performs notably worse on the Titanic dataset by 5% on accuracy and F1 score.

Additionally, OTE-boost performs similarly to OTE on all datasets. Performing marginally better on the Parkinsons and SA Heart datasets, and marginally worse on TicTacToe and Titanic.

6.4 Discussion

From the tests, the residual boosting method on the PASTE algorithm appears to benefit performance occasionally. In two instances, on the SA Heart and Titanic dataset, there are large improvements between the original method and its corresponding ‘boost’ variant. With the additional residual predictor, the residual prediction model is given more information than the original, providing additional insights and targeting the errors themselves. Importantly, predicting the residuals on a separately trained model creates an ensemble consisting of the original

model and the residual model. Both of these models combined together can individually provide insight that the other model missed, resulting in increased performance. Promisingly, there were no significant performance decreases across all datasets, showing that it is worth using since there is a chance of higher performance, without a chance of a significant decrease compared to PASTE. Specifically for the low performance datasets, PASTE appears to be an underfitted model, and the residual boosting has corrected for this. Therefore, it can be stated that residual boosting improves the performance of PASTE predictions below 90% accuracy by up to 5%. If a prediction is above 90% accurate, additional boosting will not improve the performance due to diminishing returns.

For PASTE-boost, across all datasets and metrics, it performs similarly to Random Forest and XGBoost. Although, it does not outperform these methods for any dataset. One area that some algorithms sometimes struggle with is predicting the minority classes. Without techniques to specifically account for these biases, methods will sometimes only predict the majority class, typically in order to achieve a high accuracy. PASTE and PASTE-boost, however, always predict the minority classes straight out of the box. While these predictions may not always be correct, the ability to predict the minority class is still an important aspect of the algorithms.

For CISTE, the performance gain from residual boosting still appears beneficial for lower accuracy models. Similarly to the effect from boosting the PASTE method, residual boosting has various effects depending on the base model performance. For both datasets where the base original model has an accuracy $> 95\%$, boosting will not improve the result, because any extra complexity added will likely lead to overfitting. For datasets with lower predicted accuracy, improvements can be made, but not as consistently as with the PASTE method. This implies that while the performance may be lower on certain datasets, the model has not underfitted and has reached its maximum ability with the current setup.

The performance of OTE-boost does not vary much from OTE. OTE is a powerful method which fits well to the data, therefore adding residual boosting does not give any benefit and will likely cause overfitting issues.

6.5 Conclusions about new methods

Overall, PASTE and PASTE-boost appear to be very capable methods of prediction. They work well on small datasets, generally matching or beating the performance of the original OTE method that it was based on. It would be beneficial to find alternative ways to use PASTE with larger datasets by increasing the efficiency of the method. This could be done by reducing the number of features considered at each node, similarly to random forest or parallelisation of the tree building method. Using conditional inference trees in the OTE framework appears to provide no benefit as the performance is worse, and takes longer to train. As with PASTE, further investigations into reducing training time for this would provide further information about the capability of CISTE. Residual boosting appears to improve results for low performance models, but not for high performance models. Generally, if the accuracy of a PASTE or OTE model is $< 90\%$, residual boosting should be used in order to improve the performance. For CISTE, the benefits are less clear and residual boosting should not be used in this case.

The results show that the proposed methods provide advantages over conventional tree-based methods in specific scenarios rather than with general performance improvements. Residual boosting in particular provides a method to improve models that underfit the data. This is apparent with PASTE, where the small selected tree ensemble could result in residuals containing informative structure, allowing for a learnable residual prediction. This advantage is shown most evidently on the SA Heart and Titanic datasets. Additionally, residual boosting does not notably reduce the performance compared to the no-boost variant.

For further research into this topic, an improvement to the boosting mechanism for PASTE results could provide some even higher performance results. Alternatively, using boosting within the STE framework to give each tree in the ensemble a weight value rather than removing trees that do not provide obvious benefits could provide another variation for an ensemble of trees. A further, natural improvement on this method is to adapt it to work with regression problems. Beyond this, optimising the PASTE algorithm to gain insights into larger datasets is the main improvement to be made.

Final Methodological Pipeline

7.1 Introduction

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED] Therefore, the premise of this chapter looks to explore a range of machine learning methods that can utilise the extent of these variables

to predict and give insight into what may or may not cause vehicles to fail their Audit inspection. By finding the best method, the production can benefit the most.

7.2 Data

Each station has a number of features based on factors experts believed would be influential. The specifics are shown in Chapter 4, showing how each feature was engineered and its format.

This results in 808 number of features.

These features were tested on one vehicle model, with information from 20 months, providing observations,

The response variable is a binary value,

The classes are not balanced with of the vehicle passing and failing. The company defined threshold for the binarisation of the final inspection score is at , however, this may not be the most predictive point for the threshold. Before finding which methods are most successful, the most optimal threshold was found. A 5-fold cross validation prediction using the Random Forest method was used to test various thresholds, with no feature selection [20] . The thresholds tested were . The receiver operating characteristic area under curve (ROC-AUC) was calculated for each of these thresholds and the highest value was selected as the final threshold [18] . The results from the tests can be seen in Figure 7.1. The figure shows that the best split points are . has the highest mean score at 0.65 but has the largest variability too. has a lower mean, at 0.61, and a slightly larger difference with the baseline score indicating it performs better against random chance than a threshold of . However, has a mean only 0.025 lower than that of a threshold of . Most importantly, it has the lowest variance, and since it is only marginally lower than the maximum score of all thresholds, it is

logical to continue using the company defined threshold. Therefore, the selected threshold for the analysis is $\mathbf{0.5}$.

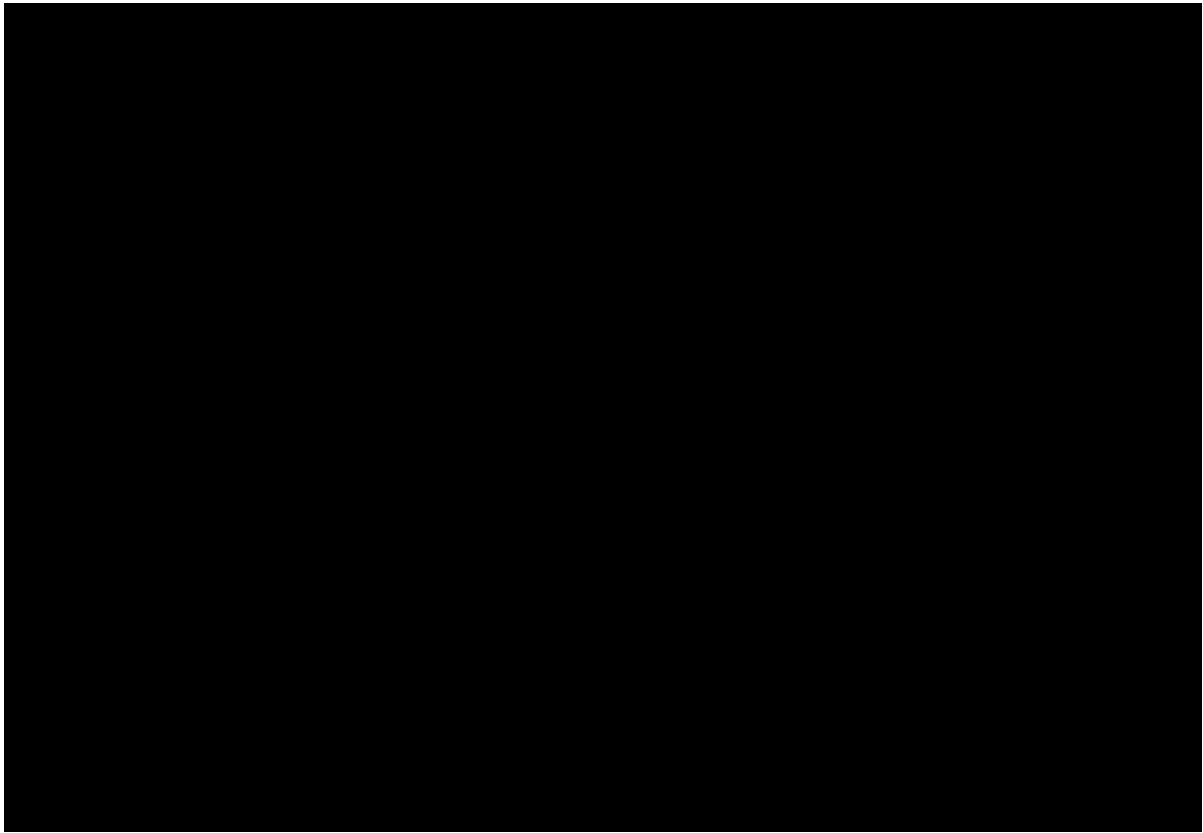


Figure 7.1: The ROC-AUC score across 5 split point thresholds; $\mathbf{0.5}$. These results are across a 5-fold cross validation with no feature selection. The baseline ROC-AUC score is shown in pink, representing the ROC-AUC score if the trained model made random predictions. The dot represents the mean value across the 5 folds. The whiskers represent the range of values from all 5 folds.

7.3 Methodological Definition

In order to test which method has the best performance for the data, all methods will be compared across various metrics: accuracy, recall, precision, F1 score, Cohen's Kappa and the ROC-AUC [28]. An incorrect prediction on a passing vehicle (a false positive) is costly as it will result in wasting time and potentially causing more issues. Whereas an incorrect prediction on a failing vehicle (a false negative) would still result in a failure, but would be no worse than no prediction. Therefore, the cost of a false positive is higher than that of a false negative. Consequently, optimising precision was most beneficial for this pipeline. Any vehicles predicted to fail are

more likely to be genuine failures. Although the failures constitute the minority class, it was deemed more valuable to be able to minimise false positives.

The machine learning methods used for this analysis are a selection of simple methods, tree based methods and neural network methods, due to their popularity, accessibility and ability. These are Random Forest, XGBoost, Light Gradient Boosting Machine (LightGBM), Logistic Regression, a neural network with backpropagation, a semi-supervised Generative Adversarial Network (SSGAN), Optimal Tree Ensemble (OTE), P-Value Adjusted Selected Tree Ensemble (PASTE), and a residual boosted version of the two latter methods: OTE-boost and PASTE-boost [20, 24, 52, 94, 65, 75, 54]. These have been discussed in more detail in Chapters 5 and 6.

To gain an insight into how the methods compared, there were two sets of tests; with no feature selection and with feature selection. Using no feature selection will give a good baseline to compare the feature selection results to. To select the features 10 hyperparameter tuned random forests are used. For each of these forests, the feature importance was found. Across the 10 models, the average feature importance was calculated and any feature with an importance value greater than 0.01 was selected. This value was chosen as it is a balance between having features with a large influence, but not having too few features, removing some of the detail. This provides a selection of features that have consistently had some influence on the model, reducing the number of features from 251 to just 25. [REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]. This selection of 25 features are the same features used for all of the tested methods.

Once the feature selection has taken place, the model testing begins. Here each tested method undergoes a 10-fold cross validation test with 15-fold hyperparameter tuning nested within. Each fold was transformed using a quantile transformer, as defined in the Sci-Kit Learn documentation [23], in order to reduce the effect of outliers and spread out more closely clustered values. Each of the results are recorded and the average and variance are shown as the output. For clarity, each of the methods used the precision as the performance metric for tuning. The hyperparameter search was a random grid search over a range of parameters in order to maintain

simplicity and efficiency. Notably, the maximum features considered at each is kept constant at $\sqrt{d}$, considerations for improving upon this are discussed in Section 8.2. For each model, a selection of hyperparameters were chosen to tune, as shown in Table 7.1. In addition to the tests with feature selection, the same tests will be completed on the original data, in order to see if the feature selection is worthwhile, and how much of an improvement it is.

Table 7.1: Hyperparameters and their ranges for each of the tested machine learning methods.

Method	Hyperparameter	Range
Random Forest	n_estimators	10 to 300
	max_depth	1 to 30
	min_samples_split	2 to 50
	min_samples_leaf	1 to 50
	class_weight	1:1 to 1:3
	max_features	$\sqrt{d}$
XGBoost	n_estimators	10 to 100
	max_depth	1 to 30
	learning_rate	0.01 to 0.2
	subsample	0.6 to 1.0
	colsample_bytree	0.6 to 1.0
	min_child_weight	1 to 10
LightGBM	n_estimators	10 to 100
	max_depth	1 to 30
	learning_rate	0.01 to 0.2
	subsample	0.6 to 1.0
	colsample_bytree	0.6 to 1.0
	min_child_weight	1 to 10
Logistic Regression	C	0.01 to 10
	penalty	l2
	solver	liblinear
	class_weight	1:1 to 1:3
Neural Network	dropout_rate	0.1 to 0.5
	learning_rate	0.0001 to 0.01
	epochs	50 to 200
SSGAN	lambda_fm	1.0 to 5.0
	lambda_entropy	0.5 to 2.0
	lambda_center	0.5 to 2.0
	batch_size	20 to 100
	num_epochs	100 to 500
OTE/OTE_boost	tree_no	10 to 20
	learning_rate	0.05 to 0.2
	depth	3 to 7
	catboost_iterations	500 to 2000
OTE_boost	catboost_iterations	500 to 2000
PASTE/PASTE_boost	tree_no	10 to 20
	learning_rate	0.05 to 0.2
	depth	3 to 7
	catboost_iterations	500 to 2000
PASTE_boost	catboost_iterations	500 to 2000

7.3.1 Extreme Thresholds

Furthermore, originally predicting around the threshold proved to be a very difficult challenge. The values surrounding the threshold, specifically the scores of [REDACTED], are very similar, making them difficult for a machine learning method to separate. Therefore, to test if the data is at all separable or predictable, the messy data near the threshold was removed, leaving only extreme values. A random forest with no feature selection was then trained across 10 folds using a range of threshold pairs, a selection of combinations between [REDACTED]. The precision, recall and F1 score were calculated for each of the fold. The mean and standard deviation of each of these tests are shown in Table 7.2. The selection of thresholds was made by analysing the different cut points in the data on or below the mean. The number of data points below the cut point was found and matched up with a cut point with a similar number of points on the other side of the mean, for example [REDACTED]

[REDACTED] This gives a range of threshold pairs which approximately have an even split of data on either side.

Threshold Pair	Precision	Recall	F1 Score	Total Positive Classes
[REDACTED]	0.900 ± 0.122	0.933 ± 0.133	0.903 ± 0.082	15
[REDACTED]	0.917 ± 0.105	0.780 ± 0.129	0.831 ± 0.068	23
[REDACTED]	0.674 ± 0.166	0.533 ± 0.097	0.593 ± 0.115	56
[REDACTED]	0.662 ± 0.117	0.320 ± 0.101	0.426 ± 0.110	56
[REDACTED]	0.524 ± 0.130	0.197 ± 0.056	0.279 ± 0.061	76

Table 7.2: Results for Random Forest tests with various thresholds of the data. The precision, recall and F1 score are calculated, and the total number of positive classes (failures in inspection) are shown. [REDACTED]

The results from the extreme threshold test show that there is some separability and predictability within the data, but the values near the mean are noisy. The precision is, once again, the important metric since it is important that any prediction of the failure class is correct. Simply removing the values of [REDACTED] does not provide a high precision or recall unfortunately.

However, the extreme values are very well predicted. The thresholds at [REDACTED] give a high recall, precision, and F1 score. An important feature to look for in the final selected model is the ability to still accurately predict these extreme values correctly, as they can be predicted well on their own. An alternative approach to consider is whether a model trained on the [REDACTED] threshold can have a high precision when given any value, including all values in between. This method was tested and proved to be very ineffective at predicting any values outside of the extremes. This was expected since the model was trained on the well separated data, and not the data near the threshold, meaning it would have no information about the decision boundary for more difficult situations. Another feature of this analysis is the fact that the worst performing threshold from Table 7.2 is undoubtedly the company defined threshold of [REDACTED], which is also the most unbalanced set of data. But as seen in Figure 7.1, this is still the highest performance on across thresholds using all of the data.

7.4 Results

The results of the tests on every method are shown below, first showing the tests without feature selection and then the tests with feature selection. Once again, precision of the failure class is the most relevant value minimising the chance of an incorrect positive prediction.

7.4.1 No Feature Selection Results

As seen in Figure 7.2, the most successful methods without feature selection are the gradient boosting machines of XGBoost and LightGBM, which are two methods to have a median at least 50%, at 0.50 ± 0.278 , and 0.50 ± 0.358 meaning they make the same number of correct positive predictions than incorrect positive predictions. The semi-supervised GAN performed well too, with one of the smallest variances, 0.50 ± 0.129 , making it the most consistently performing method. Logistic Regression, PASTE and PASTE-boost all performed extremely poorly with a median of 0.00. With all features, these methods struggled to find patterns in the noise.

One of the surprising results from the results with no feature selection was the recall of the SSGAN with a score of 0.50 ± 0.353 , 0.266 greater than the second highest, the neural

network. It is therefore much better at predicting more positive instances than others. However, the variance is large, showing discrepancies between different splits in the data. Unsurprisingly, this gives it the highest F1 score with the least variance too, of 0.47 ± 0.037 .

The kappa score gives a measure of agreement between two raters, the actual values vs the predictions, taking into account this agreement happening by chance [28]. For unbalanced datasets where predictions are focused on the minority class, as in this case, the values will be lower than expected. Therefore, the fact that the values for kappa are low is not inherently an issue. The values for Logistic Regression, OTE, OTE_boost, PASTE and PASTE_boost are around or below 0 in places which are very poor performances. SSGAN once again has the highest kappa median at 0.15.

The AUC-ROC value is the highest for XGBoost at 0.67 ± 0.078 , showing the robustness of the method. It also has a similar variance compared to the other methods too, however LightGBM and PASTE are lower showing more consistent performance, despite the lower values.

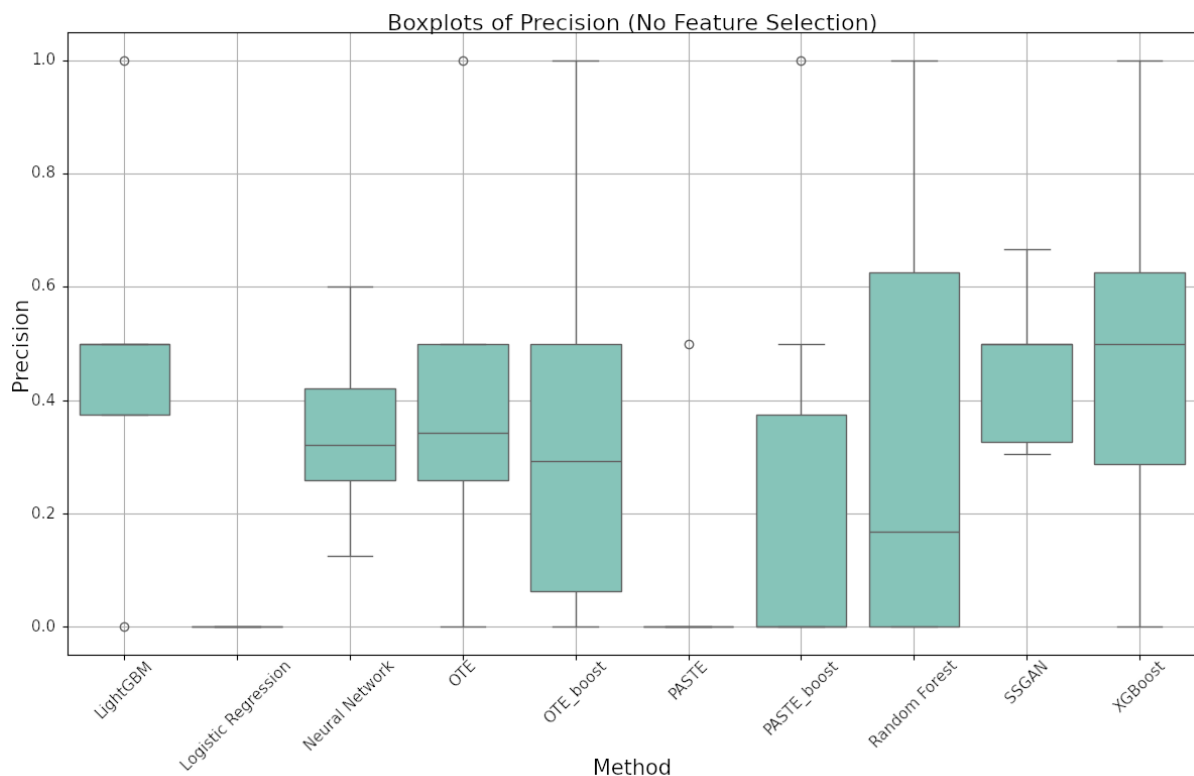


Figure 7.2: A boxplot showing the ranges of precision for each of the tested methods, when no feature selection has taken place.

7.4.2 Feature Selection Results

The results from the methods that used feature selection are the main focus. These results are expected to be more successful since there are fewer features, many of which would add noise to the learning process. Since these features were chosen because they had the largest influence on the predictability of the response variable.

As shown in Figure 7.3, Random Forest has the best performance with a median of 0.69 ± 0.292 and a low variance compared to other methods. This is much better, as there is a 69% chance that a prediction of the minority class, the failure class, is correct. Some folds achieve a precision of 100%, but this does depend on the specific data split. SSGAN and XGBoost peak higher than Random Forest, but are far more dependent on the split of the data. Interestingly, both PASTE and OTE outperform their boosting counterparts. This is likely due to the boosting methods overfitting the data, making them perform worse.

The recall of the neural network is highest at 0.5 ± 0.166 , meaning it found the most out of the all actual positive results. However, this is not necessary to note for this research question. Being able to predict just one extra vehicle correctly is more important than finding all of them, hence the precision is most important.

The kappa value is still relevant, and Logistic Regression has the highest with a low interquartile range, but a high variance due to the outliers above and below the median 0.3 ± 0.243 . The neural network, XGBoost and Random Forest are also successful here with values: 0.29 ± 0.210 , 0.24 ± 0.200 and 0.24 ± 0.161 .

These methods are also successful in term of ROC-AUC with median values between 0.6 and 0.7, with XGBoost the largest at 0.71 ± 0.090 . Random Forest is slightly lower with a similar variance 0.65 ± 0.081 . PASTE and OTE and their boosted variants are, once again, the worst of the tested methods to use. PASTE took the longest to train too making it inappropriate for this usecase.

The full table of results can be found in the appendix.

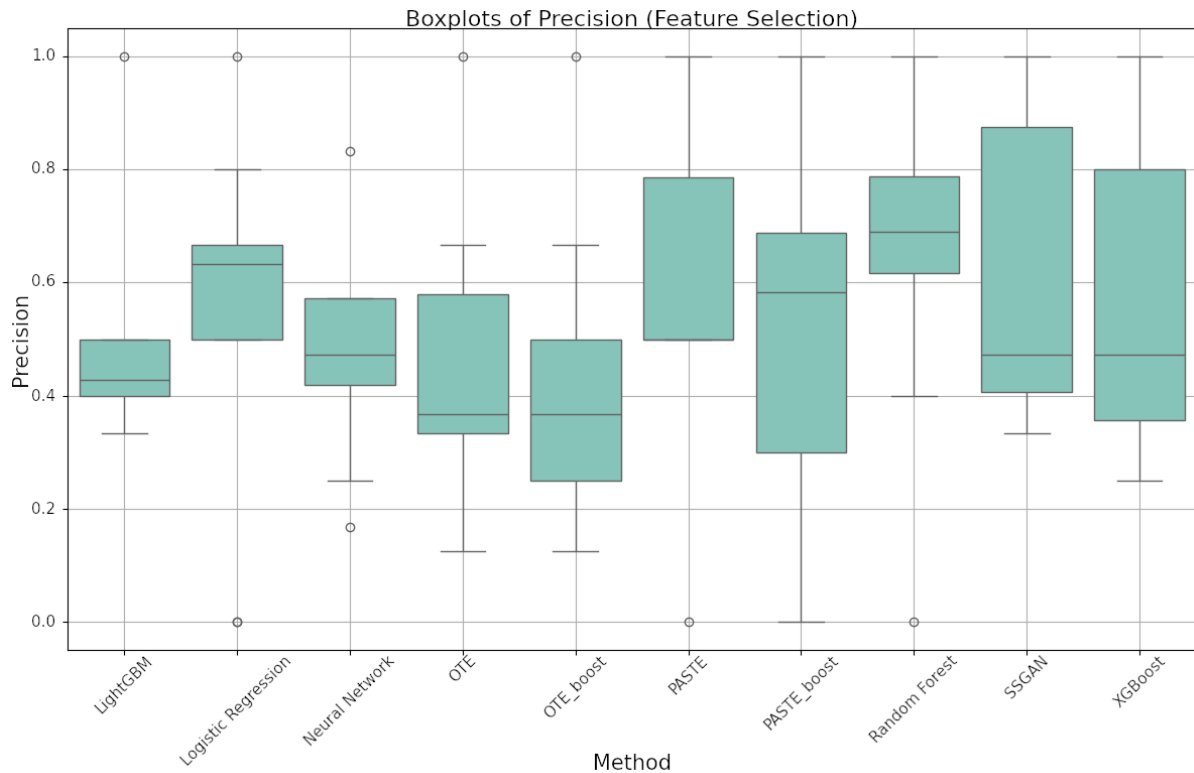


Figure 7.3: A boxplot showing the ranges of precision for each of the tested methods, when feature selection has taken place.

7.4.3 Extreme Thresholds

It was important to see how the trained Random Forest models performed on the extreme values. Therefore, each fold from the training had a separate test set in order to get an unbiased result. Within this test set, there was a subset of extreme values, in this case values [REDACTED], about [REDACTED] of the training set. The absolute size of this subset varied from [REDACTED] observations, meaning performance scores will range significantly based solely on the size of each data set, and are therefore less useful. For example, the precision is 0.50 ± 0.500 , however this depends on how NaN values are handled, in this example, precision is set to 0 if it is a NaN value, but any other value to replace the NaN is equally accurate. Removing the folds with NaN precision gives 5 remaining folds, each of which have perfect precision, 1.00 ± 0.000 . Figure 7.4 shows this, but also includes the results for all other folds, even the folds with no failure prediction. The recall is approximately 10% lower than a model specifically trained on the extreme values. As seen on the higher thresholds, the predictions become worse.

Other thresholds were also tested to view how the model predicted for different values in the

data. It is important to remember, this is the model trained using the entire data set with feature selection. The precision of the [REDACTED] predictions is reduced to 86%, which then reduces further to 72% for [REDACTED] values. For the values closest to the threshold, specifically [REDACTED], it is only just 55%. The model performs poorest on the values surrounding the threshold. There is a large imbalance in the two classes, and the data is very similar between the two. Therefore, for extreme values, if a prediction is a failure then there is a very high chance that it is correct, but as the values become less extreme, the model struggles to predict them.

7.5 Conclusion

In conclusion, the predictability of the final test results is not very high, the accuracy rarely higher than if the majority class was always predicted. However, looking at the precision, which is the more useful metric in this study, the results are more promising. Using all of the features was generally unsuccessful due to the complexity of the model and noisiness of low-influence features. The highest median precision here was only 0.5 from XGBoost. The Semi-Supervised GAN was surprisingly successful on this data with a relatively high recall and F1 score compared to the other methods. Unfortunately, it had a large variation making it unusable in its current state. Further refinements to the model could be implemented to improve the stability.

Looking at the feature selected models, the tuned Random Forests were the most successful predictions across the 10 folds with the highest median precision across all models. It was not the highest across all other metrics, where Logistic Regression, XGBoost and the Neural Networks all performed well. However, Random Forest was consistently within the top performers and maintained a low variance. Its maximum ability may be less extreme, but it continuously performed at a high level. Potentially another type of Neural Network would be able to outperform this, although with the small dataset this may limit the potential.

The Random Forest generally predicts very well, except for a small portion of the data immediately either side of the chosen threshold. For any prediction of a failure outside of these two values, there is a 72% chance that it is correct, and 69% including all values. This is significantly better than random chance, and will therefore make a positive impact on the output for the business.

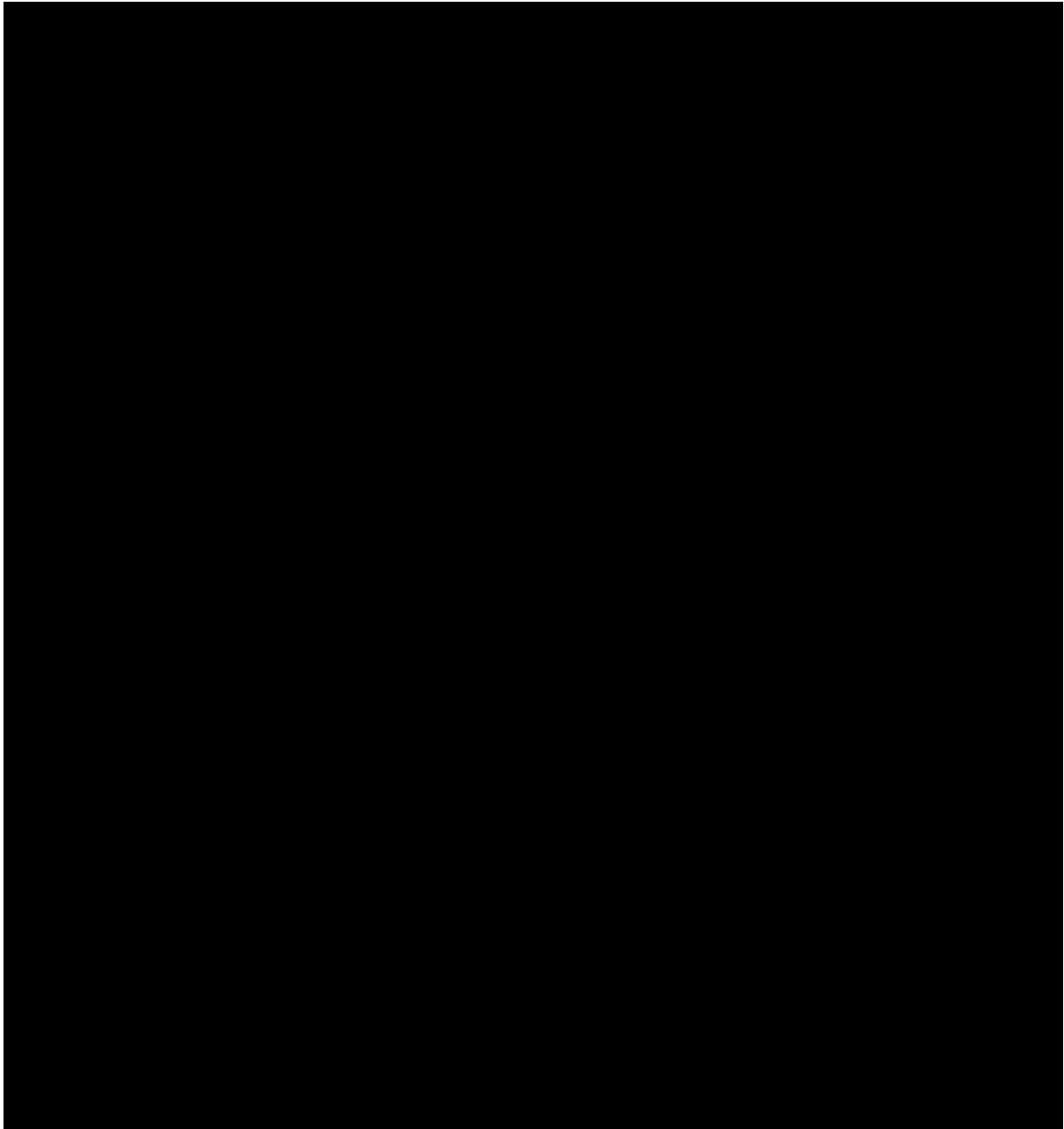


Figure 7.4: A grid of confusion matrices showing the prediction results for different cuts of the data. Each matrix is a combination of the confusion matrices for all 10 folds tested on. This is for all of the extreme thresholds mentioned in Section 7.3.1, including a confusion matrix only predicting values ██████, the closest values to the threshold.

Final Conclusions

8.1 Contributions

As mentioned in Chapter 1, a range of contributions have been made towards both the academia and industry. Combining the methods of P-Value adjusted decision trees and optimal tree ensembles created a new machine learning method with capabilities similar to that of Random Forest and XGBoost in particular cases, appearing to work best for smaller data sets with categorical data, and struggled more on larger, continuous data sets. Using conditional inference trees alongside the optimal tree ensemble framework, however, appeared to perform similarly to PASTE for most cases, except for categorical data, where it struggled far more. For both of these methods, the training time was the main hurdle restricting potential use as they both use multiple statistical tests to grow the trees which are more computationally expensive than a other methods, such as calculating the Gini impurity for a decision tree within a random forest. In addition, the pragmatic approach to improving results using residuals and training a further model to learn these after training the original model, can have a benefit, but the use-cases are limited. For methods that have already achieved accuracies above 90%, the residual boosting is likely to cause the overall model to overfit and reduce performance. If the predicted values are already close to the actual values, very little gain be gained anyway. For cases where the accuracy of a model is below 90%, using residual boosting appears to improve the performance

slightly, but importantly, it does not reduce the performance in any case. This makes using it worthwhile for poor performing models.

The industry contribution has been significant. At the start of the project, [REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

The image consists of a single, uniform black rectangle that fills the entire frame. There are no discernible features, text, or patterns other than the solid black color.

Many of the gains from the project were providing insight where insight lacked beforehand,

testing ideas and theories. It also showed what was possible with the data at the plant, insights that were previously not considered. Comparing the start and the end of the project, the attitude towards the use of data had changed drastically. [REDACTED]

[REDACTED]

[REDACTED]

Arguably, the most important contribution, combining both academic and industry, was the prediction tool. Using the best method found in Chapter 7, a prediction tool was created. The pre-processing steps and the machine learning method were integrated into an independent script provided to the company for further use. [REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

8.2 Prompts for Future Work

Naturally, there are areas for future research and questions left unanswered from this project.

Regarding the new methods, both PASTE and CISTE can be improved further by optimising the training process. Reducing the training time would make these methods more viable. However, another suggestion that was discussed during the project is using boosting methods within an optimal tree ensemble. This could take the one of two forms. Growing the trees using information gained from previous trees, creating a forest similar to XGBoost, and then culling trees that do not benefit the overall ensemble. Or the trees could be randomly grown, as originally seen in the optimal tree ensemble method, and then each tree is weighted based on a chosen metric, allowing more influential trees to have more influence on the final ensemble score, but not completely disregarding trees that do not appear to add information.

For industry, there are other improvements to suggest. [REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

In addition, given the class imbalance in the data, the default probability threshold for classifying observations into two classes of 0.5 may not be appropriate. Therefore, further work should investigate different probability thresholds using precision-recall analysis and receiver operating characteristic (ROC) curves to identify a threshold that maximises precision while maintaining an acceptable level of recall.

Furthermore, the hyperparameter search for random forests within Chapter 7 did not include one of the most influential hyperparameters, the number of features considered at each node, despite the fact that it affects the tree diversity and correlation. The default value was used, the square-root of the number of features. Instead, the number of trees was included, which generally improves stability, but has diminishing returns once a sufficiently large number of trees are grown. For future work, it would be beneficial to include the number of features considered at each node in the hyperparameter search, using a range of 2 to d , to assess whether further performance can be achieved.

Bibliography

- [1] Getty Images (US), Inc. v. Stability AI, Inc. https://cdn.vox-cdn.com/uploads/chorus_asset/file/24412807/getty_images_vs_stability_AI_delaware.pdf.
- [2] Research Alliance Ruhr. Online, Accessed 03 May 2023, Available at: <https://www.uaruhr.de/researchallianceruhr/trustworthydatascienceandsecurity.html.en>.
- [3] Research Centre Trustworthy Data Science and Security. Online, Accessed 15 June 2023, Available at: <https://rc-trust.ai/>.
- [4] Saheart dataset. <https://www.kaggle.com/datasets/emilianito/saheart>.
- [5] The Alan Turing Institute. Online, Accessed 03 May 2023, Available at: <https://www.turing.ac.uk/>.
- [6] Titanic dataset. <https://www.kaggle.com/datasets/yasserh/titanic-dataset>.
- [7] Trustworthy Digital Infrastructure for Identity Systems. Online, Accessed 19 June 2023, Available at: <https://research/research-projects/trustworthy-digital-infrastructure-identity-systems>.
- [8] Trustworthy Synthetic Data in Practice. Online, Accessed 19 June 2023, Available at: <https://research/research-projects/trustworthy-synthetic-data-practice>.
- [9] Turing Trustworthy AI forum. Online, Accessed 19 June 2023, Available at: <https://research/interest-groups/turing-trustworthy-ai-forum>.
- [10] Uci machine learning repository. <https://archive.ics.uci.edu/datasets>.

- [11] Data Protection Act, May 1973. Online, Accessed 10 May 2023, Available at: http://www.ilo.org/dyn/natlex/natlex4.detail?p_lang=&p_isn=55767&p_classification=01.
- [12] The world's most valuable resource is no longer oil, but data, 2017. Online, Accessed 03 May 2023, Available at: <https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data>.
- [13] GDPR Consent, 2018. Online, Accessed 10 May 2023, Available at: <https://gdpr-info.eu/issues/consent/>.
- [14] BAKKEN, S. The journey to transparency, reproducibility, and replicability. *Journal of the American Medical Informatics Association* 26, 3 (01 2019), 185–187. [10.1093/jamia/ocz007](https://doi.org/10.1093/jamia/ocz007).
- [15] BERRY, J. Database Marketing, Sep 1994. Online, Accessed 10 May 2023, Available at: <https://www.bloomberg.com/news/articles/1994-09-04/database-marketing?leadSource=uverify>.
- [16] BOJANOWSKI, P., GRAVE, E., JOULIN, A., AND MIKOLOV, T. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606* (2016). [10.48550/arXiv.1607.04606](https://doi.org/10.48550/arXiv.1607.04606).
- [17] BORREGO-DÍAZ, J., AND GALÁN-PÁEZ, J. Explainable Artificial Intelligence in Data Science. *Minds and Machines* 32, 3 (Sep 2022), 485–531. [10.1007/s11023-022-09603-z](https://doi.org/10.1007/s11023-022-09603-z).
- [18] BRADLEY, A. P. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition* 30, 7 (1997), 1145–1159. [10.1016/S0031-3203\(96\)00142-2](https://doi.org/10.1016/S0031-3203(96)00142-2).
- [19] BREIMAN, L. Technical note: Some properties of splitting criteria. *Machine Learning* 24, 1 (1996), 41–47. [10.1023/A:1018094028462](https://doi.org/10.1023/A:1018094028462).
- [20] BREIMAN, L. Random forests. *Machine Learning* 45, 1 (Oct 2001), 5–32. [10.1023/A:1010933404324](https://doi.org/10.1023/A:1010933404324).

- [21] BREIMAN, L., FRIEDMAN, J., STONE, C., AND OLSHEN, R. *Classification and Regression Trees*. Chapman and Hall/CRC, 1984. [10.1201/9781315139470](#).
- [22] BRIER, G. W. Verification of forecasts expressed in terms of probability. *Mon. Weather Rev.* 78, 1 (Jan. 1950), 1–3. [10.1175/1520-0493\(1950\)078%3C0001:VOFEIT%3E2.0.CO;2](#).
- [23] BUITINCK, L., LOUPPE, G., BLONDEL, M., PEDREGOSA, F., MUELLER, A., GRISEL, O., NICULAE, V., PRETTENHOFER, P., GRAMFORT, A., GROBLER, J., LAYTON, R., VANDERPLAS, J., JOLY, A., HOLT, B., AND VAROQUAUX, G. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning* (2013), pp. 108–122. [arXiv:1309.0238](#).
- [24] CHEN, T., AND GUESTRIN, C. XGBoost. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Aug. 2016), ACM. [10.1145/2939672.2939785](#).
- [25] CHLOE XANG. Scientists Increasingly Can’t Explain How AI Works, Nov 2022. Online, Accessed 10 May 2023, Available at: <https://www.vice.com/en/article/y3pezm/scientists-increasingly-cant-explain-how-ai-works>.
- [26] CLELFORD, T. The rise and rise of data science. Online, Accessed 10 May 2023, Available at: <https://www.iop.org/rise-and-rise-data-science>.
- [27] CLEVELAND, W. S. Data Science: an Action Plan for Expanding the Technical Areas of the Field of Statistics. *International Statistical Review* 69, 1 (2001), 21–26. [10.1111/j.1751-5823.2001.tb00477.x](#).
- [28] COHEN, J. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20, 1 (1960), 37–46. [10.1177/001316446002000104](#).
- [29] COMMISSION, E. Ethics Guidelines for Trustworthy AI, 2018. Online, Accessed 15 May 2023, Available at: <https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai>.

- [30] COUNCIL OF EUROPEAN UNION. Council regulation (EU) 2016/679, 1995. Available at: <https://eur-lex.europa.eu/eli/reg/2016/679/oj>.
- [31] COUNCIL OF EUROPEAN UNION. Council regulation (EU) 95/46/EC, 1995. Available at: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A31995L0046>.
- [32] DAVENPORT, T. H., AND PATIL, D. J. Data Scientist: The Sexiest Job of the 21st Century. *Harvard Business Review* 90, 10 (Oct 2012). 23074866.
- [33] DE GRAAF, R. Are All Explainable Methods Trustworthy?, 2019. Online, Accessed 12 June 2023, Available at: <https://towardsdatascience.com/are-all-explainable-models-trustworthy-4378c5b0c1a5>.
- [34] EUROPEAN ASSOCIATION FOR DATA SCIENCE. Euads mission. Online, Accessed 16 August 2023, Available at: <https://www.euads.org/euads-mission/>.
- [35] FELTEN, E., RAJ, M., AND SEAMANS, R. How will Language Modelers like ChatGPT Affect Occupations and Industries?, 2023. 10.48550/arXiv.2303.01157.
- [36] FRIEDMAN, J. H. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29, 5 (2001), 1189 – 1232. 10.1214/aos/1013203451.
- [37] GATES, B. Trustworthy Computing, Jan 2002. Online, Accessed 25 April 2023, Available at: <https://www.wired.com/2002/01/bill-gates-trustworthy-computing/>.
- [38] GLOROT, X., AND BENGIO, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010), Y. W. Teh and M. Titterton, Eds., vol. 9 of *Proceedings of Machine Learning Research*, PMLR, pp. 249–256. proceedings.mlr.press.
- [39] GOODFELLOW, I., BENGIO, Y., AND COURVILLE, A. *Deep Learning*. MIT Press, 2016. www.deeplearningbook.org.

- [40] GOODFELLOW, I. J., POUGET-ABADIE, J., MIRZA, M., XU, B., WARDE-FARLEY, D., OZAIR, S., COURVILLE, A., AND BENGIO, Y. Generative adversarial nets. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2* (Cambridge, MA, USA, 2014), NIPS'14, MIT Press, p. 2672–2680. [10.5555/2969033.2969125](#).
- [41] GRIFFIN, A. L. Trustworthy maps. *Journal of Spatial Information Science* 20 (2020), 5–9. [10.5311/JOSIS.2020.20.654](#).
- [42] HALL, W., AND PESENTI, J. Growing the artificial intelligence industry in the UK, Oct 2017. Online, Accessed 15 May 2023, Available at: <http://ftp.shujuju.cn/platform/file/2017-10-18/782c432045784854a04e458976aef0bf.pdf>.
- [43] HAYASHI, C., YAJIMA, K., BOCK, H.-H., OHSUMI, N., TANAKA, Y., AND BABA, Y. Proceedings of the Fifth Conference of the International Federation of Classification Societies (IFCS-96), Kobe, Japan, March 27–30, 1996. In *Data Science, Classification, and Related Methods* (1996). [10.1007/978-4-431-65950-1](#).
- [44] HERN, A. Streaming sites urged not to let AI use music to clone pop stars, Apr 2023. Online, Accessed 26 April 2023, Available at: <https://www.theguardian.com/business/2023/apr/12/streaming-sites-ai-copyrighted-music-copycat-tracks>.
- [45] HM GOVERNMENT. Data Protection Act 2018, c. 12., 2018. Available at: <https://www.legislation.gov.uk/ukpga/2018/12/contents/enacted>.
- [46] HOCHREITER, S., AND SCHMIDHUBER, J. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780. [10.1162/neco.1997.9.8.1735](#).
- [47] HOTHORN, T., HORNIK, K., AND ZEILEIS, A. Unbiased recursive partitioning: A conditional inference framework. *J. Comput. Graph. Stat.* 15, 3 (Sept. 2006), 651–674. [10.1198/106186006X133933](#).
- [48] JOSHI, M., AND BHARDWAJ, P. Impact of data transparency: Scientific publications. *Perspect Clin Res* 9, 1 (Jan. 2018), 31–36. [10.4103/picr.PICR_104_17](#).

- [49] JOY, T. T., RANA, S., GUPTA, S., AND VENKATESH, S. Hyperparameter tuning for big data using bayesian optimisation. In *2016 23rd International Conference on Pattern Recognition (ICPR)* (2016), pp. 2574–2579. [10.1109/ICPR.2016.7900023](#).
- [50] KAUR, D., USLU, S., RITTICHER, K. J., AND DURRESI, A. Trustworthy artificial intelligence: A review. *ACM Comput. Surv.* 55, 2 (jan 2022). [10.1145/3491209](#).
- [51] KAUSIK, B. Long Tails & the Impact of GPT on Labor, March 2023. [mpra.ub.uni-muenchen.de/id/eprint/117135](#).
- [52] KE, G., MENG, Q., FINLEY, T., WANG, T., CHEN, W., MA, W., YE, Q., AND LIU, T.-Y. Lightgbm: a highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Red Hook, NY, USA, 2017), NIPS’17, Curran Associates Inc., p. 3149–3157. [10.5555/3294996.3295074](#).
- [53] KHAN, Z., GUL, A., PERPEROGLU, A., MIFTAHUDDIN, M., MAHMOUD, O., ADLER, W., AND LAUSEN, B. Optimal trees selection for classification via out-of-bag assessment and sub-bagging. *IEEE Access* 9 (2021), 28591–28607. [10.1109/ACCESS.2021.3055992](#).
- [54] KHAN, Z., GUL, N., FAIZ, N., GUL, A., ADLER, W., AND LAUSEN, B. Ensemble of optimal trees, random forest and random projection ensemble classification. *Adv. Data Anal. Classif.* 14, 1 (Mar. 2020), 97–116. [10.1007/s11634-019-00364-9](#).
- [55] KODIYAN, A. A. An overview of ethical issues in using AI systems in hiring with a case study of Amazon’s AI based hiring tool. *Researchgate Preprint* (2019), 1–19. [42965903](#).
- [56] KOHLI, P., AND CHADHA, A. Enabling Pedestrian Safety Using Computer Vision Techniques: A Case Study of the 2018 Uber Inc. Self-driving Car Crash. In *Lecture Notes in Networks and Systems*. Springer International Publishing, feb 2019, pp. 261–279. [10.1007/978-3-030-12388-8_19](#).
- [57] LAUSEN, B., SAUERBREI, W., AND SCHUMACHER, M. *Classification and Regression Trees (CART) Used for the Exploration of Prognostic Factors Measured on Different Scales*. Springer (Physica-Verlag), Heidelberg, Germany, 1994.

- [58] LAUSEN, B., AND SCHUMACHER, M. Maximally selected rank statistics. *Biometrics* 48, 1 (1992), 73–85. [10.2307/2532740](#).
- [59] LEE, J.-G., ROH, Y., SONG, H., AND WHANG, S. E. Machine Learning Robustness, Fairness, and Their Convergence. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (2021), KDD '21, Association for Computing Machinery, p. 4046–4047. [10.1145/3447548.3470799](#).
- [60] LEONELLI, S. Locating ethics in data science: responsibility and accountability in global and distributed knowledge production systems. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374, 2083 (2016), 20160122. [10.1098/rsta.2016.0122](#).
- [61] LIAW, A., AND WIENER, M. Classification and regression by randomforest. *R News* 2, 3 (2002), 18–22. [CRAN.R-project.org/doc/Rnews/](#).
- [62] LOYOLA-GONZÁLEZ, O. Black-Box vs. White-Box: Understanding Their Advantages and Weaknesses From a Practical Point of View. *IEEE Access* 7 (2019), 154096–154113. [10.1109/ACCESS.2019.2949286](#).
- [63] LUNDBERG, S. M., AND LEE, S.-I. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Red Hook, NY, USA, 2017), NIPS'17, Curran Associates Inc., p. 4768–4777. [10.5555/3295222.3295230](#).
- [64] MAAS, A. L. Rectifier nonlinearities improve neural network acoustic models. [CorpusID:16489696](#).
- [65] MCCULLOCH, W. S., AND PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* 5, 4 (1943), 115–133. [10.1007/BF02478259](#).
- [66] McDONALD, J. *Handbook of Biological Statistics*, third ed. Sparky House Publishing, 2014. [www.biostathandbook.com/gtestgof.html](#).

- [67] MICHAEL S. POLLARD AND LOIS M. DAVIS. Decline in Trust in the Centers for Disease Control and Prevention During the COVID-19 Pandemic, 2021. Online, Accessed 12 May 2023, Available at: https://www.rand.org/pubs/research_reports/RRA308-12.html.
- [68] MIKOLOV, T., CHEN, K., CORRADO, G., AND DEAN, J. Efficient estimation of word representations in vector space, 2013. [10.48550/arXiv.1301.3781](https://arxiv.org/abs/10.48550/arXiv.1301.3781).
- [69] MILLER, R., AND SIEGMUND, D. Maximally selected chi square statistics. *Biometrics* 38, 4 (1982), 1011–1016. [10.2307/2529881](https://doi.org/10.2307/2529881).
- [70] NAIR, V., AND HINTON, G. E. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning* (Madison, WI, USA, 2010), Omnipress, pp. 807–814.
- [71] NAKAMURA, K. My Algorithms Have Determined You’re Not Human: AI-ML, Reverse Turing-Tests, and the Disability Experience. In *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility* (2019), Association for Computing Machinery, p. 1–2. [10.1145/3308561.3353812](https://doi.org/10.1145/3308561.3353812).
- [72] NASA. Hubble sees a perfect dust storm on mars, 2001. science.nasa.gov.
- [73] NATHAN YAU. Rise of the Data Scientist, June 2009. Online, Accessed 10 May 2023, Available at: <https://flowingdata.com/2009/06/04/rise-of-the-data-scientist/>.
- [74] NOER, K. U., CHADIJAH, S., AND RUDIATIN, E. There is no trustable data: the state and data accuracy of violence against women in Indonesia. *Heliyon* 7, 12 (2021), e08552. [10.1016/j.heliyon.2021.e08552](https://doi.org/10.1016/j.heliyon.2021.e08552).
- [75] ODENA, A. Semi-supervised learning with generative adversarial networks, 2016. [10.48550/arXiv.1606.01583](https://arxiv.org/abs/10.48550/arXiv.1606.01583).

- [76] ODENA, A., OLAH, C., AND SHLENS, J. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70* (2017), ICML'17, JMLR.org, p. 2642–2651. [10.5555/3305890.3305954](#).
- [77] O'HARA, K. Data trusts: Ethics, architecture and governance for trustworthy data stewardship. (*WSI White Papers, 1*) Southampton (2019). [10.5258/SOTON/WSI-WP001](#).
- [78] PROKHORENKOVA, L., GUSEV, G., VOROBEOV, A., DOROGUSH, A. V., AND GULIN, A. Catboost: unbiased boosting with categorical features. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Red Hook, NY, USA, 2018), NIPS'18, Curran Associates Inc., p. 6639–6649. [10.5555/3327757.3327770](#).
- [79] QLIK. Data Science vs Data Analytics, Apr 2021. Online, Accessed 03 April 2023, Available at: <https://www.qlik.com/us/data-analytics/data-science-vs-data-analytics>.
- [80] RIBEIRO, M. T., SINGH, S., AND GUESTRIN, C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier, 2016. [10.48550/arXiv.1602.04938](#).
- [81] ROMBACH, R., BLATTMANN, A., LORENZ, D., ESSER, P., AND OMMER, B. High-Resolution Image Synthesis with Latent Diffusion Models, 2021. Online, Accessed 02 May 2023, Available at: <https://github.com/CompVis/stable-diffusion>.
- [82] ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65, 6 (1958), 386–408. [10.1037/h0042519](#).
- [83] RUMELHART, D. E., HINTON, G. E., AND WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature* 323, 6088 (1986), 533–536. [10.1038/323533a0](#).
- [84] SALIMANS, T., GOODFELLOW, I., ZAREMBA, W., CHEUNG, V., RADFORD, A., AND CHEN, X. Improved Techniques for Training GANs, 2016. [10.48550/arXiv.1606.03498](#).
- [85] SCHUHMAN, C., VENCU, R., BEAUMONT, R., KACZMARCZYK, R., MULLIS, C., KATTA, A., COOMBES, T., JITSEV, J., AND KOMATSUZAKI, A. LAION-

- 400M: Open Dataset of CLIP-Filtered 400 Million Image-Text Pairs, 2021. [10.48550/arXiv.2111.02114](https://arxiv.org/abs/2111.02114).
- [86] SELINA SIMMONS-DUFFIN. CDC is criticized for failing to communicate, promises to do better, 2022. Online, Accessed 12 May 2023, Available at: <https://www.npr.org/sections/health-shots/2022/01/07/1071449137/cdc-is-criticized-for-failing-to-communicate-promises-to-do-better1>.
- [87] SOLAIMAN, I. The Gradient of Generative AI Release: Methods and Considerations, 2023. [10.48550/arXiv.2302.04844](https://arxiv.org/abs/2302.04844).
- [88] SOLAIMAN, I., BRUNDAGE, M., CLARK, J., ASKELL, A., HERBERT-VOSS, A., WU, J., RADFORD, A., KRUEGER, G., KIM, J. W., KREPS, S., MCCAIN, M., NEWHOUSE, A., BLAZAKIS, J., MCGUFFIE, K., AND WANG, J. Release strategies and the social impacts of language models, 2019. [10.48550/arXiv.1908.09203](https://arxiv.org/abs/1908.09203).
- [89] SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, A., SUTSKEVER, I., AND SALAKHUTDINOV, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 56 (2014), 1929–1958. [doi/10.5555/2627435.2670313](https://doi.org/10.5555/2627435.2670313).
- [90] STAHL, B. C. *Ethical Issues of AI*. Springer International Publishing, Cham, 2021, pp. 35–53. [10.1007/978-3-030-69978-9_4](https://doi.org/10.1007/978-3-030-69978-9_4).
- [91] STRASSER, H., AND WEBER, C. On the asymptotic theory of permutation statistics, 1999. [10.57938/ff565ba0-aa64-4fe0-a158-86fd331bee78](https://arxiv.org/abs/1508.03912).
- [92] STROWEL, A. ChatGPT and Generative AI Tools: Theft of Intellectual Labor? *IIC - International Review of Intellectual Property and Competition Law* (Apr 2023). [10.1007/s40319-023-01321-y](https://doi.org/10.1007/s40319-023-01321-y).
- [93] TEH, H. Y., KEMPA-LIEHR, A. W., AND WANG, K. I.-K. Sensor data quality: a systematic review. *Journal of Big Data* 7, 1 (Feb 2020), 11. [10.1186/s40537-020-0285-1](https://doi.org/10.1186/s40537-020-0285-1).
- [94] TOLLES, J., AND MEURER, W. J. Logistic regression: Relating patient characteristics to outcomes. *JAMA* 316, 5 (08 2016), 533–534. [10.1001/jama.2016.7653](https://doi.org/10.1001/jama.2016.7653).

- [95] TORRALBA, A., AND EFROS, A. A. Unbiased look at dataset bias. In *CVPR 2011* (2011), pp. 1521–1528. [10.1109/CVPR.2011.5995347](https://doi.org/10.1109/CVPR.2011.5995347).
- [96] TUCKER, C. Privacy, algorithms, and artificial intelligence. In *The economics of artificial intelligence: An agenda*. University of Chicago Press, 2018, pp. 423–437. [10.7208/chicago/9780226613475.001.0001](https://doi.org/10.7208/chicago/9780226613475.001.0001).
- [97] TUKEY, J. W. The Future of Data Analysis. *The Annals of Mathematical Statistics* 33, 1 (1962), 1 – 67. [10.1214/aoms/1177704711](https://doi.org/10.1214/aoms/1177704711).
- [98] TUKEY, J. W., ET AL. *Exploratory data analysis*, vol. 2. Reading, MA, 1977.
- [99] WEBER, M., APEL, S., AND SIEGMUND, N. White-Box Performance-Influence Models: A Profiling and Learning Approach. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)* (2021), pp. 1059–1071. [10.1109/ICSE43902.2021.00099](https://doi.org/10.1109/ICSE43902.2021.00099).
- [100] WECKERT, S. Google Maps Hacks, 2020. Online, Accessed 02 May 2023, Available at: <https://simonweckert.com/googlemaphacks.html>.
- [101] WEERTS, H., DUDÍK, M., EDGAR, R., JALALI, A., LUTZ, R., AND MADAIO, M. Fairlearn: Assessing and Improving Fairness of AI Systems, 2023. [arXiv.2303.16626](https://arxiv.org/abs/2303.16626).
- [102] WEN, Y., ZHANG, K., LI, Z., AND QIAO, Y. A discriminative feature learning approach for deep face recognition. In *Computer Vision – ECCV 2016* (Cham, 2016), Springer International Publishing, pp. 499–515. [10.1007/978-3-319-46478-7_31](https://doi.org/10.1007/978-3-319-46478-7_31).
- [103] WING, J. M. Trustworthy AI. *Communications of the ACM* 64, 10 (2021), 64–71. [10.1145/3448248](https://doi.org/10.1145/3448248).
- [104] WORSLEY, K. J. Testing for a two-phase multiple regression. *Technometrics* 25, 1 (1983), 35–42. [10.2307/1267724](https://doi.org/10.2307/1267724).
- [105] XU, F., USZKOREIT, H., DU, Y., FAN, W., ZHAO, D., AND ZHU, J. Explainable AI: A Brief Survey on History, Research Areas, Approaches and Challenges. In *Natural Language Processing and Chinese Computing* (Cham, 2019), Springer International Publishing, pp. 563–574. [10.1007/978-3-030-32236-6_51](https://doi.org/10.1007/978-3-030-32236-6_51).

- [106] ZANDER DANKO. Neon prescription... or rather, New transcription for Google Voice, Jul 2015. Online, Accessed 10 May 2023, Available at: <https://blog.google/products/google-voice/neon-prescription-or-rather-new/>.
- [107] ZUBI, Z. S. On distributed database security aspects. In *2009 International Conference on Multimedia Computing and Systems* (2009), pp. 231–235. [10.1109/MMCS.2009.5256696](https://doi.org/10.1109/MMCS.2009.5256696).



Appendix

Method	Accuracy	Precision	Recall	F1 Score	Kappa	AUC
LightGBM	0.69 ± 0.058	0.48 ± 0.358	0.12 ± 0.104	0.18 ± 0.122	0.08 ± 0.076	0.57 ± 0.045
Logistic Regression	0.68 ± 0.093	0.00 ± 0.000	0.00 ± 0.000	0.00 ± 0.000	-0.04 ± 0.080	0.52 ± 0.110
Neural Network	0.59 ± 0.127	0.35 ± 0.154	0.37 ± 0.262	0.33 ± 0.145	0.06 ± 0.173	0.52 ± 0.113
OTE	0.66 ± 0.057	0.37 ± 0.289	0.21 ± 0.181	0.30 ± 0.120	0.07 ± 0.137	0.42 ± 0.098
OTE_boost	0.66 ± 0.068	0.33 ± 0.312	0.16 ± 0.156	0.28 ± 0.121	0.02 ± 0.172	0.42 ± 0.109
PASTE	0.69 ± 0.023	0.10 ± 0.224	0.01 ± 0.040	0.20 ± 0.000	-0.02 ± 0.053	0.48 ± 0.086
PASTE_boost	0.68 ± 0.041	0.25 ± 0.382	0.04 ± 0.063	0.21 ± 0.042	-0.01 ± 0.088	0.51 ± 0.144
Random Forest	0.71 ± 0.066	0.32 ± 0.372	0.11 ± 0.130	0.16 ± 0.186	0.09 ± 0.153	0.57 ± 0.079
SSGAN	0.57 ± 0.199	0.45 ± 0.129	0.61 ± 0.354	0.44 ± 0.088	0.16 ± 0.128	0.65 ± 0.097
XGBoost	0.69 ± 0.085	0.47 ± 0.278	0.19 ± 0.089	0.26 ± 0.111	0.11 ± 0.128	0.66 ± 0.078

Table A.1: Results for the data **without** feature selection.

Method	Accuracy	Precision	Recall	F1 Score	Kappa	AUC
LightGBM	0.70 ± 0.047	0.53 ± 0.268	0.14 ± 0.083	0.20 ± 0.092	0.10 ± 0.083	0.64 ± 0.075
Logistic Regression	0.73 ± 0.109	0.54 ± 0.319	0.31 ± 0.193	0.38 ± 0.226	0.24 ± 0.243	0.67 ± 0.077
Neural Network	0.68 ± 0.096	0.48 ± 0.185	0.51 ± 0.166	0.48 ± 0.156	0.25 ± 0.210	0.65 ± 0.113
OTE	0.66 ± 0.105	0.46 ± 0.247	0.30 ± 0.162	0.34 ± 0.150	0.12 ± 0.211	0.45 ± 0.083
OTE_boost	0.64 ± 0.109	0.43 ± 0.258	0.21 ± 0.068	0.27 ± 0.103	0.06 ± 0.191	0.46 ± 0.099
PASTE	0.71 ± 0.029	0.58 ± 0.344	0.14 ± 0.161	0.32 ± 0.131	0.11 ± 0.140	0.34 ± 0.128
PASTE_boost	0.69 ± 0.066	0.52 ± 0.324	0.14 ± 0.128	0.29 ± 0.130	0.09 ± 0.179	0.40 ± 0.137
Random Forest	0.75 ± 0.050	0.66 ± 0.292	0.30 ± 0.161	0.40 ± 0.177	0.27 ± 0.161	0.67 ± 0.081
SSGAN	0.69 ± 0.064	0.60 ± 0.280	0.35 ± 0.255	0.36 ± 0.135	0.19 ± 0.086	0.59 ± 0.105
XGBoost	0.72 ± 0.079	0.58 ± 0.294	0.32 ± 0.143	0.40 ± 0.167	0.23 ± 0.200	0.71 ± 0.090

Table A.2: Results for the data **with** feature selection.