Designing a framework to gain insight into the automation of defect management at NLR

Industrial Engineering and Management

Bachelor assignment

UNIVERSITY OF TWENTE.

Author: Rick van de Water (s2297213)
g.h.vandewater@student.utwente.nl

Under supervision of: R. Brink (Rob) of NLR,
D.R.J. Prak (Dennis)
and
P.B. Rogetzer (Patricia)
of the University of Twente

November 9, 2023

Management summary

This research is performed at the Royal Dutch Aerospace Centre (NLR) in Amsterdam, the Netherlands. A knowledge center for innovation in aerospace. This research is focused on the maintenance department which is working on a project regarding continuing airworthiness. Continuing airworthiness is: "The set of processes by which an aircraft, engine, propeller or part complies with the applicable airworthiness requirements and remains in a condition for safe operation throughout its operating life." (De Florio, 2016). The main question that is researched is the following:

"How to design a framework to train defect management decisions based on real experts' decisions and obtain guidelines on the amount of input data needed to obtain acceptable results?"

In this research, a decision tree is generated based on experts' decisions, and is tried to learn this decision tree back with classification algorithms. To generate the necessary data for this problem cases have been generated which describe a defect of an aircraft and the situation in which it occurred, such as the spare parts availability and the flight schedule. Continuing airworthiness experts have decided for those cases when and where to rectify the defects. Those decisions are analyzed for a pattern with the different variables. This pattern is used to construct the decision tree which is used to generate more decisions for randomly generated scenarios. With this, four classification models are trained in Python that can decide what to do based on the input variables. This is done for six different sample sizes to see the relation between the amount of data needed and the accuracy of the model regarding learning back the decision.

The two main goals are to get insight into the performance of different classification algorithms and into the amount of data needed in order to learn back the decision tree accurately. The two best-performing classifiers are the gradient boosting classifier (GB) and the random forest classifier (RF). From our analysis, we conclude that the minimum number of data points needed is 500 for an accuracy of 99.6%. This means that the pattern in the data is almost fully learned and the model is able to decide correctly for most scenarios. Interestingly enough, the feature scores of those classifiers are not what we expect based on the decision tree used to train the model. Somehow the classifiers are able to obtain an accuracy score of 99.6% with different variable importance. When the sample size is increased to 5000, the feature importance becomes close to what we expected.

The biggest recommendation to NLR is that they should retrieve a set of data with defect management decisions to obtain reliable results. This research is fully finished, however, there is still a lot to explore with further research in this area. What can be investigated in further research is the application of the developed model in maintenance management and what the influence on the model will be when it is expanded such that all the different parts and defects are considered.

Readers Guide

Chapter 1 is the introduction and describes the company and the problem. Chapter 2 gives an introduction to defect management and how the scenarios look like in which a decision is needed. Chapter 3 summarizes theory about defect management, data generation, and classification algorithms. Chapter 4 is about the way data is retrieved, the decision tree is developed and the framework is designed. Chapter 5 provides the framework and the output of it and compares the performance of the classifiers. The last chapter 6 is about the results of the whole research, together with the discussion, limitations, and further research.

List of Figures

List of Tables

List of abbreviations

Abbreviation	Full term
AMP	Aircraft Maintenance Program
ANOVA	One-way repeated measures of analysis of variance
ADC	Air data computer
ASAM	Avionics Systems and Maintenance
ATL	Aircraft Technical Log
AUC	Area under the ROC curve
BAIT	Behavioral Artificial Intelligence Technology
CAMO	Continuous Airworthiness Management Organisation
CDS	Cabin door seal
CI	95% confidence interval
CPS	Cabin pressurization system
CWS	Cabin window shade
DM	Defect management
FFIS	Fuel flow indicating system
FN	False negative
FP	false positive
GB	Gradient boosting classifier
LR	Logistic regression
MEL	Minimum equipment list
MM	Maintenance management
NLR	Nederlands Lucht- en Ruimtevaartcentrum (Royal Dutch Aerospace Centre)
POS	Passenger oxygen system
RF	Random forest classifier
SE	Standard error
SVC	Support vector classifier
TN	True negative
TP	True positive

Contents

1 Introduction

In this chapter, the research is introduced. First, the company is described, followed by the project in this company this research belongs to, and after that the project this research contributes to, the problem identification and the research design.

1.1 Company Description

NLR is the Royal Dutch Aerospace Centre. They were founded in 1919 to improve the safety of military aviation. Because of the growth of civil aviation, they started to focus on that as well in 1937. They focused on improving the basics of scientific research into aviation. This focus on innovation in aerospace is currently still the main focus of the NLR (NLR, 2023). NLR consists of several departments. One of those departments is Avionics Systems and Maintenance (ASAM), this department is focused on innovations in maintenance techniques (NLR, 2022).

1.2 Bot project

One of the projects ASAM is working on is a bot for continuing airworthiness management operations (CAMO). Continuing airworthiness means the processes to ensure that the aircraft complies with airworthiness requirements and is in a condition for safe operations at all times (Brink, 2021). The need for this bot has to do with a future ambition of the NLR. They expect that in the upcoming future aviation will also become available for customer usage. They expect unmanned aircraft will be used as cabs (Brink, 2021). The expected increase in aircraft will also increase the total number of defects and maintenance. The current maintenance programs and decisions about rectification are all made by CAMO experts. The project of the NLR aims to develop a bot that can do all of this. The bot should also be able to keep track of the aircraft technical log (ATL), which is a document with every flight, defect, and rectification that the specific aircraft encountered during its lifetime. The bot also needs to determine the aircraft maintenance program (AMP), which describes all the upcoming maintenance tasks. The maintenance tasks in the AMP are preventive maintenance tasks and mandatory regular inspections. Next to those preventive maintenance tasks, a defect may be detected and corrective maintenance needs to be scheduled as well. When a defect is registered, the bot evaluates which maintenance is necessary and decides whether this needs to be done immediately. If immediate maintenance is necessary, the bot has to create a maintenance task. This decision process is called defect management (DM). If it is not necessary the next process starts: maintenance scheduling. Maintenance scheduling considers the AMP and schedules the maintenance task such that costs, risk, and downtime are minimal while obeying the regulations. This scheduling process is called maintenance management (MM) (Brink, 2021).

1.3 Problem Description

The tool does not exist yet and therefore the NLR is working on the development of this. The only part that already exists is a model that can create an AMP. All the other processes are not yet part of the CAMO-bot. For the bot to be able to decide in DM problems, it needs a decision framework that can decide based on the input variables what option for rectification is the best. To develop such a framework data about the defects and the corresponding decisions is needed to train the framework. In Figure 1, the problem cluster is mapped out. The blue rectangle is the problem experienced by the CAMO-organizations and the red rectangles are the problems the NLR currently runs into. To solve those problems this research investigates framework designs that can

learn decisions based on comparable input data and investigate the amount of input data needed to obtain acceptable results. Because the NLR does not have access to a data set, first a data set needs to be generated. Therefore this research also looks into how to generate a data set.

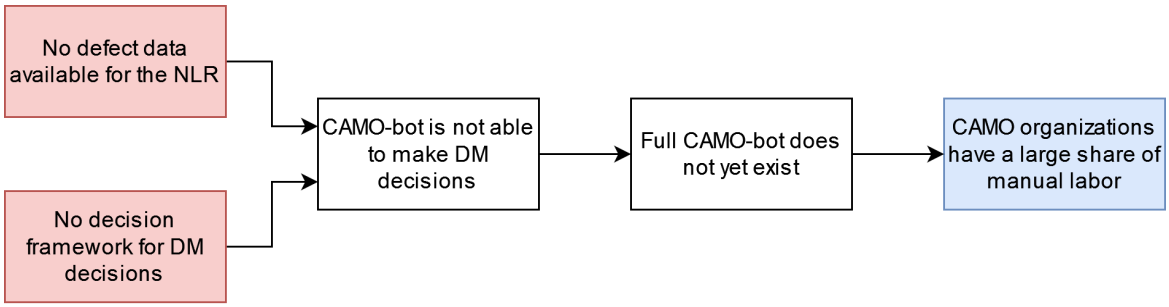


Figure 1: Problem cluster

1.3.1 Research design

The goal of this research is to gain insight into defect management (DM). The specific part this research is focused on is the decision of whether or not to continue operations. For this, a data set with possible defects is necessary, as well as the decisions that are made in those situations. Since this data is not available, this needs to be obtained. The defects are a small sample size for a specific type of aircraft and DM experts are asked to decide what to do with those defects. The list of defects and the experts' decisions together are the data set for the remainder of the research. This is necessary for this research since we want to implement supervised classification algorithms. These classifiers need a decision, which they will learn back. The expert's decisions are reviewed such that the logic behind the decisions is copied into a decision tree. This decision tree is then able to make comparable decisions as the experts did. The decision tree is used to decide on new, randomly generated, scenarios. The decision it made, is assumed to be the same as the experts should have made. This assumption can be made, since this research does not investigate the accuracy scores and feature importance of the classifiers for this specific case, but aims to prove the concept of learning back a decision tree with classification algorithms. These random scenarios with decisions are analyzed for correlation between the different variables and the final decision that is made. With those correlations known, a model is developed that can use those to make the most likely decision based on the variables. This model has an uncertainty. This uncertainty mostly depends on the number of scenarios used to determine the correlation. To get insight into this relation different number of scenarios were tested and the corresponding uncertainty is documented. Based on this a recommendation is delivered on the size of the data set that is necessary to come up with a reliable model. The final decision method needs to be able to perfectly learn the decision back and therefore an accuracy of 99% is aimed for. This means that the model is not allowed to decide wrongly in many scenarios. To achieve this, we also want to train on as many data points as possible, which is possible by using cross-validation, which is explained in Chapter 3.

1.4 Research Questions

The main research question that is answered in this thesis is the following:

"How to design a framework to train defect management decisions based on real experts' decisions and obtain guidelines on the amount of input data needed to obtain acceptable results?"

This question is answered in the following research. For that, we split it up into different underlying questions that also are answered in this research.

1. How are defect management decisions currently made?
    - What does defect management consist of?
      To develop a bot for defect management, insight into those processes is needed. This knowledge question helps to gain insight into those processes and is answered in chapter 2.1.
    - Which variables are considered by the NLR with defect maintenance?
      The bot has to decide whether maintenance should be deferred or not. To make this decision a lot of variables are considered. In chapter 2.2 the relevant variables, according to the NLR, are discussed.
2. How to develop a tool for decision-making in defect management?
    - What are defect management decisions about?
      To develop a tool to make DM decisions, understanding where those decisions are about is crucial and this is found in Chapter 3.1.
    - Which variables to consider for defect management decisions?
      The variables found with literature research to consider for making DM decisions are found in Chapter 3.2.
    - What is cross-validation and how to apply it?
      The application of cross-validation improves the performance of classification algorithms. This is discussed in Chapter 3.4.
    - Which classification algorithms exist for determining the importance of different variables?
      The classifiers determine based on known data with variables what the best decision is for unknown cases. There are multiple classifiers and this question is to determine which methods are suitable for this research and those are discussed in Chapter 3.5.
    - How to visualize the accuracy of a classifier? ROC-curves display the accuracy of classifiers and how to interpret and construct those curves is explained in Chapter 3.6.
    - When are multiple means statistically different? To determine the difference of multiple means, statistical tests can be applied. Chapter 3.7 discusses which test is best suited for the accuracy scores of different classifiers, which are tested on the same data.
3. How to develop a framework that provides insight into the importance of the variables and the accuracy of a classification algorithm?
    - How is the decision tree constructed?
      Chapter 4.1 describes how the decision tree is constructed based on the experts' decisions and how it is adapted to generate data for the framework.
    - What does the framework look like that determines the accuracy and feature importance of the different classification algorithms?
      How the model loads and prepares the data, implements the classification algorithms, and compares the classifiers' performances is described in Chapter 4.2.
4. What are the results and findings of the results?

- What is the average accuracy score of the classifiers?
  In Chapter 5.1 the accuracy scores are provided per sample size and per classifier.
- How does the accuracy score of a classifier behave?
  The different classifiers all behave differently and all provide useful insights for this research. What we learn from each classifier can be found in Chapter 5.2.
- Do the accuracy scores of the classifiers significantly differ?
  To draw conclusions on the differences, statistical tests are applied in Chapter 5.3 to investigate the significance of the differences in the classifier accuracy scores.
- Do the classifiers have a similar feature importance?
  All the classifiers rank the features by importance, but is this ranking similar and how does it change with different sample sizes? This is analyzed in Chapter 5.4

1.5 Scope

The scope of this research is about generating data based on a decision tree built from ten fictive situations and the focus is on the theoretical properties of the decision framework. To narrow down this research, only six components are considered and other variables are simplified by binarization of the values. More about the variables is described in chapter 2.2. Data generation was time-consuming and therefore only ten experts were asked and other data scenarios were constructed using a decision tree. Different types of aircraft have different functionalities and therefore other variables as important key factors. This research is narrowed down to airlines that focus on passenger transportation.

1.5.1 Deliverables

- The decision tree, based on the expert's decisions, which can decide in every scenario.
- Insight into and the performance of the different classification algorithms.
- Insight into the relation between the model accuracy and the amount of input data.
- Insight into the ability of classification models to learn back a decision tree.
- The thesis itself.

1.6 Next step

Before we start with the research, we first dive into the current situation of the development of the CAMO-bot and into what Defect Management is.

2 Current situation

Based on the research question we investigate in this chapter what the current situation is. More specifically we look into how defect management decisions are made and which variables might influence those decisions. Also, the different scenarios are constructed that are presented to experts to get insight into the decision process. Those experts decide on the scenarios and provide the influence the different variables had on their decision.

2.1 Defect management

Defect management is about making the decision whether or not to defer corrective maintenance. For this, a lot of variables can have an influence. First of all the regulations. The regulations can require immediate rectification or give constraints for deferral. Secondly, the flight schedule is important to consider. An aircraft is assigned to a number of flights to perform in the upcoming hours and days. Canceling those flights brings high costs with it. In the DM decision, it is therefore desired to consider this as well. Next to this, the availability of a maintenance crew, dock, and the needed spare parts can influence the decision as well. This does influence the time the aircraft has to be on the ground before taking off again. The airport where the defect is noticed is also of importance since an airline does not have a service contract with a maintenance provider at every single airport. When there is no service contract it takes more time, effort, and money to execute maintenance tasks on that airport. Then there are the upcoming maintenance tasks. Every aircraft has a lot of regular checks scheduled for which the aircraft has to be on the ground anyway. When any of those tasks are coming up, deferral might be more desired(Brink, 2021).

2.1.1 Immediate rectification because of regulations

The MEL states which parts must be working to be allowed to fly. If a part, or combination of parts, is a defect that is considered critical, so it is not allowed to be a defect when in operation, then immediate rectification is always the answer to the problem. Even if rectification is not yet possible because of a lack of crew, parts, time, or anything else. The aircraft has to be rectified before operating again.

2.1.2 Deferral because of no impact of the defect on operations

For some parts and defects, the MEL gives space to execute flights even without extra constraints. In those situations, it is almost always preferred to defer maintenance till later. Sometimes the best moment might be to go for immediate rectification but those circumstances are rare. That only happens if time, a maintenance crew, a maintenance dock, and the fitting spare parts are available. But even then deferral could be preferred.

2.1.3 Noncritical defects with impact on operations

For a lot of defects or combinations of defects, the decision is not clear beforehand. The regulations give some space to execute flights, but often there are operational constraints that can cause trouble in executing flights. For those situations, a choice has to be made between the disadvantages of immediate rectification and the disadvantages of executing flights with constraints. The disadvantages of immediate rectification can be last-minute cancellation, expensive replacement, and looking for ways to make the immediate rectification possible. The operational constraints on flights can be for example a capacity constraint, maximum flight height, and unpressurized flights.

2.2 Defect management variables

In the paragraph before, a lot of variables are called and is stated that those all have influence on the decision to defer maintenance. In the next parts, all those variables are described and explained. The decision is binary. Maintenance is done immediately or deferred. This decision is made by experts and they take different variables into account for their decision process. Of course, some variables have a bigger influence on the decision than others. The variables taken into account for this research are listed and explained below.

- Component: which defect occurred on which component?
- Location: on which airport is the aircraft and are needed resources available?
- Flight Schedule: when are the next scheduled flights for this aircraft?
- Maintenance tasks: when is the next scheduled maintenance?
- Circumstances: Are there weather or terrain difficulties that might impact the decision?

2.2.1 Component

The defect and which component is defective are relevant since the regulations are different for every single defect and component. In order to decide what to do in certain situations is this information crucial.

2.2.2 Location

Aircraft are in different airports throughout their lifespan. Not in all the airports their airline has a service contract. Rectification in an airport without a service contract is much more expensive and difficult to arrange. Also in airports with a service contract rectification is difficult, if the needed resources are not available like a maintenance docking station, personnel, and parts.

2.2.3 Flight schedule

Rectification of an occurred defect takes time, during which the aircraft is out of use. It is assumed that at night the gap till the next flight is big enough in order to rectify the defect, but during the day this is not the case as can be interpreted from the flight schedule displayed in table 1. Given this information, the flight schedule might be relevant and therefore is taken into account with the scenarios.

2.2.4 Maintenance tasks

Aircraft have mandatory maintenance tasks that have to be performed once in a while. Different tasks exist: some occur more regularly and others take more time. In those maintenance sessions, there is space to rectify defects. If a maintenance task is scheduled in the near future, the rectification of the defect might be deferred more likely.

2.2.5 Circumstances

For some components, the regulations described in the MEL do not allow deferring the rectification when the vision is not enough or the flight is above a certain height. In order to decide in those situations additional information about those circumstances is needed. Note that this data is only displayed for the components where the MEL says the circumstance should be considered.

2.3 Development of scenarios

Those variables that play a role in Defect Management (DM) are known, but to what extent those variables influence the decision is unknown. To get an overview of the influence of each variable on the decision a questionnaire is developed. The questionnaire consists of different scenarios. In each scenario, a defect is described where a DM decision is needed. The description contains all the variables above, such that the expert is able to decide. Next to the decision, the experts are asked to fill in the influence each variable had on the decision. For this, the options are big influence, small influence, or no influence. The developed scenarios can be seen in appendix A. Those scenarios are developed such that each of them contains an interesting trade-off that provides insight into the decision-making process. The scenarios are provided to multiple experts and therefore different decisions occur. Especially for those scenarios, it is interesting to look into the differences in the focus of the experts to see where the different outcomes came from.

2.3.1 Scenario description

We assume that all the scenarios take place for the same fictional airline on the same routes. This airline is stationed at Rotterdam The Hague Airport (RTM) and serves three other cities: London City Airport (LCY), Istanbul Airport (IST), and Adolfo Suárez Madrid–Barajas Airport (MAD). Istanbul Airport has a service contract, but the other two do not. Between those cities is a flight network such that every possible connection is flown every day both in the morning and the afternoon. For this schedule five aircraft are in use. The full schedule can be found in table 1. Note that all the times displayed are in UTC+1. There are six different components considered: the cabin pressurization system (CPS), cabin window shade (CWS), the air data computer (ADC), the passenger oxygen system (POS), the cabin door seal (CDS) and the fuel flow indicating system (FFIS).

Flights	Flight time	Morning schedule			Afternoon schedule		
	(in minutes)	Aircraft	Start	End	Aircraft	Start	End
RTM-MAD-IST	395	1	07:00	13:35	3	15:30	22:05
RTM-IST-MAD	450	2	07:00	14:30	5	15:30	23:00
IST-RTM-LCY-RTM	376	3	07:00	13:16	1	15:30	21:46
LCY-IST-LCY	451	4	07:00	14:31	4	15:30	23:01
MAD-LCY-MAD-RTM	455	5	07:00	14:35	2	15:30	23:05

Table 1: Flight schedule Airline

2.3.2 Scenarios

The scenarios that are delivered to the experts for them to make a decision can be found in appendix A. Those scenarios include all variables described in Chapter 2.2 and are chosen such that all of them provide valuable information for the development of the tools. The resources were only provided for the scenarios where the aircraft was at an airport that had a service contract. For the other scenarios, the resources were by definition unavailable because of the lack of a service contract. The resources play a significant role Since all the experts received the same set of scenarios, it might occur that some of them decide differently in the same scenario.

2.3.3 Trade-offs

What insight does each scenario provide? The scenarios in the questionnaire are developed carefully such that the number of scenarios and the length of the questionnaire are still reasonable. In a lot of possible situations, the regulations prohibit deferring rectification, and in others, it is completely unnecessary to do maintenance. For the scenarios, situations are chosen in which the decision is not clear beforehand, such that a maximum of valuable information can be retrieved from them. The interesting trade-off per scenario can be seen in table 2.

Scenario	Trade-off
1	Likely to defer, but everything for rectification available
2	Rectification preferred and convenient to rectify
3	Rectification preferred on a suitable location, but all else inconvenient
4	Rectification preferred with enough time, but all else inconvenient
5	Likely to defer, but enough time for rectification and upcoming maintenance tasks
6	Likely to defer, for rectification only not enough time
7	Likely to defer, but enough time and best location for rectification
8	Rectification is mandatory for at least one of the defect components. Which defect to rectify, or both? No time for rectification but all else is available.
9	Two failed components, but deferring is possible with constraints. When to rectify the components? Currently, enough time, but no service station, and resources are available.
10	Rectification mandatory for at least one of the defect components. Which defect to rectify, or both? Currently enough time, but no service station and available components.

Table 2: Tradeoffs

2.4 Questionnaire responses

The responses of the five experts can be found in appendix B. The data in this appendix is also summarized per scenario. Per scenario, the decision can be seen, just as the importance score of the different variables per scenario. The importance is a score from 0 to 10. This score is determined by the following formula: $Score = 2 * \#Big + 1 * \#Small + 0 * \#No$, where $\#Big$ means the number of experts that indicated that it has a big influence, $\#Small$ the number of experts that indicated a small influence, and $\#No$ the number of experts that indicated that it has no influence. Since we have five experts, the maximum score is 5*2=10 and the minimum score is 0. A score of 0, 1, or 2 is seen as no influence, 3 to 6 as a small influence, and 7 to 10 as a big influence. The summary of the response and the scores are in table 3:

Scenario	Decision	MEL	Tasks	Flights	Location	Resources
1	Immediate Rectification	7	0	6	6	7
2	Immediate Rectification	9	4	3	4	2
3	Immediate Rectification	10	0	8	7	6
4	Defer Maintenance	10	0	7	8	
5	Defer Maintenance	10	2	6	4	
6	Defer Maintenance	10	4	8	5	5
7	Defer Maintenance	10	0	6	2	4
8	Immediate Rectification	9	2	6	8	9
9	Defer Maintenance	10	2	8	6	
10	Immediate Rectification	10	1	7	5	
Average		**9.5**	**1.5**	**6.5**	**5.5**	**5.5**

Table 3: Summary of experts responses

The experts noted that in principle deferring is the desired option. However, the rules must be obeyed. If it is not possible to depart without breaking the rules, rectification must be done first. In some specific scenarios, it is possible to rectify the defect already before the next scheduled flight, but those situations will not occur that often. When that is the case, though, rectification might as well be done.

The first thing that we can notice is that the experts value the MEL as very important with a score of 9.5 out of 10. This makes a lot of sense because this is the regulation that tells under which conditions the aircraft is allowed to depart and when it needs to stay grounded. The same is true for the upcoming flights. The biggest problem with immediate rectification is that it takes time. But when there is time, because the aircraft has to wait till the next morning for departure, it can be the best solution to use the night to repair the defect, if possible. This is probably the reason why the experts value this variable so highly. However, it is not always valued with a 10. We assume that this is due to the fact that for some scenarios other variables are also very important and therefore the MEL is valued as relatively less important than in the scenarios where it scores a 10.

Next to that, it can also be noticed that the upcoming maintenance tasks are not taken into account that much for the decision with a score of only 1.5. This has to do with the fact that the maintenance tasks have to do with the whole plain and not only with the defect. Next to that, if you have to repair before you are allowed to depart, it does not matter when the next service is scheduled.

The flights, locations, and resources are more varied between the different scenarios. This is probably depending on the specific scenario and the value of the variable in that scenario.

A scenario that stands out is scenario 2. This has to do with the fact that the MEL demands to rectify immediately and therefore the other variables are ignored since they do not influence that decision anymore.

2.5 Starting point

These are the basics of defect management and we have data about the decisions. Now we need to develop a generator that is able to generate scenarios comparable to decisions as the experts did.

We need this generator to create a big data set and to test the model we have developed. This model exists of different classification algorithms and provides an accuracy score for how well it learned the decision back and what the importance of the different variables was. Before we can develop this generator, we need literature that confirms that it is possible to extrapolate your data set, based on patterns found inside the data set. We also need literature that backs up the model we developed and the method and the classification algorithms we used inside the model.

3 Literature Review

In this part, the literature research will be described. This will be the base for the rest of the research.

3.1 Defect management decisions

Defect management is everything that happens after a defect occurs, beginning with what is wrong, followed by the decision to continue operating and when and how to fix the defect. When an unexpected issue occurs with an aircraft a technician determines between arrival and departure of the aircraft whether or not the defect needs to be rectified before the next flights can be performed. The decision is made by the technician by looking into extensive maintenance manuals. There is no decision support, there is a gap between the theoretical knowledge and practical application, and the field lacks decision-making models (Koornneef, Verhagen, & Curran, 2020).

3.2 Defect management variables

Important variables in aircraft maintenance are cost, time, quality, reliability, maintainability, availability, and flexibility or replaceability (Pleumpirom, Amornsawadwatana, et al., 2012). This is mainly about maintenance scheduling. For the defect management decision, we do not take into account the quality of the maintenance and the reliability of the aircraft after the rectification. Costs are seen as an indirect consequence of other unwanted disturbances, such as maintenance on an outstation, cancellation, or delay of the flight. Part availability, whether there is enough time for rectification without disturbing the flight schedule, the upcoming other maintenance tasks, which is the maintainability and the current location of the aircraft are also considered in this research (Pleumpirom et al., 2012).

3.3 Behavioral Artificial Intelligence Technology

Based on the decisions that the experts made in the carefully created cases, the influence of each variable on the final decision will be analyzed by using the Behavioral Artificial Intelligence Technology (BAIT) method (Ten Broeke, Hulscher, Heyning, Kooi, & Chorus, 2021). The BAIT method generates cases where a decision needs to be made and a list of variables that might influence this decision. For each case, the variables are measured and decisions, made by experts, will be documented as well. When there are enough decisions for different cases, the correlation between the variables and the final decision is analyzed. How much impact does each of those variables have on the final decision? When the relation between the variables and the decision is known, a bot can be developed that uses this relation to decide on a case based on the variables that are measured. When the bot is developed it will decide on some new cases (that are not used for the development of the bot). Those cases are then submitted to experts for their decision. Is the bot decision the same as the expert decision or are there differences? Does the bot make decisions that should not be allowed? When the bot performance is good enough, a proof of concept can be provided as well (Ten Broeke et al., 2021).

3.4 Stratified k-fold cross validation

Classifiers work with training and test data. Training data is the data that the model uses to learn the decision and test data is the data used to test the performance of the classifier. The model's accuracy is the rate of correct decisions made for this test data. To improve the performance of

classification algorithms cross-validation can be applied. K-fold cross-validation splits the data set into multiple (k) parts (folds) that are all once used as test data and other times as training data. This means that the classifier is tested k times and therefore provides k accuracy scores. The total accuracy of the classifier is the average of those k accuracy scores. Since every part of the data set is used to test the model and the accuracy is an average of ten repetitions, this method provides a more reliable accuracy(Wong & Yeh, 2019). The classifier is also less likely to overfit the data set. Overfitting is memorizing the data set instead of learning the patterns in the data set. Overfitting occurs when the model learns the data set too well, but this is not likely with cross-validation since it combines the results of k different training sessions(Ghojogh & Crowley, 2023). Standard values for the number of folds k are 2, 5, and 10. 10 is the most used one in literature(Ghojogh & Crowley, 2023) and the most preferred, but the test sample needs to include at least five times both decision options. This means that a model with a binary decision variable needs at least 2 (both decision options) * 5 (required number of the decision) * 10 (preferred number of folds) = 100 data entries. To make sure that every fold contains the same amount of both decision options, a stratified k-fold could be applied. Stratified means that the data is sorted based on the decision variable and then split such that each fold does contain the same number for each decision alternative(Prusty, Patnaik, & Dash, 2022).

3.5 Classification algorithms

In this research, we want to use different classification algorithms and compare them to each other. To decide which classifiers to use, literature research is conducted to investigate which classifiers are suited in the case of this research (Gama & Brazdil, 1995).

3.5.1 Logistic regression

Logistic regression (LR) can be used when a classification problem is binary. It uses a Sigmoid function to generate a probability. This probability will be compared to a predetermined threshold to assign a label to the given problem (Gong, 2022). The function itself uses all the input variables as predictors and multiplies them with the regression coefficients. The regression coefficients are first approximated and then improved until stability is reached. With the final regression coefficients, the Sigmoid function is finished. The threshold for the function is based on the ratio of the outcomes (LaValley, 2008).

3.5.2 Random forest classifier

The random forest classifier (RF) is a collection of decision trees. Each decision tree will provide an outcome for a specific case, but the final decision will be the majority vote of all the decision trees. The different decision trees are all generated on a sample of the training data (Breiman, 2001). This classifier is applied by Kim(Kim, Ji, Kim, & Park, 2022) and Kumar(Kumar, Sharma, Muttoo, & Singh, 2022) to determine repair tasks based on the defect description.

3.5.3 Gaussian naive Bayes classifier

Naive Bayes is based on Bayes' Theorem, which is an approach that calculates the conditional probability for every single feature. This classifier assumes independence between all the different features. The final classification is done by determining for each outcome class the probability that the specific case belongs to that class. The class with the highest probability is the predicted outcome. Naive Bayes also performs well on small data sets, but the assumption of independence

between the different features is most of the time unrealistic(N. Friedman, Geiger, & Goldszmidt, 1997). Because of this assumption, this classifier does not apply to this research and is therefore not applied in the model.

3.5.4 K-nearest neighbor algorithm

The K-nearest neighbor algorithm looks for each input, which known situation is most similar, and what the decision class of that neighbor is. This algorithm works especially for data sets with a lot of continuous variables. A disadvantage of this procedure is that it assumes all variables are equally important. Another disadvantage is that the model does not provide the confidence of the decision. On the other hand, it has obtained good results for small sample sizes (Keller, Gray, & Givens, 1985). We want to have the confidence of the decision in our case and we also have a lot of binary variables. Therefore this method is not best suited to our case and will be left out of the scope of this research.

3.5.5 Support Vector Classifier

The support vector classifier (SVC) determines a border between the two different classes. This is done based on maximizing the distance from this border to all the different vectors. (Noble, 2006) This classification method is especially applicable when there are a lot of continuous variables, but does also works for binary variables (Ben-Hur & Weston, 2010). This classifier is used for repair task allocation by Kim(Kim et al., 2022) and Kumar(Kumar et al., 2022).

3.5.6 Gradient boosting classifier

Gradient boosting (GB) creates regression trees in the direction of the gradient to be more accurate. Just like with random forest, this classifier uses different regression trees, and the decision is made by voting (Lin, Yue, & Mao, 2014), (J. H. Friedman, 2002).

3.5.7 Decision tree

The solutions found by decision trees are local optima and not often also the global optimum. Inaccuracies in the training data can have a great impact on the decision method (Lin et al., 2014). A big advantage of decision tree algorithms is that the classification method (the decision tree) can be extracted and investigated (Ochodek, Hebig, Meding, Frost, & Staron, 2022). Since Gradient boosting and random forest also use decision trees, this advantage applies to those classifiers as well (Lin et al., 2014).

3.6 Visualization of classifier performance

Classifiers determine for the test data in which class it belongs. If a positive is correctly predicted it is called a true positive (TP), but if this prediction is wrong it is a false positive (FP). If a negative is correctly predicted it is called a true negative (TN), but if it is positive then it is called a false negative (FN)(Hoo, Candlish, & Teare, 2017). This is visualized in Figure 2.

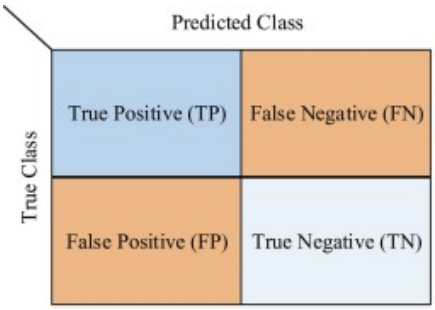


Figure 2: Confusion matrix (Demir, 2022)

The sensitivity, or true positive rate, is the percentage of correct positive predictions: $Sensitivity = \frac{TP}{TP+FN}$. The specificity, or false positive rate, is the percentage of false positive predictions to all negative values: $Specificity = \frac{FP}{FP+TN}$. The accuracy is the percentage of correct predictions: $Accuracy = \frac{TP+TN}{TP+FN+FP+TN}$(Demir, 2022).

The receiver operating characteristic (ROC) curve is a way to visualize the classifier performance. This is done by plotting the sensitivity on the y-axis and 1-the specificity on the x-axis. The accuracy is visible in this graph as the area on the ROC curve (AUC). When the performance of the classifier is random, the diagonal line is expected to be the result. This also mean that the accuracy of a random performance (not trained classifier) is expected to be around 50%(Fawcett, 2006). An example of a ROC-curve can be seen in Figure 3, where the straight diagonal is a random classifier with an accuracy of 50%.

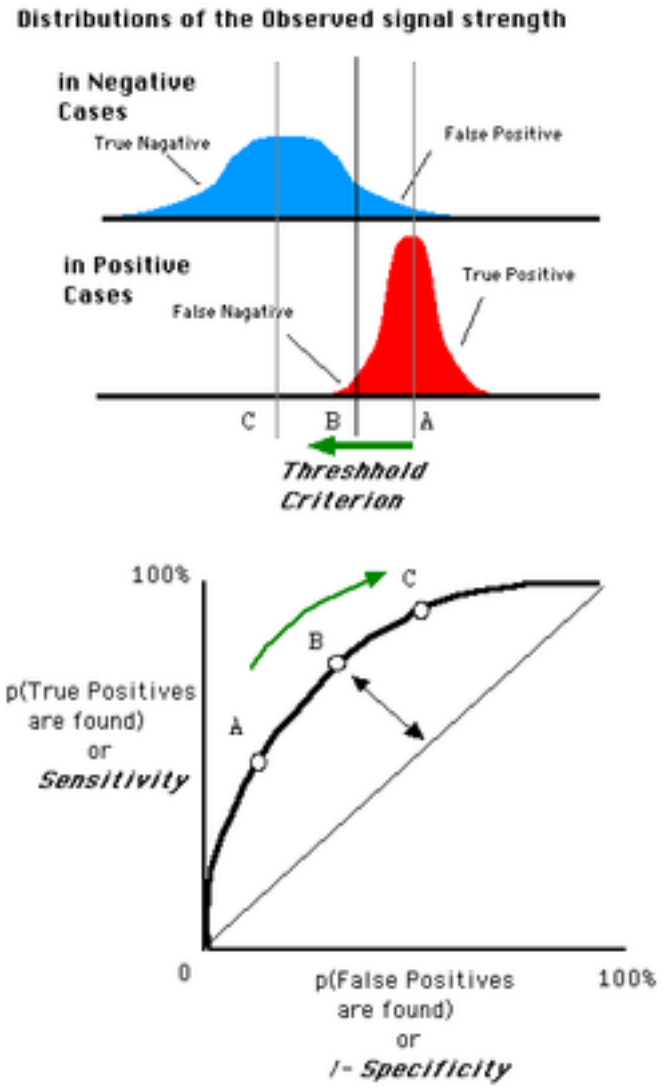


Figure 3: ROC-curve visualization (Fawcett, 2006)

3.7 One-Way Repeated Measures ANOVA

How to compare different samples with each other? What is the test that fits this comparison? How to interpret the results? The One-way repeated measures analysis of variance (ANOVA) tests whether three or more related samples significantly differ from one another on a chosen variable of interest. Related samples mean that the same unit of observation (the variable of interest) is measured on the same data points multiple times under slightly different occasions. The variable of interest should be continuous, normally distributed, and have a similar spread across the different samples. The samples should all measure this variable, have the same predictors, and exist at least five values per sample(Smaga, 2021).

ANOVA has the null hypothesis that the samples do not differ from each other. It tries to determine if one of the samples scores significantly different from the other samples on the variable of interest. For this ANOVA provides a F-statistic and a p-value. The F-statistic is the measure of difference and the p-value is the chance of scoring this F-statistic when there is no difference. If the

p-value is below alpha, which is chosen to be 0.05, the difference is statistically significant and can be trusted to not be due to chance. When this is the case, it means that at least two samples are significantly different, but further investigation is needed to determine which(Smaga, 2021). For this further investigation a paired sample T-test can be applied. With the paired sample T-test, a confidence interval for the mean difference is constructed. If this confidence interval does not contain 0, the null hypothesis of no difference is rejected(Miao & Chiou, 2008).

3.8 Conclusion

Out of this research, we take the following into account for the methodology. First of all, the variables described by Pleumpirom are comparable with the variables considered in this research. Secondly, we use a small data set, evaluate the decisions, and extrapolate the learned decision to new scenarios to create a bigger data set, like is done with BAIT. This more extensive data set will be used to train four classification algorithms: logistic regression, random forest classifier, support vector classifier, and gradient booster classifier. Gaussian naive Bayes and k-nearest neighbor algorithm are not applied in this research, since those are not suitable. Before we implement those classifiers, we apply stratified k-fold cross-validation to our data set to improve the performance of the classifiers. We use k=10 since ten folds perform the best according to the literature. The performance of the classifiers is visualized in ROC curves and compared to each other using the ANOVA statistical test on differences to investigate if the classifiers' differences are statistically significant. If that is the case, confidence intervals for mean differences are determined.

4 Methodology

To develop a model that can decide what decision is the best to make in the different scenarios, the classification algorithms need to be implemented. For this algorithm, the decisions of known scenarios and the variables of the scenarios are needed as input. This is what the model has to learn. This input is the scenarios with the decisions of the experts. Since this is not enough data for the model to train with, we first develop a decision tree that can make comparable decisions. This decision tree we use to generate a lot of decisions for randomly generated scenarios. Those scenarios are the training data and test data for the model. We use k-fold cross-validation to retrieve more reliable output data. The output data is the accuracy of the model and the importance of the different input variables. To choose which of those algorithms performs the best, those output variables are compared, and based on those the best-suited algorithm is determined.

4.1 Decision tree

With the answers to the questionnaire, a decision tree is developed that can decide for every single possible scenario. This tree a variable and based on the value checks the next variable until it is able to make the decision. For this decision tree, it is necessary to know the patterns in the answers to the questionnaire. The most important variables should be checked first and so all the variables should be checked until the whole decision tree is developed. The decision tree should provide for every scenario the correct decision for it to work.

This decision tree is developed such that it can decide for all the ten known scenarios what the right decision is. This is done only based on the variables such that it can also decide for every other scenario with the same variables but different values. The base decision is to defer maintenance unless it is mandatory or preferred to rectify immediately. This logic is also applied and visible in the decision tree.

4.1.1 Design of decision logic

When looking at the DM decision process, the first thing that is checked is the MEL. Those regulations decide for many defects that maintenance is mandatory or can be deferred, in those cases that will almost always be the decision to make. When one of those scenarios is not the case, but the MEL provides some extra conditions or restrictions that need to be maintained the situation changes. Then the situation is evaluated more specifically. In those scenarios, the defect can be deferred if the additional restrictions are met and the conditions can be fulfilled. This does mean that maintenance will only be chosen if it is mandatory or the circumstances are such that maintenance does not cause any trouble for the flight schedule, the aircraft is located in an airport with a service contract and all the resources needed for the rectification are available.

Taking all the expert opinions into account, we develop the decision tree. For this decision tree, all the variables of the questionnaire are used, except the maintenance tasks, since the experts did not value the influence of this variable as significant on the decision. With the other variables, the following logic is found: first, the MEL and conditions in it are looked up and when that requests maintenance or only flights with specific constraints that are hard to meet, the decision is to rectify first. When that is not the case, the other variables are checked and if all of them are positive for rectification the decision is also to repair. Otherwise, the decision is to defer the rectification to a later moment, so that we can schedule the maintenance. The whole decision tree can be seen in Figure 4.

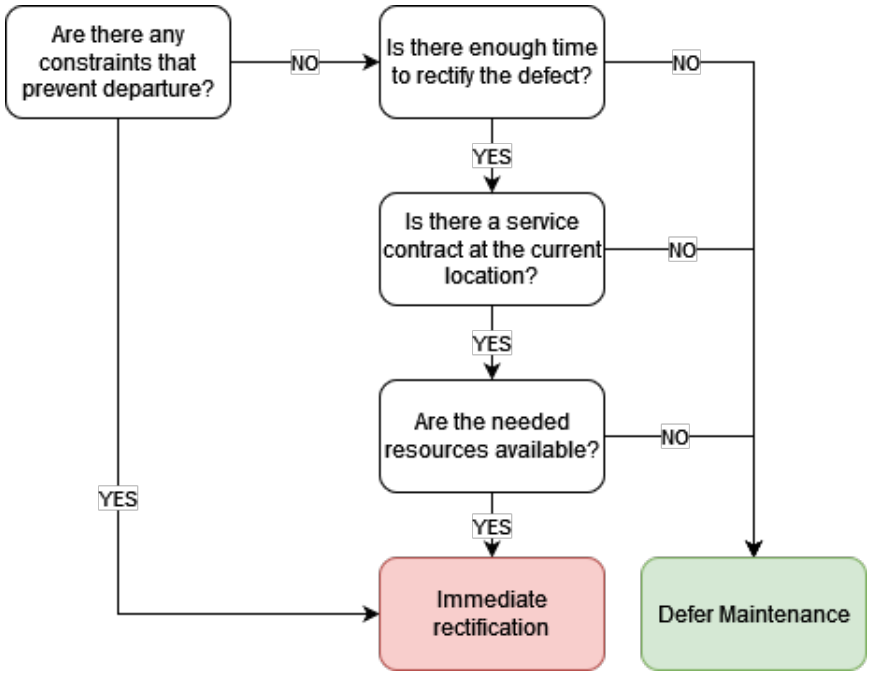


Figure 4: Developed decision tree

When this logic is applied to the cases provided to the experts, it matches all ten scenarios. This was the aim of the development of this decision tree as well, which is positive.

4.1.2 Data generation

This decision tree is used to generate a data set. The previous data set consists only of binary input variables. For the classification model, continuous input variables are desired as well to show that the model is able to handle those as well. The random generated scenarios do therefore have a slightly different form compared with the scenarios provided to the experts. The randomly generated scenarios consist of the variables in table 4.

Variable	Format	Description
Component	text	the six different components
Constraint	text	dark or mountains
Maintenance tasks	number between 0 and 100	hours until the first maintenance task
Flight schedule	time between 0 and 10 hours	time until the first scheduled flight
Current location	text	the four different locations
Resources	text	available or unavailable

Table 4: The variables in the data set

Those different variables also have an impact on the decision tree. The decision tree stays roughly the same, but some small things are added or changed. For the components, the locations, and the resources nothing has changed, and for the other variables the changes are the following:

- The specific constraints mentioned earlier in this paragraph are the following: for the air data computer: darkness and for the cabin pressurization system and the passenger oxygen system

a flying height constraint. For the other components, the constraints do not matter. The decision tree does now check those constraints for those components.

- The maintenance tasks are added to the data set, but still not considered by the decision tree.
- The decision tree still checks whether there is enough time to perform the maintenance. For each component a different amount of time is needed, varying from 1 hour and 45 minutes to 8 hours. Note that this is a fictional value, that for the remainder of this research is assumed to be true.

The decision tree adapted to those changes is visible in Figure 5:

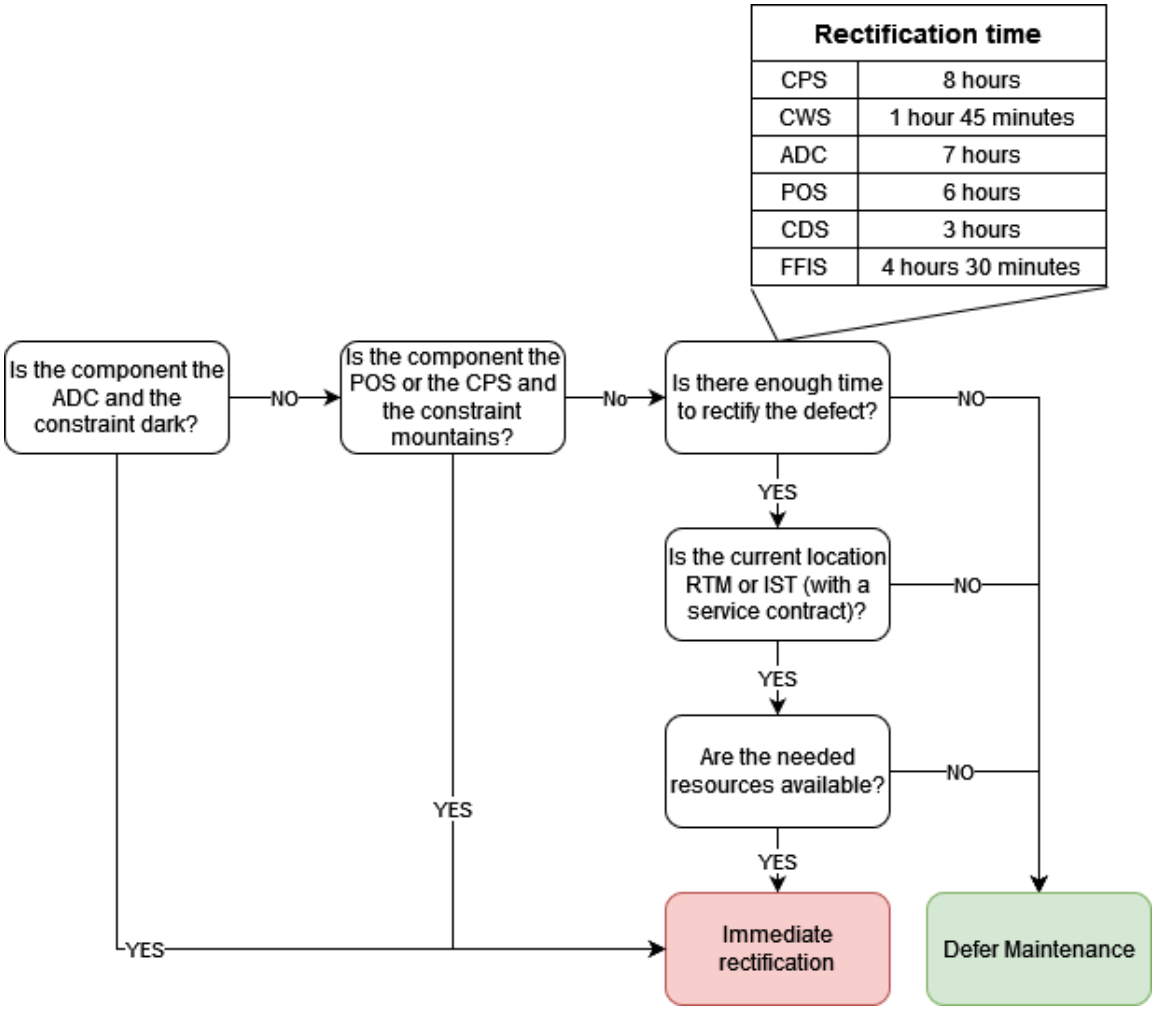


Figure 5: Decision tree used for the data generation

This decision tree is used to generate the data set. This consists of all the variables with a randomly assigned value and based on those values the decision tree makes a decision.

4.2 Classification model

To fit the classifiers to this data-set and compare their performance a Python model is developed. This model uses the following libraries: Pandas for data analysis[1], Matplotlib for plotting Figures[2], NumPy for mathematical calculations[3], and SciPy for statistical formulas[4]. This model can roughly be divided into three different parts, which are all explained in detail:

1. Preparation of the data
2. Implementation of the classification algorithms
3. Comparison of the classifiers

4.2.1 Preparation of the data

The generated data set from the decision tree is loaded. Of this data set the rows with missing values are deleted. The column with the decision is taken apart since this is the variable the classifiers have to learn. The other variables are used as the predictors. Before the classification algorithms are implemented, dichotomization is applied to the categorical predictors such that these predictors are split into binary predictors for all the possible values the categorical predictor could take.

Stratified k-fold cross-validation is applied, by using the python scikit-learn library[5] and with this, the data set is split into ten equal-sized folds that all contain a similar amount of both decision options. All those folds are once used to test the classifiers when the classifiers are trained on the other nine folds combined.

4.2.2 Implementation of the classification algorithms

The model fits and evaluates all the classifiers separately. The classifier is fit to the training set and tested on the test set. The accuracy of this test and the importance of all the features are stored. This is done ten times, such that all the folds are used as test sets and have provided their accuracy score and feature importance. From this the average accuracy of the classifier is calculated and a ROC curve is plotted, in which the accuracy is visualized together with the sensitivity and specificity. The average feature importance is plotted in a bar chart, with the range from the lowest to highest importance.

To implement the classification models in Python the scikit-learn library[6] is used. From this library, the *LogisticRegression, RandomForestClassifier, SVC,* and *GradientBoostingClassifier* modules are used to implement the classifiers. The input needed is the randomly generated data, divided into ten folds by applying stratified k-fold cross-validation, and the corresponding decision for each data input. This data is used to train all the different classification models. The output of this model is for each classification model: the accuracy of the decision per fold and the importance of the different predictors, which from now on is called the feature importance, per fold as well.

The model works as follows: the input data is split into ten samples, each exactly ten percent of the data set. Every sample will be used as test data when the model is trained on all the other

[1] Retrieved from: `https://pandas.pydata.org/`
[2] Retrieved from: `https://matplotlib.org/`
[3] Retrieved from: `https://numpy.org/`
[4] Retrieved from: `https://scipy.org`
[5] Retrieved from `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html`
[6] Retrieved from `https://scikit-learn.org/stable/index.html`

samples. Each classification algorithm looks into the training data and learns the decision back by determining the feature importance for every single predictor. Using this feature importance it can decide what to do in the remaining scenarios. So for all the test data, the classification algorithms decide what to do in those scenarios. Those decisions are then compared with the decision made with the decision tree. When all the scenarios are done, a confusion matrix is composed. This matrix provides an overview of the number of scenarios in which the right decision is made and the number of scenarios with wrong decisions. The average percentage of scenarios in which the model decided correctly over the five different cross-validations is the model's accuracy.

4.2.3 Comparison of the classifiers

The classifiers' accuracy exists of ten values since every fold provides an accuracy score. Those scores are tested to see whether a significant difference exists between the classifiers. For this, an ANOVA test is applied. The accuracy score of four different classifiers (LR, RF, SVC, GB) is the variable of interest. ANOVA can be applied, since the accuracy is a continuous variable, since it can take any value. The different values of the accuracy follow roughly the normal distribution since most values are around the average accuracy and the positive outlier is as likely as a negative outlier. The stratified k-fold selected the data points randomly. There is enough data because the accuracy exists of ten repetitions (the folds) per classifier. The accuracy scores are expected to vary similar for all the classifiers. The four samples are related since the accuracy score is measured on the same data points (folds) with different classifiers.

To test which classifiers differ, confidence intervals are constructed on the accuracy of all the possible combinations of classifiers. Based on the type of data set we have the test is a T-test. For this, the difference between the accuracy per fold should first be calculated. This is done by subtracting the accuracy of one of the classifiers from another classifier for each of the ten folds. This gives ten values and from those values, the mean ($\hat{p}$) and standard error (SE) are calculated. The mean is calculated with this formula: $\hat{p} = \frac{1}{10} * \sum_{n=1}^{10} p_n$ and the standard error with this formula: $SE = \sqrt{\frac{\sum_{n=1}^{10}(p_n - \hat{p})}{10-1}}$ where p_n is the difference for fold n, with a maximum of n = 10. Based on this mean and standard error the 95% confidence intervals (CI) are constructed. $CI = (\hat{p} - \frac{t*SE}{\sqrt{n}}, \hat{p} + \frac{t*SE}{\sqrt{n}})$, where $\hat{p}$ is the mean, SE is the standard error, n is the number of splits which is 10 and t is 2.262, which is the value from the t-table with $degrees of freedom = 10 - 1 = 9$ and $\alpha = 0.05$. The two classifiers have a significant different accuracy when this confidence interval does not contain 0(Miao & Chiou, 2008).

Since all the classifiers are tested on the same data set with the same predictors and all of them provide the accuracy of the prediction and the importance of the features, comparing those provides a lot of insight into the differences between the classifiers' performances. Therefore a graph with the feature importance of the classifiers is constructed. To compare the importance with each other, they must have the same representation. Both the random forest classifier and gradient boosting classifier provide importance as a percentage, such that the sum is equal to 1 (Breiman, 2001)(J. H. Friedman, 2002). However, logistic regression and the support vector classifier normalize the feature scores (LaValley, 2008)(Noble, 2006). Therefore the scores of logistic regression and the support vector classifier are converted into percentages as well. The current score of the features is a z-score. This means that the area to the left of this z-score is the probability and also the importance. The bigger this area, the more important the feature. Calculating a percentage score from this z-value is done by taking this probability as a percentage of the probabilities of all the features combined.

4.2.4 Experiments

The model runs several times for all the classification algorithms. Each time with a different size for the input data. For each run, the k-fold cross-validation is used with k = 10. It starts with 100 scenarios, split into ten equal parts of 10 scenarios that are all once used as test data and nine times as training data. After that, the model runs with 200, 500, 1000, 2000, and 5000 scenarios. Each time the data set is split into 10 folds. When the model has more test data it is expected to better learn the pattern and relation between the predictors. Therefore the accuracy is expected to increase when more data is used. Since the test sets become bigger as well it is also expected that the variance of the accuracy per fold will decrease.

4.2.5 Output

The output of those experiments is gathered and combined. This resulted in graphs with the average accuracy plotted out against the sample size, in which the classifiers can be compared. Next to that is for all the classifiers plotted how the accuracy and uncertainty change when the sample size increases. This output is displayed in the next chapter with the results.

5 Results

In this chapter, the results of the model are described. First of all, the average accuracy scores are provided and plotted. Secondly, those accuracy scores are analyzed and described per classifier. Based on that conclusions are drawn about the best-suited classifier and the desired sample size of the data for reliable results. Next to this, the feature importance is compared between the classifiers and other outstanding results are highlighted and discussed.

5.1 Comparison of the classifiers

The average accuracy is plotted in Figure 6 and written down in table 5. Out of those results, the logistic regression and the support vector classifier do not achieve the desired accuracy level of 95% as the random forest classifier and the gradient boosting classifier. Those two classifiers do already achieve this level with most of the folds when n=200, and with all the folds when n=500, where the logistic regression and support vector classifier only achieve a score of 83% for n=5000.

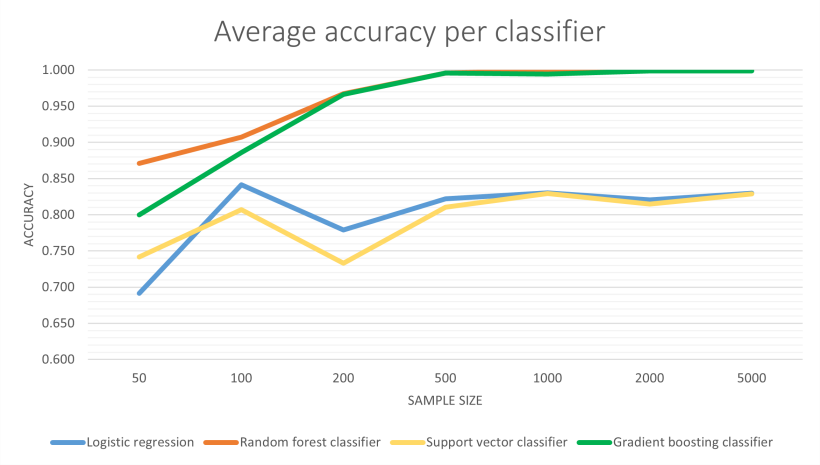


Figure 6: Average accuracy of the classifiers

Accuracy per classifier per sample size						
Classifier	**100**	**200**	**500**	**1000**	**2000**	**5000**
Logistic regression	0.842	0.779	0.822	0.830	0.820	0.830
Random forest classifier	0.907	0.967	0.996	0.999	1.000	1.000
Support vector classifier	0.807	0.733	0.811	0.829	0.815	0.829
Gradient boosting	0.886	0.967	0.996	0.994	0.999	0.999

Table 5: Average accuracy per sample size and classifier

5.2 Analysis of accuracy per classifier

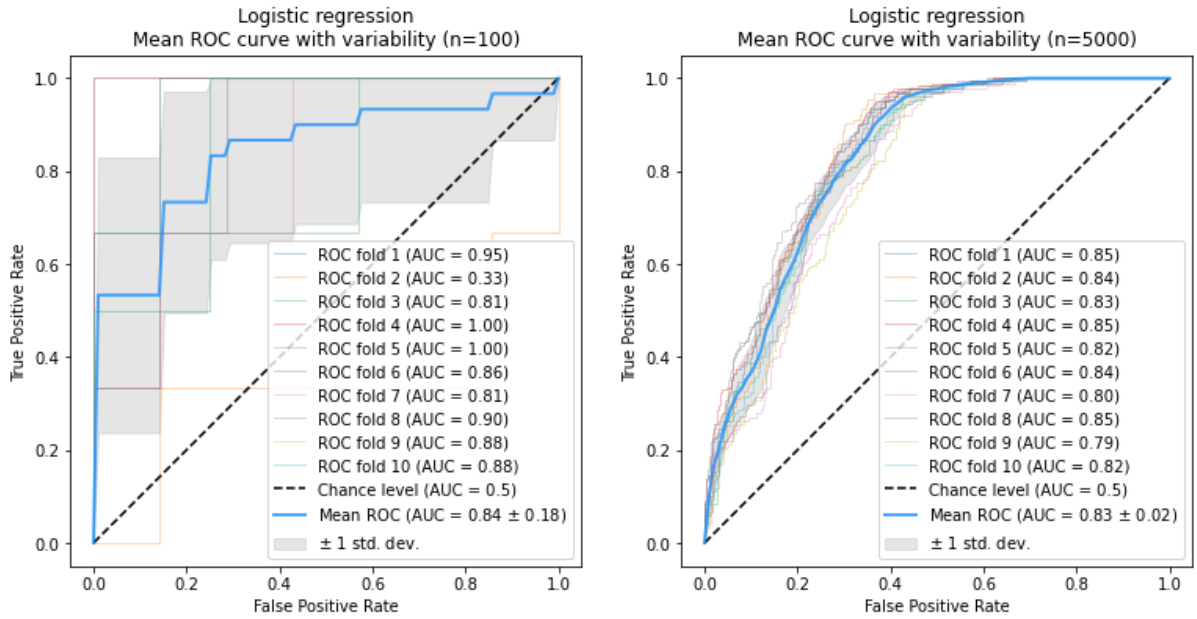
With all experiments done, the relation between the sample size and the accuracy of the classifier is analyzed. This is done by plotting the average accuracy score and the range from the lowest to the highest accuracy score per sample size. This way the improvement regarding the sample size is visualized per classifier. Next to that is for all the sample sizes the ROC-curve plotted with the accuracy per fold visualized.

5.2.1 Accuracy of logistic regression

Logistic regression achieves an accuracy of around 83% for all the sample sizes. The folds provide more similar results when they increase in size, which means that the variance decreases and the reliability of the accuracy increases. However, the accuracy itself does not seem to increase.

Figure 7: Accuracy of logistic regression with variance

With a sample size of 100, at least one fold achieves an accuracy of 100%, which is not unlikely since the test size per fold is only 10. This big variety between the folds is also visible in Figure 8a. What is interesting about the logistic regression is that the performance with 1000 observations has a bigger variance than with 500 and the variance with 2000 is much smaller. The difference between sample sizes 100 and 5000, visualized in Figure 8, is mostly the decrease of the standard deviation, but not an increase of the accuracy.



(a) ROC-curve with sample size 100

(b) ROC-curve with sample size 5000

Figure 8: ROC-curves of logistic regression

5.2.2 Accuracy of the random forest classifier

The random forest classifier already scores an accuracy of 90% on average for n=100 and this approaches 100% when more data is added. For n=500, the worst score already is 98.5%, which means that the random forest does learn the pattern almost perfectly with $n \geq 500$. Adding more data does not improve the accuracy score of the model.

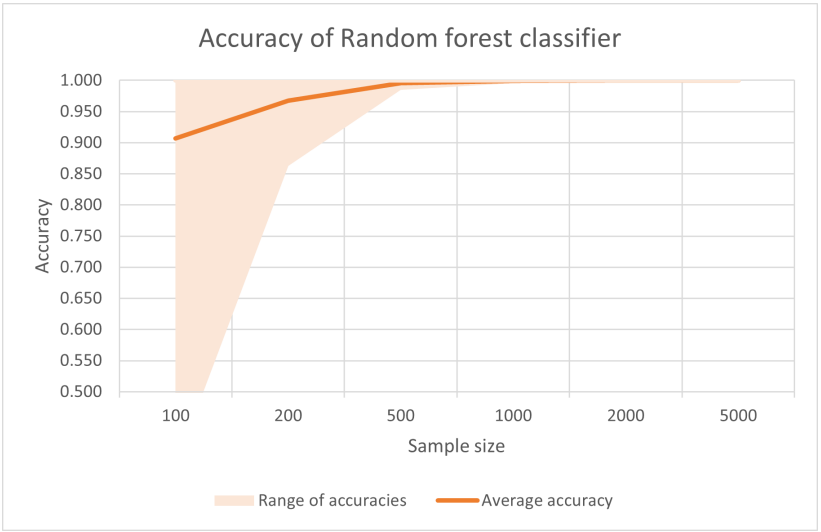


Figure 9: Accuracy of the random forest classifier with variance

The ROC-curve for n=500 can be seen in Figure 10 and there the curve is indeed in the top left corner. The fact that this classifier performs well is not surprising since this classifier uses decision

trees to search for patterns in a data set and this data set was generated with a decision tree as well.

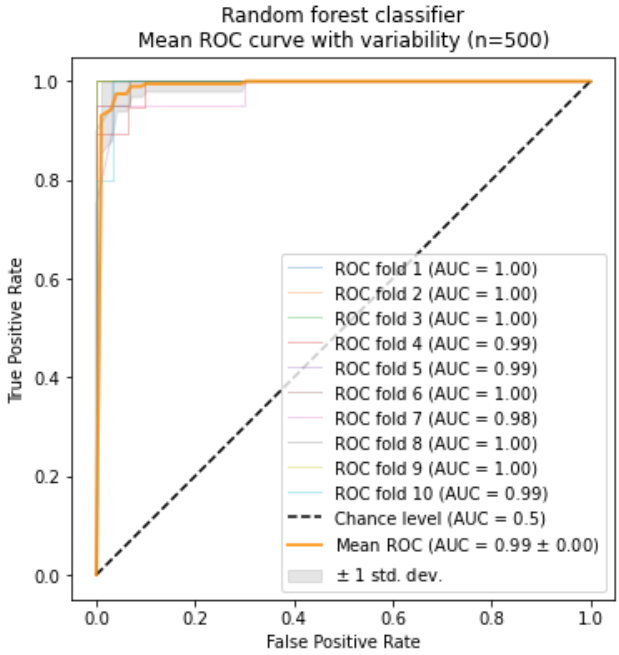


Figure 10: ROC-curve of logistic regression with sample size 500

5.2.3 Accuracy of the support vector classifier

The support vector classifier approaches an accuracy score of 83%, while the variance decreases for every increase in sample size.

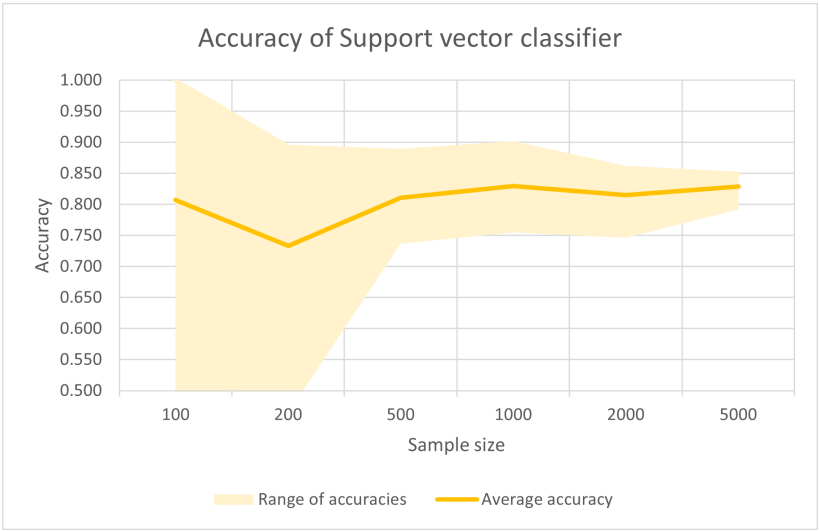
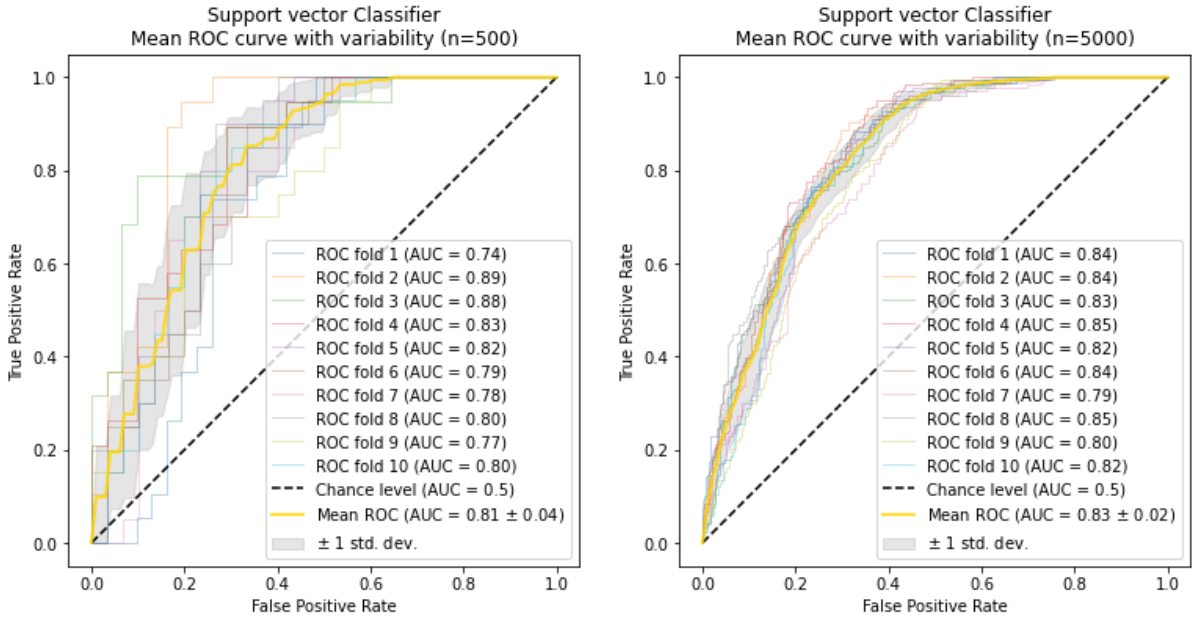


Figure 11: Accuracy of the support vector classifier with variance

The variance decreases a lot between n=200 and n=500, which means that the different folds

all approximate a similar accuracy score. When comparing the ROC curves of n=500 and n=5000, the biggest difference is the decrease in the deviation from 0.04 to 0.02. However, the accuracy does improve slightly as well from 0.81 to 0.83.



(a) ROC-curve with sample size 500

(b) ROC-curve with sample size 5000

Figure 12: ROC-curves of the support vector classifier

5.2.4 Accuracy of the gradient boosting classifier

The gradient boosting classifier has an accuracy score of 99.6% over the data set with size 500, which means that only two scenarios were evaluated wrong. This accuracy score means that this model is already able to learn the pattern in the data set with 500 entries. That this classifier performs well, is not surprising since this classifier uses decision trees to search for patterns in a data set and this data set was generated with a decision tree as well.

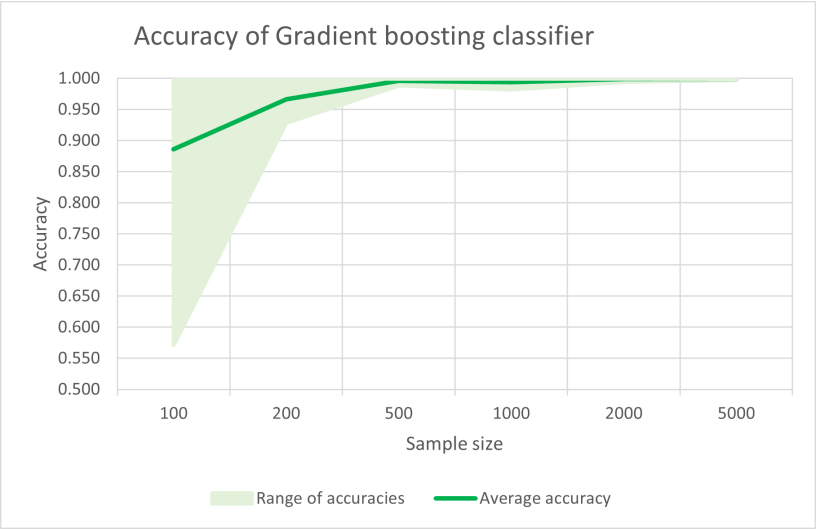
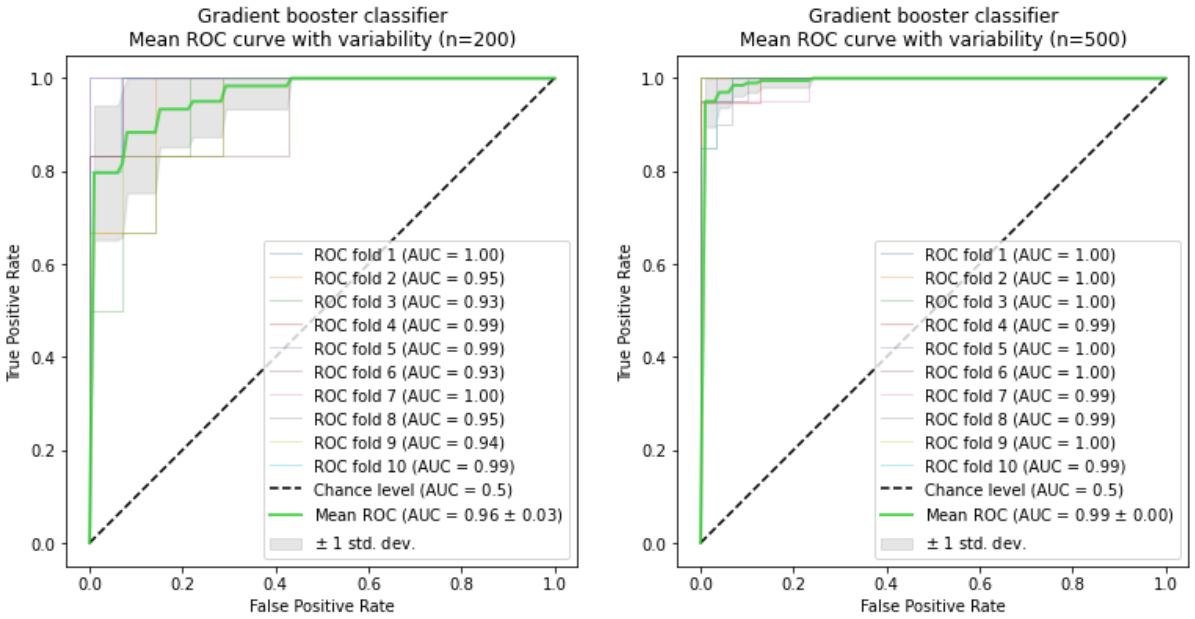


Figure 13: Accuracy of the gradient boosting classifier with variance

With 200 data points the variance is still significant and are not all the accuracy scores optimal as can be seen in Figure 14a, especially compared with Figure 14b.



(a) ROC-curve with sample size 200

(b) ROC-curve with sample size 500

Figure 14: ROC-curves of the gradient boosting classifier

5.3 Confidence intervals on the accuracy differences

In the Figures with the accuracy and their range is visible that the variance of the classifiers is big for sample sizes 100 and 200. Therefore it is hard to tell whether the differences between the classifiers are already significant. To investigate the differences an ANOVA test is performed for all

the sample sizes to test whether there are at least two samples that statistically significantly differ. The results of those ANOVA tests can be found in table 6.

Results of One-Way Repeated Measures ANOVA						
Sample size	**100**	**200**	**500**	**1000**	**2000**	**5000**
F-statistic	5.18	24.47	163.03	152.47	351.88	661.92
p-value	0.0059	0.0000	0.0000	0.0000	0.0000	0.0000

Table 6: ANOVA test statistics per sample size on a different accuracy between the classifiers

Since all the p-values are below 0.05, on a 95% confidence level all the null hypotheses of no differences are rejected. So, it can statistically be trusted that for all sample sizes, at least two classifiers have different accuracy. To investigate which classifiers differ, two-sample paired T-tests are conducted on all the possible combinations of classifiers for n=100 and n=500. Based on those T-tests, 95% confidence intervals on mean differences are constructed. Those can be found in Figure 15 and 16 for n=100 and n=500 respectively.

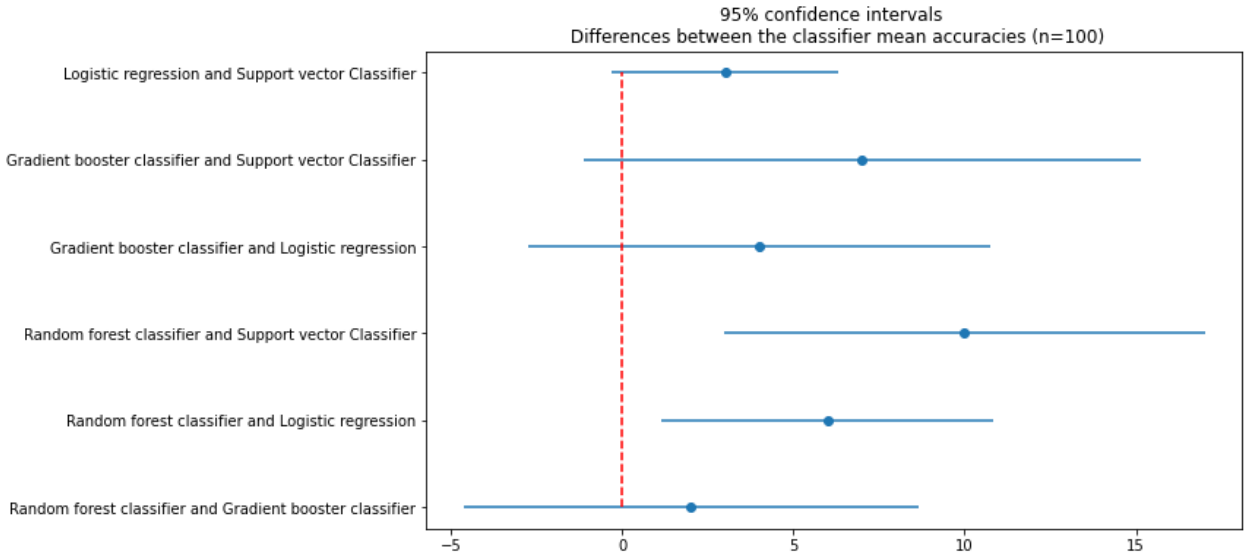


Figure 15: Confidence interval on accuracy difference for sample size 100

For n=100 only two confidence intervals do not contain 0. This means that with a significance level of 95%, the accuracy of the random forest classifier statistically differs from the accuracy of the support vector classifier and that of logistic regression. For all the other differences, on a 95% level of significance, the null hypothesis of no difference can not be rejected.

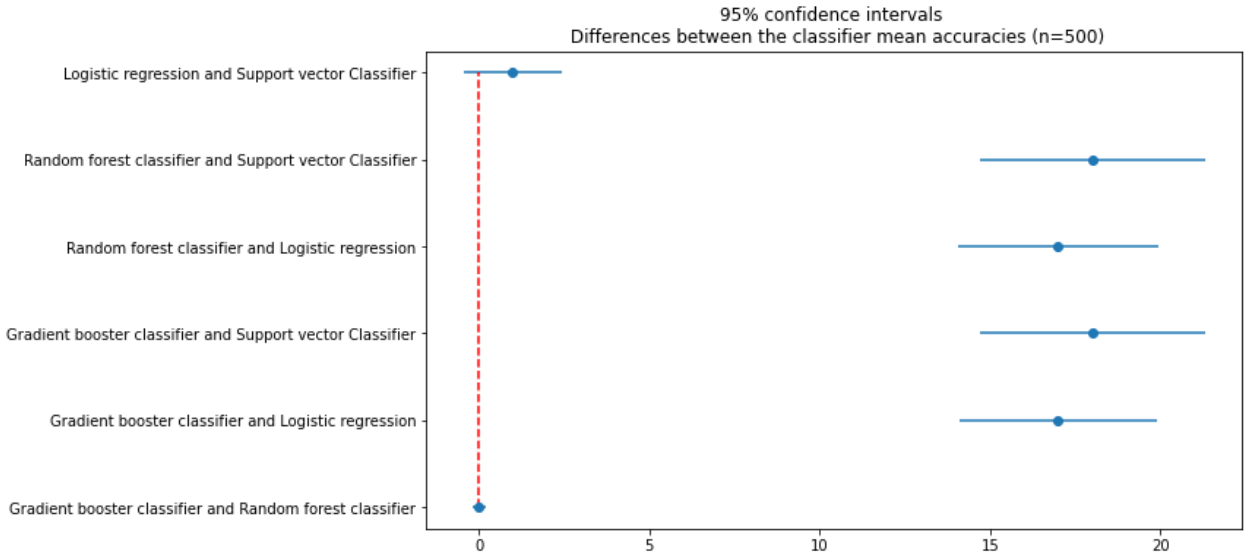


Figure 16: Confidence interval on accuracy difference for sample size 500

The confidence intervals do decrease when the sample size increases. This relation makes sense since the width of the interval is calculated by dividing by the square root of the sample size. Most of the differences are statistically significant on a 95% level of confidence. However, there are two exceptions which are both interesting. First of all, logistic regression and the support vector classifier do not statistically differ on a 95% level of confidence. In Figure 6 it can be seen that those classifiers indeed score similarly. The same is true for the random forest classifier and the gradient-boosting classifier. That those are not statistically different is expected for n=500 since both score almost the perfect accuracy score of 100%.

5.4 Feature importance comparison

Not only the accuracy is evaluated and considered, but also the features. Those results are compared between the classifiers as well. Since the random forest classifier and the gradient boosting classifier perform significantly better on the accuracy, they are evaluated in more detail. Since the sample size of 500 is recommended, we take the feature importance that is obtained with this sample size as well.

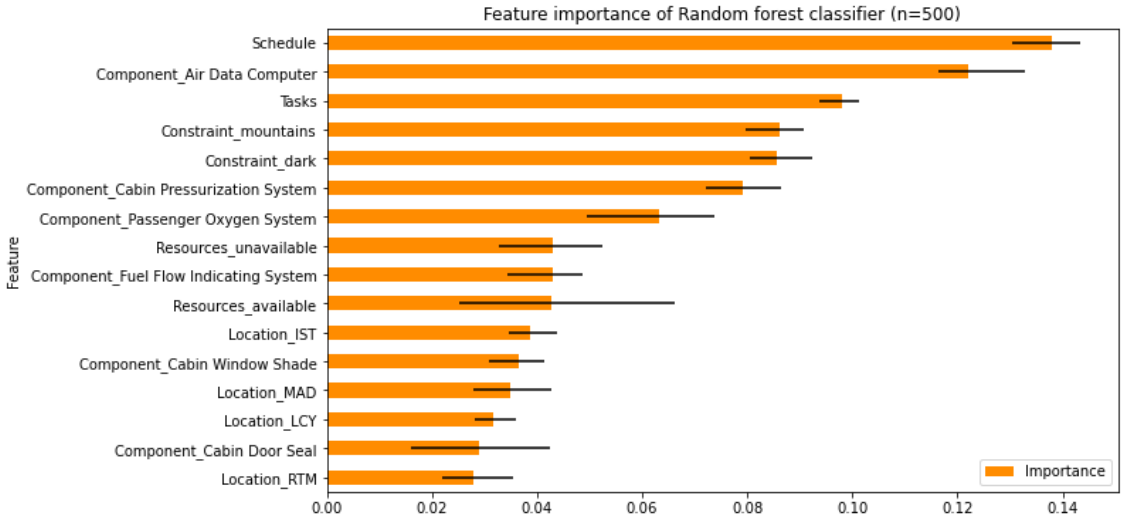


Figure 17: Feature importance of the random forest classifier with 500 scenarios

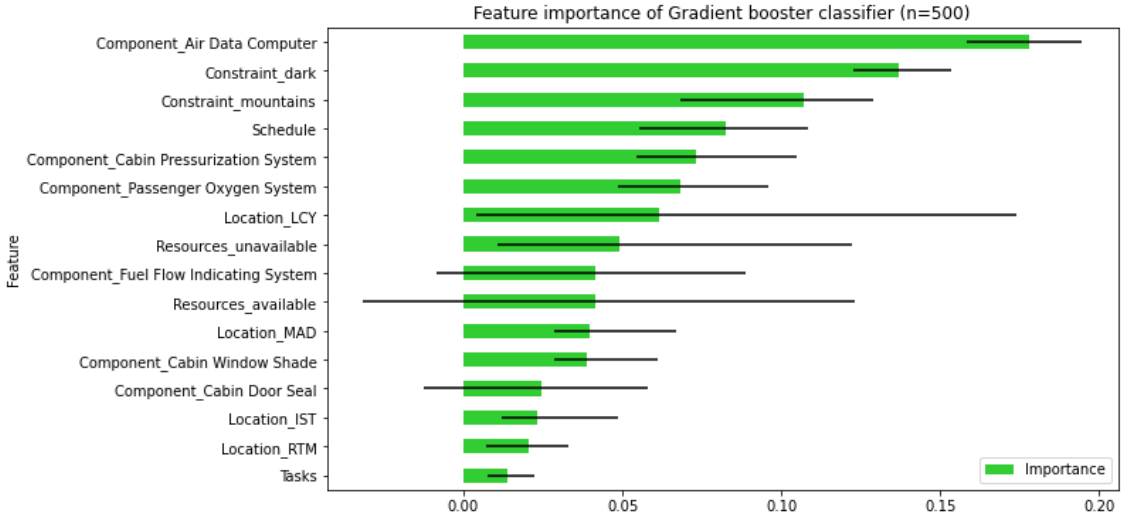


Figure 18: Feature importance of the gradient boosting classifier with 500 scenarios

All the features are also compared in one figure: Figure 19.

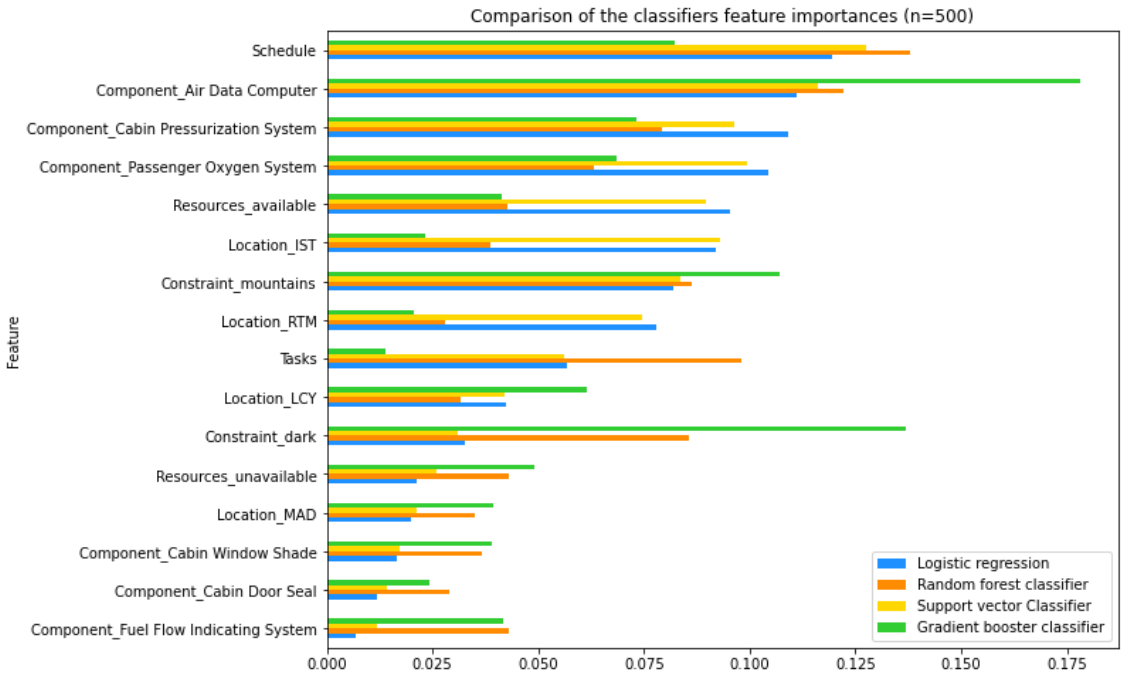


Figure 19: Feature importance of the classifiers with 500 scenarios

In Figure 17 and Figure 18 the feature importance can be seen that is determined by the random forest classifier and the gradient boosting classifier respectively. Since both classifiers achieve high accuracy, it was expected that the feature importance would be similar, but in Figure 19 can be seen that this is not the case. Some interesting observations that are made when observing the feature importance:

- 'Maintenance tasks' is the second most important feature for the RFC, but the least important for the GB. The original decision tree did not take this variable into account with deciding, which makes the result of the RFC very interesting.
- The locations are valued differently by the classifiers. RTM and IST should be similar, just as LCY and MAD. However, both classifiers did not value anything different here.
- The components are exactly ranked the same by both classifiers with the air data computer as the most important.
- Both classifiers rank the constraints as very important, which makes sense since those require maintenance in certain cases.
- The locations, resources, and flight schedule should all be valued similarly according to the original decision tree.

When the sample size is increased to 5000, those differences become smaller, as can be seen in Figure 20, 21, and 22.

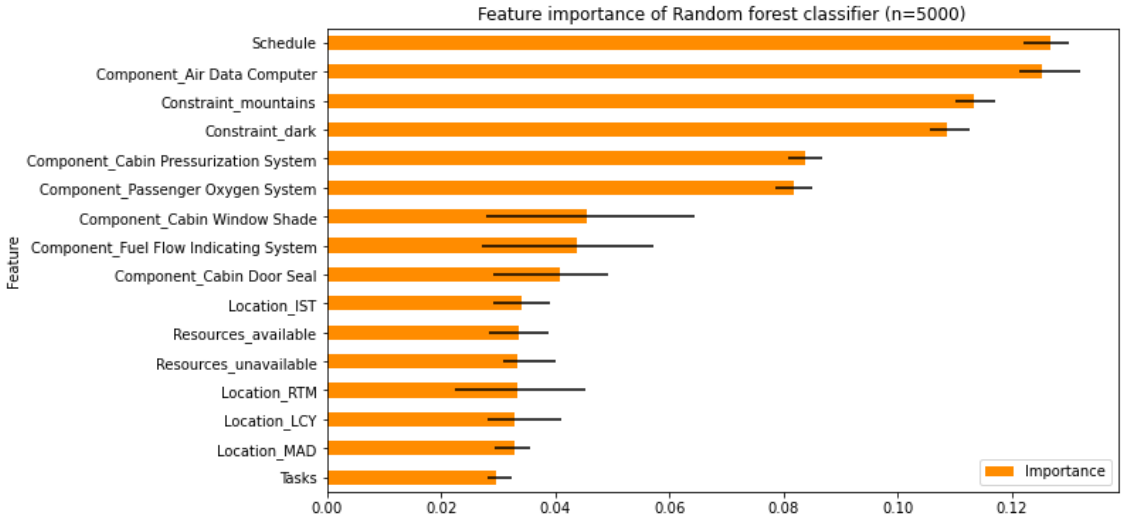


Figure 20: Feature importance of the random forest classifier with 5000 scenarios

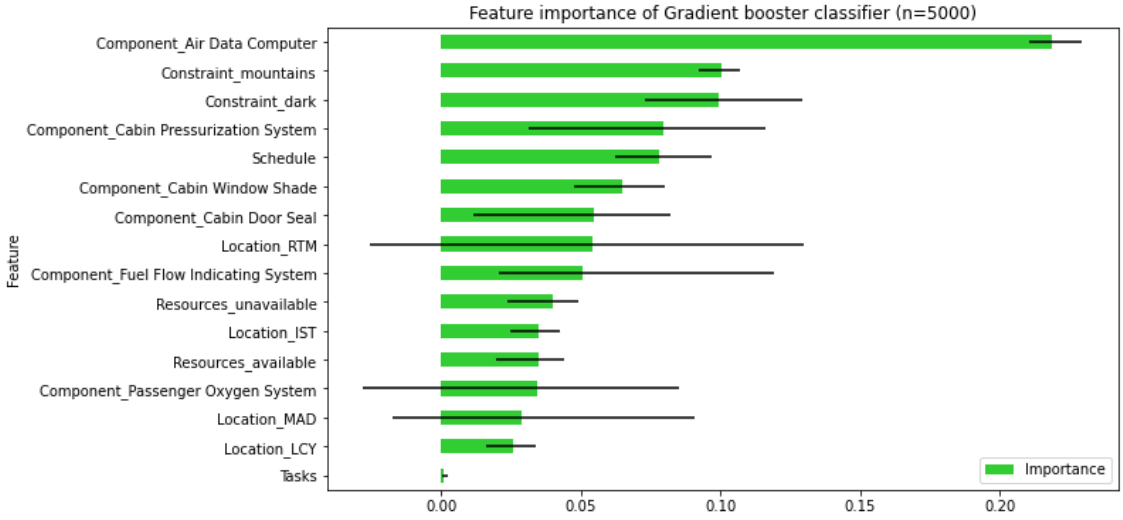


Figure 21: Feature importance of the gradient boosting classifier with 5000 scenarios

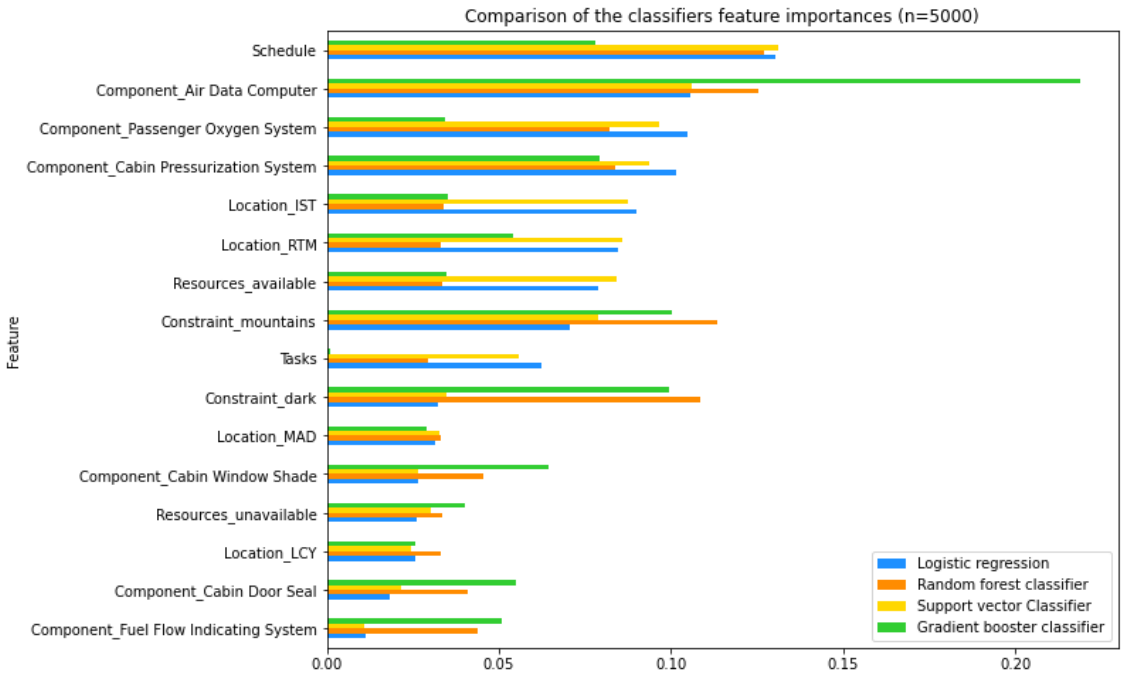


Figure 22: Feature importance of the classifiers with 5000 scenarios

- 'Maintenance tasks' is according to the RFC now also the least important variable, but still has some influence.
- RTM and IST are now similar and more important than the other two locations, just as LCY and MAD. Both classifiers did find some relation here.
- The components are now ranked differently by the gradient boosting classifier. However, this change is bad according to the decision tree that is used to generate the data set.
- Both classifiers still rank the constraints as very important, which makes sense since those require maintenance in certain cases.
- The GB and RFC both value the locations, resources, and flight schedule more similarly, which is according to the original decision tree.

Another interesting observation is that with both n=500 and n=5000, the feature importance of the SVC and LR are very similar, which can be seen in Figures 19 and 22. This might explain the similar accuracy. However, further research is needed to investigate why the features are so similar for those classifiers, without achieving high accuracy.

6 Conclusion

The conducted research was about proofing the concept of catching a decision situation with a decision tree and applying classification algorithms to learn back this decision tree and to investigate the amount of data different classifiers need in order to learn this decision tree. To prove this concept and construct this model, data was gathered for the development of this bot that can decide in defect management what decision to make. Chapter 3 describes the research in defect management that is done and based on that research ten different cases are developed, such that they will provide a large amount of data despite the limited size. This small amount of cases is then provided to experts in defect management. Their decisions are the norm for the model. This small set of data is then analyzed in order to find a logical pattern between the variables and the decision that holds for all those cases. The experts' responses made clear that the variable of 'upcoming maintenance tasks' does not need to be considered for the DM decision and the variable 'constraints' is most important. With the MEL, the current location and available resources, the constraints, and the flight schedule the experts were able to decide in each scenario what to do. It can be concluded that with those variables known, it is possible to make defect management decisions.

This logic pattern is described in a decision tree such that this tree can decide in randomly generated scenarios that are similar to the scenarios provided to the experts. With this decision tree data sets of different sizes are generated. Per sample size is analyzed with four classifiers how accurately they can predict the made decisions based on the different variables.

Using stratified k-fold cross-validation with k=10, and the random forest classifier or the gradient boosting classifier an accuracy of 99.6% can be obtained with a data set with 500 data points. The accuracy of those two classifiers is not statistically different from each other. With n=200 an accuracy of around 96.7% can be achieved, but this is, with a 95% confidence level, not significantly higher than the desired level of 95%. Therefore, a sample size of at least 500 for the random forest classifier and gradient boosting classifier is recommended to achieve this desired accuracy level for sure. Logistic regression and the support vector classifier do not achieve an accuracy of 95%, but score max 84%, and are therefore not recommended to use for a data set like this one. The null hypothesis that those classifiers are different can not be rejected on a confidence level of 95%.

Scoring an accuracy of 99.6%, interestingly enough, does not mean that the classifiers have learned back the decision tree perfectly. The analysis of the feature importance of these models provided the insight that some of the variables were incorrect, according to the decision tree used to generate the data, estimated to be very important for the decision-making process. When the sample size was increased to 5000 the features were much more in line with the decision tree. So, in order to learn the decision tree itself back perfectly, 5000 is not even enough.

6.1 Discussion

This research has delivered insights for the NLR on how to develop the CAMO bot. First of all, this research shows the importance of data for developing a machine-learning tool. Without data, it is not possible to construct a decision tree or to train a classification algorithm.
It is possible to take a limited set of cases to gain a lot of data if the cases are constructed in a way that a lot of insight is gathered into the decision-making process. In this small set of data, a logic pattern can be found that is able to decide based on the different variables what the decision is, but it is hard to tell how reliable this pattern is since the data set is such limited.
Based on a logic pattern, it is not complex to generate a big data set with decisions. This generated bigger data set can be used to test different classifiers and to find the amount of data that is needed

in order to find significant conclusions, but not to find straight facts, since some assumptions are made in order to develop the final model.
Another insight provided by this research is that different classifiers use different methods and will therefore not all be as accurate and therefore suited for each different scenario. Besides, if a complex decision tree is used to construct a data set, it is hard for classifiers, even for ones that use decision trees, to learn this decision tree back. This is visible in the fact that the feature importance of the RF and the GB do not align with the decision tree used to generate the data, however, the bigger the training set becomes, the better the approximation of the decision tree is.

6.2 Recommendations

First of all, it is recommended to continue with the development of the CAMO-bot that is able to tell which maintenance tasks need to be performed and when. For the development of this bot, more expert opinions and decisions are needed to develop a reliable model. Thus is recommended that the NLR contact an external party like an airline or a CAMO organisation that has this data available. For the development of the decision-making part, a decision tree is ideal. Therefore, it is recommended to map out the decision logic of defect management. This decision tree should be able to decide in every scenario like the developed decision tree in this research but then applied to every possible defect on every possible component. In order to investigate the importance and influence of different variables, machine learning provides useful insight into the importance of these factors and their influence on the defect management decision process. This knowledge could be applied where possible to prevent unnecessary delays, such as having spare parts for the most crucial components. Therefore, sharing the results with parties like CAMO organizations is recommended as well.

6.3 Limitations

Experts decided about each component only twice, such that it is not possible on this data set to tell the influence per component in detail, which limits the results about the feature importance of the components. The variable resources are the staff, equipment, and docking station at the airport. For this research, it is assumed that these are all available, or not. In reality, this is not binary but more complex. The rectification times of the components, used to construct a more complex decision tree, are fictional, these values could be adapted to the real values. Next to that, rectification time is not a definite value, but a statistically deviating value. The only two options for this research were to rectify the defect immediately or to defer the rectification. However, mostly this is not the case, since the best option might be to fly with some constraints to the next airport where the rectification should be done immediately. This is for now not an option in this simplified case. Because of these simplifications and some assumptions, the reliability of the developed decision tree is unknown, the only thing we know is that it can decide correctly on the ten known scenarios provided to the experts. Because of this, nothing can be said about the coefficients of the variables and the accuracy scores.

6.4 Further research

For now, only six components and two different kinds of locations are taken into account. This is recommended to be expanded. The current model considers some relation between variables and shared impact on the decision, but there might be more (complex) relations which is recommended to investigate. This model can decide whether to defer or to rectify, but after that decision different decisions have to be made and tasks have to be performed. The bot should be able to provide those as well and that will be helpful since that will also be different for every scenario. These tasks should

be researched and combined with the defects in order to improve maintenance operations. This model might be possible to use in maintenance management as well, but this has to be researched.

References

Ben-Hur, A., & Weston, J. (2010). A user's guide to support vector machines. *Data mining techniques for the life sciences*, 223–239.

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5–32.

Brink, R. (2021). *Interview with rob brink about airworthiness.*

De Florio, F. (2016). Chapter 2 - airworthiness. In F. De Florio (Ed.), *Airworthiness (third edition)* (Third Edition ed., p. 5-6). Butterworth-Heinemann. Retrieved from `https://www.sciencedirect.com/science/article/pii/B9780081008881000021` doi: https://doi.org/10.1016/B978-0-08-100888-1.00002-1

Demir, F. (2022). 14 - deep autoencoder-based automated brain tumor detection from mri data. In V. Bajaj & G. Sinha (Eds.), *Artificial intelligence-based brain-computer interface* (p. 317-351). Academic Press. Retrieved from `https://www.sciencedirect.com/science/article/pii/B9780323911979000138` doi: https://doi.org/10.1016/B978-0-323-91197-9.00013-8

Fawcett, T. (2006). An introduction to roc analysis. *Pattern recognition letters*, *27*(8), 861–874.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational statistics & data analysis*, *38*(4), 367–378.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine learning*, *29*, 131–163.

Gama, J., & Brazdil, P. (1995). Characterization of classification algorithms. In *Portuguese conference on artificial intelligence* (pp. 189–200).

Ghojogh, B., & Crowley, M. (2023). *The theory behind overfitting, cross validation, regularization, bagging, and boosting: Tutorial.*

Gong, D. (2022). Top 6 machine learning algorithms for classification. *Accessed: Aug*, *4*, 2022.

Hoo, Z. H., Candlish, J., & Teare, D. (2017). What is an roc curve? *Emergency Medicine Journal*, *34*(6), 357–359. Retrieved from `https://emj.bmj.com/content/34/6/357` doi: 10.1136/emermed-2017-206735

Keller, J. M., Gray, M. R., & Givens, J. A. (1985). A fuzzy k-nearest neighbor algorithm. *IEEE transactions on systems, man, and cybernetics*(4), 580–585.

Kim, E., Ji, H., Kim, J., & Park, E. (2022). Classifying apartment defect repair tasks in south korea: a machine learning approach. *Journal of Asian Architecture and Building Engineering*, *21*(6), 2503–2510.

Koornneef, H., Verhagen, W. J., & Curran, R. (2020). A decision support framework and prototype for aircraft dispatch assessment. *Decision Support Systems*, *135*, 113338.

Kumar, S., Sharma, M., Muttoo, S., & Singh, V. (2022). Autoclassify software defects using orthogonal defect classification. In *International conference on computational science and its applications* (pp. 313–322).

LaValley, M. P. (2008). Logistic regression. *Circulation*, *117*(18), 2395–2399.

Lin, L., Yue, W., & Mao, Y. (2014). Multi-class image classification based on fast stochastic gradient boosting. *Informatica*, *38*(3).

Miao, W., & Chiou, P. (2008). Confidence intervals for the difference between two means. *Computational Statistics & Data Analysis*, *52*(4), 2238-2248. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0167947307002940` doi: https://doi.org/10.1016/j.csda.2007.07.017

NLR. (2022). *Mission, vision & strategy.* Retrieved from `https://www.nlr.org/about-us/mission-and-vision/`

NLR. (2023). *Geschiedenis.* Retrieved from `https://www.nlr.nl/nlr-100-jaar/`

Noble, W. S. (2006). What is a support vector machine? *Nature biotechnology*, *24*(12), 1565–1567.

Ochodek, M., Hebig, R., Meding, W., Frost, G., & Staron, M. (2022). Chapter 8 recognizing lines of code violating company-specific coding guidelines using machine learning. In *Accelerating digital transformation: 10 years of software center* (pp. 211–251). Springer.

Pleumpirom, Y., Amornsawadwatana, S., et al. (2012). Multiobjective optimization of aircraft maintenance in thailand using goal programming: A decision-support model. *Advances in Decision Sciences*, *2012*.

Prusty, S., Patnaik, S., & Dash, S. K. (2022). Skcv: Stratified k-fold cross-validation on ml classifiers for predicting cervical cancer. *Frontiers in Nanotechnology*, *4*, 972421.

Smaga, Ł. (2021). One-way repeated measures anova for functional data. In T. Chadjipadelis, B. Lausen, A. Markos, T. R. Lee, A. Montanari, & R. Nugent (Eds.), *Data analysis and rationality in a complex world* (pp. 243–251). Cham: Springer International Publishing.

Ten Broeke, A., Hulscher, J., Heyning, N., Kooi, E., & Chorus, C. (2021). Bait: a new medical decision support technology based on discrete choice theory. *Medical Decision Making*, *41*(5), 614–619.

Wong, T.-T., & Yeh, P.-Y. (2019). Reliable accuracy estimates from k-fold cross validation. *IEEE Transactions on Knowledge and Data Engineering*, *32*(8), 1586–1594.

Appendix

Appendix A: Developed scenarios

Case number	Component	Current Location	Upcoming flights	Upcoming Maintenance Tasks	Extra information
1	21-30-01 Cabin Pressurization System	RTM Resources: available	Current time: 23:05 7 hours 55 minutes at RTM RTM-MAD-IST: 7:00 – 13:35 1 hour and 55 minutes at IST IST-RTM-LCY: 15:30 – 21:46 9 hours and 14 minutes at RTM	Oil filter element and SOAP Sample (in 87 flight hours) 300-hour check (in 287 flight hours) 1200-hour check (in 587 flight hours)	Flights over the mountains of Spain (max elevation: 8000 feet) and the Balkan (9500 feet).
2	25-20-04-01 Cabin Window Shade	IST Resources: available	Current time: 21:46 9 hours and 14 minutes at IST IST-RTM-LCY-RTM: 7:00 – 13:16 2 hours and 14 minutes at RTM RTM-MAD-IST: 15:30-22:05 8 hours and 55 minutes at IST	Oil filter element and SOAP Sample (in 7 flight hours) 300-hour check (in 107 flight hours) 1200-hour check (in 1007 flight hours)	
3	34-10-01 Air Data Computer	IST Resources: Unavailable at IST, but available at RTM	Current time: 10:04 45 minutes at IST IST-MAD: 10:49 – 14:30 1 hour at MAD MAD-LCY-MAD-RTM: 15:30 – 23:05 7 hours and 55 minutes at RTM	Oil filter element and SOAP Sample (in 87 flight hours) 300-hour check (in 287 flight hours) 1200-hour check (in 287 flight hours)	This flight is in the winter, so it gets dark early
4	34-10-01 Air Data Computer	LCY	Current time: 23:00 8 hours at LCY LCY-IST-LCY: 7:00 – 14:30 1 hour at LCY LCY-IST-LCY: 15:30-23:00 8 hours at LCY	Oil filter element and SOAP Sample (in 87 flight hours) 300-hour check (in 287 flight hours) 1200-hour check (in 887 flight hours)	It is summer, so it stays light for a long enough time to perform the scheduled flights before darkness sets in.
5	35-20-01 Passenger Oxygen System	MAD	Current time: 23:00 8 hours at MAD MAD-LCY-MAD-RTM: 7:00 – 14:35 55 minutes at RTM RTM-IST-MAD: 15:30 – 23:00 8 hours at MAD	Oil filter element and SOAP Sample (in 7 flight hours) 300-hour check (in 107 flight hours) 1200-hour check (in 707 flight hours)	Flights over the mountains of Spain (max elevation: 8000 feet) and the Balkan (9500 feet).
6	52-10-02 Cabin Door Seal	RTM Resources: available	Current time: 10:04 45 minutes at RTM RTM-LCY-RTM: 19:19 – 21:46 9 hours and 14 minutes at RTM RTM-MAD-IST: 7:00 – 13:35 1 hour and 55 minutes at IST	Oil filter element and SOAP Sample (in 7 flight hours) 300-hour check (in 107 flight hours) 1200-hour check (in 1007 flight hours)	
7	73-30-01 Fuel Flow Indicating System	RTM Resources: unavailable	Current time: 21:46 10 hours and 14 minutes at RTM RTM-MAD-IST: 7:00-13:35 1 hour and 55 minutes at IST IST-RTM-LCY-RTM: 15:30-21:46 10 hours and 14 minutes at RTM	Oil filter element and SOAP Sample (in 87 flight hours) 300-hour check (in 287 flight hours) 1200-hour check (in 587 flight hours)	
8	35-20-01 Passenger Oxygen System & 21-30-01 Cabin Pressurization System	IST Resources: available	Current time: 10:23 45 minutes at IST IST-LCY: 11:08 – 14:31 59 minutes at LCY LCY-IST-LCY: 15:30 – 23:01 7 hours and 59 minutes at LCY	Oil filter element and SOAP Sample (in 7 flight hours) 300-hour check (in 107 flight hours) 1200-hour check (in 107 flight hours)	Flights over the mountains the Balkan (max elevation: 9500 feet).
9	52-10-02 Cabin Door Seal & 21-30-01 Cabin Pressurization System	MAD	Current time: 23:00 8 hours at MAD MAD-LCY-MAD-RTM: 7:00 – 14:35 55 minutes at RTM RTM-IST-MAD: 15:30-23:00 8 hours at MAD	Oil filter element and SOAP Sample (in 7 flight hours) 300-hour check (in 107 flight hours) 1200-hour check (in 407 flight hours)	Flights over the mountains of Spain (max elevation: 8000 feet) and the Balkan (9500 feet).
10	35-20-01 Passenger Oxygen System & 52-10-02 Cabin Door Seal	LCY	Current time: 23:00 8 hours at LCY LCY-IST-LCY: 7:00 – 14:30 1 hour at LCY LCY-IST-LCY: 15:30 – 23:00 8 hours at LCY	Oil filter element and SOAP Sample (in 87 flight hours) 300-hour check (in 287 flight hours) 1200-hour check (in 887 flight hours)	Flights over the mountains of the Balkan (max elevation9500 feet).

Figure 23: Developed scenarios

Appendix B: Responses questionnaire

Scenario	What is the decision you should make in this situation?	How much did the variables below influence the decision made? [Minimum Equipment List]	How much did the variables below influence the decision made? [Upcoming maintenance tasks]	How much did the variables below influence the decision made? [Upcoming flights]	the variables below influence the decision made? [Current location of aircraft (outstation or	How much did the variables below influence the decision made? [Availability resources]
1	Immediate rectification	Big influence	No influence	Big influence	Small influence	Small influence
	Immediate rectification	No influence	No influence	No influence	Small influence	Big influence
	Defer maintenance	Big influence	No influence	Big influence	No influence	No influence
	plan troubleshooting in RTM to identify cause	Small influence	No influence	Big influence	Big influence	Big influence
	Immediate rectification	Big influence	No influence	No influence	Big influence	Big influence
2	Defer maintenance	Big influence	Big influence	No influence	Big influence	No influence
	Defer maintenance	Small influence	Small influence	Big influence	No influence	No influence
	Immediate rectification	Big influence	No influence	No influence	No influence	No influence
	Immediate rectification	Big influence	Small influence	Small influence	Big influence	Big influence
	Immediate rectification	Big influence	No influence	No influence	No influence	No influence
3	due to MEL limitations.	Big influence	No influence	Big influence	Big influence	Small influence
	Immediate rectification	Big influence	No influence	Big influence	Big influence	Big influence
	RTM to MAD, swap ADC in MAD, continue	Big influence	No influence	Big influence	Big influence	Big influence
	Immediate rectification	Big influence	No influence	Big influence	Small influence	Small influence
	Immediate rectification	Big influence	No influence	No influence	No influence	No influence
4	night at LCY.	Big influence	No influence	Big influence	Big influence	
	Immediate rectification	Big influence	No influence	No influence	Big influence	
	Defer maintenance	Big influence	No influence	Small influence	Big influence	
	Defer maintenance	Big influence	No influence	Big influence	Small influence	
	Defer maintenance	Big influence	No influence	Big influence	Small influence	
5	Defer maintenance	Big influence	Big influence	Big influence	Big influence	
	Defer maintenance	Big influence	No influence	Small influence	Small influence	
	MEL	Big influence	No influence	No influence	No influence	
	Defer maintenance	Big influence	No influence	Big influence	No influence	
	Immediate rectification	Big influence	No influence	Small influence	Small influence	

Figure 24: Responses to questionnaire

6	Defer maintenance	Big influence	Big influence	Big influence	No influence	No influence
	Immediate rectification	Big influence	No influence	Big influence	Big influence	Big influence
	Defer maintenance	Big influence	No influence	Big influence	No influence	No influence
	in RTM	Big influence	Big influence	Small influence	Big influence	Big influence
	Defer maintenance	Big influence	No influence	Small influence	Small influence	Small influence
7	Defer maintenance	Big influence	No influence	Big influence	Small influence	Big influence
	Defer maintenance	Big influence	No influence	Big influence	No influence	No influence
	Immediate rectification	Big influence	No influence	No influence	No influence	No influence
	Defer maintenance	Big influence	No influence	Big influence	Small influence	Big influence
	Defer maintenance	Big influence	No influence	No influence	No influence	No influence
8	Pressurization System	Big influence	Small influence	Big influence	Big influence	Big influence
	Pressurization System	Big influence	No influence	Small influence	Big influence	Big influence
	Defer maintenance	Big influence	No influence	Small influence	Small influence	Small influence
	Immediate rectification of both components	Small influence	Small influence	Big influence	Big influence	Big influence
	Immediate rectification of both components	Big influence	No influence	No influence	Small influence	Big influence
9	RTM and rectification of both components	Big influence	Small influence	Big influence	Small influence	
	Immediate rectification of both components	Big influence	No influence	Big influence	No influence	
	Defer maintenance	Big influence	No influence	Small influence	Big influence	
	Defer maintenance	Big influence	Small influence	Big influence	Big influence	
	Defer maintenance	Big influence	No influence	Small influence	Small influence	available or not
10	only	Big influence	No influence	Big influence	Big influence	
	only	Big influence	No influence	Big influence	No influence	
	Unpress anyway and accept any delays	Big influence	No influence	Small influence	No influence	
	Defer maintenance	Big influence	Small influence	Big influence	Big influence	
	only	Big influence	No influence	No influence	Small influence	

Figure 25: Responses to the questionnaire (part 2)