TUM Data Innovation Lab

MTU Aero Engines AG

Understanding and Modeling of Aircraft Engine Shop Visits

Helge Brügner
Moritz Heimbächer
Céline Marquet

Matthias Berger
Peter Schlick

Prof. Dr. Massimo Fornasier (Supervisor)
Dr. Ricardo Acevedo Cabra (Project Lead)

July 17, 2018

Summer Semester 2018

Technische Universität München

Abstract

This project was conducted as part of the Data Innovation Lab (DI-Lab) in the summer semester 2018. This project was split into two use cases. First, the goal was to predict shop visit intervals of aircraft engines (i.e. the time between two subsequent shop visits). A shop visit dataset was used to train random forest model for this prediction and up to 73% accuracy was achieved. The second use case was the prediction of engine module work scopes during shop visits and for simplicity, only a single engine module was considered. This use case was further split into two tasks, the first being the prediction of work scopes for individual engines and the second being the prediction of fleet work scopes. For the first, models were developed with around 70% prediction accuracy. It was attempted to solve the second task using Kaplan-Meier survival estimates. However, it was concluded that this model is not sufficient, and further research directions were proposed.

We would like to thank the supervisors at MTU for the cool project, the great mentoring and their patience with us.

Furthermore, we would like to thank Ricardo for organizing the DI-Lab and giving us the opportunity to apply our skills in practice.

Please refer to Table A.1 for the list of contributions in this report.

Contents

List of Figures

List of Tables

Abbreviations

ADS-B Automatic Dependent Surveillance – Broadcast.

CSN Cycles Since New.
CSO Cycles Since Overhaul.
CSV Cycles Since Visit.

EIS Entry Into Service.

FHA Flight Hour Agreement.
FR24 Flightradar24.

HSR Hot Section Refurbishment.

IAE International Aero Engines.
ICAO International Civil Aviation Organization.
IQR Interquartile Range.

LLP Life-limited Part.
LPT Low Pressure Turbine.

OOB Out-of-Bag.

SV Shop Visit.

T&M Time and Material.
TSN Time Since New.
TSO Time Since Overhaul.

XGB Extreme Gradient Boosting.

Chapter 1

Introduction

This project is concerned with the prediction of aircraft engine maintenance intervals and damage patterns. The following chapter thus introduces basic concepts of aircraft engines and engine maintenance concepts in Aircraft Engines and Engine Maintenance. Subsequently, the datasets that are used in this project are introduced in Datasets.

1.1 Aircraft Engines and Engine Maintenance

To understand the mechanical concepts used in this report, this section first introduces the technology behind aircraft engines. Additionally, details of the V2500 engine program are provided. Subsequently, engine maintenance concepts are introduced.

1.1.1 Aircraft Engines

As described in Ackert (2011, p. 3), engines used for commercial aircraft are turbofan engines. The main role of an engine is generating thrust. This is done by the fan which accelerates the inlet air. The core engine compresses some of the inlet air and mixes it with fuel. In this way a high temperature exhaust gas is generated to power the acceleration of the fan.

In this project the IAE V2500 engine is considered because MTU Aero Engines has many maintenance contracts for these engines. It is built by International Aero Engines (IAE), a group of four different aircraft engine manufacturers. The engine mostly powers the Airbus A320 family which consists of short- to medium-range passenger airliners (International Aero Engines, 2018).

There are five different models of the V2500 engine as can be seen in Table 1.1. Most of the operating V2500 are A5 models. Some of them were updated to the S1 model in the past 10 years. By now only the models S1 and S2 are still in production.

When a new model enters the market there are often some early troubles and engines have to visit the shop earlier than planned. These problems can be solved for later engines of the model during production which results in longer periods until the first major shop visit.

An engine has a specific thrust rate. The newer models usually have higher thrust rates which causes more power. The engine itself consists of several modules. This makes the maintenance easier because one broken module can usually be repaired or exchanged without touching other parts of the engine (Ackert, 2011, p. 5).

In Figure 1.1 a turbofan engine is shown with its modules. In the high pressure section (also hot section) between the High Pressure Compressor and the High Pressure Turbine the deterioration is the highest and the exposure of parts in this section is expensive (Ackert, 2011, p. 6). Therefore it is an important part of maintenance. There are some parts in the modules which cannot be

Engine Model	Year of first flight	Thrust rates
A1	1988	25K
A5	1993	23K-30K
D5	1995	25K-28K
SelectOne (S1)	2008	24K-33K
SelectTwo (S2)	2015	27K-33K

Table 1.1: Engine Models (International Aero Engines, 2018)

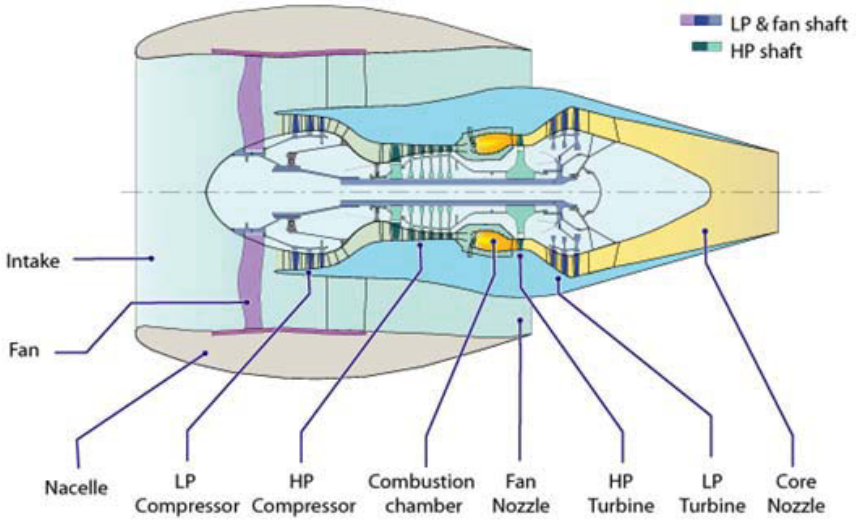


Figure 1.1: Turbofan Engine and its modules (LP: Low Pressure, HP: High Pressure) (Ackert, 2011, p.3)

contained if they fail. These are called Life-limited Part (LLP) because they have to be replaced after a certain on-wing time (Ackert, 2011, p. 8). For the V2500, the Life-limited Part (LLP)s are generally limited to 20000 cycles. A cycle is one flight, that is the time from start to landing. It is limited to cycles and not to on-wing time because the crucial part of deterioration happens during start and landing.

1.1.2 Engine Maintenance

When an engine is removed from the wing the airliner gets a backup engine. A removal can have different causes. There could have been some unforeseen event causing a performance degradation or serious damage. Many removals are planned in advance by the operator to do a regular inspection or replace some worn out modules. Further reasons can be that the engine is taken out of service or the leasing contract between owner and operator ends.

A run of an engine is the time between two core shop visits. A core shop visit is also called Hot Section Refurbishment since at least one module of the hot section is exposed. The first run of an engine often consists of more cycles than subsequent runs (also called mature runs) (Ackert, 2011,

p. 16).

An airline (also called operator) usually has several engines which are called the fleet of the operator. It consists of two engines per airliner and some spare engines. An airline plans shop visits in advance. There are different possible strategies. A run cannot be longer than 20000 cycles since then the Low Pressure Turbine (LPT)s have to be replaced. Engines which have short stage lengths (flight time of one cycle) can do up to 20000 cycles in one run. Most common are 2- or 3-stop strategies where the 20000 cycles are reached after two or three runs. The strategy depends on some more criteria. For example, the region in which the engine is operating in. If airports are near or in deserts the sand can have a huge impact on the deterioration. Of course these strategies not always work. After an unforeseen damage the engine has to be repaired and the operator has to adjust its strategy.

Maintenance costs are approximately 10-15% of the operating expenses of an airline. 35-40% of these maintenance costs are engine-related (Ackert, 2011, p. 9). As this is not a marginal part of their costs it is important for the operators to know how much money they have to plan on maintenance.

There are different contract types for maintenance. In Time and Material (T&M) contracts, the customer pays for the actual cost of labor and material within the defined scope of maintenance work. In Flight Hour Agreement (FHA) contracts, pays a fixed amount per hour flown by the engine (Ackert, 2011, pp. 25).

1.2 Datasets

This section describes the available data sets. First, the Shop Visit Dataset is introduced, which is the primarily used data source in this project. Secondly, a description of the Flightradar24 Dataset is provided that is later used to extract hubs for operators found in the shop visit dataset.

1.2.1 Shop Visit Dataset

The shop visit data set (SVData) consists of information about shop visits of V2500 engines in the time period from 1992 to 2017. It contains several thousand data points. A full list of parameters is provided in Appendix B.

1.2.2 Flightradar24 Dataset

The second dataset stems from the global flight tracking service Flightradar24 (FR24). It uses the Automatic Dependent Surveillance – Broadcast (ADS-B) technology to track flights (Flightradar24, 2018). Aircraft broadcast their signals, which can then be received by other aircraft or ground receivers (Richards, O'Brien, & Miller, 2010). FR24 uses a network of about 17000 such ground receivers to track more than 150000 flights per day. Past flight records are stored (Flightradar24, 2018).

The dataset has been purchased from Flightradar24 (FR24) and contains flights from January through April 2018. Each flight in the dataset has features such as airline name, origin and destination airport names and their ICAO codes, airplane type and engine type.

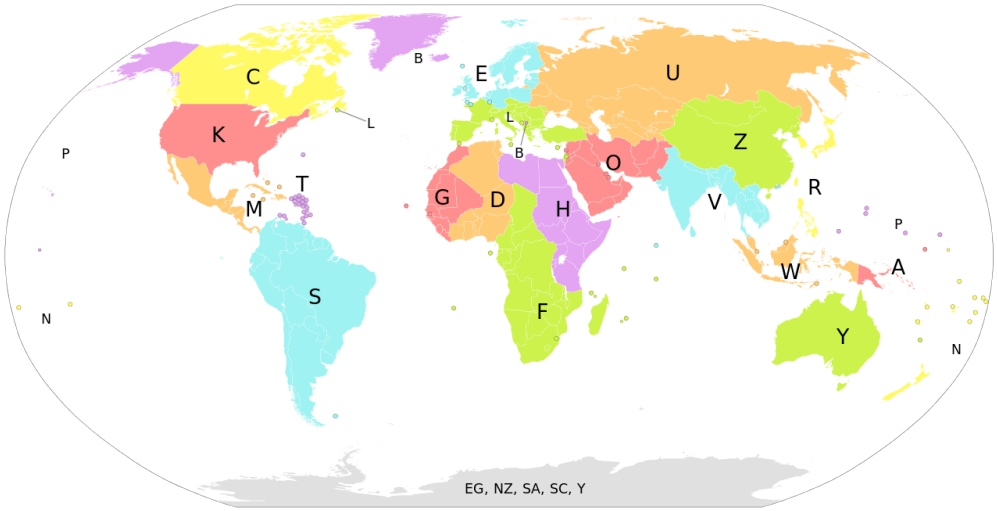


Figure 1.2: First letter of ICAO codes on a world map (Wikimedia Commons, 2015)

ICAO airport codes are published by the International Civil Aviation Organization (ICAO) (ICAO, 2018). The codes consist of four letters to uniquely identify airports and other aviation-related facilities. The letters of the code are increasingly specific and provide geographical context. The first letter represents a region as shown in Figure 1.2. Continents are assigned more than one first letter and a single first letter can be part of multiple continents. Europe, for instance, has codes E, L, B and U, where the latter is also part of Asia. The combination of first and second letter usually identifies a country. For example, German airports start with the letters ED (Wikipedia, n.d.).

Chapter 2

Data Science Methodology

This chapter introduces fundamental data science concepts that are necessary to understand this report. There are multiple ways to describe the process of developing a comprehensive data analysis model with machine learning, but generally it is seen as an iterative, non-linear process. To ensure a reasonable course of action and to reach the predefined goals in a timely manner, the use cases are structured according to the setup in Peng and Matsui (2016, p. 5ff.), which can be seen in Figure 2.1.

Figure 2.1: Epicycles of Data Analysis with five main activities:
- Stating and refining the question - Exploring the data - Building formal statistical models - Interpreting the results - Communicating the results (Peng & Matsui, 2016, p. 5)

2.1 Exploratory Data Analysis

Based on the first step in Figure 2.1, the beginning of the task is exploratory data analysis. The American statistician John W. Turkey established the concept of "exploratory data analysis" and describes it as detective work, where the researcher uses graphs to find the unexpected. Turkey is seen as the founder of several graphical analysis tools like the boxplot. In fact, he stated that "exploratory data analysis [...] does not need probability, significance or confidence" (Turkey, 1997). Following that statement, the methods for the primary analyses of the shop visit variables are descriptive and visual and will be described in the next subsection.

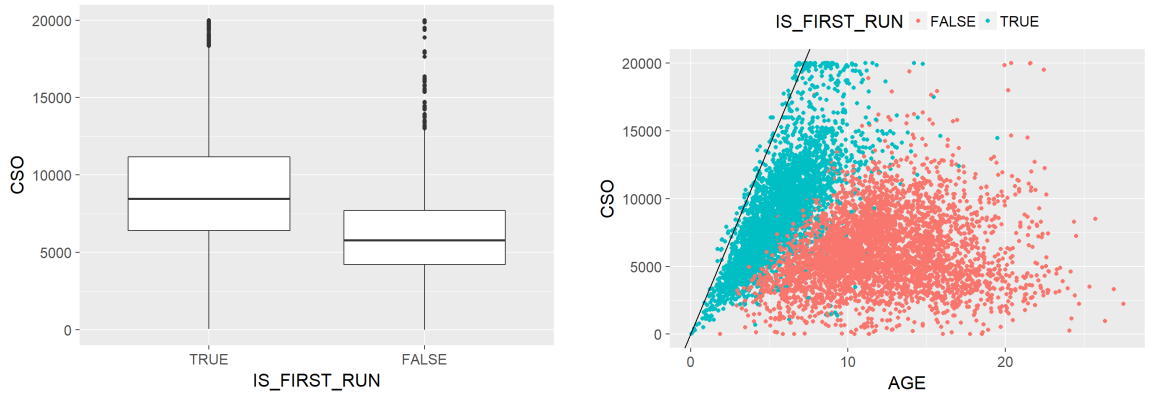
Description of Stochastic Distributions

Frequency distributions of variables can be detected by the graphic analysis tools mentioned above. The associated location parameters describe the characteristics of a distribution with numerical values. Commonly known parameters to describe the data center are the arithmetic mean, the median and the mode. However, for a comprehensive description of a distribution, further information on the dispersion of the data is needed. This can be numerically represented by "p-quantiles".

For the proportion p of data larger / smaller than x_p, $0 < p < 1$:

$$\frac{amount(x - values \leq x_p)}{n} \geq p \quad \text{and} \quad \frac{amount(x - values \geq x_p)}{n} \geq 1 - p \tag{2.1}$$

The 25% quantile is also called the "lower quartile" ($x_{0,25}$) and the 75% quantile ($x_{0,75}$) is also called the "upper quartile". The Interquartile Range (IQR) as defined by ($IQR = x_{0,75} - x_{0,25}$) is a measure of dispersion that allows conclusions about the distribution. At a low value, the data is close to the median, while a high value indicates a large variance (Fahrmeir, Künstler, Pigeot, & Tutz, 2010, p. 29ff.).



(a) Example of a box plot: Two groups (on x-axis). The medians of the groups differ and so does the IQR (the height of the box). Both plots have outliers as marked by the points above the whiskers

(b) Example of a scatter plot where the points are colored according to a third variable. Clear clusters are visible for the colored variables. The turquoise clusters possesses an upper bound (indicated by the black line)

Figure 2.2: Examples of box and scatter plots

Scatter Plot

A scatter plot shows the relationship of two variables (x_i, y_i), $i = 1, \ldots, n$ by determining the position of an element in a cartesian (x, y)-coordinate system. Often, the points are drawn as circles at their respective position. The aim is to recognize a dependency structure or correlation of the variables through graphical patterns. These may include clusters or linear structures (Fahrmeir et al., 2010, p. 128). An example is shown in Figure 2.2b.

Box Plot

A box plot is a visual tool to describe the distribution and quantiles of variables on an (at least) ordinal scale. An example box plot comparing two groups is shown in Figure 2.2a. Common elements of box plots include (Turkey, 1997, p. 52):

- A scale (parallel to the main axis of the box plot)
- A box between the lower and upper quartiles
- A horizontal line marking the median and the 1.5-fold IQRs respectively
- A connection (whisker) from the box to the horizontal lines at the extreme values
- Points for values outside of whiskers (outliers)

Density Plot

Similar to boxplots and histograms, a density plot is a tool to visualize a distribution. Clearly, in histograms the width of the bars is crucial for the look of the graph. The idea of density plots is that the histogram is smoothed such that individual data points contribute to different classes. This allows to produce a smooth function rather than a bar plot and is particularly useful in the case that the variable on the x axis is numeric. Normalizing the integral of the function to a value of 1 results in an estimate of a probability density function (Heumann & Schomaker, 2017, p. 29f.). An example of a density plot is shown in Figure 2.3.

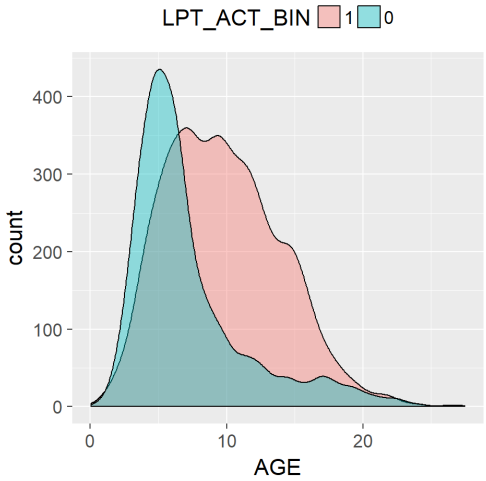


Figure 2.3: Density plot showing two groups scaled by their respective size. The blue group is concentrated around 5 on the x-axis, the red group is more spread and peaks at a value around 10

2.2 Machine Learning Models

In 1959, Arthur Samuel defined the term "machine learning" as a field of research that gives computers the ability to learn without being programmed explicitly. Machine learning is a part of data analytics and describes the development of mathematical models and algorithms that can learn from data and make predictions based on it. The practical implementation of machine learning takes place with the help of algorithms that can be grouped according to their learning processes.

There are two main groups of algorithms that learn either supervised or unsupervised. The latter is also known as data mining, which will not be presented in this documentation as this project is based on supervised learning models. In such models, a function is developed by learning associations or regularities with the help of variable pairs (namely (x_i, y_i)). These given pairs consist of an input x_i and an output y_i, which in statistical literature are classically referred to as an independent (also explanatory) variable and dependent (also explained) variable. The goal of supervised learning is to recognize a basic structure that allows to predict new, unknown dependent variables after training the algorithm (Suthaharan, 2016).

2.2.1 Random Forest

Leo Breiman (2001) introduced the extension of traditional decision tree models, referring to works by Ho (1995), Amit and Geman (1997) and Dietterich (2000), among others. Since then, the algorithm has been experiencing great popularity (Hastie, Tibshirani, & Friedman, 2009, p. 587): Strata Conference (Howard & Bowles, 2012) named Random Forest the "most successful universal algorithm of today" and for a long time the Random Forest was by far the leading algorithm in *Kaggle*[1] competitions (Goldbloom, 2015).

The model is based on the "CART-algorithm" (Breiman, Friedman, Stone, & Olshen, 1984) and "bagging" (short for bootstrap aggregation), which was was also developed by Breiman (1996) and is a method of combining multiple predictions from regression or classification models using the mean or most common classification. The goal is to increase the stability and accuracy of the prediction, i.e. to reduce the dependency on the structure of the training data set and thus counteract overfitting.

One prediction f_i, $i = 1, \ldots, M$, is made for M bootstrap samples[2] and the total prediction $\hat{F}$ looks as follows:

$$\hat{F}(X) = \frac{1}{M} \sum_{i=1}^{M} f_i \tag{2.2}$$

For more information regarding bagging and CART refer to Breiman (1996) and Breiman et al. (1984).

The Random Forest approach deviates somewhat from the original procedure of bagging. The variance of a bagged prediction depends on the ρ correlation of the individual trees:

$$Var\Big[\hat{F}(X) = \frac{1}{M} \sum_{i=1}^{M} f_i\Big] = \rho\sigma^2 + \frac{1-\rho}{M}\sigma^2 \tag{2.3}$$

As M increases, the second segment disappears, but the first one always depends on the correlation. Thus, the correlation of the individual trees limits the success of bagging. Random Forest's approach is to reduce the correlation by randomization in each split and thus exploit the variance reduction through bagging independent of the correlation.

This method is called "Random Subspace Method". In addition to bootstrap samples, it also randomizes the selection of predictive variables at each node. Instead of all p input variables, only $m \leq p$ random variables are considered potential split variables. The smaller m is chosen, the lower the correlation between two trees. The following recommendations are given in Breiman, 2001 regarding the size of m:

- for classification: $m = \sqrt{p}$ with a minimal node size of 1
- for regression: $m = \frac{p}{3}$ with a minimal node size of 5.

[1] Leading Platform for Data Science Competitions

[2] Based on repeated random sampling with replacement from the observed data

Random Forest prediction $\hat{F}_{rf}$ is analogous to equation 2.3 and calculated by the mean of all regression trees T_i:

$$\hat{F}_{rf}(X) = \frac{1}{M}\sum_{i=1}^{M} T_i \tag{2.4}$$

In contrast to the CART-algorithm there is no pruning on the random forest and the trees T_i are fully grown.

The optimal number of M bootstrap samples can be determined by the Out-of-Bag (OOB) error. For sampling with replacement one can generally say that about a third of the data points are not chosen. To calculate the OOB error, a prediction for each observation of the training data $z_j = (x_j, y_j)$ is made, using only the trees $T_{withoutj}$. For the construction of these trees z_j was not used. The OOB error is the average error rate for all z_j applied to $T_{withoutj}$ and is similar to the error obtained with k-fold cross-validation.

In general, it can be more difficult to understand the decision rules of a Random Forest compared to other methods. One way to evaluate the contributions of each explanatory variable to the goodness of fit is to calculate the importance of the variable based on permutation of the OOB observations. To determine the importance of a variable z_j, the prediction quality of a tree T_b is determined with the OOB observations.

Afterwards, a permutation of the OOB observations of z_j follows, along with the determination of the prediction quality of T_b on these permuted observations. The average quality decrease through permutation across all trees is the measure of the variable importance of z_j.

2.2.2 Extreme Gradient Boosting

The idea of Extreme Gradient Boosting (XGB) is based on a paper by Jerome H. Friedman (2001) and was first introduced by Tianqi Chen. The algorithm started to gain popularity after Chen introduced it in a Kaggle challenge and with contributions from other developers, he later published packages in multiple languages and a paper on the topic (see xgboost (n.d.), Chen and Guestrin (2016)).

"Boosting" is a nonlinear, adaptive method that tries to combine the output of many weak qualifiers to produce one strong outcome. In that sense it is similar to bagging, but the approach is fundamentally different. A weak classifier is one that only slightly outperforms an outcome achieved by random guessing. Those are sequentially applied to repeatedly modified versions of the data, resulting in a sequence of weak classifiers: $G_m(x)$, $m = 1, \ldots, M$. The final classifier is combined through a weighted majority vote:

$$G(x) = sign\Big(\sum_{m=1}^{M} \alpha_m G_m(x)\Big) \tag{2.5}$$

The weights α_i are computed by the boosting algorithm and represent the accuracy of each classifier. Furthermore, the algorithm performs data modifications at each boosting step. The weight w_i of every observation (x_i, y_i), i=1,...,N is updated in respect to whether the observation was classified correctly in the previous step. Misclassfications will result in an increased w_i whereas correct classifications are decreased. Throughout the iterative process, the classifier is thereby forced to evaluate

those observations that are more difficult to classify and were missed by the previous classifier (Hastie et al., 2009, p. 337f.).

Similar to the random forest, the Extreme Gradient Boosting (XGB) is a CART-tree ensemble method. However, as mentioned above, the learning approach differs greatly. The underlying method of the random forest aims to reduce variance of the predictor by averaging over multiple, fully-grown, independent trees (low bias, high variance). The XGB also reduces the variance of the prediction by aggregating the output of many models. But mainly it minimizes the bias of shallow trees (high bias, low variance) by growing one tree at a time and each tree solving for the net error of the previous trained tree.

$$\hat{F}_{xgb}(X) = \sum_{i=1}^{M} T_i \tag{2.6}$$

To learn the independent tree structures f_k, the following regularized objective is minimized:

$$\begin{aligned} \mathcal{L}(\Phi) &= \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \\ \text{with: } \Omega(f_k) &= \gamma t_k + \frac{1}{2}\lambda ||w_k||^2 \end{aligned} \tag{2.7}$$

In order to measure the error of a tree, the function l is a differentiable convex loss function. The second term Ω penalizes the complexity of the functions i.e. weights w and the number of leaves t, which helps to smooth the learned weights and avoid overfitting. The equation 2.7 cannot be optimized by traditional methods and therefore the model is trained in an additive way with a gradient boosting algorithm. For further information on training XGB-trees please refer to Chen and Guestrin (2016).

2.3 Survival Analysis and Kaplan-Meier Estimation

Survival analysis is the study of time-to-event data. It considers the expected time until one or more events happen. A classical example is the analysis of a group of people suffering from a disease. In this case the events are the deaths of the people. One problem of the field is that often the data is incomplete, i.e. some subjects leave the study before the event happens and therefore it is unknown if and when it happens.

The Kaplan-Meier estimator is used for this purpose. It estimates for one or more subjects if an event does not happen until a given time and can handle censored data. It was first defined by Kaplan and Meier (1958). There are $n \in \mathbb{N}$ subjects which are observated in a time period $T = [0, t_f]$. A subject can have three different states: at risk, censored and dead (the event already happened). At $t = 0$ all subjects are at risk. The Kaplan-Meier estimate is defined by:

$$\begin{aligned} \hat{S}(t) &= \prod_{t_i \leq t} \frac{r_i - d_i}{r_i} \\ \text{with } \hat{S}(0) &= 1 \end{aligned} \tag{2.8}$$

d_i is the number of events at time t_i and r_i is the number of subjects at risk before t_i. If a subject is censored at time t, $\hat{S}(t)$ remains the same. For t_i, $\frac{r_i - a_i}{r_i}$ is the fraction of subjects surviving this time step. Therefore $\hat{S}(t)$ is the product of all the fractions before t which is the fraction of subjects at risk at time t_i to all subjects at the beginning. The resulting estimate is a step function as shown in Figure 4.8, for example.

Chapter 3

Use Case 1: Shop Interval Prediction

3.1 Task Description

The first task of this project comprises the analysis and prediction of shop visit intervals. For MTU as a maintenance provider it is important to make reasonable assumptions of these intervals. Since Flight Hour Agreement (FHA) contracts are priced according to the estimated interval lengths, wrong estimations can cause revenue losses.

To this point, predictions of shop visit intervals are entirely based on expert knowledge. Such experts search an operator "similar" to their new customer in the past shop visit data. This operator's shop visit behavior is then used as the customer's prediction. "Operator similarity", in this context, is not based on a commonly agreed upon similarity measure, and thus the prediction results are highly subjective and not deterministic in their outcome.

In order to introduce deterministic results and support the expert's decision making, the goal of this task is to model the shop visit intervals based on historical data. During the modeling process, multiple machine learning models are compared and their final accuracy is assessed on a test set. The model is supposed to predict shop visit intervals in the target variable Cycles Since Overhaul (CSO), stated in the unit "flight cycles", and report an expected standard deviation for the prediction.

Two datasets are available for model training: shop visit data (see subsection 1.2.1) and FR24 data (see subsection 1.2.2). The target variable is included in the shop visit data. section 3.2 Data Preparation describes how the data is prepared for the model training. To ensure predictability, the shop visit dataset is restricted to Hot Section Refurbishment (HSR) events. Furthermore, end-of-lease shop visits and small fleets with less than 10 engines are excluded in the training to further increase predictability.

Table 3.1 shows the list of independent variables that are provided by the stakeholders and are expected to influence the target variable. During the project these variables are referred to as "mission parameters". The three year range of the AGE_BAND variable is required by the stakeholders. section 3.3 Descriptive Analysis analyzes how said parameters influence the target variable Cycles Since Overhaul (CSO).

Subsequently, section 3.4 Model Development describes which models are considered in the analysis and how the final models are selected. Among other aspects it is also analyzed how additional parameters not contained in Table 3.1 can positively influence the prediction. These models are then compared to a baseline model consisting of the mean CSO per operator. Lastly, section 3.5 provides a final assessment of the task along with possible future directions of research.

3.2 Data Preparation

The Shop Visit (SV) dataset does not contain all the parameters specified in the use case. This section describes which data points have been filtered out and how the missing variables are inferred from the data provided.

Parameter	Unit	Description	Example
ENGINE_MODEL		Type of an engine	A5, S1
RATING		Thrust rating of an engine	27K
AGE	Year	Engine age at the time of the shop visit	4 years
AGE_BAND	Interval	Age of an engine in groups of three years where "a-b years" means the interval $[a, b)$	3-6 years
UTILIZATION	Hours/Year	Average flight hours per year of an engine	2000 hrs/year
STAGE_LENGTH	Hours/Cycle	Average flight hours per flight cycle of an engine	2.1 hrs/year
IS_FIRST_RUN	Boolean	True, if the engine came from its first run	
FLEET_SIZE		Number of engines per operator	
REGION		Operation region of an operator	Europe, Asia Pacific
HUB		First letter of the operator's hub airport ICAO code	E, Z

Table 3.1: Mission parameters

3.2.1 Data Filtering

The first step is to filter out irrelevant and invalid data points. According to the use case definition, the SV dataset is restricted to HSR and non-End of Lease shop visits. Furthermore, data points with Time Since New (TSN), Cycles Since New (CSN), Time Since Overhaul (TSO) or CSO below 0 are invalid, and so are data points with CSO larger than 20000. The latter is due to the LLP restrictions, and thus shop visits have to occur after at most 20000 cycles per run.

Table 3.2 shows the filter operations and how many data points get lost per filter. The operations are applied sequentially from top to bottom. That is, for instance, the outliers are removed first and then it is filtered for HSR shop visits. Therefore, the number of data points remaining after it is filtered for HSR depends on both the outlier and the HSR filter.

Filter	Data Points Lost	Data Points Left
(originally)		13692
remove outliers	-6	13684
HSR only	-5213	8468
remove end of lease	-263	8205
remove fleets with < 10 engines	-121	8084

Table 3.2: Filter operations

3.2.2 Computation of Additional Variables

After filtering the data, the next step is to infer the missing variables from the data. The formulas presented below can be read as follows. If $[\text{VARIABLE\_NAME}]_i$ is used, the statement is applied to a single shop visit i of the shop visit dataset. If the subscript is omitted, the operation is applied to an entire column.

The age as well as the age band of the engines at the time of the shop visit is not contained in the data. However, date of the shop visit and Entry Into Service (EIS) date are provided. Thus, the age in years can be calculated with the formula $\text{AGE}_i := \text{SV\_DATE}_i - \text{EIS\_DATE}_i$. By grouping the age into bands of 3 years (0-3 to 27-30) the age band is obtained as well.

The stage length is defined as the average flight hours per flight cycle. The SV dataset contains the variables TSN in hours and CSN. Therefore, the stage length over the lifetime of the engine can be estimated by calculating $\text{STAGE\_LENGTH}_i := \text{TSN}_i/\text{CSN}_i$.

Similarly, the utilization, defined as the flight hours per year, can be estimated using TSN and the engine age at the time of the shop visit. Thus, calculating $\text{UTILIZATION}_i := \text{TSN}_i/\text{AGE}_i$ leads the desired result.

To determine whether the run leading up to the current shop visit is the engine's first run it is possible to compare CSO and CSN (or, equivalently, TSO an TSN). Since TSO and CSO are reset after ever HSR shop visit, it is sufficient to compare those values with TSN and CSN, respectively. If these values are equal, the run prior to the shop visit was the engine's first run, and otherwise it was a subsequent run. Thus, it is possible to calculate $\text{IS\_FIRST\_RUN}_i := (\text{TSN}_i == \text{CSN}_i)$, where the $(x == y)$ operator returns true if and only if x and y are equal and false otherwise.

Besides the classification into first and subsequent shop visits as described above a counter of shop visits per engine is required as well. Such a count can be derived from the engine serial number and the shop visit date. First, the data is grouped by serial number and sorted by shop visit date in ascending order. Then, a counter starting from 1 for the first shop visit in this group is incremented by one for every shop visit. Assigning this number to the individual shop visits yields the desired result.

1. Group the shop visit dataset by SERIAL_NUMBER
2. Sort the groups by SHOP_VISIT_DATE in ascending order
3. Calculate: $\text{SHOP\_VISIT\_DATE}_i := (\textit{position within group})$

The fleet size of an airline is its number of engines at a certain point in time. Since the shop visit dataset contains both engine serial numbers and airline names, grouping by airline name and counting the number of unique engine serial numbers results in an approximation of fleet sizes for all operators. Hence, the numeric fleet size is calculated in two steps. First, the fleet size per operator is computed. Then, for every shop visit, the fleet size is retrieved according to the operator of the engine at the shop visit.

1. Group the shop visit dataset by OPERATOR
2. Calculate the fleet size per operator group: $\text{FLEET\_SIZE(operator)} := (\textit{size of group})$
3. Calculate: $\text{FLEET\_SIZE\_NUM}_i := \text{FLEET\_SIZE}(\text{OPERATOR}_i)$

The numeric fleet size as computed above is then used to determine the grouped fleet size of the operators. The grouping as listed below is based on expert knowledge. MTU Maintenance assumes that the engine fleet management within the listed groups is very similar. To hopefully capture the effect of fleet management on the runtime, this grouping is adopted.

$$\text{FLEET\_SIZE}_i := \begin{cases} \text{“}\textit{small}\text{”}, & \text{if } \text{FLEET\_SIZE\_NUM}_i < 25 \\ \text{“}\textit{medium}\text{”}, & \text{if } 25 \geq \text{FLEET\_SIZE\_NUM}_i < 50 \\ \text{“}\textit{large}\text{”}, & \text{if } \text{FLEET\_SIZE\_NUM}_i \leq 50 \end{cases}$$

As stated in the use case, it is required to compare the performance of rating and rating classes as well as engine model and engine model classes. The used classifications are provided by the stakeholders. This

grouping is also used within MTU Maintenance to classify engines. The classifications are determined by the following two formulas.

$$\text{RATING\_CLASS}_i := \begin{cases} \text{“}low\text{”}, & \text{if RATING}_i \in \{\text{“}22K\text{”}, \text{“}24K\text{”}\} \\ \text{“}mid\text{”}, & \text{if RATING}_i \in \{\text{“}27K\text{”}, \text{“}27M\text{”}, \text{“}27E\text{”}\} \\ \text{“}high\text{”}, & \text{if RATING}_i \in \{\text{“}30K\text{”}, \text{“}33K\text{”}\} \\ \text{RATING}_i, & \text{else} \end{cases}$$

$$\text{ENGINE\_MODEL\_CLASS}_i := \begin{cases} \text{“}Select\text{”}, & \text{if ENGINE\_MODEL}_i \\ & \in \{\text{“}S1\text{”}, \text{“}S2\text{”}\} \\ \text{ENGINE\_MODEL}_i, & \text{else} \end{cases}$$

3.2.3 Operator Hub ICAO Codes

The REGION feature of the shop visit dataset provides a rough classification of airlines by their main operation region. Since environmental conditions within such large regions are expected to vary a lot, a finer regional classification of airlines is required. A mapping of an airline to its hub (i.e. the airport with most of the airline's flight operations) can be used to identify the main area of operation. As described in subsection 1.2.2, the first letter of the International Civil Aviation Organization (ICAO) code can be used to map an airport to a region.

The FR24 dataset contains airline names and origin as well as destination ICAO codes. Therefore, a mapping of airlines to their hub's ICAO code can be obtained by finding the airline's most frequently visited airport. Since it can be assumed that airlines fly from their hub to a destination and then back to the hub, it is sufficient to count either origin or destination airports. In this case, the origin ICAO codes are counted per airline and a list is created with the most frequent airport ICAO code per airline.

However, a direct joining of the shop visit data with the hub mapping is not possible due to either incorrect airline names in the shop visit dataset or missing airlines in the FR24 data. In the first case, parts of the names are often omitted, added or changed (e.g. “China Eastern” instead of “China Eastern Airlines”, “Atlas Jet” instead of “Atlasjet”) or the capitalization is incorrect (e.g. “SHARJAH RULER'S FLIGHT” instead of “Sharjah Ruler's Flight”, “Cobaltair Ltd” instead of “COBALT”). Most of these mismatches can be averted with a “fuzzy” matching (i.e. take the name with the fewest different letters).

Due to the limited range of the FR24 dataset, airlines that ceased to exist before 2017 may be part of the shop visit dataset but are not contained in the FR24 data. To find the hub ICAO code for such airlines, a manual matching is required. This is done by finding an airline from the FR24 data that operates within a similar ICAO code region. With this process it is possible to match 206 out of 214 airlines present in the shop visit dataset.

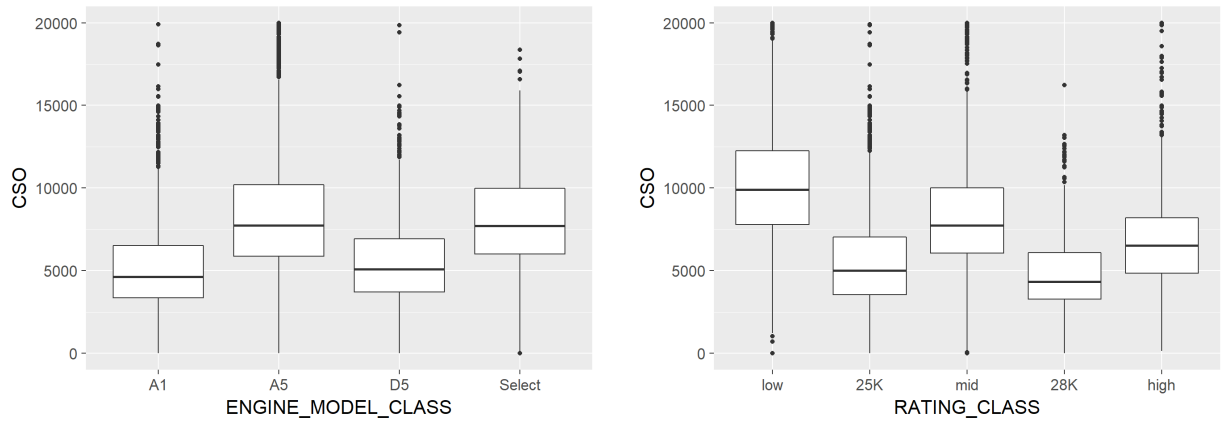
3.2.4 Shifting of Variables

The mission parameters as listed in Table 3.1 are recorded per shop visit. Some of these variables are constant with respect to the run length (CSO). These include all operator-related variables such as fleet size, region, hub, utilization and stage length as well as variables that describe engine properties (engine model, rating).

However, the variables age and consequently, age band depend on the predicted length of the run. This is because stage length, utilization and CSO determine the duration of the run in years. The following equations hold by the definition of the involved variables.

$$\text{RUN\_DURATION} = \frac{\text{CSO} \times \text{STAGE\_LENGTH}}{\text{UTILIZATION}}$$

$$\text{AGE}_{new} = \text{AGE}_{old} + \text{RUN\_DURATION}$$



(a) Engine Model Class

(b) Rating Class

Figure 3.1: Engine model class and rating class versus CSO. Newer models (A5, Select) perform significantly better than their older companions and a higher rating causes a shorter runtime. The ratings 25K and 28K perform since they belong to the models A1 and D5

An example illustrates this relation. Assume that an engine subject to prediction has average utilization of 2000 hours per year and a stage length of 2 hours per cycle, and further assume that the current age of the engine is 3 years. Now let the model predict two different run lengths: 1000 and 12000 cycles, respectively. The age at the two different predicted shop visits is therefore determined as follows.

1. $\text{AGE}_{1,new} = 3 + (1000 \times 2)/2000 = 3 + 1 = 4$
2. $\text{AGE}_{2,new} = 3 + (12000 \times 2)/2000 = 3 + 12 = 15$

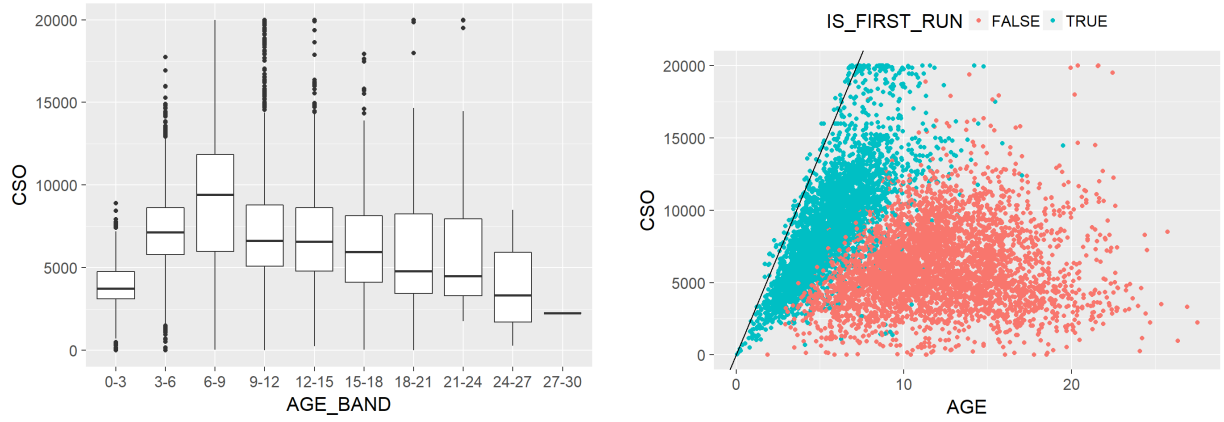
This implies that the age of an engine at a shop visit cannot be used to predict the CSO at that shop visit, since otherwise the model indirectly learns the target variable. This also holds for age band as well as CSO, TSN and CSN as used in the advanced models (see section 3.4). Similarly, the shop variable cannot be used for a prediction since the shop that performs the maintenance is not known prior to the shop visit.

Although these variables cannot be used as provided in the dataset, it is possible to use their value at the previous shop visit. These values are not dependent on the target variable but may influence the prediction. A similar approach as used for the shop visit date (see subsection 3.2.2) can be used to infer these values.

1. Group the shop visit dataset by SERIAL_NUMBER
2. Sort the groups by SHOP_VISIT_DATE in ascending order
3. For each [VARIABLE], calculate:

$$\text{[VARIABLE]\_AT\_LAST\_SV}_i := \begin{cases} \text{[VARIABLE]}_{i-1}, & \text{if } \exists\ (i-1)\text{-th shop visit} \\ \text{(default)}, & \text{else} \end{cases}$$

Since it is possible that the previous shop visit does not exist, meaningful defaults (i.e. the values of "(default)") are required for the variables. In most of the cases, the previous data point is missing if the engine had its first run. Thus, the "last" shop visit was the assembly, and a value of 0 can be assumed for the variables CSO, CSN, TSN and AGE. Consequently, the default for the age band is "0-3". For the shop, the default is set to "none" since this value is not included in the shop column.



(a) Age Band

(b) Age

Figure 3.2: Age band and age versus CSO. The left plot uncovers a negative influence of the age on the runtime. The left plot shows that the peak in run length in the age band 6-9 is caused by the first run shop visits

3.3 Descriptive Analysis

This section provides a descriptive, summarizing analysis of the shop visit dataset. As outlined in subsection 1.1.2 Engine Maintenance, general engine properties as well as an engine's operating conditions are deemed to be the factors most influencing the engine runtime. While the engine properties are known, there is no direct way to classify an engine's operating conditions. The hypothesis is that there exist proxy variables for the operating conditions.

To show the factors' influences on the runtime (CSO), this section is split into two parts. The first subsection analyzes variables related to engine properties, such as model and age. The second subsection shows why the operator cannot be used as an influencing variable and then analyses the influence of operator proxy variables like fleet size and region on the engine runtime.

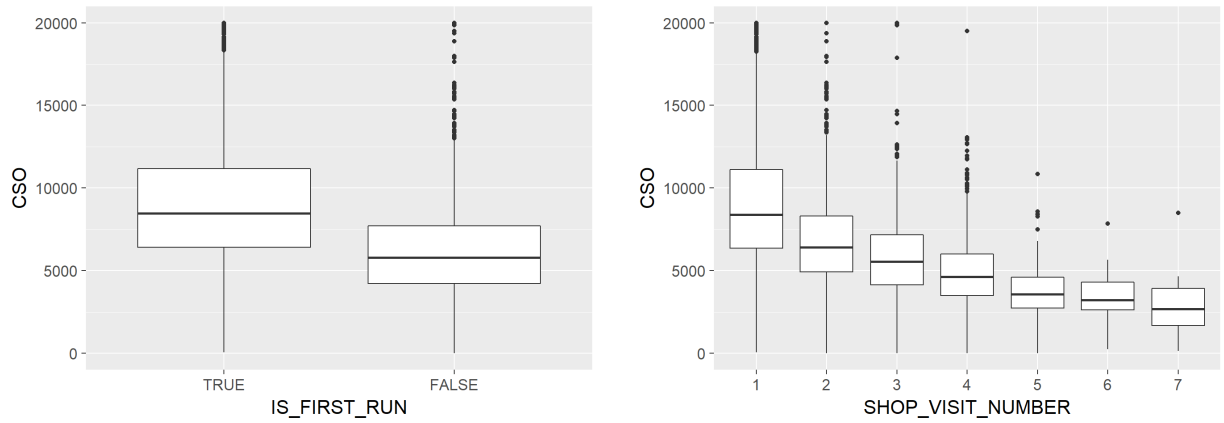
3.3.1 Engine-Related Variables

Figure 3.1a shows that the newer releases of the engine (A5, Select) perform significantly better than earlier ones with a median CSO of 7500. Both A1 and D5 perform similarly, with median CSO of around 5000. This difference is expected due to the improvements made to the newer models.

The plots in Figure 3.1b provide a comparison of models with respect to their runtime. The plots uncover a negative correlation of rating (higher to the right) and CSO if the classes "low", "mid" and "high" are considered. The ratings 25K and 28K correspond to A1 and D5 and thus they behave worse than the remaining classes.

Figure 3.2 depicts the relation of engine age at the shop visit and CSO. Figure 3.2a shows a peak in runtime for engines in the age band 6-9, whereas for younger and older engines the expected runtime declines. The low runtime for old engines is due to accumulating irreparable damages, whereas the low runtime for young engines is caused by the maximum number of cycles an engine can fly a year. Multiplying the maximum utilization of engines in the shop visit dataset with their minimum stage length results in an upper bound on cycles an engine can fly a year. This theoretical bound is depicted by a black line in the diagram.

The second plot in Figure 3.2b plots the engine age against the CSO. A point on the plot represents a single shop visit. Coloring the points by whether a first run or a subsequent run engine is serviced allows for two observations. First, engines serviced after their first run are much younger than engines coming from



(a) First and subsequent run

(b) Shop visit number

Figure 3.3: First/subsequent run and shop visit count versus CSO. First run engines can clearly expect a longer runtime than subsequent run engines. In fact, this negative trend continues within the group of subsequent runs (i.e. with a shop visit number ≥ 2) as visible in the right plot

a subsequent run. Second, new engines (i.e. engines in their first run) can expect a higher runtime than engines in later runs.

The latter claim is also supported by the plot in Figure 3.3a. Engines in their first run can expect a median runtime of around 7500 cycles whereas engines in subsequent runs can expect a median CSO of 5000. This behavior is expected as described in subsection 1.1.2. By instead plotting the shop visit number against the CSO (see Figure 3.3b) it is possible to show that this downwards trend continues even within subsequent run engines. The expected runtime for engines in their second run is with a median of over 5000 cycles more twice as high as the median runtime in the 7^{th} run.

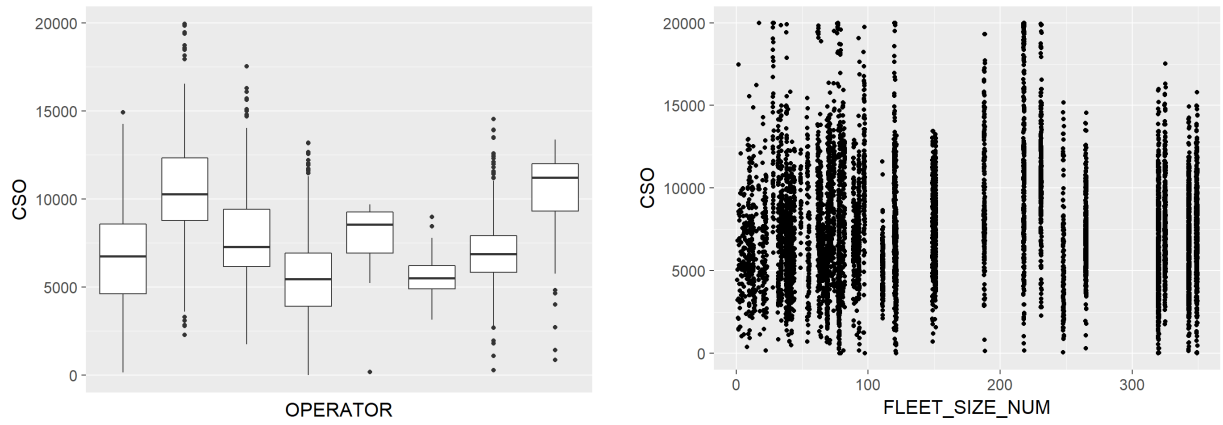
3.3.2 Operator-Related Variables

Figure 3.4a shows boxplots of the CSO a selection of unnamed operators and boxplots of their CSO. The plots show that the operator has a strong influence on the runtime. However, the operator name is not included in the list of mission parameters shown in Table 3.1. Reason for this is that if the operator were included, it would be impossible for the model to predict values for new, unseen operators. Thus, the remaining part of this subsection is concerned with the analysis of possible proxy variables for the operator and their influence on the runtime.

As described in subsection 1.1.2, it is expected that engines from an operator with better fleet management have longer runtimes. Since fleet management quality is not measurable, a suitable proxy variable could be the operator's fleet size: a larger fleet may imply better fleet management, which in turn improves the runtime.

To check this claim, Figure 3.4b plots the fleet size of an operator against its CSO. The shop visits are displayed as points. Since individual operators mostly have distinct fleet sizes (especially in the range above 100 engines - individual operators are single vertical line of shop visits), this variable does not suit as a proxy for the operators. A binning of fleet sizes is therefore necessary and described in section 3.2. However, plotting this binned fleet size against the CSO reveals that there is no noticeable influence. The plot is therefore omitted.

As expected, a finer resolution of airline operation locations via the hub ICAO code better captures runtime differences between regions. In Figure 3.5a, especially the Americas, Asia Pacific, China and Europe appear to behave similarly. However, even within the Americas (ICAO codes C, K, M, T, S), there appear



(a) A selection of operators and their CSO

(b) Numeric Fleet Size vs. CSO

Figure 3.4: Operator (anonymized) and numeric fleet size versus CSO. Clearly, operators have a high variance, but using the numeric fleet size as a proxy for operators is not possible. The well-separated vertical lines are individual operators and thus the model would overfit

to be large differences in runtime (see Figure 3.5b).

The plots for stage length and utilization are omitted here due to the lack of influence. The stage length has a slight inverse impact on the run length (i.e. a longer stage length leads to a shorter run length), whereas the utilization positively affects the run length (i.e. longer run time causes a longer run length).

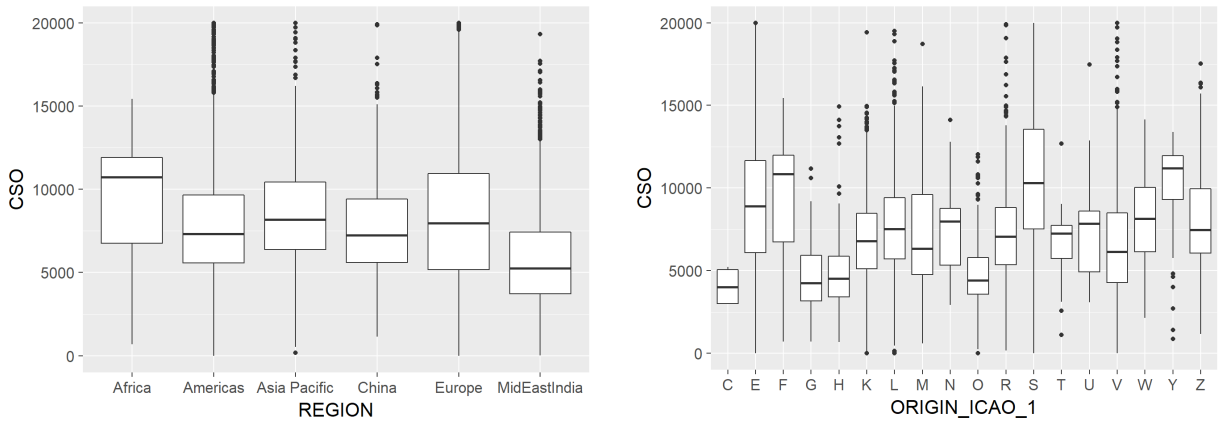
3.4 Model Development

The main goal of this use case is to verify that it is possible to predict the shop visit intervals of individual engines in the shop visit dataset. In order to achieve this, three random forest models are introduced with separate input variable combinations:

- Model I: mission parameters as in Table 3.1
- Model II: mission parameters, but engine model and rating are grouped as can be seen in subsection 3.2.2
- Model III: advanced model with the additional variables:
  - Numeric fleet size
  - CSO at last shop visit
  - Shop at last shop visit
  - TSN at last shop visit
  - CSN at last shop visit
  - Shop Visit Number

The random forest method can be recommended for this task, because it is a model that does not necessarily need data separation into a training and test set, which is advantageous considering time limitations. Additionally, it can deal with numerical and categorical input as well as missing values and it is a method known for it's good prediction results. The random forest models developed for this use case are optimized automatically and the missing values are replaced by column medians or most frequent factors.

To gain a preliminary understanding of the performance of the different random forest models, an easy baseline model, as required by the stakeholders, is computed. For this purpose, the mean shop visit interval



(a) Region vs. CSO

(b) Hub ICAO vs. CSO

Figure 3.5: Engine operating regions versus CSO. It is visible that finer resolution (by ICAO code) captures more differences

per airline is calculated and matched to all engines of that airline. Any more complex model that can be considered for the use case prediction should be able to outperforming this baseline model.

Model	**% Variance Explained**	**Mean Std. Deviation [CSO]**
Baseline-Model	/	2253.29
Model with ungrouped mission parameters	69.08	2019.13
Model with grouped mission parameters	68.9	2022.76
Advanced Model	73.68	1864.17

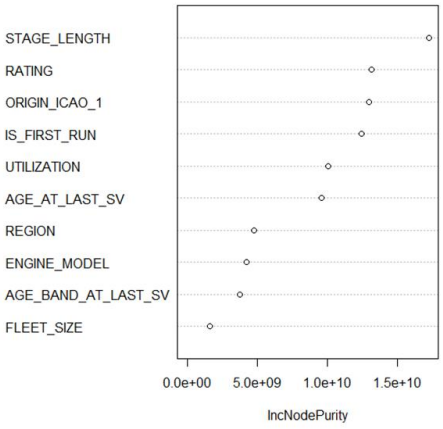
Table 3.3: Comparison of shop visit prediction models

The table 3.3 shows the results of the different models developed for the use case and are defined as mentioned above. The mean standard deviation is stated by request of the stakeholders but is not used for a profound statistical comparison of the models. However, it shall be sufficient enough to show that the more complex models are superior to the baseline model in terms of the standard deviation. The second column "% Variance Explained" represents the OOB-error of the random forest, which was explained in 2.2.1. For the baseline model this value is not calculable without a further segmentation of the data set (e.g. by cross validation). As there is no intention of the stakeholders to apply this model to practical purposes, this factor is not retrieved from the data.

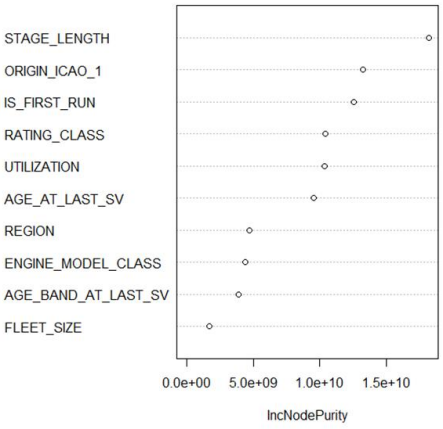
It can be observed that there is no significant difference in forecasting with grouped or ungrouped mission parameters and therefore it is suggested to use grouped parameters for an easier application in production use. Apart from that, the advanced model outperforms all other options and should be used if the necessary input data is available. In a next step the variable importance for each model is analyzed.

The label on the x-axis in the plots of 3.6 IncNodePurity relates to the loss function by which best splits are chosen. For a regression tree the loss function is the mean squared error. More useful variables achieve higher increases in node purities and an optimal node with perfect purity would split all data into classes of the same type.

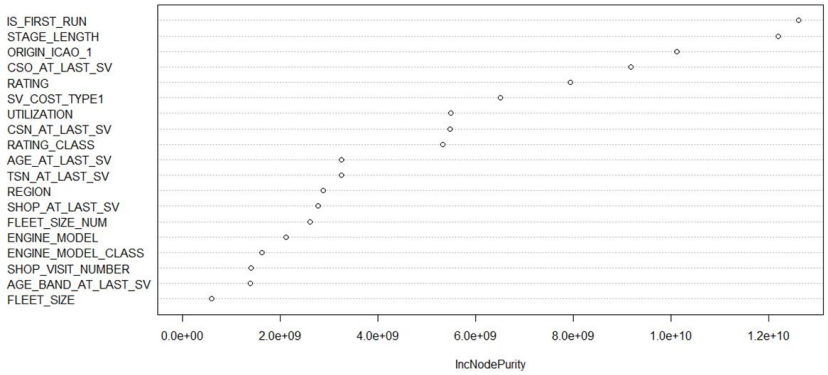
The plots show that the variables stage length and ratings are relatively important, which is consistent



(a) Variable importance of a random forest with mission parameter input



(b) Variable importance of a random forest with grouped mission parameter input



(c) Variable importance of a random forest with all available input variables

Figure 3.6: Variable Importance of shop interval prediction models: All plots show realistic results with variables describing the age of an engine and the utilization as the most important.

with technical principles that influence shop visit interval lengths. In addition, the location where the engines operates also shows main importance. This can be attributed to environmental factors like heat or sand. On the other hand, the fleet size does not indicate a prime importance, even though it was suspected that engines of operators with similar fleet size have similar shop intervals.

3.5 Summary and Outlook

Reaching the predefined goal of the first use case to develop a prediction model for shop visit intervals is possible with satisfying results. Furthermore, throughout a descriptive analysis of the given data sets, a lot of helpful information on the variables and their relationships among one another can be found and used to support the understanding of shop visit coherences.

Nevertheless, the accuracy of the prediction leaves room for improvement. This could be approached by analyzing the existing models and their residuals in more depth to identify structures among particularly

high residuals. Furthermore, retraining the models on current data, once it is available, will improve the forecast. That effect will occur not only because the available data set will enlarge, but also because the data basis will present reality more precisely.

In fact, it is possible to assume that the current models underestimate the shop visit intervals for two reasons. First, many new engine models have not had a shop visit yet and therefore cannot be found in the data. With an extension to the data this effect would be mitigated. Second, engines that were produced early in the model series were prone to early defects. Since these are included in the training data, the prediction of new fleets is rather pessimistic.

Another apparent possibility to improve the prediction is the implementation of different models, where one suggestion would be the development of a Quantile Regression Forest[1]. This method is a generalization of the random forest and it allows a non-parametric estimation of conditional quantiles instead of the conditional mean of the response variable. As a consequence, the applicability of the prediction model would become more reliable as it allows for a better assessment of possible deviations from the prediction.

[1] For more information please refer to Meinshausen (2006)

Chapter 4

Use Case 2: Work Scope Prediction

4.1 Task Description

After providing models to estimate the expected shop visit intervals for a fleet in chapter 3 Use Case 1: Shop Interval Prediction, the second part of this project is concerned with the prediction of module damage patterns for shop visits. As described in subsection 1.1.1, aircraft engines consist of multiple modules that are possibly taken apart during a shop visit.

Depending on what work was done during such a shop visit, every module is rated on a scale from 0 to 3. A module is rated 0 if it was not inspected and 3 if it was fully refurbished. In the MTU Maintenance division, a module with a work scope ≥ 2.3 is also said to be "exposed". Currently, it is not possible to predict this module exposure of an engine in a shop visit. However, this is important for two reasons:

1. optimal pricing of FHA contracts
2. optimal pre-ordering of replacement parts while an engine is being shipped to the shop

In the first case, while a FHA contract is set up for a fleet of a new customer, it is crucial to know the damage patterns of the customer's fleet. Competitive yet profitable pricing of such contracts is only possible if these exposure patterns can be predicted accurately. If the damage patterns are predicted pessimistically, the contract proposal may not be competitive enough. If the damage patterns are predicted optimistically, the contract may be competitive but not profitable for the maintenance firm.

In this scenario, a possible workflow is to first predict the fleet's expected runtime according to chapter 3 and then estimate the damage pattern based on the predicted CSO. Thus, input parameters are the mission parameters as listed in Table 3.1 together with CSO. section 4.3 describes how the mission parameters influence the exposure. Furthermore, this scenario requires an estimate for a fleet (i.e. multiple engines). Thus, the prediction target is the "exposure rate" defined as the ratio of exposed versus the total number of engines.

In the second case, predicting module damages after an engine was removed from the wing may lead to an optimized maintenance process. Currently, replacement parts are ordered after the engine was inspected in the shop. Since a shop receives a repair order right after the engine was removed, this information could be used to predict damaged modules. Appropriate parts could then be ordered while the engine is being shipped to the shop. This would result in a faster maintenance process and thus lower overall maintenance costs.

Since this scenario is engine-specific, the prediction outcome must be the likelihood that a certain module will be exposed given the engine conditions. This also makes it possible to use engine-specific influencing variables together with the mission parameters. Such variables may include TSO, CSN and TSN.

Again, both the shop visit and the FR24 datasets are available for training and only HSR shop visits are to be predicted, but for training both HSR and non-HSR shop visits can be used to increase the overall size of the dataset. The preparation of the dataset is outlined in section 4.2.

To reduce the complexity of the analysis, this task considers only the Low Pressure Turbine (LPT) out of the 13 modules contained in the V2500 engine. The LPT of the V2500 engine is currently produced by MTU and thus plays a particularly important role. Furthermore, the LPT is part of the cold section. It

is therefore suitable for a prediction, since hot section modules are very likely to be exposed at every HSR shop visit. Lastly, the LPT is an interesting module since it is theoretically a two-run module (i.e. in theory, it gets exposed only every second shop visit).

Two different prediction approaches are considered in this task. First, to predict work scopes for individual engines, similar models as used in the first task are trained and evaluated in section 4.4. It is concluded that these do not suffice to predict expected work scopes for fleets. Thus, in section 4.5 it is attempted to model the exposure rates using survival analysis. Kaplan-Meier estimation is used to model the LPT survival probability. Finally, a summary and an recommendations for future research directions are provided in section 4.6.

4.2 Data Preparation

The following data preparation is done with all modules. However, for simplicity, the description focuses on the LPT. Therefore all steps are done with the work scopes of the LPT. The additional parameters as defined in use case 1 are used again (see section 3.2) and three additional variables are introduced here.

The notation used here is analog to use case 1. The subscript $[\text{VARIABLE\_NAME}]_i$ means a formula is applied to a row (i.e. a shop visit) and an omitted subscript implies column-wise operations.

The task is to predict if a module will be exposed in a shop visit. As defined in section 4.1, a work scope ≥ 2.3 is called exposed. Therefore, the binary exposure of a module is defined as follows:

$$\text{LPT\_ACT\_BIN}_i := \begin{cases} 1, & \text{if LPT\_ACT}_i \geq 2.3 \\ 0, & \text{else} \end{cases}$$

As in use case 1, data points with any of TSN, CSN, TSO or CSO < 0 or > 20.000 are disregarded. Furthermore, there are also missing values for the work scopes. These are mostly from the most recent or the oldest shop visits. The former case applies to engines with a shop visit at the end of 2017 and early 2018. When the data set was provided to MTU, the corresponding engines were already removed from the wing but the shop visit was either not yet done or not recorded in the data. Furthermore, some old data points have no work scope on record. Thus, overall 1467 data points are disregarded.

After some discussion it was decided to only consider HSRs because that are the shop visits the MTU mostly plans beforehand and therefore is interested in the work scopes of those. In this step 3838 miscellaneous shop visits get lost.

The shop visit dataset does not contain work scope history records. However, this may be relevant for the prediction. Such a history can be calculated with the serial number of the engine and the shop visit date. First, the work scope of the LPT at the last shop visit (LPT_AT_LAST_SV) is calculated. It is derived from grouping by the serial number of the engines and sorting the groups by the shop visit date in ascending order. Then, the last LPT work scope of the engine can be assigned to the current shop visit. If the last shop visit does not exist (e.g. if the current shop visit is the first of an engine), LPT_AT_LAST_SV is set to 1. This is a reasonable assumption since the LPT is new once an engine enters into service.

1. Group the SV dataset by SERIAL_NUMBER
2. Sort the groups by SHOP_VISIT_DATE in ascending order
3. Calculate: $\text{LPT\_AT\_LAST\_SV}_i := \begin{cases} \text{LPT\_ACT\_BIN}_{i-1}, & \text{if} \quad i > 1 \wedge \exists\ (i-1)\text{-th shop visit} \\ 1, & \text{else} \end{cases}$

Not only the last shop visit could be meaningful, also the time of the last exposure. It is counted in cycles because as stated in subsection 1.1.1 start and landing have the biggest impact on the deterioration of the modules. For an engine CS_LPT_OH (Cycles Since LPT Overhaul) is defined by the sum of all CSVs of the engine since the module's last exposure. Therefore the dataset is again grouped by the serial

Filter	Data Points Lost	Data Points Left
no filter		13692
without missing LPT WS	-1467	12225
only HSR	-3838	8387
without missing LPT_AT_LAST_SV	-380	8007
without missing CS_LPT_OH	-268	7739

Table 4.1: Filter operations

number and sorted by the date of the shop visit. Then the row with the last exposure is identified and all the CSV-entries of the rows after the exposure until the current one are summed up.

1. Group the SV dataset by SERIAL_NUMBER
2. Sort the groups by SHOP_VISIT_DATE in ascending order
3. Calculate: $\text{CS\_LPT\_OH}_i := \sum_{j \in J} CSV_j$
   with $J = \{j \leq i \mid \max_{k<i}\{k > 0 \mid \text{LPT\_ACT}_k = 1\} \cup \{0\}\}$

To calculate these variables the work scopes from previous shop visits are needed. Some of these are missing as mentioned before and so 380 data points are lost if LPT_AT_LAST_SV is used and another 268 data points for the use of CS_LPT_OH. The filter operations and amount of data points removed can be seen in Table 4.1.

The data has to be split in two parts, one to train the model and one to test it. The training set consists of 80% of the data points. The splitting is done randomly but with the requirement that the distribution of the LPT_ACT_BIN variable remains the same in both sets.

Finally, the variables are prepared for the survival probability estimation in section 4.5. This method uses all data points (also the Miscellaneous) because otherwise we would lose some deaths. The Kaplan-Meier estimation cannot be applied to numeric variables. Each unique value of the variable would constitute a group, and consequently, the group sizes would become too small. Therefore, age, stage length and utilization are cut into two groups each such that the groups have approximately the same size.

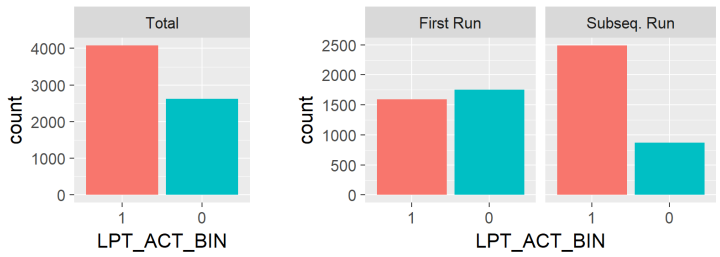
$$\text{AGE\_GROUP}_i := \begin{cases} \text{“}young\text{”}, & \text{if AGE}_i < 10 \\ \text{“}old\text{”}, & \text{if AGE}_i \geq 10 \end{cases}$$

$$\text{STAGE\_LENGTH\_GROUP}_i := \begin{cases} \text{“}low\text{”}, & \text{if STAGE\_LENGTH}_i < 2 \\ \text{“}high\text{”}, & \text{if STAGE\_LENGTH}_i \geq 2 \end{cases}$$

$$\text{UTILIZATION\_GROUP}_i := \begin{cases} \text{“}low\text{”}, & \text{if UTILIZATION}_i < 2500 \\ \text{“}high\text{”}, & \text{if UTILIZATION}_i \geq 2500 \end{cases}$$

4.3 Descriptive Analysis

The purpose of this section is to provide a descriptive analysis of the mission parameters in the context of the second task. As mentioned in the task description, the analysis and prediction of work scopes is reduced to the LPT. To simplify the visualization, the binary work scope classification as described in section 4.2 is used. This analysis was performed on the training set only. Note that for the sake of conciseness only the most revealing plots are shown; the remaining diagrams can be found in Appendix D.



(a) All work scopes

(b) Work scopes for first and subsequent run

Figure 4.1: These plots show the work scopes for all shop visits (left) and grouped by first and subsequent run (right). "1" are the shop visits with an LPT exposure and "0" are the shop visits without. The left plot shows that the data set is not balanced with respect to the two classes. The second plot reveals that in subsequent runs the exposure rate is significantly higher

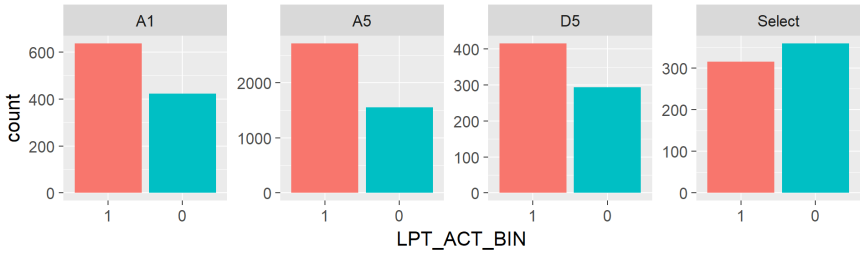


Figure 4.2: This plot shows the work scopes by engine model class. It is visible that A1 through D5 perform similarly, whereas the Select engines have a lower exposure rate. The latter is possibly since most of the Select engines only had a single shop visit so far

The histograms can be interpreted as follows. For each plot a certain grouping variable is used and a subplot is generated for each unique value that the grouping variable takes. This value is displayed in the gray bar above the plot. The two bars within each subplot then represent the number of exposed (1, red) and unexposed (0, turquoise) modules per group. Such a plot layout makes it possible to compare the exposure rates of different groups and thus uncover possible relations.

First, Figure 4.1a uncovers that the dataset is unbalanced. 4084 shop visits exist with an exposed LPT (approximately 61%), whereas in 2625 shop visits the LPT was not exposed (the remaining 39%).

Figure 4.1b displays work scopes depending on whether the engine had its first or a subsequent run. In the first run, the number of exposed and unexposed LPT modules is almost equal. In contrast, subsequent runs have a much higher rate of exposed to unexposed modules. This observations supports the claim that the LPT is a two-run module, since more than 50% of the modules remained unexposed in the first run. IS_FIRST_RUN thus clearly influences the work scope.

The diagram in Figure 4.2 displays the work scopes depending on the engine model class. The class allows for better comparison of groups since the S2 model has only around 100 observations. Note that the y-scales of the plots differ to allow for better comparison of small and large groups. Overall, A1, A5 and D5 follow a similar distribution of work scopes. In contrast, Select has a significantly lower exposure rate. Since the Select class is the newest member of the V2500 engine family, two reasons may be the causes of this behavior. First, due to improvements made to the Select engines, these may perform significantly better than the older models. Second, it may be possible that so far primarily First Run shop visits of Select engines were observed. The second seems most likely due to the similarity to the First Run subplot in Figure 4.1b. Also note that the distributions of A1 through D5 resemble the plot of all work scopes shown in Figure 4.1a.

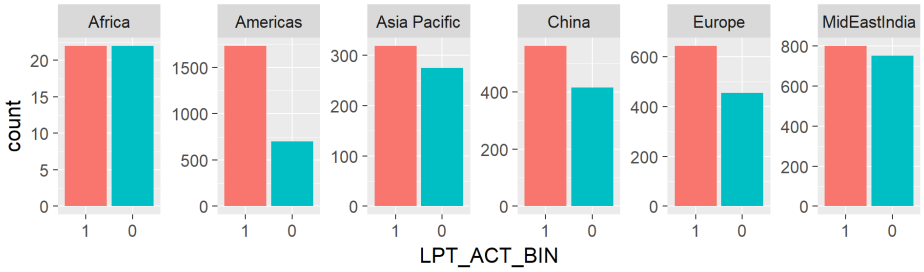
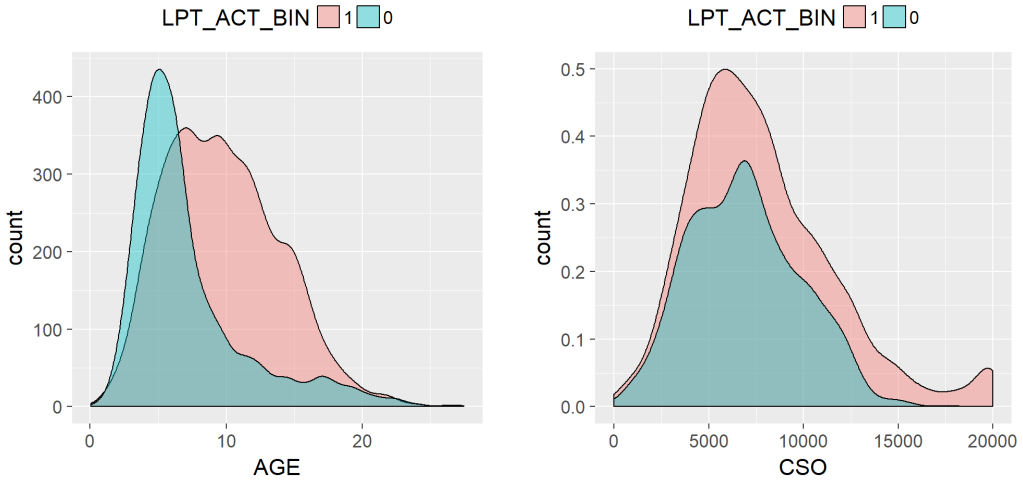


Figure 4.3: This plot shows the work scopes per region. "Middle East / India" has been shortened to "MidEastIndia" for an improved readability. Engines operating in the Americas apparently has a much higher exposure rate than engines in other regions



(a) Work scope densities by age

(b) Work scope densities by CSO

Figure 4.4: This plot shows density estimates for exposed and non-exposed shop visits by age (left) and CSO (right). The exposure rate is highest for engines between 7 and 20 years of age. Furthermore, runs with a length of 7000 or above 17000 cycles can expect a high exposure rate

Figure 4.3 displays the exposures by operator region. Notice that Africa has the lowest overall exposure rate, followed by Middle East/India. However, with a comparatively small sample size the Africa group is not representative. In contrast, the Americas have by far the highest exposure rate of all regions. China and Europe behave similarly, while Asia Pacific performs slightly better. The region is a clear influencing factor of the work scopes.

Density plots are used to show the influence of continuous variables on the work scopes. The plots are scaled by their group size to ensure comparability. Again, turquoise represents shop visits with unexposed LPT, whereas red shows shop visits with exposure. The densities are plotted "layered" (i.e. both densities are plotted and the area underneath is filled).

The plot in Figure 4.4a reveals that the LPT is most likely to be exposed in shop visits of engines with an age between 10 and 16 years. Clearly, the age is an influencing factor for the work scopes.

Finally, Figure 4.4b displays the work scope densities depending on the run length (CSO). The exposure rate increases for a run length greater than 5000 cycles. The slight increase of the count for runs approaching 20000 cycles is due to the LLP expiry. As noted in subsection 1.1.2, mandatory replacement limits of V2500 parts are 20000 cycles. Again, the CSO is an influencing factor for work scopes.

4.4 Modeling: Individual Engine Exposure

The first task of the second use case is a binary prediction for the exposure of individual engines regardless of corresponding fleet. Analogous to the shop interval prediction of the first use case a random forest is also applied here. However, as the data is more familiar and the data preparation is less time consuming, another model is introduced to offer the possibility of comparing different, equally complex methods. The following variable combinations were each used for both methods:

- Model I: mission parameters as in 3.1 and additionally CSO
- Model II: variables of Model I and additional variables available in the shop visit dataset:
  - Numeric fleet size
  - Cost Type
  - Removal Reason
  - Shop Visit Number
- Model III: variables of Model II and additional variables as calculated in 4.2, but less data points due to missing values

The models were all implemented with automated hyper parameter optimization and therefore no manual tuning was necessary.

Models	**Accuracy on test set in %**	**95%-Confidence Interval**
Random Forest		
Model I	73.5	(71.3, 75.6)
Model II	73.5	(71.3, 75.6)
Model III	73.3	(71.0, 75.5)
Extreme Gradient Boosting		
Model I	72.9	(70.7, 75.1)
Model II	73.4	(71.2, 75.5)
Model III	73.3	(71.0, 75.5)

Table 4.2: Accuracy of different Random Forest and Extreme Gradient Boosting Models

As it can be seen in table 4.2, all models[1] achieve fairly similar results on the test set. It can be assumed that Model III would outperform the other models if it was possible to add the missing data points back to the training and test set. This would need some further analysis of problems with the computation of these variables. As no form of the RF or the XGB can be singled out as the best model at this point, it is suggested to use any variant of the random forest. The advantage of the random forest is that it can construct trees parallel while gradient boosting is a sequential approach and therefore random forest has faster computational speed.

The analysis of variable importance in this use case is less straightforward, because XGB needs dummy variables in order to include factors into the model. Hence, factors in training and test data were expressed dummy variables and applied for RF and XGB. In consequence, the results of variable importance do not take the overall variables into account, but reveal the importance of every categorical feature in the data. The following examplatory plot 4.5 shows the importance of the top twenty categorical features / numerical input variables. The importance of a characteristic is expressed relative to the most important variable.

[1] Input variables of models as stated before

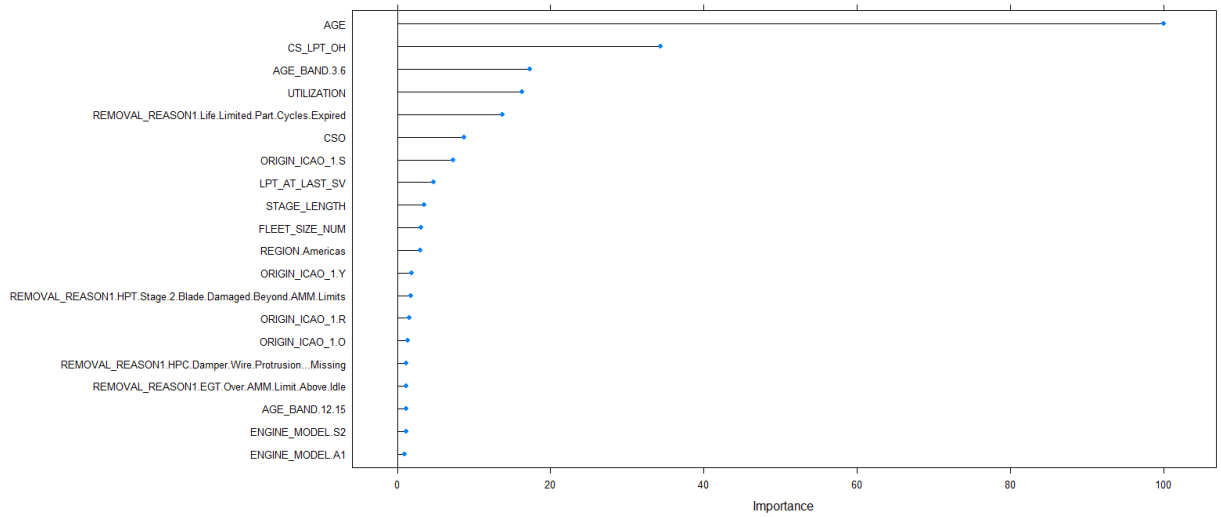


Figure 4.5: Variable Importance of an XGB model for individual engine exposure with all available input variables (Model III). The plot shows realistic results with variables describing the age of an engine and the utilization as the most important. It is also comprehensible that the LLP-expiration is a strong indicator for the module exposure.

Although it is possible to construct satisfying models for the prediction of single engine work scopes in this manner, both the random forest and the extreme gradient boosting algorithms failed to offer a way to predict the exposure rate for an entire fleet. This is mostly due to the fact that there is no target figure describing the exposure rate in the data and by computing this value manually, too many data points are conflated and there is too little data left for a comprehensive application of the above models.

The next section 4.5 will show a different approach to the prediction of an exposure rate for an entire fleet.

4.5 Modeling: Fleet Exposure Rate

As an alternative approach to the models evaluated in section 4.4 this section discusses the applicability of survival analysis to the exposure rate prediction. In particular, the survival plots presented here are based on Kaplan-Meier estimates of the survival function. The importance of groups is determined via the log-rank test. Refer to section 2.3 Survival Analysis and Kaplan-Meier Estimation for details.

First, it is described how the prediction of exposure rates can be formulated as a survival problem. Then, the variables are analyzed individually to find proper grouping factors using the log-rank test. Finally, it is attempted to model the survival problem based on said grouping factors.

4.5.1 Exposure Rate Prediction as a Survival Problem

The estimation of LPT exposure rates in shop visits can be formulated as a survival problem as follows. First, an event in the survival model corresponds to a shop visit in the work scope prediction scenario. In one of these events, a module either “survives” (not exposed, work scope of 0) or it “dies” (it was exposed, work scope of 1). Thus, it can be assumed that the “life” of an LPT module starts when it is mounted on the engine and ends when it is removed. If the work scope was classified as 1, the module was either replaced or overhauled. Thus, subsequent runs thus fly with a “newborn” LPT.

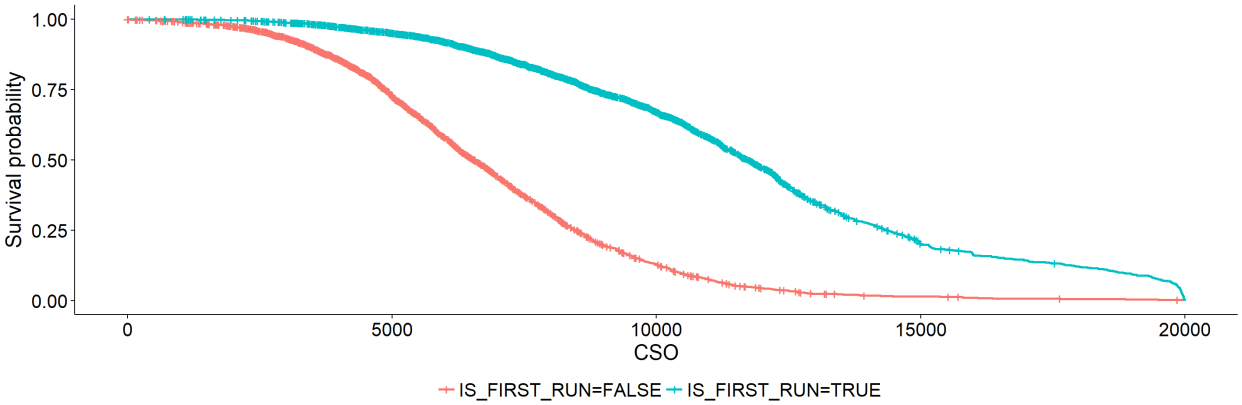


Figure 4.6: Survival function estimates for first and subsequent run. As expected, the exposure rate for subsequent run engines is higher (the curve is shifted to the left)

Second of all, CSO can be assume to be the time variable. This assumption is reasonable since the lifetime of an engine is most influenced by its cycles flown and not the total flight time (see also subsection 1.1.2). Furthermore, CSN is reset on every HSR shop visit and thus measures the lifetime of the module. Neither CSN nor Cycles Since Visit (CSV) are an option for the time variable. CSN is not never reset and thus only models the lifetime of the whole engine, and CSV is reset at every shop visit, even when the module is not exposed.

Analog to groups of patients, the survival curves are now estimated for groups of LPT modules that behave in a similar fashion. How relevant grouping variables and factors are found is described in Variable Grouping.

Once the survival curve is estimated, it is possible to predict the fleet exposure rate as follows. First, the group of the fleet subject to the prediction is determined. Then, the corresponding survival function estimate is retrieved. The CSO can now either be guesses via a model from use case 1 or it is known prior to the prediction, and the corresponding value, p, of survival function is the survival probability. $1 - p$ then corresponds to the exposure rate of an engine (the functions estimate the survival probability, but the rate of "dead" LPT modules is needed).

4.5.2 Variable Grouping

As mentioned above, Kaplan-Meier survival functions are estimated by groups. It is important to consider the trade-off between group size and estimation accuracy. Small groups will cause a bad function estimate, but omitting influencing variables may also negatively impact the prediction. Therefore, similar groups should be merged to maintain a high number of samples. This subsection is concerned with finding optimal grouping variables and factors. For both engine model and rating the classified variables (see section 3.2) are used to reduce the overall number of groups. Again, note that the most important plots are shown in this section. The diagrams for the remaining variables can be found in Appendix E.

Figure 4.6 shows the survival function for the shop visits grouped by first and subsequent runs. The curves are well-separated and the result is as expected: the LPT tends to survive longer in a first run than in subsequent runs. IS_FIRST_RUN clearly is a significant grouping variable and is kept as-is.

The survival curve estimates for the engine models are displayed in Figure 4.7. The plot suggests that A5 and Select as well as A1 and D5 behave similarly, where A5 and Select perform better. This is proved by log-rank test, which results in values 0.834 and 2.35×10^{-3} for the A5/Select and A1/D5, respectively. Therefore, the respective pairs are merged into single groups to increase their sizes.

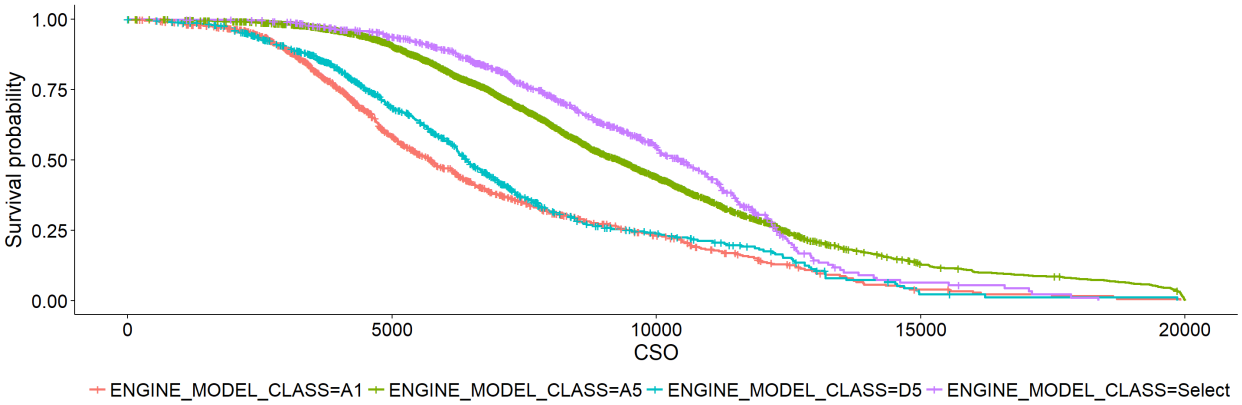


Figure 4.7: Survival function estimates per engine model class. A1 and D5 perform significantly worse than A5 and Select engines

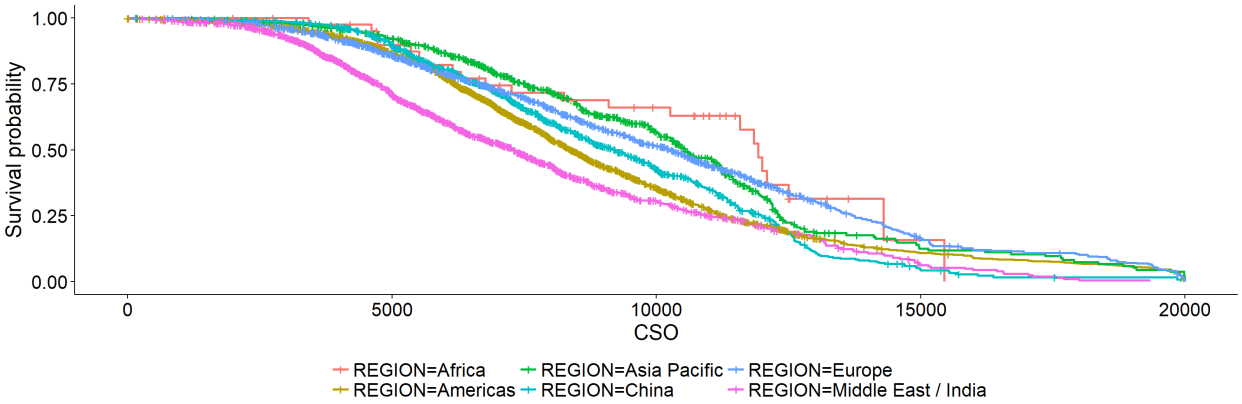


Figure 4.8: Survival function estimates by region. The steps in the Africa plot are clearly visible since the group only contains 89 data points. Engines operating in Middle East/India perform worst

Figure 4.8 plots the survival curves by main operating region. The function for Africa clearly constitutes an outlier. This also visualizes the problem of small groups – the steps of the function are clearly visible since the Africa group only contains 89 data points. However, comparing Africa, Asia Pacific and Europe with the log-rank test results in values $\geq 1.482 \times 10^{-2}$. Similarly, comparing China and the Americas gives a log-rank value of 1.164×10^{-2}. Hence, Africa/Asia Pacific/Europe and China/Americas are merged into two groups.

The survival curves for the age groups are shown in Figure 4.9. A clear separation of the groups is visible. For simplicity, the survival diagrams for utilization and stage length are omitted. Both age group and stage length group show strong statistical significance with a log-rank value of 1.004×10^{-209} and 4.716×10^{-103}, respectively. A log-rank value of 1.371×10^{-4} for the utilization group shows that it does not prove significant. Hence, the stage length and age groups are kept and the utilization group is rejected.

Finally, the fleet size is rejected due to insignificance with log-rank values $\geq 4.225 \times 10^{-4}$. The rating class is kept with all its factors since pairwise log-rank tests result in values $\leq 5.388 \times 10^{-6}$.

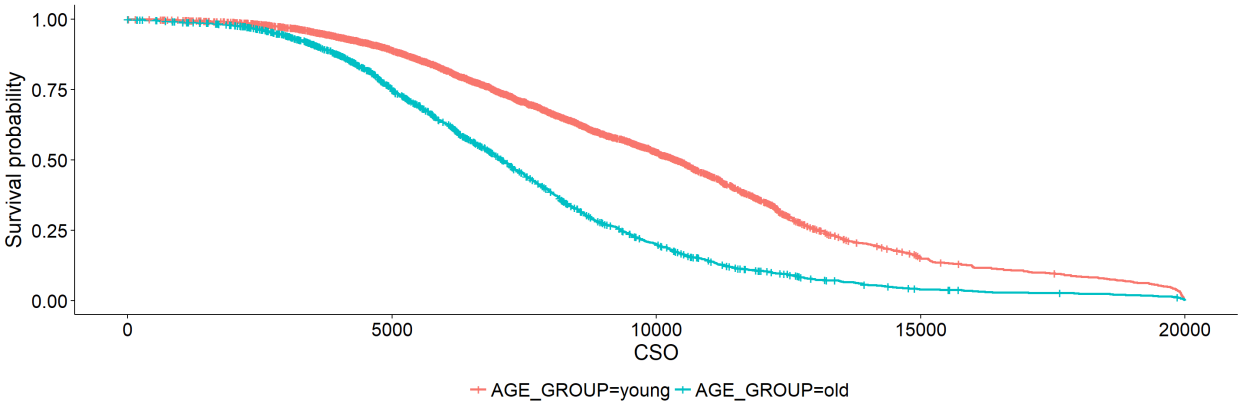


Figure 4.9: Survival function estimates by age (grouped). Young engines have a lower exposure rate than old engines

4.5.3 Modeling

The merged groups can now be used to model the survival problem. In particular, based on the previous analysis, the following variables are selected.

- IS_FIRST_RUN
- RATING_CLASS
- AGE (grouped)
- STAGE_LENGTH (grouped)
- ENGINE_MODEL_CLASS (merged)
- REGION (merged)

This results in a total number of 112 groups. Optimally, with a training set of around 10500 data points, this should result in groups with approximately 94 data points each. However, it turns out that the group sizes are not uniformly distributed. 25 of the 112 groups have size ≤ 5, and 18 groups consist of ≤ 3 shop visit observations. Clearly, such small groups lead to bad survival curve estimates.

A possibility to mitigate this issue is to select less grouping variables. A likely candidate here is IS_FIRST_RUN since the curves for first and subsequent runs are well-separated. However, it remains unclear how to select the remaining grouping variables for an optimal trade-off between group size and accuracy. The basic form of the log-rank test as used above is not sufficient for such a comparison.

Furthermore, it is unclear how the survival function estimates are tested. The classic machine learning approach to estimate the function on a training dataset and evaluating its accuracy on a test set is not possible since the target value (the exposure rate) is not contained in the training set. It is not a valid approach to predict the most likely exposure label (i.e. predict 1 if the probability is ≥ 0.5 and 0 otherwise). Such an error measure is not meaningful since for a predicted probability < 1 there also exists the chance that the module is not exposed.

4.6 Summary and Outlook

In this second use case it was attempted to model engine module work scopes. The LPT was considered as a representative for all modules to simplify the analysis. The descriptive analysis identified the most important influencing factors for work scopes. Subsequently, the modeling of individual engine work scopes showed promising results. The exposure was predicted correctly in around 70% of the cases.

Since the exposure rate is not part of the provided data sets, it was concluded that the "classic" models do not suffice to predict exposure rates for fleets. It was therefore attempted to model the exposure with Kaplan-Meier survival analysis. although the results seemed promising, this modeling was not completed since it remained unclear how a relevant grouping could be found and how the prediction model can be tested.

To tackle these challenges, further research could explore clustering methods to find groups of similar shop visits in the data. Here, a constraint for the clustering algorithm should be a reasonably large group size. If it is ensured that the group size remain large enough, the Kaplan-Meier estimation can be used to model the respective exposure rates.

To test these models, it could be possible to group the engines contained in the test dataset according to the grouping developed during training. The exposure rates of these test groups can then be estimated by computing the ratio of exposed to non-exposed LPT modules. The prediction error can then be determined by comparing the predicted survival probability and the calculated exposure rate. It remains to be shown whether the groups in the test data set are large enough to provide meaningful results.

Furthermore, as an alternative to Kaplan-Meier estimates, proportional hazards models such as Cox Regression can be explored. This method operates under the assumption that an influencing factor is proportional to the death hazard of a particular group and that said factors are time-independent (Cox, 1972). For instance, switching from one drug to another may half a patient's hazard rate. Cox Regression can both incorporate categorical as well as numerical variables. Furthermore, it does not require grouping as it estimates a single hazard function for the data (Cox, 1972). However, for the model to be applicable, it is required to check the conditions posed by the model.

Chapter 5

Summary

The purpose of this chapter is to first discuss organizational aspects of this project(section 5.1). Second, suggestions for a usage of the models in practice by MTU Maintenance are provided (section 5.2). Finally, recommendations for further improvement of the models are presented in 5.3.

5.1 Organizational Aspects

The organizational aspects of working on use cases for internal clients revealed some challenges, but also a great learning experience. From the outset, an extensive documentation of the group meetings was created, which incidentally accelerated the writing of the final report. The primary function, however, was to record the use case requirements, group progress or the assignment of individual tasks.

Especially the documentation of use case requirements should be improved in the future. Even though the interviews with stakeholders were documented, an official confirmation of meeting notes by the owners was not conducted. This lead to some subsequent changes of the goals that were time-consuming and should have been avoided. For example, it was only later discovered that the cut point for exposure should have been a work scope of 2.3 instead of 2.0 which lead to a large number of changes. Most importantly, all involved parties should be present at the first interview and the objective target should be approved by all.

This more precise strategy would have been especially useful in the second use case. The difficulties coming with a prediction target value that is not in the data were underestimated and while relying on the successful approach of the first use case, the literature research was not extensive enough. Due to a combination of the lack of approved goals as mentioned before and a limited project period, the exposure rate prediction could not be entirely fulfilled.

Overall, the organization of the project was nonetheless productive and target-aimed. The communication with all involved parties worked well and the results were satisfying. Especially the descriptive data analysis emerged to be greatly informative the for stakeholders. Even though it was not a primary goal, this analysis added crucial value to the project results.

5.2 Suggestions for Practical Applications

Once the models are developed, considerations have to be made on how to use them productively. The goal is to make the use as easy as possible for both developers and end users. The developer point of view is considered first with the focus on the distribution of the models as R packages. Afterwards, a possible combination of both use cases into a single model is presented.

5.2.1 Deploying and Distributing the Models

MTU Maintenance currently primarily uses the programming language R[1] for data analysis and thus the models were developed in R as well. To make the models easily reusable, an R package was developed for the

[1] https://www.r-project.org/about.html

first use case that contains the data preparation, the model training as well as the resulting trained models. Such a package has many benefits for the developer, including (Leisch, 2008):

- Predefined structure of R packages: conventions ensure that R packages must have the same layout. Thus, a developer familiar with this structure can easily make changes to the package (e.g. update or change the models)
- Versioning: R packages can be equipped with a version number. Developers and users can therefore easily see which version of the package they use
- Documentation: functions in R packages can be documented in a standardized fashion, thus easily enabling users and developers to understand the functionality provided
- Easy use within R: the model can be used for predictions without any knowledge of the models
- Meaningful default values: if certain parameters of the model are unknown, the package can automatically set the optimal default values for the missing parameters to still produce useful predictions
- Validate the input: the package can ensure that the input values are provided in the correct format and return error messages otherwise

Furthermore, with the developed package it is possible to achieve reproducible results. In his blog, Pete Warden, a senior developer of the machine learning framework Tensorflow, noted that such reproducibility is crucial in data science (Warden, 2018). By including the training data as well as the training procedures in the package, it is possible for developers to reproduce the results presented in this package. This is particularly crucial in this project since it is planned to include the models in business decisions. Another benefit is that if new training data becomes available, the models can be updated which possibly improves the prediction accuracy.

Furthermore, the package was equipped with tests[2] that can be invoked after changes are made to the models. This way, errors during package development are reduced and proper operation is ensured.

Similarly to use case 1, the results of the second use case should be included in a package as well. This way, the results provided here are preserved for future use within MTU.

5.2.2 Combining the Use Cases

The two use cases are related in the sense that they assist MTU Maintenance in the planning of shop visits and the expected work scopes of modules. In fact, the model of the second use case is dependent on the first. For a new customer, the run length (CSO) that is required to predict the work scopes is unknown. By coupling the first and the second model as shown in Figure 5.1, it is possible to automatically predict both the run length and work scope for an engine or a fleet.

This is possible by first using one of the models from use case 1 to predict the CSO using the mission parameters. Subsequently, the predicted CSO together with the mission parameters is inserted into the model of the second use case to predict the work scopes. Both is returned to the user as a result. This greatly improves the usability of the models for the users.

Currently, however, it is only possible to predict both CSO and work scopes for an individual engine since the exposure rate prediction for fleets in the second use case is missing. However, if a model for exposure rate prediction is developed, the setup can also be used for fleet prediction.

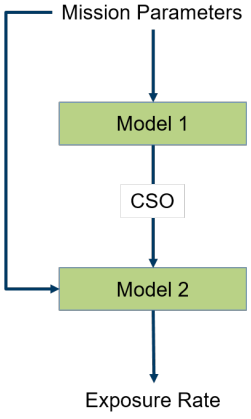


Figure 5.1: Combination of Use Cases 1 and 2 into a single model

[2] the package "testthat" was used; refer to CRAN for further information

5.3 Recommendations for Further Model Improvements

For this project a dataset with 13692 shop visits were given. In classical machine learning this is a very small number and therefore the quality of a prediction is limited. Obviously, it is not possible to get many more data points since there have not been more shop visits of the V2500.

However, it would have been possible to get better results if the given data had been recorded more accurately. Additionally for future consideration it could be helpful to not only analyze shop visit data but also on-wing data of the engines. FR24 has a lot of useful data but it currently can not be mapped to the shop visit data since it does not contain the serial number. If it would be possible to identify the engines in FR24, more data could be obtained, for example the exact flight regions. This could, for example, be done by mapping an engine serial number to the tail number of an aircraft.

Bibliography

Ackert, S. (2011). *Engine maintenance concepts for financiers*. Aircraft Monitor.

Amit, Y. & Geman, D. (1997). Shape quantization and recognition with randomized trees. *Neural Comput. 9*(7), 1545–1588. doi:10.1162/neco.1997.9.7.1545

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140. doi:10.1007/BF00058655

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. (1984). *Classification and regression trees*. The Wadsworth statistics/probability series. CRC Press.

Chen, T. & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794). KDD '16. San Francisco, California, USA: ACM. doi:10.1145/2939672.2939785

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, *34*(2), 187–220. Retrieved from http://www.jstor.org/stable/2985181

Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Mach. Learn. 40*(2), 139–157. doi:10.1023/A:1007607513941

Fahrmeir, L., Künstler, R., Pigeot, I., & Tutz, G. (2010). *Statistik: Der weg zu datenanalyse* (6.). Springer-Lehrbuch. Springer Berlin Heidelberg. Retrieved from http://www.springer.com/de/book/9783642019388

Flightradar24. (2018). About. Retrieved June 28, 2018, from https://www.flightradar24.com/about

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Ann. Statist. 29*(5), 1189–1232. doi:10.1214/aos/1013203451

Goldbloom, A. (2015). How to win a kaggle competition. Retrieved July 17, 2018, from https://www.import.io/post/how-to-win-a-kaggle-competition/

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2.). Springer Series in Statistics. New York: Springer-Verlag New York. Retrieved from http://www.springer.com/gp/book/9780387848570

Heumann, C. & Schomaker, M. (2017). *Introduction to statistics and data analysis: With exercises, solutions and applications in r*. Springer International Publishing. Retrieved from https://books.google.de/books?id=4mQBDgAAQBAJ

Ho, T. K. (1995). Random decision forests. In *Proceedings of the third international conference on document analysis and recognition (volume 1) - volume 1* (pp. 278–). ICDAR '95. Washington, DC, USA: IEEE Computer Society. Retrieved from http://dl.acm.org/citation.cfm?id=844379.844681

Howard, J. & Bowles, M. (2012). *The two most important algorithms in predictive modeling today*. In Strata Conference: Santa Clara. Retrieved from http://strataconf.com/strata2012/public/schedule/detail/22658

ICAO. (2018). Location indicators (doc 7910/167). Retrieved June 28, 2018, from https://store.icao.int/location-indicators-doc-7910-167th-edition-multilingual-printed.html

International Aero Engines. (2018). Products. Retrieved July 12, 2018, from http://i-a-e.com/products.html

Kaplan, E. L. & Meier, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, *53*(282), 457–481. doi:10.1080/01621459.1958.10501452. eprint: https://www.tandfonline.com/doi/pdf/10.1080/01621459.1958.10501452

Leisch, F. (2008). Creating r packages: A tutorial. In P. Brito (Ed.), *Compstat 2008 - proceedings in computational statistics*. Heidelberg, Germany: Physica-Verlag. Retrieved from http://nbn-resolving.de/urn/resolver.pl?urn=nbn:de:bvb:19-epub-6175-3

Meinshausen, N. (2006). Quantile regression forests. *J. Mach. Learn. Res.* *7*, 983–999. Retrieved from http://dl.acm.org/citation.cfm?id=1248547.1248582

Peng, R. & Matsui, E. (2016). *The art of data science*. Lulu.com. Retrieved from https://books.google.de/books?id=ZDH9DAEACAAJ

Richards, W. R., O'Brien, K., & Miller, D. C. (2010). w Air Traffic Surveillance Technology. Boeing. Retrieved June 28, 2018, from https://www.boeing.com/commercial/aeromagazine/articles/qtr_02_10/pdfs/AERO_Q2-10_article02.pdf

Suthaharan, S. (2016). *Machine learning: Models and algorithms for big data classification* (1.). Integrated Series in Information Systems 36. Springer US. Retrieved from http://www.springer.com/us/book/9781489976406

Turkey, J. W. (1997). *Exploratory data analysis*. Behavioral Science: Quantitative Methods. Reading, Mass. USA: Addison-Wesley.

Warden, P. (2018). The machine learning reproducibility crisis. Retrieved July 15, 2018, from https://petewarden.com/2018/03/19/the-machine-learning-reproducibility-crisis/

Wikimedia Commons. (2015). Icao first letter. Retrieved June 28, 2018, from https://en.wikipedia.org/wiki/File:ICAO_FirstLetter.svg

Wikipedia. (n.d.). Icao airport code. Retrieved June 28, 2018, from https://en.wikipedia.org/wiki/ICAO_airport_code

xgboost. (n.d.). Scalable and flexible gradient boosting. 2015-2016 DMLC. Retrieved July 17, 2018, from https://xgboost.readthedocs.io/en/latest/

Appendix A

Contribution

The purpose of the table in Table A.1 is to report the contributions to this report as requested by the DI-Lab supervisors. The "Section" column references sections in the text and the "Main Contributor" column lists the primary contributor to the respective section in the report. Note that naturally throughout the development of such a report each section has multiple contributors and this list only contains the primary one.

Section	**Main Contributor**
subsection 1.1.1 Aircraft Engines	Moritz
subsection 1.1.2 Engine Maintenance	Moritz
subsection 1.2.1 Shop Visit Dataset	Moritz
subsection 1.2.2 Flightradar24 Dataset	Helge
section 2.1 Exploratory Data Analysis	Céline, Helge
section 2.2 Machine Learning Models	Céline
section 2.3 Survival Analysis and Kaplan-Meier Estimation	Moritz
section 3.1 Task Description	Céline
section 3.2 Data Preparation	Helge
section 3.3 Descriptive Analysis	Helge
section 3.4 Model Development	Céline
section 3.5 Summary and Outlook	Céline
section 4.1 Task Description	Helge
section 4.2 Data Preparation	Moritz
section 4.3 Descriptive Analysis	Helge
section 4.4 Modeling: Individual Engine Exposure	Céline
section 4.5 Modeling: Fleet Exposure Rate	Helge
section 4.6 Summary and Outlook	Helge
section 5.1 Organizational Aspects	Céline
section 5.2 Suggestions for Practical Applications	Helge
section 5.3 Recommendations for Further Model Improvements	Moritz

Table A.1: Contributions to this report

Appendix B

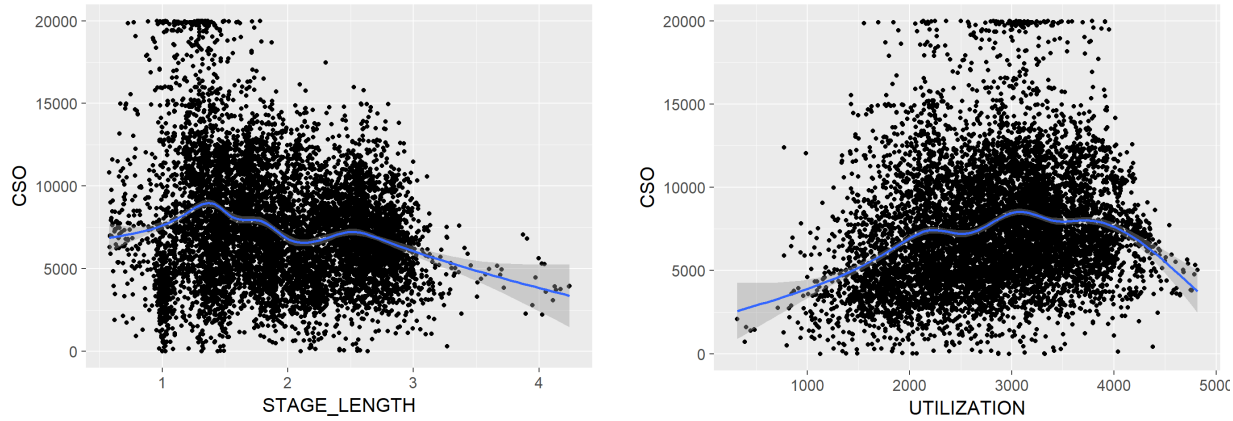
Shop Visit Dataset: Variables

Parameter	Unit	Description	Example
SHOP_VISIT_DATE	date	The date the shop visit is registered	14.12.2016
SERIAL_NUMBER		Specific id of an engine	
ENGINE_MODEL		Engine model as described in subsection 1.1.1	A5
RATING		Thrust rating of the engine	33K
OPERATOR		Operating airline of the engine at the time of the shop visit	Lufthansa
TSN	hours	Time Since New, i.e. flight hours since the entry into service	7291
CSN	cycles	Cycles Since New, i.e. number of cycles since the entry into service	12577
TSV	hours	Time (flight hours) Since last shop Visit	5306
CSV	cycles	Cycles Since last shop Visit	10291
TSO	hours	Time (flight hours) Since last Overhaul (HSR)	8978
CSV	cycles	Cycles Since last Overhaul (HSR)	12731
SHOP		Shop where the maintenance took place	HA9
REMOVAL_REASON		Reason for the shop visit	Oil Leakage
SV_CLASS		Classified by work scope, generally either Miscellaneous or Hot Section Refurbishment	
[MODULE]_ACT		Work scope of the [MODULE], between 0-3, where 0 means there was no work at all and 3 is a complete exposure	2.3
REGION		derived from OPERATOR	Europe
EIS_DATE	date	Date of entry into service	05.01.2004
AGE_BAND		Age of the engine grouped in bands of three to four years	3-6 years

Table B.1: List of most important variables in the shop visit dataset

Appendix C

Descriptive Analysis of Run Lengths



(a) Stage Length vs. CSO

(b) Utilization vs. CSO

Figure C.1: Stage length and utilization versus CSO. The purpose of the blue lines is to guide the eye. Stage length has a negative influence on the run length whereas utilization appears to have a positive influence

Appendix D

Descriptive Analysis of Work Scopes

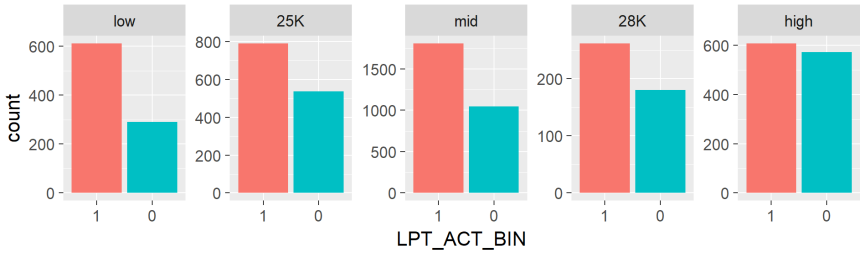
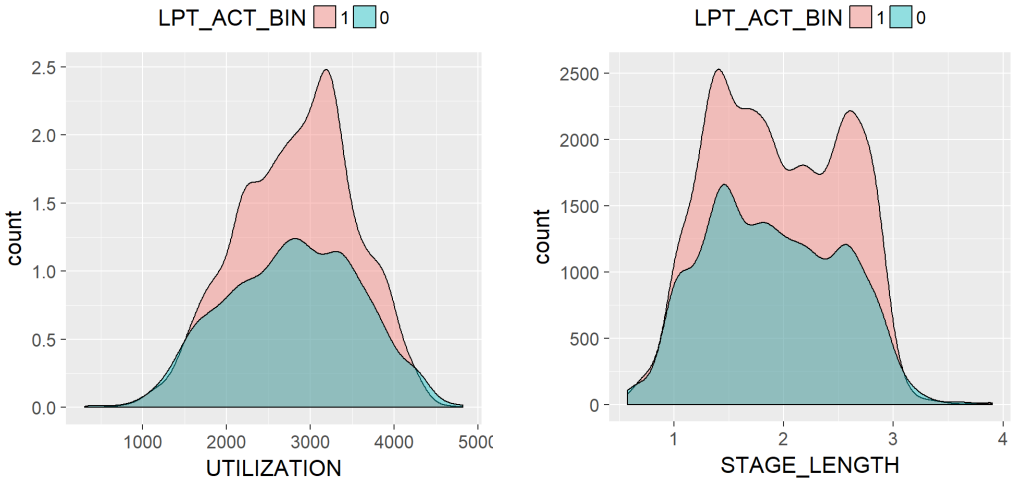


Figure D.1: Work scopes by engine rating class. Clearly, engines with a higher rating can expect a lower exposure rate



(a) Work scope densities by utilization (b) Work scope densities by stage length

Figure D.2: Density plots of work scopes by utilization and stage length. These two factors have little influence on the work scopes since the exposure rate is almost proportional to the group size

Appendix E

Survival Analysis of Work Scopes

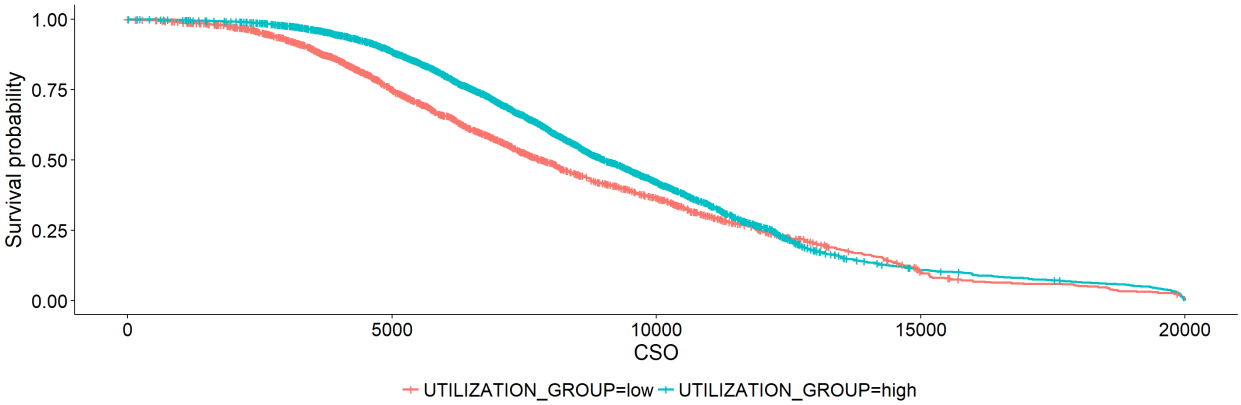


Figure E.1: Survival functions for utilization groups. Both groups behave similarly, and thus utilization is not a strong influencing factor

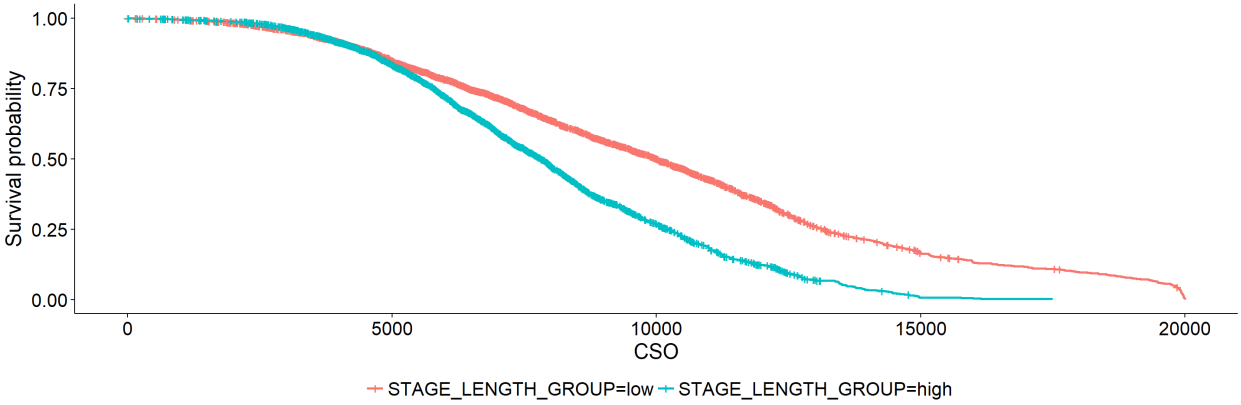


Figure E.2: Survival functions for stage length groups. A higher stage length seems to decrease the survival rate of LPT modules

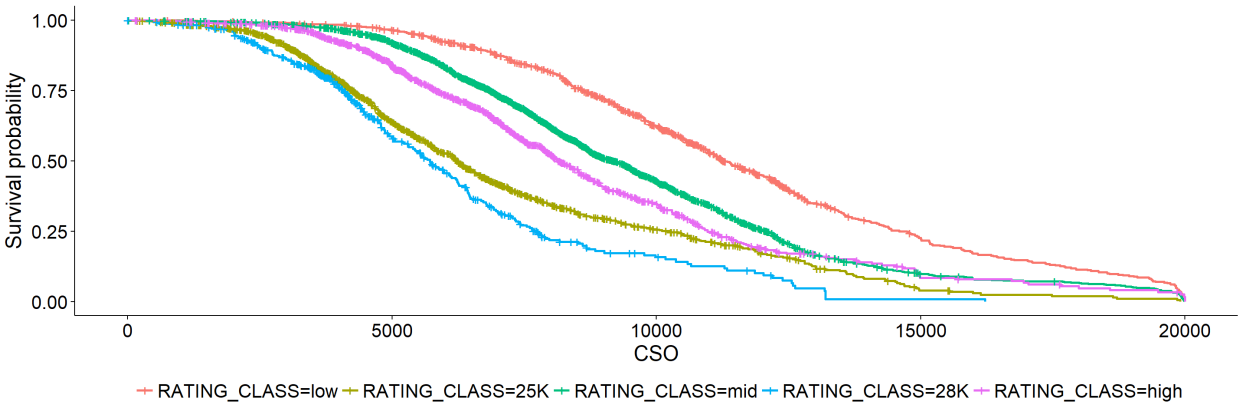


Figure E.3: Survival functions for rating classes. LPT modules in engines with a low rating perform best (red), whereas a rating of 28K (blue) performs worst

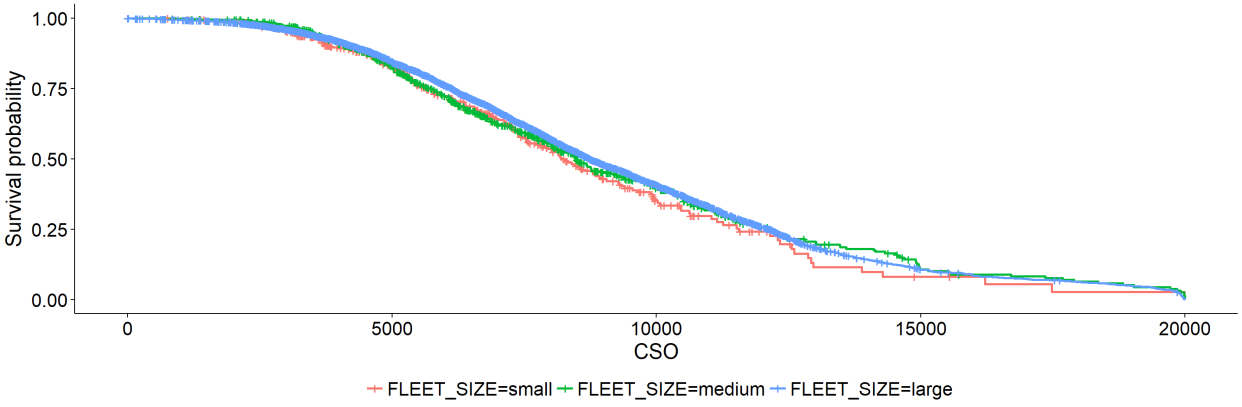


Figure E.4: Survival functions for fleet sizes. The fleet size does not influence the exposure rate since all curves behave the same

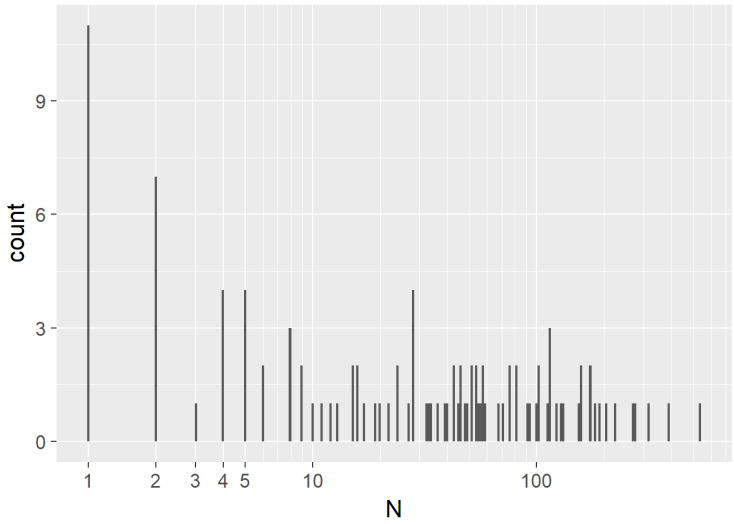


Figure E.5: Survival analysis group size distribution, the x scale is logarithmic. There are many groups with small sizes (i.e. ≤ 5 shop visits). Kaplan-Meier is therefore not applicable