Predicting the need for turbocharger inspection based on maintenance records

Master Thesis in MSc Business Analytics & Management

Rotterdam School of Management
Erasmus University Rotterdam

Yanfei Chen
566549yc
yfc1685@gmail.com

Supervisors Erasmus University Rotterdam:
Coach: Dr. Jan van Dalen
Co-reader: Dr. Müge Tekin

Supervisors Company:
Coach: Alexander Mik

Date: June 2021

Acknowledgment

> Things have their root and their branches. Affairs have their end and their beginning. To know what is first and what is last will lead near to what is taught in the Great Learning.
>
> *Great Learning, Book of Rites*
> *Written ca. 500 B.C.E.*
> *Translated by James Legge*

This thesis concludes my Master in Business Analytics & Management, and serves as a prelude to my doctoral degree.

I would like to express my sincere gratitude to Prof. Dr. Jan van Dalen for his timely, detailed, and constructive feedback, which lit my path when I was walking in the darkness. Furthermore, I am very grateful to Prof. Dr. Müge Tekin, who provided me with many useful tips, and pointed out insufficiencies in my thesis. Also, I thank Alexander Mik for his business insight and the opportunity on which my thesis is based.

Last but not least, I want to thank my family and friends in China. Even though the pandemic increases the distance between the two ends of Eurasia, I feel that they are beside me when we pick up our phones. Without their support, I could not have completed this thesis.

Preface

Executive summary

Increasing vehicle uptime is in the interest of transport companies, Original Equipment Manufacturers (OEMs), and society. Transport companies are seeking new ways of staying competitive; OEMs are making a profit by selling transport services; and society is looking for environment-friendly solutions to transportation capacity crunch.

Predictive maintenance answers this need. Predictive maintenance techniques estimate the health status of a component, and find a preferred time to repair or replace it (Sakib and Wuest, 2018). Vehicle uptime gets increased, and cost savings can be obtained by both transport companies and transport service providers. However, predictive maintenance is often associated with sensors, telematics, cloud platform and complicated analytics, discouraging practitioners from stepping forward.

This thesis proves that predictive maintenance can be introduced based solely on maintenance records, a prevalent data source kept by almost every garage and OEM. The proposed solution predicts turbocharger failure, increases truck uptime, and brings cost savings. Detailed steps of analysis and advice on execution are presented. This thesis is especially applicable for companies that are either about to launch pilot projects or unaware of the business value that predictive maintenance can create.

Contents

List of Tables

List of Figures

1 Introduction

In recent years, predictive maintenance, i.e., maintenance based on estimates of the health status of equipment (Susto et al., 2015), has been introduced in many industries, especially where optimization of asset management is desired. By taking advantage of predictive maintenance techniques, companies can prevent unexpected equipment downtime, improve service quality for customers, and reduce the additional cost caused by over-maintenance in preventive maintenance policies (Wang et al., 2017). Choosing an optimal time to replace or repair a part based on the prediction given by predictive maintenance models can optimize a trade-off between maintenance and performance cost, as well as increase availability and reliability of assets (Sakib and Wuest, 2018). For example, it is observed that by implementing predictive maintenance techniques, the potential for companies in heavy industries to increase asset availability is 5 to 15 percent, and reduce maintenance costs 18 to 25 percent (Bradbury et al., 2018).

Despite the promising outlook of predictive maintenance, the levels of adoption of predictive maintenance across different industries are still relatively low. A study of 280 companies from Belgium, Germany and the Netherlands (Haarman et al., 2017) shows that two thirds of the surveyed companies still adopt periodic inspections, and draw conclusions based either solely on inspector's expertise or on a combination of inspector's expertise and instrument read-outs. Haarman et al. (2017) also observe that among all critical success factors of predictive maintenance implementation, such as data availability, budget, culture, technology, and data security, data availability is considered as the most important one. This can be partly explained by the fact that it is still technologically challenging to collect sensor data from assets in real-time (Haarman et al., 2017), whereas unawareness of the potential business value which maintenance data can create may also stop companies from stepping forward.

Geissbauer et al. (2016) surveyed over 2,000 senior executives from industrial products companies worldwide, and showed that transportation and logistics, among all surveyed industries, has the highest percentage of industry experts who think that data and analytics are of high importance to decision-making. An important reason is that transportation and logistics is a low margin industry (Tipping and Kauschke, 2016) where sudden vehicle breakdown quickly turns profit to loss (Prytz et al., 2015). Thus, increasing vehicle uptime is a preoccupation for transport and logistics companies seeking new ways of being competitive (Prytz, 2014). However, it is estimated that freight transport across Europe will increase by 60% by 2050 (European Commission, 2019). Meanwhile, European Commission (2019) points out that capacity crunch is already being felt in relevant sectors, and that solutions which have a negative impact on greenhouse gas emission reductions, like deploying more vehicles, are not favored. These concerns call for advanced techniques which improve vehicle uptime while contribute to sustainable development.

Closely associated with the transportation and logistics industry is the automotive industry, in which original equipment manufacturers (OEMs) are heading towards selling transport services, such as maintenance and uptime guarantee, alongside with vehicles (Prytz et al., 2015). If a vehicle is under warranty (service costs covered by a fixed monthly fee), its sudden breakdown affects not only the fleet operator but also the OEM, because expenses, such as towing, disruption of garage workflow, costs of components, and rent of replacement vehicle, are borne by the latter (Prytz et al., 2013).

Typically, in these two industries, a fault is fixed only after it becomes an issue affecting vehicle's performance (Prytz et al., 2013). One of the key reasons for the huge gap between current situations and the future promised by predictive maintenance techniques is the inherent complexity of these industries. Wireless communication becomes less reliable when vehicles operate in remote areas, on country borders or in harsh environments, resulting in fewer opportunities for continuous monitoring (Prytz et al., 2015). Meanwhile, it is not worth installing sensors purely for diagnostic purposes, as they are usually expensive and need to fulfil rigorous safety standards (Prytz et al., 2015).

Nevertheless, the immaturity of continuous monitoring should not be an obstacle to embracing predictive maintenance. Predictive maintenance can be implemented even without real-time sensor data. Maintenance records, which record the details of maintenance operations of vehicles, are widely available in workshops and garages. Patterns that correlate with failure can be detected not only from real-time sensor data, but also from maintenance records (Haarman et al., 2017). It is hardly cost-effective for relevant companies to overlook existing data sources and chase for more complex prediction systems which are not necessarily mature in their situations. For these companies, exploiting maintenance records is a means to quickly introduce predictive maintenance solutions (Prytz et al., 2015), as well as a feasibility study of more advanced systems.

The present study presents a data-driven approach that predicts the health status of truck turbocharger. Models are based on maintenance records from a truck dealership[1] which sells trucks as well as maintenance services to fleet operators. These maintenance records are generated when trucks visit the garage, due to either scheduled maintenance or sudden breakdown. The predictive maintenance solution proposed in the present study is designed to be adopted as an aid to technicians (Prytz et al., 2013), reminding them to focus on the trucks which are labeled by the model as in need of turbocharger inspection. The outcomes of the present study will enable the company to increase vehicle uptime, establish better customer relationships, and cut maintenance costs for both its clients and itself.

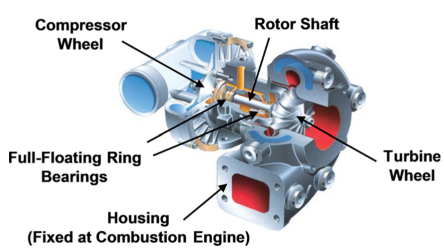
The remainder of the thesis is structured as follows. Section 2 states the research objective and research questions. Section 3 introduces the theoretical background of the present study. Section 4 explains the data and methods which lay a foundation for subsequent analysis. Section 5 presents the steps of analysis. Section 6 reflects and concludes this thesis.

[1]The name of the company is masked for privacy reasons.

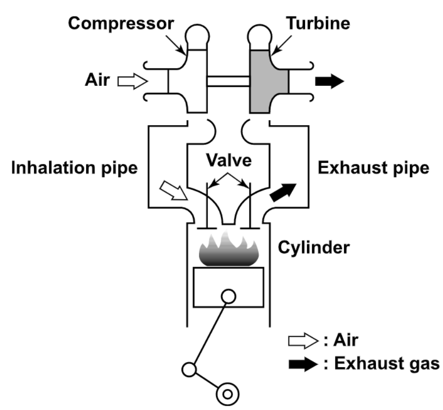
2 Objective and research questions

2.1 Research objective

The objective of the present study is to increase truck uptime and bring cost savings to fleet operators by predicting turbocharger failure based solely on maintenance records. Among the myriad of components inside a truck, some components are more crucial than others. Failure of these crucial components will almost always lead to a truck's sudden breakdown. Predicting when they fail is at the core of improving truck uptime. Praveenkumar et al. (2018) address the use of vibration signal for automated fault diagnosis of automotive gearbox. Liljefors (2020) presents a recurrent neural network-based method for predicting turbocharger failure in heavy-duty trucks based on sensor data and maintenance records. Truck compressor failure is studied by Prytz et al. (2013; 2015) and Nowaczyk et al. (2013) based on a Volvo internal database. Mashhadi et al. (2020) propose a stacked ensemble model to predict the remaining useful life of truck turbocharger based on the same Volvo database.



(a) 3D sketch of automotive turbocharger (Schweizer, 2010)



(b) Illustration of the action of a passenger vehicle turbocharger (Tetsui, 2002)

Figure 1: Turbocharger

The present study focuses on turbocharger (see Figure 1). Turbocharger is 'a device in which heat energy from engine exhaust gas turns a turbine along with a compressor on the same axis, such that inflow air is pressurized by the compressor and supplied to the engine, thereby improving the engine's combustion efficiency' (Tetsui, 2002, p. 582). When a turbocharger fails, it causes significant collateral damage to the truck (Prytz et al., 2013). Although turbochargers are usually designed to last the full lifetime of the truck or scheduled for planned maintenance, this is not enough to prevent all unanticipated failures (Prytz et al., 2013).

Most studies on predictive maintenance are solely or partly based on sensor data. Choudhary et al. (2009) study 150 papers that are related to the use of data mining in manufacturing, and conclude that almost all research considers the case when continuous monitoring of devices is possible. Prytz et al. (2013) also argue that it is common in fault prediction research to be able to continuously monitor the device in question. Later, Prytz et al. (2015) demonstrate the scarcity of academic literature on optimizing maintenance strategy based on maintenance records. A recent study (Carvalho et al., 2019) also shows that vibration signals are the most common data used to design predictive maintenance models.

However, since maintenance records are so ubiquitous, it is interesting to know whether predictive maintenance can be realized based solely on maintenance records. To achieve this objective, the following research questions are developed.

2.2 Research questions

Understanding the failure patterns of turbocharger is conducive to a solid study. Therefore, the first research question is generated as follows.

RQ1: What are the failures patterns of turbocharger?

To predict turbocharger failure, machine learning models are built. These models should learn from maintenance records, and predict whether the status of a turbocharger is healthy or unhealthy, i.e., whether a turbocharger has a long way to fail or will fail soon. This leads to the second research question.

RQ2: How well can turbocharger failure be predicted?

To create business values, maintenance operations should be considered from not only a technical perspective, but also an economic perspective. For fleet operators, when not replacing a turbocharger incurs more costs than replacing it, the economic life of the turbocharger is depleted, and replacement should be immediately carried out for the purpose of cost savings. Hence, the last research question is developed as follows.

RQ3: For fleet operators, to what extent can maintenance costs be reduced when the predictive maintenance model is implemented?

Apart from the above research questions, the present study also explores the relationship between the health status of turbocharger and the replacement of other components. When predicting truck compressor failure, Prytz et al. (2015) consider pumped air volume since last compressor change, mean compressed air per distance, and air compressor duty cycle, all of which are directly related to compressor usage pattern. However, maintenance records restrict the present study, as there is no information directly related to turbocharger usage pattern. To extract more features, the ages of other components are considered.

2.3 Academic relevance

Table 1 lists papers particularly relevant to the present study. All these studies use data-driven methods to predict the failure of a specific truck component. However, none of them are based solely on maintenance records.

Table 1: Most relevant papers to the present study

Reference	Component	Data[a]
Prytz et al. (2013)	compressor	SD
Nowaczyk et al. (2013)	compressor	SD
Prytz et al. (2015)	compressor	MR and SD
Liljefors (2020)	turbocharger	MR and SD
Mashhadi et al. (2020)	turbocharger	SD

[a] The data types are Maintenance Records (MR) and Sensor Data (SD).

The present study addresses the scarcity of literature on the use of maintenance records in predicting truck component failure, and is one of the pioneering studies of predicting truck component failure based solely on low-quality, less advanced, but more common data. The feature engineering techniques employed in the present study contribute to the field of data mining by providing new ideas of extracting information from maintenance records, which have limited numerical columns.

2.4 Managerial relevance

The predictive maintenance solution proposed in the present study benefits the company in three ways. First, customer satisfaction is increased, as the fleet operators associated with the company can enjoy longer truck uptime and suffer less from truck breakdown. Second, as fewer sudden breakdowns happen, the company can manage its labors and physical assets more efficiently, reducing cost from both aspects. Third, the solution can act as a pilot project inside the company, laying a foundation for more advanced and complicated prediction systems.

Meanwhile, the present study encourages more transport and automotive companies to introduce predictive maintenance, as the presented methods only rely on maintenance records, one of the most prevalent data sources in these industries. Successful implementation of predictive maintenance reduces operational costs, helping these businesses to stay competitive.

Society can also benefit from a wide adoption of predictive maintenance, because total vehicle uptime gets increased without more vehicles being deployed. This is in line with sustainability.

3 Theoretical background

This section outlines the theoretical background that makes predictive maintenance happen, beginning with the illustration and comparison of the current three maintenance strategies. The mathematical principles behind failure prognostics and some failure reasons are then presented, as they are the fundamentals of prediction models. Next, practical aspects of data are reviewed. An overview of the statistical and machine learning techniques used in predictive maintenance, particularly in the present study, is given. This section concludes with cost savings, an indicator which can determine whether a predictive maintenance solution is accepted by the company.

3.1 Maintenance strategies

Different nomenclatures and groups of maintenance strategies exist (Carvalho et al., 2019). The present study adopts the classification of maintenance strategies proposed by Susto et al. (2015) into: run-to-failure (R2F), preventive maintenance (PvM), and predictive maintenance (PdM).

Under the run-to-failure strategy, maintenance interventions are performed only after failure happens (Carvalho et al., 2019). Run-to-failure is obviously the simplest maintenance strategy, hence frequently adopted (Susto et al., 2015; Prytz et al., 2013). However, sudden truck breakdown can increase the lead time of a consignment, and unanticipated failure of a mechanical or electronic component can paralyze a production line. Besides, the costs of interventions and the ramifications of downtime are by and large much more substantial than those associated with preventive maintenance (Susto et al., 2015).

Preventive maintenance, also known as scheduled maintenance, is a maintenance strategy under which maintenance actions are carried out according to a planned schedule based on time or process iterations (Susto et al., 2015). In this way, failures are usually reduced to varying degrees, but unnecessary replacements are inevitably performed (Susto et al., 2015). With the rapid development of technology, industrial products are becoming more and more sophisticated, while better quality and reliability are required (Jardine et al., 2006). These facts challenge preventive maintenance, as it has become a major expense of many industrial companies (Jardine et al., 2006).

A good maintenance strategy needs to overcome the drawbacks in the above two strategies, and is supposed to both improve equipment condition and minimize maintenance costs (Carvalho et al., 2019). Predictive maintenance, which leverages predictive tools to determine when maintenance actions are necessary (Carvalho et al., 2019), is the strategy that stands out among others (Jezzini et al., 2013). Its advantages include: maximizing equipment uptime, delaying or reducing maintenance activities, and minimizing labor and material costs (Carvalho et al., 2019). Regarded as closely associated with Industry 4.0 (Kumar et al., 2019), predictive maintenance is attracting more and more scholarly attention. This is demonstrated by the fact that the number of published articles on this topic grows steadily in the last decade (Carvalho et al., 2019).

3.2 Failure prognostics

Prognostics is a technique aiming at predicting failures before they occur (Jardine et al., 2006). There are two main prediction types in equipment prognostics (Jardine et al., 2006). The most widely used type of prognostics is to predict the remaining useful life (RUL) of a piece of equipment, i.e., how much time is left before failure occurs given the current equipment condition and past operation profile (Jardine et al., 2006). Since remaining useful life is a random variable, its expectation can be expressed as

$$E[T - t | T \geq t, Z(t)], \tag{3.1}$$

where T denotes the random variable of equipment lifetime, t is the current age, and $Z(t)$ is the past condition profile up to the current time (Jardine et al., 2006).

However, in some situations, especially where a failure is catastrophic (e.g., nuclear power plant), it is more desirable to estimate the probability that a piece of equipment stays functional up to some point (e.g., next inspection or planned downtime) given the current equipment condition and past operation profile (Jardine et al., 2006). This motivation makes it necessary to estimate the *posterior probability*

$$P(T - t \leq \Delta | T \geq t, Z(t)), \tag{3.2}$$

where T denotes the random variable of equipment lifetime, t is the current age, $Z(t)$ is the past condition profile up to the current time, and Δ is the time horizon of interest (Prytz et al., 2013).

3.3 Failure reasons

In reliability engineering, failure patterns are often analyzed by using the failure rate function, which can be interpreted as the probability (risk) of failure in an infinitesimal unit interval of time (Finkelstein, 2008). Arguably the most popular failure rate function is the 'bathtub curve' (Klutke et al., 2003), see Figure 2, where $\lambda(t)$ is the failure rate.

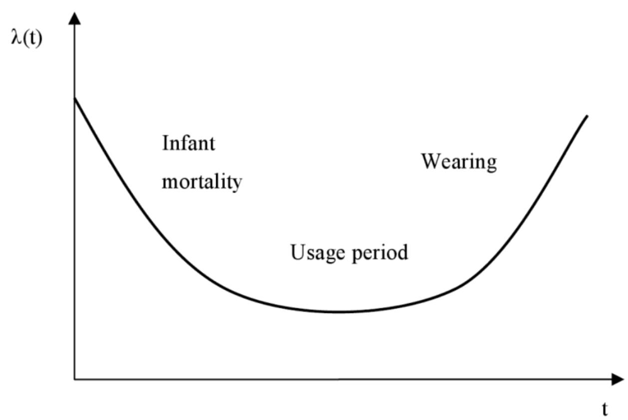


Figure 2: The bathtub curve (Finkelstein, 2008)

The bathtub curve suggests that age is associated with failure rate, and that their relationship is nonlinear. The risk of failure is relatively high at the early and late stages.

Khare et al. (2012) show that most failures in the automotive systems depend on age and accumulated usage. Besides, temperature can affect the working condition of a vehicle component. Pesaran et al. (2013) show that both cold and hot weather can damage Li-Ion batteries and even result in safety incidents.

It should be noted that, repairing or substituting some of the components in a complex system does not necessarily improve the condition of the overall equipment or a certain component. Louit et al. (2009) define two types of maintenances: (i) hazardous maintenance, in which the condition of the focused component is worsened after maintenance and (ii) imperfect repair, in which the condition of the focused component is improved after maintenance.

3.4 Data types and sources

There are two types of data used in predictive maintenance: real data and synthetic data (Carvalho et al., 2019). Synthetic data are often generated when building or evaluating models, such as when using SMOTE (Synthetic Minority Over-sampling TEchnique) to amplify the minority class (Chawla et al., 2002). However, most academic papers use real data (89%) instead of synthetic data (11%), and this may occur due to the fact that synthetic data are not able to represent a real-world application in most cases (Carvalho et al., 2019).

Data can come from different parties and are stored in various entities. The present study adopts the nomenclature of data sources in Prytz (2014). In a predictive maintenance project, data mainly come from two sources: on-board data and off-board data. The more prevalent data source is off-board data, which span from maintenance records to vehicle usage statistics (Prytz, 2014). In particular, maintenance records contain repair information and are kept by OEM (Original Equipment Manufacturer) authorized workshops and garages (Prytz et al., 2015). Since maintenance records are seldom designed for the purpose of data mining, they are by and large of low quality, consisting of errors and inconsistencies (Prytz et al., 2015). For example, it is sometimes hard to tell whether a specific date refers to the date of diagnosis or the date of repair (Prytz et al., 2015), as the technicians who input the data were not able to foresee that these data would be used for data mining. Thus, maintenance records pose a dilemma in which people either select a data set with inconsistencies and deal with missing values or limit the analysis to vehicles that have the desired features (Prytz et al., 2015).

On-board data include vehicle usage patterns and ambient conditions collected by sensors (Prytz et al., 2013; 2015; Hashemian, 2010). These data are either transmitted through a telematics gateway on the cloud or to the OEM, or stored inside a vehicle for retrieval at a workshop (Prytz, 2014). However, both approaches have limitations. The former approach is limited by unreliable real-time wireless communication, as discussed earlier. The latter approach is limited by the memory and computing power of the devices inside a vehicle. As a result, the data stored inside a vehicle are often simple statistics of various kinds (Prytz et al., 2015).

3.5 Data construction

To construct a data set suitable for estimating the posterior probability in Expression 3.2, the data construction techniques proposed by Prytz et al. (2013) and Prytz et al. (2015) are reviewed.

A new parameter, *time-to-repair*, is constructed. This parameter captures the time interval between when a truck visits the garage and when its component is replaced. It is straightforward to calculate time-to-repair for trucks on which the component fails (hereinafter referred to as 'faulty trucks'). For trucks on which the component does not fail (hereinafter referred to as 'non-faulty trucks'), when their components fail is not known. This part of data (the majority part) is right-censored. To overcome this, for a non-faulty truck, its component is assumed to fail immediately after its latest visit to the garage.

Estimating the posterior probability is a classification problem where the prediction model returns a value between 0 and 1. The response (target variable, dependent variable) is constructed by comparing time-to-repair and the prediction horizon, i.e., Δ in (3.2). Prediction horizon is defined as the period of interest for the predictive algorithm (Prytz et al., 2015). A binary variable, the response equals 1 (positive) when time-to-repair falls within the prediction horizon and 0 (negative) when time-to-repair is outside the prediction horizon. Obviously, constructing time-to-repair is for labeling examples as positive or negative, not for including a new feature into the models.

Prytz et al. (2013) mention that if some trucks have many occurrances in the data set, machine learning algorithms learn to almost exclusively consider individual idiosyncrasies, instead of indicators of component failure. To address this problem, Prytz et al. (2013) suggest that each vehicle can contribute at most two examples: one positive and one negative. However, the drawback is that the size of available data is reduced (Prytz et al., 2013). Prytz et al. (2015) propose another way to prevent models from learning individual patterns. They argue that data should be split into training set and test set on a *per vehicle* basis: if one or more examples from a given vehicle belong to the test set, no examples from the same vehicle can go to the training set, and vice versa.

3.6 Corrgram and the two-sample Kolmogorov-Smirnov test

Corrgram is a visual display of the correlation matrix of numerical data (Friendly, 2002). Correlation matrix is a square, symmetric matrix with rows and columns corresponding to variables, in which the off-diagonal elements are correlations between pairs of variables, and elements on the main diagonal are unity (Everitt and Skrondal, 2002, p. 107). Pearson correlation and Spearman's rank correlation are two widely used correlations. Unlike Pearson correlation, which assumes linear relationship, Spearman's rank correlation assesses monotonicity, thus being more robust and sensitive to non-linear relationships (Croxton and Cowden, 1939; Aitken, 1957; Dietrich, 2017). Correlation matrix can be ordered using various agglomerative clustering algorithms, including complete linkage, single linkage, and average linkage (Hastie et al., 2009, p. 523). Specifically, the complete linkage agglomerative clustering algorithm calculates the dissimilarity of two groups by finding that of the furthest pair (Hastie et al., 2009, p. 523).

The two-sample Kolmogorov-Smirnov test (two-sample K-S test) determines whether two samples appear to follow the same distribution (Berger and Zhou, 2014). Therefore, for any classification problem, the two-sample K-S test can be used to explore whether a numerical feature has different distributions in the positive and negative groups. This provides more insight into the data, and can be used as a tool for feature selection. For example, Prytz et al. (2015) decide whether a feature is included in the predictive maintenance model by the results given by the two-sample K-S test.

3.7 Class imbalance

The class imbalance problem arises when modeling data sets with discrete classes, where the proportions of the classes are highly skewed (Liljefors, 2020). This problem has been extensively studied due to its prevalence in the real world (Japkowicz and Stephen, 2002). An overview of the various techniques to tackle class imbalance can be found in He and Garcia (2009).

Many predictive maintenance use cases are troubled by class imbalance. The number of failure (positive) records is extremely small compared to that of normal (negative) records. It is even argued that the class imbalance problem in the field of predictive maintenance would never disappear, since as the aim of maintenance is to make failures rare events, one could expect that as maintenance improves, fewer failures should occur (Louit et al., 2009). When built on a data set whose target variable is hugely imbalanced, most algorithms will be biased towards the majority class and hence, perform poorly on the minority class (Prytz et al., 2015).

SMOTE (Synthetic Minority Oversampling TEchnique) is a well-known technique to fight class imbalance (Chawla et al., 2002; Branco et al., 2016). SMOTE over-samples the minority class by creating new synthetic data points, each of which is generated as a convex combination of a selected minority class instance and one of its k minority class neighbors (He and Ma, 2013). Compared to techniques which only under-sample the majority class, a combination of under-sampling the majority class and over-sampling the minority class can achieve better classifier performance (Chawla et al., 2002). In the area of predictive maintenance, studies that use SMOTE to address class imbalance include Prytz et al. (2015), Cerqueira et al. (2016), and Ramentol et al. (2016).

SMOTE has two design parameters: the number of neighbors to consider k, and the percentage of synthetic data to create (Prytz et al., 2015). The lower the value of k, the more similar synthetic data are to existing data. The second parameter determines to what degree the minority class is amplified.

3.8 Machine learning algorithms

Machine learning methods have been emerged as a promising tool in predictive maintenance use cases (Carvalho et al., 2019). However, the performance of predictive maintenance applications relies on the appropriate choice of the machine learning algorithm (Carvalho et al., 2019). According to Carvalho et al. (2019), the most employed machine learning algorithm in predictive maintenance is random forest (33%), followed by neural network-based methods (27%), support vector machine (25%), and k-means (13%), see Figure 3.

Lasso, random forest, and gradient boosting are reviewed below. These three machine learning algorithms are used to predict turbocharger failure in the present study.

3.8.1 Lasso

According to Everitt and Skrondal (2002, p. 242), the lasso estimate is defined by

$$\hat{\beta}^{lasso} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}. \tag{3.3}$$

Lasso is closely related to least squares regression (Everitt and Skrondal, 2002, p. 242), as shown in the first term in 3.3. However, lasso introduces a penalty term $\lambda \sum_{j=1}^{p} |\beta_j|$ which shrinks coefficients. $\lambda \geq 0$ is the tuning parameter and

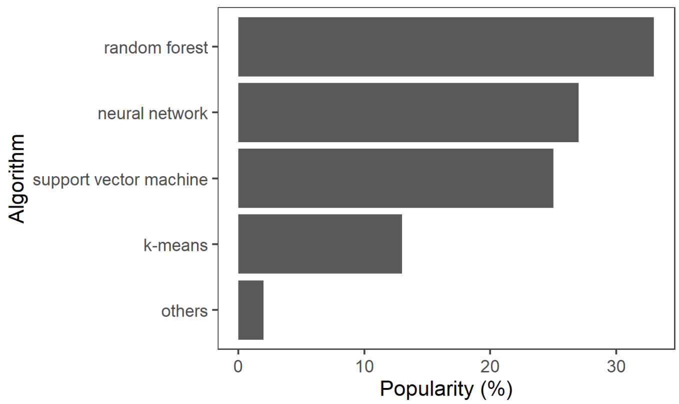


Figure 3: Most employed machine learning algorithms in predictive maintenance (Carvalho et al., 2019)

the penalty term, showing the complexity of the model. The larger the value of λ, the greater the amount of shrinkage is, and the less complex the model becomes (Hastie et al., 2009, p. 63). A unique feature of lasso is that when λ increases, some coefficients will shrink to exactly zero, and all coefficients will be zero when λ reaches a certain value (Hastie et al., 2009, pp. 69-73). This allows lasso to perform feature selection (Everitt and Skrondal, 2002, p. 242), making the model more interpretable (Hastie et al., 2009, p. 57). Moreover, lasso could increase prediction accuracy by sacrificing a little bit of bias to reduce the variance of the predicted values (Hastie et al., 2009, p. 57).

3.8.2 Random forest

Random forest is a collection of individual tree predictors (Breiman, 2001), each of which is built on a random subset of features and uncorrelated with any other trees. Because of the Kolmogorov's strong law of large numbers (Loeve, 1977), the out-of-sample error converges a.s. to a limit when the number of trees increases, avoiding the risk of overfitting (Breiman, 2001; James et al., 2013, p. 321). Although the bias of a random forest is the same as that of any of the individual trees, combining a number of individual trees can reduce variance, thus improving prediction (Hastie et al., 2009, p. 600). The built-in feature selection mechanism decorrelates individual classifiers. It is shown that in random forest, decorrelation is associated with lower out-of-sample error (Breiman, 2001).

Random forest estimates the out-of-sample error by computing the so-called *out-of-bag error*. For each observation in the training data set, its own random forest predictor is constructed such that all individual trees are built on bootstrap samples in which the observation does not show up (Hastie et al., 2009, p. 593). Despite the fact that both out-of-sample error and out-of-bag error converge as the number of trees increases, a random forest model can only have a finite number of trees, and out-of-bag error always fluctuates before convergence. Growing a forest with an enormous number of trees is both time-consuming and unnecessary.

When splitting a node, $m \leq p$ out of all p features are considered. If $m = p$, the unique feature selection mechanism no longer plays a part, and random forest degrades into bagging (Montillo, 2009). The inventors recommend that the default value for m is $\sqrt{p}$ for classification problems (Hastie et al., 2009, p. 592). However, if the ratio of relevant features to all available features is small, the chance can be small that the relevant features are selected at each split (Hastie et al., 2009, p. 596). It is argued that the best value of m depends on the specific problem, and should be considered as a tuning parameter (Hastie et al., 2009, p. 596).

For each of the individual trees in a random forest, fewer splits could result in lower variance and higher bias (James et al., 2013, p. 307), making tree depth a parameter worth tuning. Segal (2004); Duroux and Scornet (2018) and Duroux and Scornet (2018) show that regulating tree size can result in gains in prediction performance, and that doing so incurs almost no additional cost due to the intrinsic recursive nature of random forest. However, Hastie et al. (2009, p. 596) suggest that using full-grown trees seldom costs much, and reduces computation time as a result of less tuning parameters.

3.8.3 Gradient boosting

Similar to that for random forest, the motivation for boosting is a procedure that combines the outputs of many base learners to produce a powerful ensemble (Natekin and Knoll, 2013; Hastie et al., 2009, p. 337). In most cases, decision trees are selected as the base learner of boosting methods because of their interpretability (if they are small), built-in feature selection mechanism, invariance under strictly monotone transformations of features, and fast construction (Hastie et al., 2009, p. 352). However, while all individual trees are independently built in a random forest, boosting methods build new trees sequentially.

For each iteration, the weights of the observations that are misclassified in the previous iteration get increased, and those of the observations that are correctly classified get decreased (Hastie et al., 2009, p. 338). Thus, each successive tree is forced to focus on the observations that are wrongly classified by previous trees in the sequence (Hastie et al., 2009, pp. 338-339).

Boosting seeks to empirically minimize a loss function (Zhang et al., 2005). In its original version as well as AdaBoost, one of the family members of boosting methods, this is realized by numerical optimization in a greedy fashion (Zhang et al., 2005; Vezhnevets and Vezhnevets, 2005; Hastie et al., 2009, p. 356). In contrast, gradient boosting performs numerical optimization by constructing a tree at the latest iteration in a way that the predictions given by the new tree are maximally correlated with the negative gradient of the loss function (Natekin and Knoll, 2013; Hastie et al., 2009, pp. 359-360).

Learning rate ν and the number of boosting iterations M are two of the tuning parameters of gradient boosting algorithm. Learning rate ν is defined such that the contribution of each tree is scaled by a factor $0 < \nu \leq 1$ when the latest tree is added to the current ensemble (Hastie et al., 2009, p. 364). It is shown both theoretically and empirically (Friedman, 2001; Ridgeway, 1999; Breiman, 1999; Hastie et al., 2009, p. 365) that small values of ν are conducive to lower generalization error. Hastie et al. (2009, p. 365) argue that the best strategy appears to be a combination of a very small $\nu < 0.1$ and an M chosen by early stopping, a regularization technique which tackles overfitting by stopping iterations before global minimum is approached (Hastie et al., 2009, p. 364).

Gradient boosting framework has various implementations, including XGBoost (Chen and Guestrin, 2016), CatBoost (Dorogush et al., 2018) and LightGBM (Ke et al., 2017). XGBoost is claimed by its authors to be a "scalable end-to-end tree boosting system" (Chen and Guestrin, 2016).

3.9 Model evaluation metrics

In a binary classification problem, the receiver operating characteristic (ROC) curve and the precision-recall (PR) curve are two commonly-used methods to assess and compare binary classifier performance (Cao et al., 2020). However, for some problems, researchers face the challenge of evaluating classification with imbalanced response, where one class (usually negative) greatly outnumbers the other (Cao et al., 2020). When class imbalance arises, ROC curve has been shown to be misleading (Kleinbaum et al., 2002; Davis et al., 2005; Davis and

Goadrich, 2006; Jeni et al., 2013; Saito and Rehmsmeier, 2015; Prytz et al., 2015). Recently Cao et al. (2020) prove that PR curve is ineffective on some imbalanced data sets.

Finding a perfect evaluation metric for predictive maintenance is not straightforward. It is desirable to discover as many unhealthy components as possible, but labeling healthy components as unhealthy incurs additional costs. Thus, both precision and recall need to be as high as possible. However, it is tempting to use one metric instead of two, not only for conciseness, but also for comparing candidate models. F1-score synthesizes precision and recall by calculating their harmonic mean (Guns et al., 2012), but treats them as equally important. The general version of F-score, F_β, is calculated as

$$F_\beta = (1+\beta)\frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}. \tag{3.4}$$

F_β "measures the effectiveness of retrieval with respect to a user who attaches β times as much importance to recall as precision" (Van Rijsbergen, 1974). When predicting air compressor failure, Prytz et al. (2013) use F_β as the evaluation metric, and set β to 0.5 based on the fact that precision is more important than recall.

Another evaluation metric for classification problem is the Matthews correlation coefficient (MCC), which is calculated as

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(FP+TN)(TN+FN)(FN+TP)}}. \tag{3.5}$$

To get a high score in MCC, the model has to make correct predictions both on the majority of the negative cases and on the majority of the positive cases, independently of their ratios in the overall data set (Chicco and Jurman, 2020; Cao et al., 2020). MCC is easy to interpret: a coefficient of +1.0 indicates perfect prediction; 0.0 means random guessing; -1.0 indicates total disagreement between prediction and observation (Matthews, 1975; Cao et al., 2020). It is encouraged that for any binary classification problem, each test performance should be evaluated through the Matthews correlation coefficient, instead of F-score and accuracy (Chicco, 2017).

3.10 Cost savings

Prytz et al. (2013) propose a formula to estimate the cost savings when a prediction model is adopted. The cost savings C_{save} is calculated as

$$C_{save} = TP \cdot (C_u - C_p) - FP \cdot C_p, \tag{3.6}$$

where TP denotes true positive (correctly discovering an unhealthy component), FP denotes false positive (erroneously labeling a healthy component as unhealthy), C_u denotes costs of sudden breakdown, and C_p denotes costs of replaced component. There are no cost savings associated with true negative (correctly identifying a healthy component), but the cost savings associated with false negative (erroneously labeling an unhealthy component as healthy) is indirect. Failing to discover an unhealthy component is equivalent to maintaining the *status quo*, neither incurring additional costs nor saving costs.

4 Data and methods

This section first describes the data used in the present study, and the procedures to transform the original data into what can be used for modeling. Modeling approaches are then presented. Finally, a formula for estimating the cost savings is given.

4.1 Data

4.1.1 Data source

The data set used in the present study is retrieved from the company's garage, where trucks are inspected and repaired. The original data set contains more than 400,000 rows and more than 20 columns. The data set contains information regarding historical maintenance events. When a truck is sent to the garage for maintenance, invoices which record that specific maintenance event are generated. Each row represents a certain component which is replaced in a maintenance event. In most cases, one maintenance event is represented by multiple rows, for it is rare that only a single component is replaced in a maintenance event.

The columns contain maintenance information and truck information. Maintenance information includes invoice number, invoice creation date (maintenance date), invoice status, the component name, component amount, component price, invoice status, and invoice description. Truck information includes engine code, license plate, license plate acquisition date, age, cumulative mileage, engine type, vehicle type, and whether a truck's maintenance services are covered by a monthly fee.

4.1.2 Data cleaning

Like Prytz et al. (2015), the present study also faces data quality problem. Duplicate rows account for 12.8% of the data set. The reason for duplication is unknown. However, even if duplication means a larger amount of units, the present study only cares about what, instead of how many, components are replaced in a single maintenance event. Hence, these duplicate rows are removed.

Whether an invoice is actually executed is shown in the invoice status column. Rows whose invoice status belongs to one of "depreciated", "canceled", and "expired" are not associated with actual maintenance operations. These rows account for 0.7% of the data set, and are removed.

According to the Netherlands Vehicle Authority (Dutch: Rijksdienst voor het Wegverkeer, RDW), a license plate identifies the vehicle, and it remains unchanged even if the owner is changed (RDW, 2021). Thus, a truck can only have one license plate and the acquisition date associated with the license plate. The license plate acquisition date indicates the first day that a truck operates on the road. However, some trucks in the data set have more than one license plate acquisition date. Human errors are likely to be the reason. Since license plate acquisition date is used to calculate the time interval between when a truck first operates on the road and when a truck is maintained at the garage, it plays an important role in feature engineering. 1.6% of the rows are associated with trucks with abnormal license plates. To ensure data quality, all these rows are removed from the data set.

In the majority of cases (76%), invoice creation date is preceded by license plate acquisition date, meaning that a truck is maintained after it operates on the road. But in some cases (24%), it is the other way around. This happens when a truck is maintained before it operates on the road, or loosely, before it is sold to a client. These maintenance events are filtered out, as they are not associated with on-road operations.

Some trucks in the data sets have more than one vehicle type, which makes no sense and is definitely due to input error. 0.2% of the rows are associated with this kind of truck. These rows are removed for data quality. Besides, vehicle type is one of the features in the models.

Some rows in the data set do not have cumulative mileage. Moreover, in some cases, two rows which have the same invoice number have disparate mileage readouts. As cumulative mileage is a feature in the present study, these rows (22.5%) are removed.

Trucks that have only one maintenance event in the data set are discarded, because they are not representative (Prytz et al., 2013), and account for 0.1% of the data set. Turbocharger is usually replaced at most once during the whole lifetime of a vehicle (Prytz et al., 2013), and only 8.7% of the faulty trucks in the data set have more than one turbocharger replacement. Thus, it is reasonable to focus on the first turbocharger replacement. All maintenance events which take place after the first turbocharger replacement are not considered.

4.2 Methods

4.2.1 Feature engineering

From the textual description which accompanies each maintenance event, most turbocharger replacements are associated with safety incidents, such as engine failure and stranding. Few turbocharger replacements are due to reasons like diagnosis or general periodic inspection (Dutch: algemene periodieke keuring). These two types of reasons are regarded as equivalent – they are all labeled as turbocharger failure. The reason is that the vast majority of trucks do not have their turbochargers replaced. This fact accentuates the meaning of turbocharger replacement, whatever reason the replacement is due to.

Cumulative mileage, month, and age are included in the analysis. Defined as the time between the license plate acquisition date and the maintenance date, age indicates how long a truck has operated on the road. As weather condition changes all year round, the month of the maintenance event is included as a substitute for ambient environment.

There are four distinct vehicle types in the data set. Two of these types account for a very small portion of the data (0.8%) compared to the other two major types. To address the imbalance in vehicle type, the two types with the smallest numbers are combined to form a new category. This approach does not mean that these two types are so similar that they can be regarded as one category. Categories are also combined in the column of engine type, but based on similarity in engine.

Driving behaviors could change when a truck's maintenance services are covered by a monthly fee, thus having a different impact on the mechanical systems. Hence, whether a truck is warranted, *subscription*, is included as a feature.

When a truck is maintained, no matter how many components are replaced, the overall mechanical and electronic systems are renewed, and this could have an impact on the health status of turbocharger. Considering this, the time interval between two consecutive maintenance events is included as a feature.

The present study relies heavily on expert features, i.e., truck components that are assumed to correlate with turbocharger failure by technical experts at the company. Twenty-seven components are identified and included. Not all of them indicate one specific model of a component. Some models are similar to each other and are grouped into one component. For example, air filters of different specifications are all considered as air filter. Components that belong to the same system or share the same purpose are also grouped. For example, coolant pipe, coolant pump and other coolant-related components are grouped into one category, as all of them are responsible for circulating coolant. For each truck, the time interval between the previous time when a certain component is replaced and the current maintenance event, or in other words, the age of a certain component, is calculated.

There are 34 features to be considered, with four categorical and 30 numerical. The response, *unhealthy*, is constructed using the method proposed by Prytz et al. (2013), see Section 3. The prediction horizon is set to eight weeks (56 days). This time horizon gives the company and its clients sufficient time to plan and coordinate, thus is a reasonable choice from a business perspective.

The data set is then grouped by maintenance event so that each row represents a single event. The final data set is constructed in a way that each faulty truck offers one positive example, and each non-faulty truck offers one negative example. The final data set has 3,284 rows coming from 3,284 unique trucks. 80% (2,626 rows) of the data are used as the training set, and the rest (658 rows) as the test set. The test set has 10 positive examples, accounting for 1.5% of the data set.

4.2.2 Modeling

The present study uses three machine learning models: lasso, random forest, and gradient boosting. To avoid the influence of unit of measurement, all numerical features in the training and test sets are normalized based on the values in the training set.

Since positive examples only account for 1.4% of the data set, Synthetic Minority Over-sampling TEchnique (SMOTE) is used to amplify the portion of positive examples. The number of neighbors to consider, k, is set to nine. Synthetic data are created, making the size of the minority class five times larger. The size of the majority class is halved for under-sampling. After these, the new training set consists of 1,475 examples, with 1,295 (87.8%) negative and 180 (12.2%) positive. One fifth of the negative examples are real data, while the rest are synthetic data.

In lasso, the shrinkage parameter λ in 3.3 is tuned. The number of trees and the number of features to consider at each split are tuned in random forest. While the number of trees and tree depth are tuned in gradient boosting, early stopping is introduced. The present study uses XGBoost (Chen and Guestrin, 2016), one of the implementations of gradient boosting framework.

To find the best combination of tuning parameters (in lasso there is only one), 10-fold cross validation is performed on the training set to calculate the MCC (Matthews correlation coefficient) for each combination. For each algorithm, the most parsimonious model whose error is no more than one standard error above the error of the best model is selected as the final model (Hastie et al., 2009, p. 244). The MCC's of three finalized models are compared on the test set where the threshold varies.

4.2.3 Estimation of cost savings

When a fleet operator adopts the prediction model, the estimated cost savings can be imputed as

$$C_{save} = \left(\beta - \frac{1}{\text{precision}}\right) \frac{n \cdot C_p \cdot TP}{TP + FP + TN + FN}, \tag{4.1}$$

where C_{save} denotes cost savings; n is the number of trucks that the fleet operator owns; C_p is the cost of planned turbocharger replacement; β is the ratio of cost of sudden breakdown to cost of planned turbocharger replacement; and TP, FP, TN and FN denote the four elements of the confusion matrix on the test set.

Equation (4.1) is based on an important premise: the fault frequency of turbocharger is equal to the proportion of positive examples in the test set. The fault frequency of turbocharger refers to the percentage of turbochargers that fail *within a given time* (Prytz et al., 2013). Hence, when the time interval increases, fault frequency also increases. This means that the ratio of positive to negative examples in the test set can be set to an arbitrary value, so long as it makes sense from a business perspective. But in the present study, fault frequency, i.e., the ratio of positive to negative examples in the test set, is not adjusted. Derivation of Equation (4.1) is given in Appendix A.

5 Analysis

After finalizing the data set, summary statistics of the whole data set (before splitting, over-sampling and under-sampling) are presented. The best tuning parameters for each algorithm are obtained by performing 10-fold cross validation on the training set. The test set is used to compare the three tuned models in terms of Matthews correlation coefficient (MCC) and cost reduction.

5.1 Data description

The data set contains 3,284 rows, with 46 positive (unhealthy) examples and 3,238 negative (healthy) examples. A positive example indicates that the turbocharger will fail within 56 days. There are four categorical features and thirty numerical features. The distributions of the categorical features in the positive and negative groups are shown in Figure 4. It is observed that no categorical feature has significantly different distributions in the two groups.

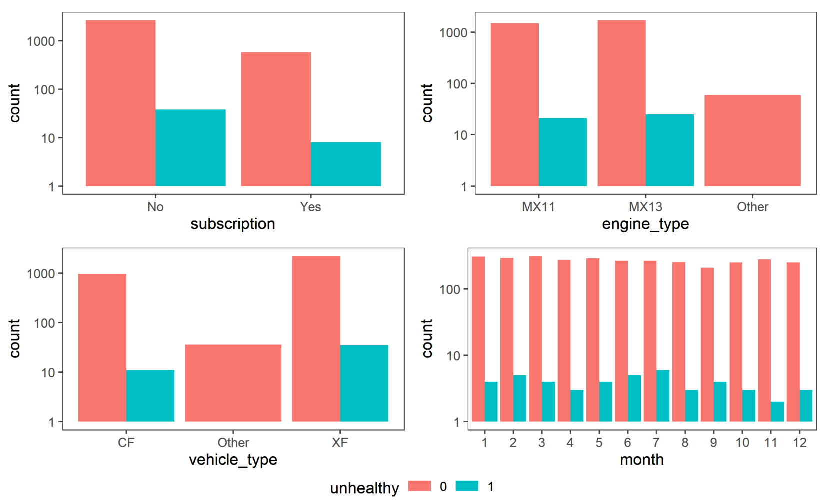


Figure 4: Distributions of categorical features

Summary statistics of the numerical features are presented in Appendix B. A feature whose name begins with '*prv*' refers to the age of the component. Figure 5 shows the corrgram of all numerical features. The correlation matrix is calculated using the Spearman's rank correlation coefficient. All correlation coefficients are positive, meaning that for every pair of features, an increase in the value of one feature is by and large associated with an increase in the value of the other feature. This is due to the fact that all components are getting older when a truck operates on the road. For example, the age of wear indicator is highly correlated with the age of brake pad set, suggesting that these two parts are frequently replaced together.

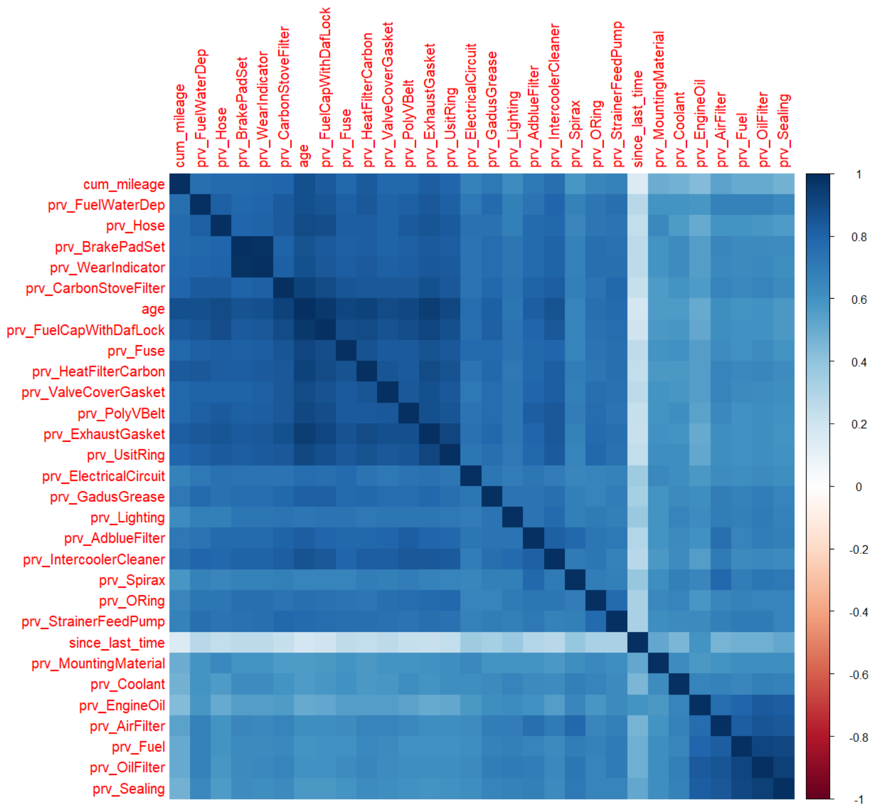


Figure 5: Corrgram of numerical features

To detect the features which are distributed differently in the positive and negative groups, the two-sample Kolmogorov-Smirnov test (two-sample K-S test) is performed for each numerical feature. The results of the two-sample K-S test is given in Table 2. Ten features are significant at the 0.05 level, i.e., they are distributed differently in the positive and negative groups. Figure 6 shows the kernel density estimations (KDEs) of cumulative mileage and the age of fuel cap with DAF lock. On average, positive examples have higher cumulative mileages than do negative examples. Examples with cumulative mileages higher than 500,000 km account for a small portion of the negative group, but a considerable portion of the positive group. Examples with age of fuel cap with DAF lock older than 2,500 days account for a negligible portion of the negative examples, but a visible portion of the positive examples. Both the two-sample K-S test and the KDEs show that maintenance records can provide information that can be used to separating positive and negative examples.

Table 2: Results of the two-sample Kolmogorov-Smirnov test

Feature	p-value
cum_mileage	0.0004
prv_FuelCapWithDafLock	0.001
prv_CarbonStoveFilter	0.001
prv_HeatFilterCarbon	0.001
age	0.002
prv_Coolant	0.018
prv_BrakePadSet	0.027
prv_WearIndicator	0.032
prv_Fuse	0.034
prv_GadusGrease	0.048

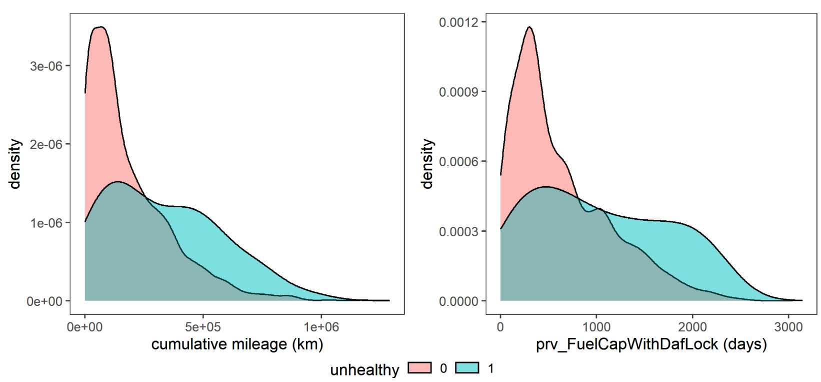


Figure 6: KDEs of cumulative mileage and fuel cap with DAF lock

5.2 Failure patterns

Figure 7 shows the failure patterns of turbocharger. As there are 46 faulty trucks in the data set, the sample size is 46. All turbochargers fail within 2,500 days after license plate acquisition date, but no time period has a significantly higher or lower turbocharger failure rates than do other age periods, given the small sample size. No turbocharger failure happens after 1 million km, and most turbocharger failures happen before a cumulative mileage of 700,000 km is reached. To know whether turbocharger failures are concentrated in certain months, Pearson's chi-squared test is performed. The null hypothesis is that turbocharger failure happens equally likely across all months. The result given by the Pearson's chi-squared test ($p = 0.981$) fails to reject the null hypothesis at the 0.05 level of significance, suggesting that no month sees more or fewer turbocharger failures than do other months.

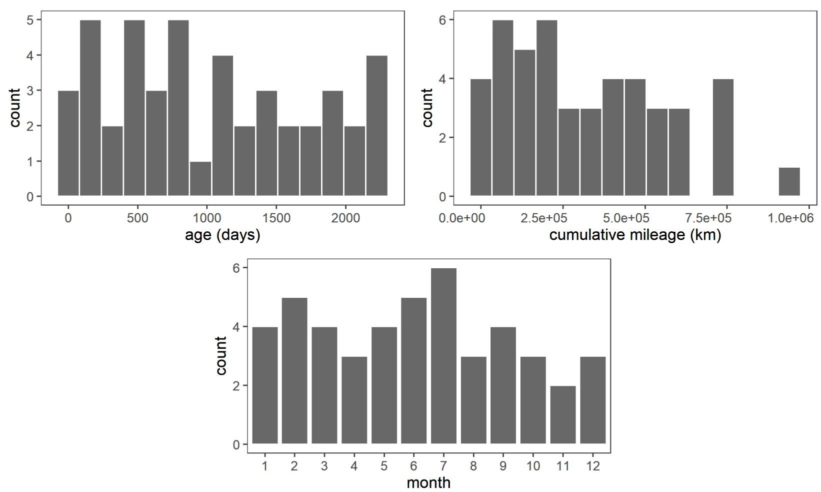
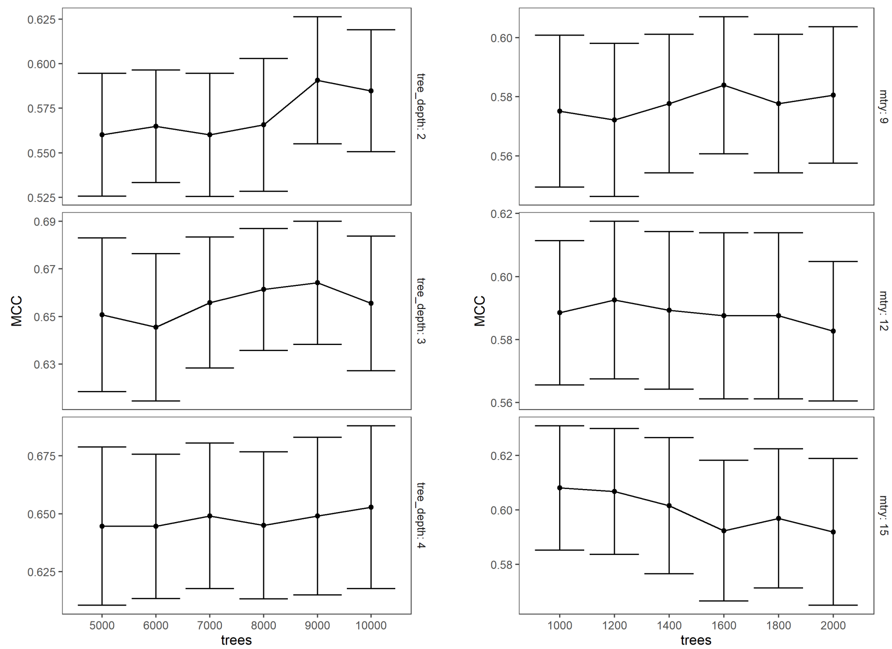


Figure 7: Failure patterns of turbocharger

5.3 Model performance

Figure 8 shows the impact of the tuning parameters on MCC of the training set. For gradient boosting, the best performing tree depth is three, with which five out of six models have an MCC above 0.65. In terms of MCC, the best performing gradient boosting model has 9,000 trees and a maximum tree depth of three, and achieves an MCC of 0.664. However, using the one-standard-error rule, the selected gradient boosting model has 5,000 trees and a maximum tree depth of three, and achieves an MCC of 0.651.

Compared to gradient boosting, random forest by and large performs worse on the training set, with only three models achieving an MCC above 0.6. The best number of features to consider at each split, *mtry*, is 15. The best performing random forest model has 1,000 trees and an *mtry* of 15, and achieves an MCC of 0.608. However, using the one-standard-error rule, the selected random forest model has 1,000 trees and an *mtry* of 12, and achieves an MCC of 0.589.



(a) Gradient boosting

(b) Random forest

Figure 8: Impact of tuning parameters on MCC of the training set

Figure 9 shows the impact of the penalty in lasso on the MCC of the training set. As penalty increases, MCC increases slightly, and reaches its peak shortly before penalty reaches 10^{-3}. After this, MCC decreases significantly, and dives below zero as penalty continues to increase. The reason why MCC is missing when penalty is between 10^{-2} and 10^{-1} is that the lasso model labels all observations as negative, thus making the denominator in 3.5 zero. The best performing lasso model has a penalty of 3.17×10^{-4}, and achieves an MCC of 0.316. However, using the one-standard-error rule, the selected lasso model has a penalty of 9.78×10^{-4}, and achieves the same MCC, i.e., 0.316.

The best performing algorithm on the training set is gradient boosting, followed by random forest. Compared to these two algorithms, lasso performs much worse on the training set. The best lasso model achieves an MCC approximately half of those achieved by the best gradient boosting and random forest models.

The test set is used to examine the three selected models. Three selected

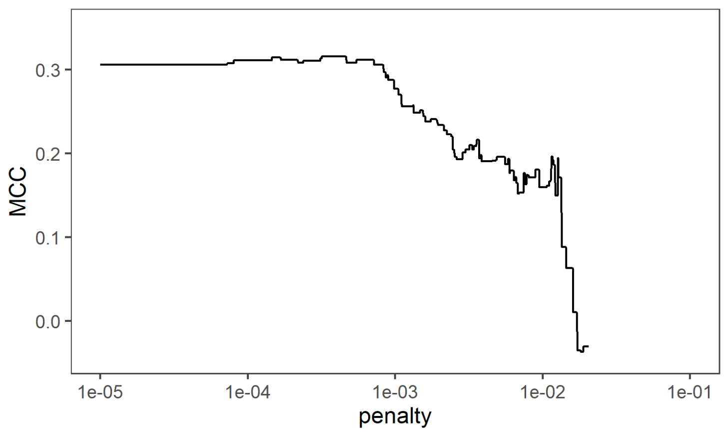


Figure 9: Impact of penalty on MCC of the training set

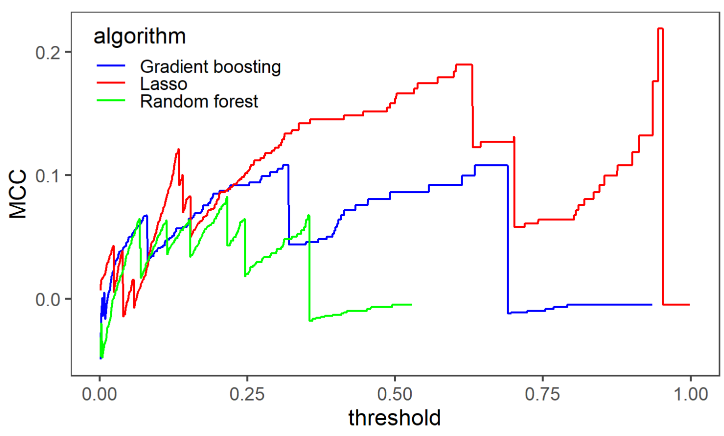


Figure 10: Impact of threshold on MCC of the test set

models are compared by MCC on the test set where the threshold increases from 0.001 to 0.999. When the threshold increases, both the number of true positives and that of false positives decrease. The impact of threshold on their performances is shown in Figure 10. Except when the threshold is slightly above zero, all three algorithms achieve an MCC above zero. However, all three algorithms perform worse on the test set than the training set. This is mainly due to two reasons. First, the test set is not used for modeling. Second, and more importantly, the proportion of positive examples in the test set (1.5%) is much smaller than that in the training set (12.2%). The worst performing algorithm is random forest, which achieves the lowest MCC most of the time and does not generate positive predictions when the threshold is higher than 0.5. Gradient boosting outperforms random forest, and achieves an MCC above 0.1. The best performing algorithms is lasso, which has the highest MCC among all algorithms when the threshold is higher than 0.2. Lasso keeps generating positive predictions until the threshold reaches 0.96. The highest MCC in Figure 10 is above 0.2, achieved by lasso when the threshold is set to 0.95.

Table 3 shows the confusion matrix for the test set based on the lasso with a threshold of 0.95. One tenth of the unhealthy turbochargers are discovered, making the recall 0.1. Half of the retrieved turbochargers are actually healthy, making the precision 0.5. The lasso model with the optimal threshold achieves a high precision, but there is room for improvement on recall.

Table 3: Confusion matrix for the test set (lasso, threshold=0.95)

	Predicted	
Actual	Negative	Positive
Negative	647	1
Positive	9	1

Figure 11 shows the ten most important features in lasso. The ages of some components are positively correlated with turbocharger failure, while those of others negatively correlated. This might be partly explained by hazardous maintenance (Louit et al., 2009), which means that the health condition of turbocharger is worsened after certain components are maintained.

Lasso, the worst performing algorithm on the training set, outperforms the others on the test set. Although it is not straightforward to conclude whether random forest and gradient boosting overfit, a plausible explanation is that the

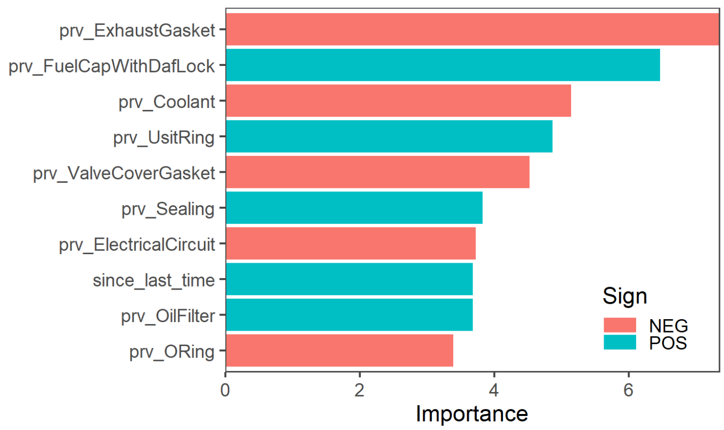


Figure 11: Most important features in lasso

relationship between the *odds* of turbocharger failure and available features can be approximated by linearity.

5.4 Cost reduction

To make the estimation general, in Equation (4.1), n is set to 50, the number of trucks that a typical fleet operator owns. The exact cost of planned turbocharger replacement cannot be disclosed due to privacy reasons, but is set to €3,000 for calculation. The cost of sudden breakdown is difficult to estimate, as it depends on several factors. As the company's garage is situated in the Netherlands, the cost is much higher when the truck breaks down in France than in Belgium, because of the toll roads and mandatory recovery vehicles. The cost is also higher when the truck is carrying live animals than standard containers. Besides, evenings and weekends are more expensive time to break down. To get an average estimate, the cost of sudden breakdown is set to three times as that of planned turbocharger replacement, i.e., $\beta = 3$. Since the test set has 658 rows, $TP + FP + TN + FN = 658$.

After inserting the above numbers, Equation (4.1) can be written as

$$C_{save} = 228 \cdot TP \cdot \left(3 - \frac{1}{\text{precision}}\right). \tag{5.1}$$

Equation (5.1) shows that cost savings are subject to two (interconnected) factors: the number of true positives TP, and the precision. Specifically, precision determines whether the cost savings are larger than zero, thus having a larger impact on whether the company adopts the proposed predictive maintenance solution.

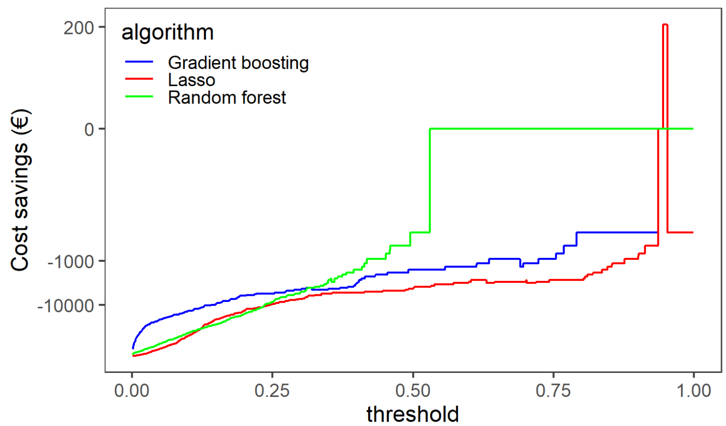


Figure 12: Impact of threshold on cost savings

The impact of threshold on cost savings is shown in Figure 12. When the threshold is low, cost savings are below zero due to a large number of false positives, i.e., unnecessary replacements. As random forest stops generating positive predictions when the threshold is higher than 0.5, the cost savings level off at zero. Neither random forest nor gradient boosting offer cost reduction. Lasso can bring cost savings more than €200, but it can only offer benefits when the threshold is around 0.95. As turbochargers are expensive, cost savings can only be realized when the algorithm is very sensitive to unhealthy examples and indifferent to healthy examples.

The algorithm which achieves the highest MCC in Figure 10, lasso, also offers the largest cost savings in Figure 12. This proves the effectiveness of MCC, as it evalutates a model based on its performance on both classes.

6 Discussion and conclusion

6.1 Main findings

The failure patterns of turbocharger can be discovered through the analysis above. All turbocharger failures happen before the trucks reach 2,500 days old. Most turbocharger failures happen before a cumulative mileage of 750,000 km, and all failures happen before a cumulative mileage of 1 million km. No certain age period, no certain cumulative mileage period, and no month see significantly more or fewer turbocharger failures.

It is hardly possible to predict turbocharger failure by gaining insight solely from faulty trucks. However, both the results given by the two-sample Kolmogorov-Smirnov test and the kernel density estimations show that maintenance records are able to provide information associated with turbocharger failure. Since all machine learning algorithms achieve an Matthews correlation coefficient (MCC) above zero, it can be concluded that factors correlated with turbocharger failure can be detected by machine learning algorithms. On a data set in which 1.5% of the examples are positive, lasso can achieve an MCC of 0.219. One tenth of the unhealthy turbochargers can be discovered, while half of the retrieved turbochargers are actually healthy.

Within a time interval in which fault frequency (see Section 4) equals 1.5%, lasso can bring cost savings of €228 for a fleet operator who owns 50 trucks. It should be noted that, the estimated cost savings in Figure 12 is conservative, as the estimations are based on the premise that turbocharger replacement is executed completely according to prediction results. The predictive maintenance solution proposed in the present study should be adopted as an aid to technicians, who can reduce the number of false positives based on their expertise. In this case, the threshold could be set to lower values to discover more unhealthy turbochargers.

6.2 Discussion

The present study reaffirms the argument that failure patterns can be detected from maintenance records by machine learning algorithms (Haarman et al., 2017). Compared to Prytz et al. (2013) and Prytz et al. (2015), the present study realizes cost savings under more restricted condition – maintenance records are the only data source. Besides, Prytz et al. (2013) construct the test set in a way that positive examples account for 5% of the test data, but the ratio in the present study is 1.5%. Thus, it is more difficult to achieve cost savings in the present study, as cost savings are sensitive to false positives. Mashhadi et al. (2020) only consider data related to faulty trucks, whereas the present study takes both faulty and non-faulty trucks into account, making the results more general and closer to real-world applications.

The best performing machine learning algorithm in the present study, lasso brings cost savings to the company when the other two algorithms do not. However, lasso is not among the most popular machine learning algorithms in the field of predictive maintenance (Carvalho et al., 2019), see Figure 3. The present study shows that lasso can effectively perform feature selection, and improve prediction results.

Unlike the studies in Table 1, the present study does not use information directly related to the working condition of turbocharger. Nevertheless, machine learning algorithms still successfully detect patterns that correlate with turbocharger failure. The results show that the working condition and health status of different components inside a truck are interconnected, and that it is worthwhile considering other components when modeling a certain component.

The evaluation metric is area under the receiver operating characteristic curve (AUC) in Liljefors (2020), and F-score in Prytz et al. (2013), Nowaczyk et al. (2013), and Prytz et al. (2015). Different from these studies, the present research uses MCC as the evaluation metric, and proves its effectiveness. When the lasso model achieves the highest possible MCC in Figure 10, the threshold also maximizes the cost savings, see Figure 12.

6.3 Managerial implications

It has been proved that the maintenance records which the company owns can lead to a feasible predictive maintenance solution, and that cost savings can be achieved even under strict assumptions. Hence, it is advised that the introduction of predictive maintenance should be put on the agenda by the company.

A committee needs to be established to oversee the introduction of predictive maintenance. It is imperative that data quality should be improved. Manual data entry should be either standardized or automated to reduce errors and facilitate data mining.

The list of components provided by the technicians turns out to be crucial to modeling turbocharger failure. Their expertise by no means plays a less important role in the age of data analytics. Instead, the cooperation between technicians and data scientists is the key to a successful predictive maintenance application.

The prediction model needs to be rerun to renew the estimated health status of turbocharger, and the time interval should be as small as possible. When a turbocharger is predicted as unhealthy, the company needs to contact the fleet operator and invites her to send the truck to the garage within eight weeks. Technicians then inspect the turbocharger as well as other components. Whether to replace the turbocharger or other components is at the discretion of the technicians. Technicians' domain knowledge is likely to reduce unnecessary replacements, and bring more cost savings.

The present study lowers the barrier to the entry of fleet operators, automotive companies, and relevant practitioners into the field of predictive maintenance. Compared to relevant studies, the present study relies on more prevalent data source, and uses less complicated techniques for data modeling. It is encouraged that relevant companies launch pilot projects to assess their capacities and prepare for more advanced predictive maintenance solutions.

6.4 Limitations and future research

Although the present study shows that cost savings can be obtained, it is not entirely clear how long a certain faulty frequency stands for. Thus, the results are not able to estimate the monthly, quarterly or annual cost savings for a fleet operator. Besides, the estimated cost savings are for the fleet operators. The cost savings that the proposed solution brings to the company are not estimated.

The present study assumes that all trucks are maintained only at the company's garage. This assumption is reasonable for trucks under warranty, but the degree to which it holds for trucks not under warranty is hard to estimate. When a truck visits another garage, the reliability of the data related to subsequent maintenance events is at question.

The time interval between two consecutive maintenance events varies greatly, making it sometimes hard to determine what rows belong to a certain maintenance event. It is likely that a truck stays at the garage for more than one day, but chances are that a truck needs to be maintained shortly after it leaves the garage. To simplify analysis, no lower limit for time interval is stipulated, but there is room for improvement here.

One direction for future research is to take textual descriptions into account.

Textual descriptions, input by technicians when performing maintenance tasks, are common in maintenance records. However, due to lack of standardization of data entry, textual descriptions are usually messy, including ambiguous abbreviations and typos. Extracting information from these textual descriptions requires domain knowledge, but it is likely that information related to component failure is hidden in these descriptions.

Another direction is to add more variables into cost savings estimation. The cost of sudden breakdown depends on several factors, each of which can be approximated by a continuous or discrete probability distribution. The extent to which false positives are reduced by domain knowledge can also be introduced. In this case, the estimated cost savings for a certain threshold are a distribution rather than a single number.

The potential of maintenance records has not been fully exploited, as the present study focuses only on turbocharger. However, there are other crucial components whose failure leads to sudden breakdown. Future research could take all crucial components into account to further increase truck uptime and cost savings.

6.5 Conclusion

The present study proves that it is feasible to introduce predictive maintenance based solely on maintenance records. It has been shown that factors correlating with turbocharger failure exist in the maintenance records, and that turbocharger failure can be predicted by machine learning algorithms. Truck uptime gets increased, and cost savings can be obtained by fleet operators even under strict assumptions. When the proposed predictive maintenance solution is adopted, the company will be able to increase customer satisfaction, and find a new way of being competitive within the industry. Detailed and tractable steps of analysis are presented. The present study also recommends relevant companies to introduce predictive maintenance, and calls for more research into the role of maintenance records in predictive maintenance.

References

Aitken, A. C. (1957). *Statistical mathematics*, Oliver & Boyd.

Berger, V. W. and Zhou, Y. (2014). Kolmogorov–Smirnov test: Overview, *Wiley StatsRef: Statistics Reference Online* .

Bradbury, S., Carpizo, B., Gentzel, M., Horah, D. and Thibert, J. (2018). Digitally enabled reliability: Beyond predictive maintenance, *Technical report*, McKinsey & Company.

Branco, P., Ribeiro, R. P. and Torgo, L. (2016). UBL: An R package for utility-based learning, *arXiv preprint arXiv:1604.08079* .

Breiman, L. (1999). Prediction games and arcing algorithms, *Neural Computation* **11**(7): 1493–1517.

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

Cao, C., Chicco, D. and Hoffman, M. M. (2020). The MCC-F1 curve: A performance evaluation technique for binary classification, *arXiv preprint arXiv:2006.11278* .

Carvalho, T. P., Soares, F. A., Vita, R., Francisco, R. d. P., Basto, J. P. and Alcalá, S. G. (2019). A systematic literature review of machine learning methods applied to predictive maintenance, *Computers & Industrial Engineering* **137**: 106024.

Cerqueira, V., Pinto, F., Sá, C. and Soares, C. (2016). Combining boosted trees with metafeature engineering for predictive maintenance, *International Symposium on Intelligent Data Analysis*, Springer, pp. 393–397.

Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique, *Journal of Artificial Intelligence Research* **16**: 321–357.

Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system, *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 785–794.

Chicco, D. (2017). Ten quick tips for machine learning in computational biology, *BioData Mining* **10**(35).

Chicco, D. and Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation, *BMC Genomics* **21**.

Choudhary, A., Harding, J. and Tiwari, M. (2009). Data mining in manufacturing: A review based on the kind of knowledge, *Journal of Intelligent Industry* **20**(501).

Croxton, F. E. and Cowden, D. J. (1939). *Applied general statistics*, Prentice-Hall, Inc.

Davis, J., Burnside, E. S., de Castro Dutra, I., Page, D., Ramakrishnan, R., Costa, V. S. and Shavlik, J. W. (2005). View learning for statistical relational learning: With an application to mammography., *IJCAI*, Citeseer, pp. 677–683.

Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and ROC curves, *Proceedings of the 23rd International Conference on Machine Learning*, pp. 233–240.

Dietrich, C. F. (2017). *Uncertainty, calibration and probability: The statistics of scientific and industrial measurement*, Routledge.

Dorogush, A. V., Ershov, V. and Gulin, A. (2018). CatBoost: Gradient boosting with categorical features support, *arXiv preprint arXiv:1810.11363* .

Duroux, R. and Scornet, E. (2018). Impact of subsampling and tree depth on random forests, *ESAIM: Probability and Statistics* **22**: 96–128.

European Commission (2019). *Transport in the European Union*.

Everitt, B. and Skrondal, A. (2002). *The Cambridge dictionary of statistics*, Cambridge University Press.

Finkelstein, M. (2008). *Failure rate modelling for reliability and risk*, Springer Science & Business Media.

Fletcher, G. S. (2019). *Clinical epidemiology: The essentials*, Lippincott Williams & Wilkins.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine, *Annals of Statistics* pp. 1189–1232.

Friendly, M. (2002). Corrgrams: Exploratory displays for correlation matrices, *The American Statistician* **56**(4): 316–324.

Geissbauer, R., Vedso, J. and Schrauf, S. (2016). Industry 4.0: Building the digital enterprise, *Technical report*, PricewaterhouseCoopers.

Guns, R., Lioma, C. and Larsen, B. (2012). The tipping point: F-score as a function of the number of retrieved items, *Information Processing & Management* **48**(6): 1171–1180.

Haarman, M., Michel, M. and Costas, V. (2017). Predictive maintenance 4.0: Predict the unpredictable, *Technical report*, PricewaterhouseCoopers and Mainnovation.

Hashemian, H. M. (2010). State-of-the-art predictive maintenance techniques, *IEEE Transactions on Instrumentation and Measurement* **60**(1): 226–236.

Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction*, Springer Science & Business Media.

He, H. and Garcia, E. A. (2009). Learning from imbalanced data, *IEEE Transactions on Knowledge and Data Engineering* **21**(9): 1263–1284.

He, H. and Ma, Y. (2013). *Imbalanced learning: Foundations, algorithms, and applications*, John Wiley & Sons.

James, G., Witten, D., Hastie, T. and Tibshirani, R. (2013). *An introduction to statistical learning*, Vol. 112, Springer.

Japkowicz, N. and Stephen, S. (2002). The class imbalance problem: A systematic study, *Intelligent Data Analysis* **6**(5): 429–449.

Jardine, A. K., Lin, D. and Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance, *Mechanical Systems and Signal Processing* **20**(7): 1483–1510.

Jeni, L. A., Cohn, J. F. and De La Torre, F. (2013). Facing imbalanced data – recommendations for the use of performance metrics, *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, IEEE, pp. 245–251.

Jezzini, A., Ayache, M., Elkhansa, L., Makki, B. and Zein, M. (2013). Effects of predictive maintenance (PdM), proactive maintenace (PoM) & preventive maintenance (PM) on minimizing the faults in medical instruments, *2013 2nd International conference on advances in biomedical engineering*, IEEE, pp. 53–56.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree, *Advances in Neural Information Processing Systems* **30**: 3146–3154.

Khare, V. R., Bandyopadhyay, P. and Waldo, M. B. (2012). Automotive field failure analysis based on mileage — feasibility benefits, *2012 IEEE Conference on Prognostics and Health Management*, pp. 1–8.

Kleinbaum, D. G., Dietz, K., Gail, M., Klein, M. and Klein, M. (2002). *Logistic regression: A self-learning text*, Springer.

Klutke, G.-A., Kiessler, P. C. and Wortman, M. A. (2003). A critical look at the bathtub curve, *IEEE Transactions on Reliability* **52**(1): 125–129.

Kumar, A., Chinnam, R. B. and Tseng, F. (2019). An HMM and polynomial regression based approach for remaining useful life and health state estimation of cutting tools, *Computers & Industrial Engineering* **128**: 1008–1014.

Liljefors, F. (2020). Time dependent modeling of turbocharger failure using machine learning.

Loeve, M. (1977). Elementary probability theory, *Probability theory I*, Springer, pp. 1–52.

Louit, D. M., Pascual, R. and Jardine, A. K. (2009). A practical procedure for the selection of time-to-failure models based on the assessment of trends in maintenance data, *Reliability Engineering & System Safety* **94**(10): 1618–1628.

Mashhadi, P. S., Nowaczyk, S. and Pashami, S. (2020). Stacked ensemble of recurrent neural networks for predicting turbocharger remaining useful life, *Applied Sciences* **10**(1): 69.

Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme, *Biochimica et Biophysica Acta (BBA)-Protein Structure* **405**(2): 442–451.

Montillo, A. A. (2009). Random forests, *Lecture in Statistical Foundations of Data Analysis* .

Natekin, A. and Knoll, A. (2013). Gradient boosting machines, a tutorial, *Frontiers in Neurorobotics* **7**: 21.

Nowaczyk, S., Prytz, R., Rögnvaldsson, T. and Byttner, S. (2013). Towards a machine learning algorithm for predicting truck compressor failures using logged vehicle data, *12th Scandinavian Conference on Artificial Intelligence, Aalborg, Denmark, November 20–22, 2013*, IOS Press, pp. 205–214.

Pesaran, A. A., Santhanagopalan, S. and Kim, G.-H. (2013). Addressing the impact of temperature extremes on large format Li-Ion batteries for vehicle applications.

Praveenkumar, T., Saimurugan, M., Krishnakumar, P. and Ramachandran, K. I. (2018). Fault diagnosis of automobile gearbox based on machine learning techniques, *Procedia Engineering* **97**: 2092–2098.

Prytz, R. (2014). *Machine learning methods for vehicle predictive maintenance using off-board and on-board data*, Licentiate dissertation, Halmstad University Press.

Prytz, R., Nowaczyk, S., Rögnvaldsson, T. and Byttner, S. (2013). Analysis of truck compressor failures based on logged vehicle data, *9th International Conference on Data Mining, Las Vegas, Nevada, USA, July 22–25, 2013*, CSREA Press.

Prytz, R., Nowaczyk, S., Rögnvaldsson, T. and Byttner, S. (2015). Predicting the need for vehicle compressor repairs using maintenance records and logged vehicle data, *Engineering Applications of Artificial Intelligence* **41**: 139–150.

Ramentol, E., Gondres, I., Lajes, S., Bello, R., Caballero, Y., Cornelis, C. and Herrera, F. (2016). Fuzzy-rough imbalanced learning for the diagnosis of High Voltage Circuit Breaker maintenance: The SMOTE-FRST-2T algorithm, *Engineering Applications of Artificial Intelligence* **48**: 134–139.

RDW (2021). Over rdw | rdw.
URL: *https://www.rdw.nl/over-rdw*

Ridgeway, G. (1999). The state of boosting, *Computing Science and Statistics* pp. 172–181.

Saito, T. and Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets, *PLOS One* **10**(3): e0118432.

Sakib, N. and Wuest, T. (2018). Challenges and opportunities of condition-based predictive maintenance: A review, *Procedia CIRP* **78**: 267–272.

Schweizer, B. (2010). Dynamics and stability of turbocharger rotors, *Archive of Applied Mechanics* **80**(9): 1017–1043.

Segal, M. R. (2004). Machine learning benchmarks and random forest regression.

Susto, G. A., Schirru, A., Pampuri, S., McLoone, S. and Beghi, A. (2015). Machine learning for predictive maintenance: A multiple classifier approach, *IEEE Transactions on Industrial Informatics* **11**(3): 812–819.

Tetsui, T. (2002). Development of a TiAl turbocharger for passenger vehicles, *Materials Science and Engineering: A* **329**: 582–588.

Tipping, A. and Kauschke, P. (2016). Shifting patterns: The future of the logistics industry, *Technical report*, PricewaterhouseCoopers.

Van Rijsbergen, C. J. (1974). Foundation of evaluation, *Journal of Documentation* .

Vezhnevets, A. and Vezhnevets, V. (2005). 'Modest AdaBoost' – Teaching AdaBoost to generalize better, *Graphicon*, Vol. 12, pp. 987–997.

Wang, J., Li, C., Han, S., Sarkar, S. and Zhou, X. (2017). Predictive maintenance based on event-log analysis: A case study, *IBM Journal of Research and Development* **61**(1): 11:121–11:132.

Zhang, T., Yu, B. et al. (2005). Boosting with early stopping: Convergence and consistency, *The Annals of Statistics* **33**(4): 1538–1579.

Appendices

A Derivation of cost savings formula

Prytz et al. (2013) propose the following formula to estimate cost savings.

$$C_{save} = TP \cdot (C_u - C_p) - FP \cdot C_p \tag{A.1}$$

TP denotes true positive (correctly discovering an unhealthy component); FP denotes false positive (erroneously labeling a healthy component as unhealthy); C_u denotes cost of sudden breakdown; and C_p denotes cost of replaced component.

In Equation A.1, TP and FP are the number of true positive and false negative cases in the testing set. In practice, however, each fleet operator has a different number of operating trucks, making the equation unscalable. Suppose that the fleet operator who will adopt the prediction model has n trucks, and since the size of the testing set is equal to $TP + FP + TN + FN$, Equation A.1 can be written as follows.

$$C_{save} = \frac{n \cdot [TP \cdot (C_u - C_p) - FP \cdot C_p]}{TP + FP + TN + FN} \tag{A.2}$$

Precision is defined as the proportion of positive results that is true positive (Fletcher, 2019).

$$\text{precision} = \frac{TP}{TP + FP} \tag{A.3}$$

Let β denote the ratio of cost of sudden breakdown to cost of planned turbocharger replacement.

$$\beta = \frac{C_u}{C_p} \tag{A.4}$$

Equation A.2 can be finally written as follows.

$$C_{save} = \left(\beta - \frac{1}{\text{precision}}\right) \frac{n \cdot C_p \cdot TP}{TP + FP + TN + FN} \tag{A.5}$$

B Summary statistics of numerical features

Table 4: Summary statistics of numerical features

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
age	3,284	650	523	1	259.8	982.2	3,140
cum_mileage	3,284	185,088	178,640	1	54,334.5	273,871	1,286,181
since_last_time	3,284	198	299	1	34	229	2,344
prv_AdblueFilter	3,284	538	463	1	216	724.2	3,140
prv_AirFilter	3,284	431	435	1	153	529.2	2,573
prv_BrakePadSet	3,284	585	490	1	225	851	3,140
prv_CarbonStoveFilter	3,284	605	521	1	229	886.2	3,140
prv_Coolant	3,284	446	422	1	147	615	2,573
prv_ElectricalCircuit	3,284	513	456	1	180	718	3,140
prv_EngineOil	3,284	379	430	1	98	447.2	2,477
prv_ExhaustGasket	3,284	621	513	1	238.8	931.8	3,140
prv_Fuel	3,284	419	424	1	138	532.2	2,573
prv_FuelCapWithDafLock	3,284	630	515	1	246.8	939	3,140
prv_FuelWaterDep	3,284	592	515	1	203	898.5	2,573
prv_Fuse	3,284	600	507	1	227	880	3,140
prv_GadusGrease	3,284	551	494	1	186	791.2	2,573
prv_HeatFilterCarbon	3,284	615	527	1	224	924	3,140
prv_Hose	3,284	581	488	1	227	831.2	2,573
prv_IntercoolerCleaner	3,284	577	497	1	218	832	3,140
prv_Lighting	3,284	510	467	1	175	711.2	3,140
prv_MountingMaterial	3,284	406	401	1	123	552	2,520
prv_ORing	3,284	527	468	1	188	731.2	2,573
prv_OilFilter	3,284	432	441	1	140	556	2,520
prv_PolyVBelt	3,284	593	484	1	236	867.2	3,140
prv_Sealing	3,284	412	429	1	133	507.8	3,140
prv_Spirax	3,284	475	437	1	175.8	637	2,573
prv_StrainerFeedPump	3,284	534	482	1	191	728	3,140
prv_UsitRing	3,284	603	502	1	231	892	3,140
prv_ValveCoverGasket	3,284	586	509	1	206.8	867.2	3,140
prv_WearIndicator	3,284	591	493	1	227.8	863	3,140