PREDICTING THE OCCUPANCY RATES OF TRUCK PARKING LOCATIONS. A MACHINE LEARNING APPROACH

Bachelor Thesis

BSc Industrial Engineering and Management

S. I. Slavova (Stefani)

August 2021

provincie Overijssel

UNIVERSITY OF TWENTE.

Colophon

Document:	Bachelor Thesis
Title:	Predicting the occupancy rates of truck parking locations. A machine learning approach
Author:	S. I. Slavova (Stefani)
Date:	August 2021
Educational institution:	**University of Twente** Drienelolaan 5 7522 NB Enschede
Educational program:	Industrial Engineering and Management
First supervisor:	Ing. J.P.S. Piest MSCM MBA BHRM (Sebastian)
Second supervisor:	Dr. ir. W.J.A. van Heeswijk (Wouter)
Host organization:	**Provincie Overijssel** Luttenbergstraat 2 8012 EE Zwolle
External supervisor:	R. Schasfoort (Robert)
First contact person:	L. Mollink (Lennart)
Second contact person:	G. Kuiper (Gerard)

Preface

Dear Reader,

Before you lies the thesis I have been working on in the final semester of my bachelor's degree in Industrial Engineering and Management at the University of Twente. The research was supervised by the province of Overijssel and took place from February until July 2021.

During one of my minors, I developed a strong passion for smart city engineering and was fascinated by the concept of using information and communication technology to optimize the efficiency of city operations and services. After completing this minor, I had no doubts that I want to write my thesis in a similar domain. I am beyond grateful that I was provided with the opportunity to carry out an assignment on such an interesting but at the same time challenging topic, which is machine learning.

I would like to thank my external supervisor, Robert Schasfoort, for this opportunity and for giving me the freedom to shape my assignment in a direction I was particularly interested in. In addition, I wish to thank Gerard Kuiper and Lennart Mollink, who were always welcoming and supportive and provided me with all the necessary resources throughout this project.

Secondly, I would like to express my gratitude to my first university supervisor, Sebastian Piest, for the excellent guidance and support during this process, and to my second university supervisor, Wouter van Heeswijk, for the insightful feedback. Their input and expertise helped me considerably with progressing forward with my thesis and improving the quality of my work.

Even though I had to conduct my research mostly at home, I had an inspiring and fulfilling learning experience, for which I wish to thank everyone involved in my research. Last but certainly not least, I wish to thank my family, who has always been of great support, and all my friends in Enschede.

This thesis marks the end of my bachelor's studies at the University of Twente. I am forever grateful for the opportunity to study abroad and the last three years will leave a mark forever.

I hope you enjoy reading my thesis.

Stefani Slavova

Bulgaria, August 2021

Management summary

With the increasing amount of freight carried by trucks over land, comes the responsibility of the governments for providing a supporting infrastructure and legislation. Neglecting these responsibilities leads to several issues, including a shortage of safe and secure truck parking places. When truck drivers are unable to find a suitable parking place, however, they park at illegal locations or continue driving without rest, which causes further nuisance and traffic unsafety.

This study is conducted under the supervision of the province of Overijssel, which has engaged in a long-term program aimed at tackling truck parking issues. Problem identification revealed the potential for improving the utilization of existing parking infrastructure, by providing truck drivers with information about the expected occupation of the parking lots at the time of their arrival. Following this, the main objective of this study was to determine an approach for the development of a prediction model, which could be further implemented into an integrated information system.

A literature review revealed the potential of adopting a machine learning approach for predicting parking occupation, however, knowledge gaps about predicting the occupation of truck parking lots were found. The literature did not give a full overview of which variables can be used to forecast truck parking occupancy or which machine learning algorithms have the potential to generate the most prominent predictions. This provided an opportunity to use this thesis as a means to fill in the identified gaps, by focusing on **truck parking occupancy prediction through machine learning**.

For the development of the machine learning model, **historical data** spanning 1.5 years from a single truck parking lot in Overijssel was utilized. Based on insights from the literature about car parking occupancy and further experimentations with different configurations, the best-performing model configuration was selected. The model uses **time-dependent** features as input and information about the **previous occupancy** of the parking lot. Thus, all inputs can be derived from the same dataset i.e., data incoming from the **electronic toll system** of the parking lots.

Evaluation of the model shows promising results, however, due to data limitations, there was no possibility to apply the approach to other truck parking areas and validate it. However, the transferability of the system towards other truck parking lots was assessed by determining the minimum amount of training data needed. When the volume of the data was decreased by 96.9%, the accuracy only decreased by 18.9%. The outcomes show that only **16 days** of data are needed for a model that performs slightly worse than a model trained with 1.5 years of data, which is highly promising regarding the ease of implementation in the future.

Furthermore, the added value of providing the model with information about the occupancy from earlier in the day was determined, showing an accuracy decrease of 275%, when this variable is omitted from the model. Nevertheless, a model with time-dependent features only produces satisfactory results and has the advantage of being able to produce a forecast for a longer period, although with lower accuracy.

Regarding delivering the outputs of the model to truck drivers and allowing them to use these in their decision-making process, a conceptual model for the integration of the model into a comprehensive system was proposed, as well as a brief implementation plan. An in-depth analysis of the expected benefits from the implementation of the system revealed advantages for a wide range of stakeholders: For example, it is expected that the work lifestyle of truck drivers will improve, road haulage companies

and goods owners will encounter lower costs, parking infrastructure owners will increase their revenues, and others.

Considering the promising results and the expected benefits, the continuation of the research is recommended, by focusing on the possibilities of data collection and expansion of the system, as well as its deployment. To evaluate the performance of the proposed system in terms of accuracy and reliability, the main work in the future should focus on developing a working prototype and performing further experiments.

Contents

List of Figures

List of Tables

List of Abbreviations

ANN:	Artificial Neural Networks
ARIMA:	AutoRegressive Integrated Moving Average
CRISP-DM:	Cross Industry Standard Process for Data Mining
DOW:	Day Of the Week
DRIPs:	Dynamic Route Information Panels
DT:	Decision Tree
IT:	Information Technology
KNN:	K-Nearest Neighbour
lightGBM:	light Gradient Boosting Machines
MAE:	Mean Absolute Error
MASE:	Mean Absolute Scaled Error
MSE:	Mean Squared Error
RMSE:	Root Mean Squared Error
RQ:	Research Question
SVM:	Support Vector Machines
SVR:	Support Vector Regression
Xgboost:	eXtreme Gradient Boosting

Chapter 1 Introduction

In this chapter, we treat the background of this research. Section 1.1 provides a brief context of the problem, followed by an introduction of the involved parties in section 1.2. Section 1.3 continues with a detailed problem description, followed by determining the research objectives in section 1.4. Next, the main research question along with the research sub-questions are formulated in section 1.5. Finally, we introduce the methodology to be followed in the research and the report outline in sections 1.6 and 1.7 respectively.

1.1 Background

Road freight is the dominant mode of transport in the intra-European trade and logistics sector, accounting for 53.4%[1], followed by maritime and rail transport with 29.6% and 12.3% respectively (Eurostat, 2021). Goods worth billions of euros are transferred daily on the Trans European Road Network, which constitutes the backbone of trade and commerce on the European continent. This shows how important trucks are for the European economy. Today about 6.5 million trucks are circulating throughout the EU (ACEA - European Automobile Manufacturers' Association, 2017). Indisputably, the sector performs successfully in terms of volume, however, the high number of trucks requires supporting infrastructure and legislation.

A survey conducted in 2018 (European Commission, 2019) shows that 83% of truck drivers believe that there is an insufficient number of safe and secure truck parking areas in Europe. This comes as no surprise considering that a study led by the European Commission revealed the astonishing shortage of 400,000 secure parking spaces (European Commission, 2019). As road freight transport continues to grow (Eurostat, 2019), providing sufficient safe and secure truck parking areas will get more challenging in the future, and has been listed as a top priority by the European Commission (Directive 2010/40/EU, 2010). The main objectives concern increasing the overall capacity of truck parking areas and optimizing the existing capacity so that truck parking locations are more efficiently utilized.

1.2 Involved parties

This thesis is executed in collaboration with the province of Overijssel, which has been facing truck parking shortages for years (Provincie Overijssel, 2020). These shortages result in nuisance, traffic unsafety, and other issues. As a mission to develop a network of safe and secure truck parking areas, the province together with partners i.e., municipalities, Rijkswaterstaat[2], logistics business organizations, and business park managers, has started a *multi-year program (2020-2030)* aimed at optimizing truck parking in the area. This research is executed as a part of the program, whose ambition is to:

- Ensure that drivers can rest according to the European regulations;
- Ensure that truck parks are safe and secure and more efficiently utilized;
- Prevent nuisance in other places.

[1] Percentage share in tonne-kilometers of the transactions performed within the boundaries of the European Union.
[2] Rijkswaterstaat is the executive agency of the Dutch Ministry of Infrastructure and Water Management, responsible for the design, management, and maintenance of the main infrastructure facilities in the Netherlands.

1.3 Problem identification

To get a better understanding of the underlying issues related to truck parking, we start this research with thorough problem identification. We investigate the existing problems, as well as their causes, to reach a core problem. Proposing a solution to the core problem is the main objective of the research.

1.3.1 Problem context

1) Unsafe traffic conditions

Due to the driving time limits and mandatory rest period imposed by the EC Regulation No 561/2006 (2006), a driver that is unable to find a suitable parking space might choose to either park somewhere illegally or continue driving illegally. However, the fines for violating the rules are high and the tachographs providing records of the hours driven can be traced back 28 days. Therefore, experiences show that drivers prefer to park illegally in a non-designated parking area that is potentially dangerous rather than to break the obligatory rest periods (Nagy & Sandor, 2012). Preferred illegal locations among truck drivers are highway access ramps, emergency lanes, and public roads on business parks. However, parking at these locations creates unsafe traffic conditions and poses safety hazards to other motorists and truck drivers themselves.

2) Unsafe driving

As mentioned above, the second alternative truck drivers have is to continue driving illegally and tired, which imposes safety risks for all participants in traffic. Firstly, several studies show that fatigue is associated with increased accident risk (European Commission, 2015). Furthermore, according to different surveys worldwide (Australia, France, Ireland, Netherlands, USA), over 50% of long-haul drivers report having at some point almost fallen asleep while driving (ETSC, 2001 as cited by European Commission, 2015). Finally, a study by the AAA Foundation for Traffic Safety in the USA revealed that 21% of all accidents in which a person died involved a fatigued driver (Tefft, 2014). This shows how dangerous drowsy driving might be.

3) Increased pollution

Due to the shortages of parking spaces, truck drivers need to drive searching for a parking spot and/or park at illegal locations, and both actions lead to unnecessary fuel consumption and CO_2 emissions. While at most legal parking locations, truck drivers can connect to the grid and use the necessary utilities, such as electricity to charge their phones or to power electric cookers, no illegal locations can provide that. Subsequently, when truck drivers stop at illegal locations and need these services, they are forced to idle[3], which can sometimes create as many emissions as a moving vehicle (Burgess et al., 2009). In case that the trucks are parked in local streets, this leads to decreased air quality and health of the residents living in proximity (Palaniappan et al., 2005; de Almeida Araujo Vital et al., 2020).

4) Unnecessary costs

As explained above, the lack of parking spaces impacts fuel consumption, due to the unnecessary time spent driving or time spent idling. Firstly, a significant share of the operational costs in the trucking industry is incurred by fuel costs (Murray & Glidewell, 2019). Moreover, a study by the University of California, Davis has found that 8.7% of the total fuel consumption of trucks is caused by idling (Lutsey et al., 2004). Secondly, idling is also associated with increased maintenance costs and engine wear, as it

[3] Idling is associated with keeping the vehicle's engine running when the vehicle is not in motion.

causes additional wear to the internal parts compared to driving at normal speeds (Air Resources Board, 2017).

5) Cargo crime and social insecurity

Another issue resulting from the shortage of parking locations relates to trucks becoming an attractive target for vandalism and cargo crimes. Such actions lead to considerable financial and reputational losses to supply chain operators. It is estimated that most thefts happen when trucks are parked and the direct losses resulting from them exceed 8.2 billion euros per year (van den Engel & Prummel, 2007 as cited by European Commission, 2019). According to a survey among European truck drivers providing international road transport, only 12.8% of the participants indicated that they feel safe in the parked vehicle during the night. 23.5% of them had already been robbed (Poliak et al., 2020). Hence, it can be concluded that that the shortage of safe parking places leads to higher insecurity among truck drivers, resulting in a worsened work lifestyle.

6) Unutilized parking infrastructure

The last problem, which will be discussed in this section relates to the suboptimally utilized parking infrastructure. This is caused by the unequal distribution of trucks over parking areas, the lack of information about parking areas and their facilities, and the lack of information about the occupancy of the parking places. Firstly, this leads to unrealized revenue for legal parking operators. Secondly, as the demand for truck parking locations is increasing, it is of high importance that the existing infrastructure is optimally used, to minimize the cost incurred by building new infrastructure, as it is estimated that the investment cost for one parking place is 70,000–120,000€ (Poliak et al., 2020).

1.3.2 Core problem

For identifying the core problem, we further mapped the identified inventory of problems and examined their causes and effects, visualized as a problem cluster in Figure 1-1. The analysis shows that the identified problems emerge from *overcrowding* of trucks and *unequal distribution* over parking areas. Further analysis of the causes shows that there are five potential core problems. Firstly, truck drivers are not aware of the occupancy status of the parking lots before their arrival, leading to implications when the lots are full at the time of arrival. Second, trucks are unequally distributed due to truck drivers' preference to park at certain areas, such as close to shippers and clients or at parking lots with lower fees. Thirdly, the situation is affected negatively by the limited supply of parking lots with the appropriate facilities. Another cause is the imposed mandatory driving time and rest periods by the European Union. Finally, due to a truck traffic ban on Sundays and national holidays in Germany, the parking lots in Overijssel[4] accumulate a higher number of trucks, leading to more shortages.

From the problem cluster, we can deduce that the core problem is *the lack of insight into the occupation of truck parking areas* for truck drivers. This cause is furthest from the initial problems and is not the effect of another cause. In contrast to the other potential core problems, this problem influences multiple other problems and is influenceable, which makes it the most appropriate core problem in the context of this research. Finally, we reformulate the problem as follows:

[4] The province of Overijssel is located on the border with Germany, making it a preferred parking spot for truck drivers on Sundays and national holidays due to the imposed German truck traffic ban.

In Overijssel, there is no applied method to monitor the occupation of truck parking areas, which hinders the information provision to truck drivers and the efficient utilization of the parking infrastructure.

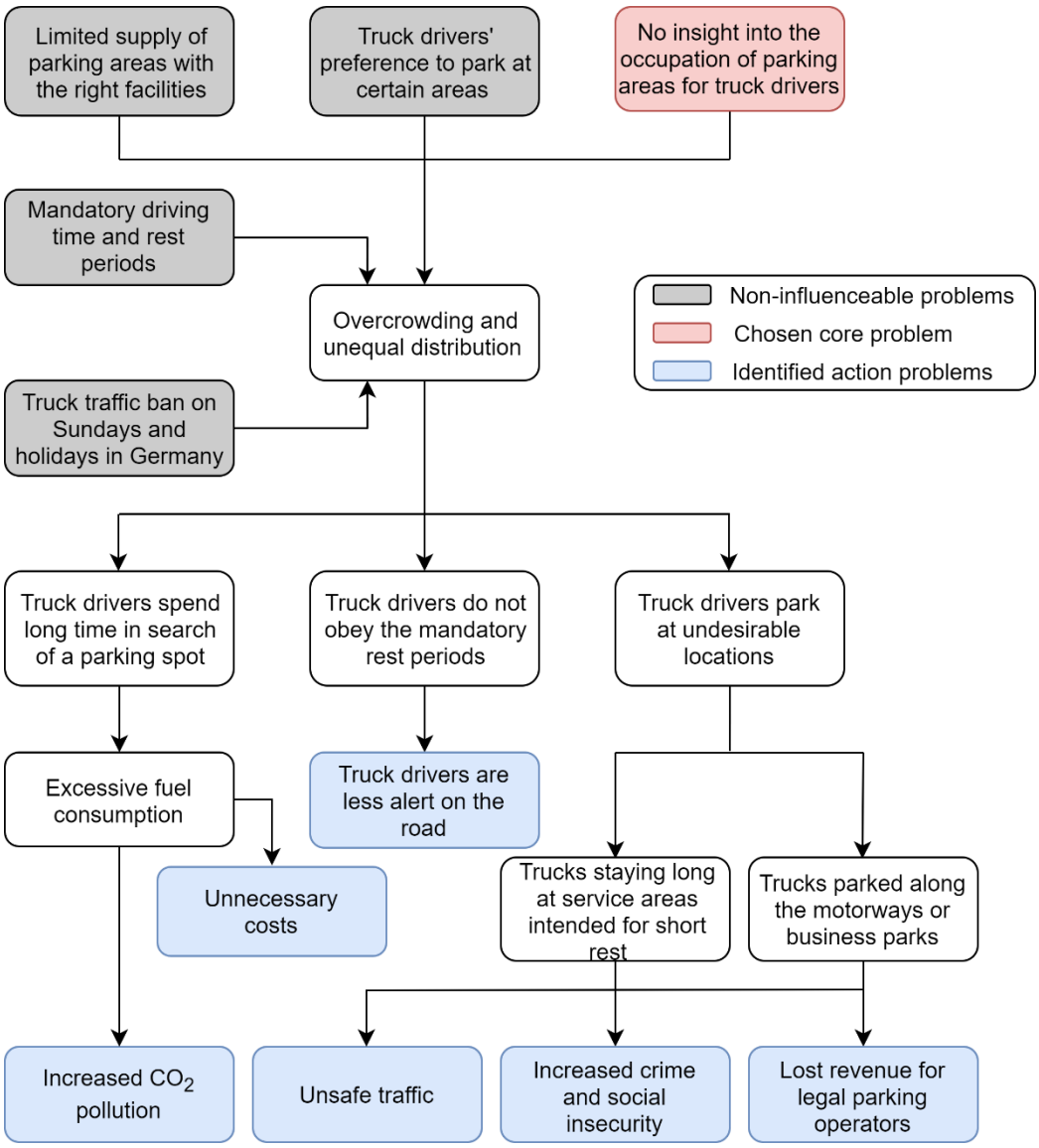


Figure 1-1 Problem cluster

1.4 Research objectives

The problem identification revealed the need for a method to monitor the occupation of truck parking areas in Overijssel. A way to achieve this is to use historical data to train a model that predicts the occupancy rates of truck parking areas in real time. In the literature, short-term forecasting techniques are classified into four main categories: statistical techniques, artificial intelligence techniques, knowledge-based expert systems, and hybrid techniques (Sadek, Martin & Shaheen, 2020). A systematic

literature review revealed that researchers apply artificial intelligence techniques[5] more frequently than the other techniques for forecasting parking occupation based on historical data. Hence, this approach will be adopted in the study. Following this, the main objective of the research is translated into developing a machine learning model that predicts the occupancy rates[6] of truck parking areas. For the research, historical data from preselected parking lots will be used. However, since the truck parking problem is wider than the locations involved in the study, it should also be determined to what extent the proposed approach is generalizable and transferable to other truck parking locations.

1.5 Research questions

To achieve the selected research objectives, we formulate the main research question as follows:

How can an accurate and reliable machine learning model be developed that determines the real-time occupancy rate of truck parking areas situated in the province of Overijssel based on historical data?

Furthermore, to answer the main research question, twelve sub-questions are determined:

1. Who are the relevant stakeholders of the prediction model and what are their anticipated benefits from implementing the solution?
2. Which machine learning methods are known in the literature for making predictions of numerical outputs?
3. Which variables are most relevant to be used as input in the predictive model according to the literature?
4. Which forecasting techniques are known in the literature for predicting (truck) parking occupancy rates?
5. Which evaluation metrics can be used to assess the model's performance?
6. What differences and similarities are there between predicting car parking occupancy and truck parking occupancy?
7. What data is available that is relevant for predicting the occupancy rates of truck parking areas in Overijssel?
8. How should the dataset, which will be fed to the model, be configurated?
9. What are the characteristics of the prediction model(s)?
10. What are the performance indicators of the proposed prediction model(s)?
11. To what extent is the resulting model transferable towards other truck parking areas?
12. How can the outputs of the model be communicated to the relevant stakeholders?

1.6 Research methodology

Since the research will focus on data mining problems, we will apply the CRoss Industry Standard Process for Data Mining (CRISP-DM) method (Chapman et al., 2000). The framework is published in 1999 by an association formed by the companies Daimler Chrysler AG, SPSS Inc., and NCR Systems Engineering, aiming to standardize data mining processes across industries (Chapman et al., 2000). It is referred to as the most frequently used methodology when it comes to data science projects (Saltz, 2020). Due to its popularity in practice, this framework is chosen for this research. Furthermore, as the nature of the research is data mining oriented, the methodology will be easy to apply and will give a clear structure to

[5] Artificial intelligence techniques are based on machine learning/deep learning algorithms.
[6] The occupancy rate shows the fraction of occupied parking spaces.

the planning of the project. CRISP-DM consists of six phases and Figure 1-2 shows a schematic overview of the process. The sequence of the phases is not rigid: the output of each phase affects the input of the following phase, nevertheless, shifting back and forth between the phases is often necessary.

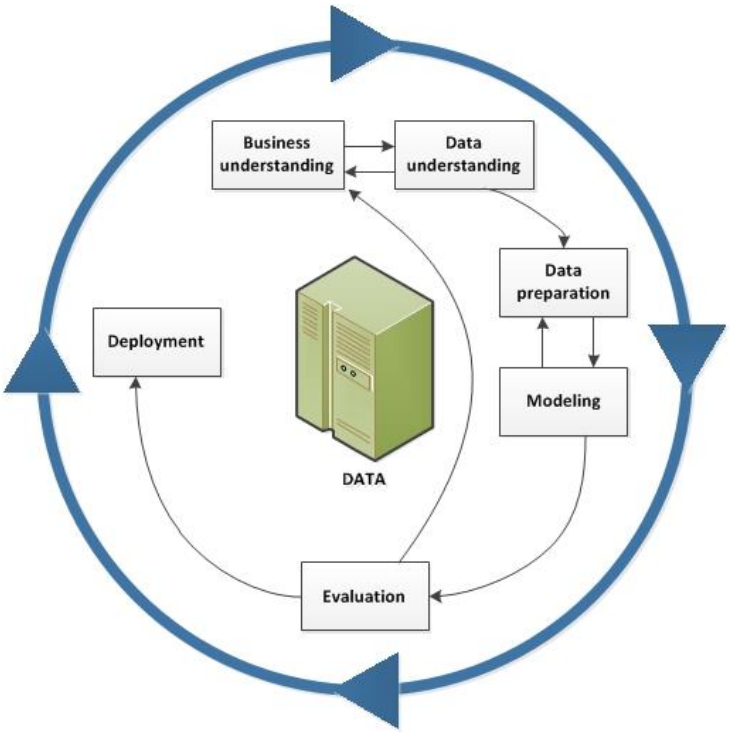


Figure 1-2 The CRISP-DM Cycle (Source: www.ibm.com)

1.7 Report structure

The remainder of the thesis is structured as follows: Each phase of the CRISP-DM cycle roughly represents one chapter in the report. Chapter 3 is an exception since it is not part of the CRISP-DM cycle but a theoretical chapter including the literature review. Table 1-1 outlines how the research sub-questions are distributed between chapters. After all research questions are answered, the report ends with a discussion of the results, limitations and recommendations, and a conclusion.

Table 1-1 Structure of the report

Research sub-question	Research phase	Treated in
1. Who are the relevant stakeholders of the prediction model and what are their anticipated benefits from implementing the solution?	Business understanding	Chapter 2
2. Which machine learning methods are known in the literature for making predictions of numerical outcomes?	Literature review	Chapter 3
3. Which variables are most relevant to be used as input in the predictive model according to the literature?	Literature review	Chapter 3
4. Which forecasting techniques are known in the literature for predicting (truck) parking occupancy rates?	Literature review	Chapter 3
5. Which evaluation metrics can be used to assess the model's performance?	Literature review	Chapter 3

6. What differences and similarities are there between predicting car parking occupancy and truck parking occupancy?	Literature review	Chapter 3
7. What data is available that is relevant for predicting the occupancy rates of truck parking areas in Overijssel?	Data understanding	Chapter 4
8. How should the dataset, which will be fed to the model, be configurated?	Data preparation	Chapter 5
9. What are the characteristics of the prediction model(s)?	Modelling	Chapter 6
10. What are the performance indicators of the proposed prediction model(s)?	Evaluation	Chapter 7
11. To what extent is the resulting model transferable towards other truck parking areas?	Evaluation	Chapter 7
12. How can the outputs of the model be communicated to the relevant stakeholders?	Deployment	Chapter 8

Chapter 2 Business understanding

The main goal of this research is to develop an information system that predicts the occupancy rates of truck parking locations situated in the province of Overijssel. When developing a new product, which aims to solve a problem, analyzing the involved stakeholders is an essential first step. For the success of the initiative, one should aim at engaging all involved parties and strive towards establishing proper communication and collaboration opportunities. Following this, this chapter will focus on answering RQ 1: *Who are the relevant stakeholders of the prediction model and what are their anticipated benefits from implementing the solution?* We will start by listing all relevant stakeholders in section 2.1. Following this, section 2.2 will focus on analyzing the expected benefits of implementing the system from the perspective of each stakeholder. Finally, implementing the system may also have some drawbacks, which will be addressed in section 2.3

2.1 Involved stakeholders

The truck parking problem in Overijssel is a complex issue, involving many parties. Therefore, implementing a predictive system for information provision will affect all involved stakeholders to some extent. Before determining the effects of the solution, one should start by identifying the involved stakeholders. By brainstorming, using domain knowledge about the logistics sector, and through discussions with the province and Rijkswaterstaat, we identified 14 stakeholders. We included the parties who have an influence or power over the project, who have an interest in its implementation and the ones who are affected by the implementation. A comprehensive list is provided in Table 2-1.

Table 2-1 Involved stakeholders

№	Stakeholder
1	Truck drivers
2	Road users
3	Road haulage companies
4	Goods owners
5	Parking infrastructure owners
6	Business-park managers
7	The province of Overijssel
8	Rijkswaterstaat
9	Road authorities
10	Nearby[7] communities
11	The environment
12	Software application developers
13	System admins
14	System support staff

[7] The word 'nearby' is used to describe the communities situated in a proximity to where trucks are found to park illegally.

2.2 Expected benefits

Now that the main groups of stakeholders have been identified, we will further analyze their relationship to the proposed system by determining how each group of stakeholders is **expected** to benefit from the implementation. The benefits are derived based on brainstorming, domain knowledge of the logistics sector, and discussions with involved stakeholders. To get a good understanding of the relationship of the relevant stakeholders to the project goal and the relationships between stakeholders, we use a *stakeholder onion diagram*. A stakeholder onion diagram distinguishes itself from other types of stakeholder analysis visualizations because its emphasis is on the project goal rather than the project itself or key stakeholders only (Olson, 2013). It consists of four layers: The center represents the solution that is delivered by the project. The second layer contains the stakeholders who interact with it directly. The next layer is populated with the parties that control the project solution. The final layer contains all stakeholders which are outside the organization but are still important to consider. Arrows indicate the relationships, and a stakeholder can be related to the previous layer or other stakeholders. The results are visualized in Figure 2-1.

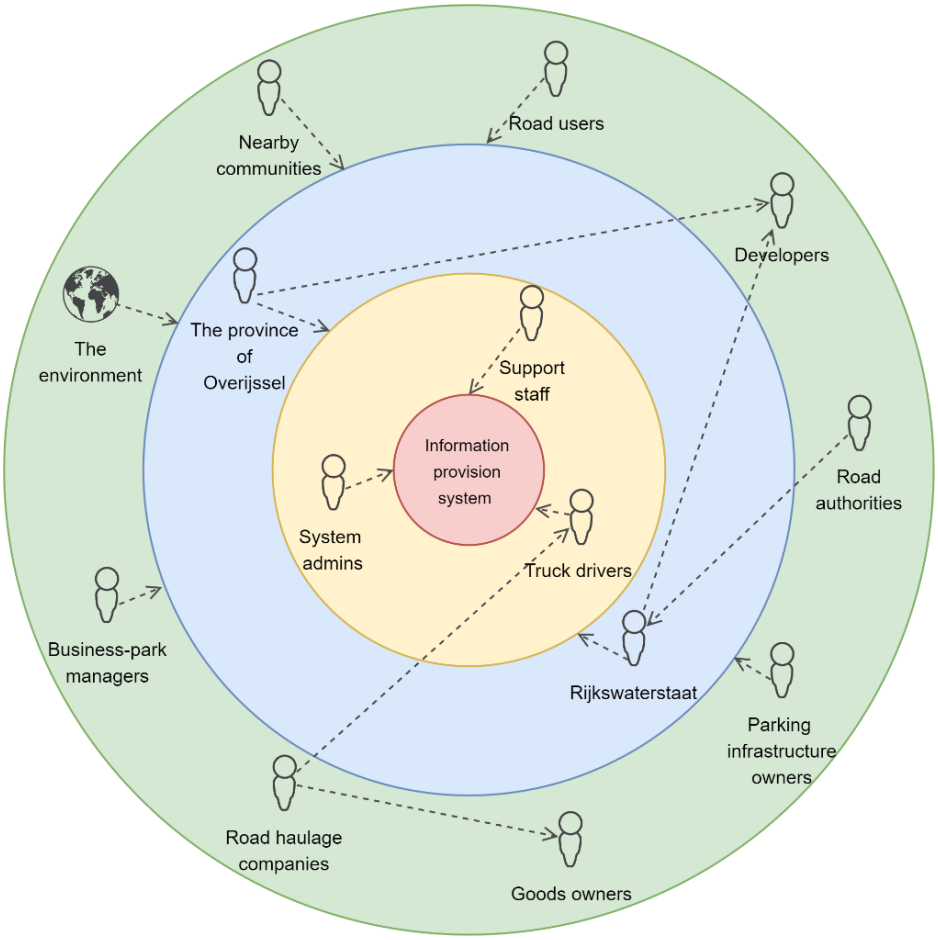


Figure 2-1 Stakeholder onion diagram

1. Truck drivers

First and foremost, truck drivers are the main end-user of the anticipated system. A system that predicts the truck parking occupation and further communicates these predictions to truck drivers will help them

make better-informed decisions when planning their route based on expected availability at the time of their arrival. More effective route planning will lead to a reduced number of stressful situations for drivers, caused by the pressure on compliance with statutory breaks. Furthermore, parking the trucks at safe and secure parking lots will lead to increased perceived safety and security for the drivers. This will increase their sleep quality and thus, reduce the chances of causing accidents due to fatigue. Overall, truck drivers will benefit from the system by experiencing less stress, improving their quality of sleep, and reducing the feeling of social insecurity, which in the long term will lead to a better work lifestyle.

2. Road users

The next identified stakeholder group concerns all road users and participants in traffic. Firstly, more efficient route planning will reduce the time for searching for a parking spot, which will lead to fewer wandering vehicles and an improved traffic flow. Secondly, helping truck drivers find designated parking locations will result in fewer illegally parked trucks and thus, fewer accidents because of roadside parking. Finally, a better quality of rest and sleep will lead to more alert truck drivers on the road and hence, fewer accidents caused by fatigue, which is a benefit for all participants in traffic.

3. Road haulage companies

Road haulage companies are another stakeholder that will be affected positively by implementing the proposed system. Firstly, trucks parked in a safe location will increase the perceived safety of the vehicle owners. Secondly, fewer accidents because of roadside parking or fatigued truck drivers will cause less associated costs from damaged vehicles. Third, less time spent searching for parking or idling means less fuel consumption and slower vehicle depreciation, which will decrease the corresponding costs. Finally, fewer cargo crime accidents and vandalism will lead to fewer costs due to damages to the vehicles and their contents. Overall, the main benefits to road haulage companies relate to a decrease in the expenditures of keeping the vehicles operational.

4. Goods owners

The fourth identified stakeholder is goods owners. The benefits for them are similar to those of road haulage companies. Their perceived safety will increase due to knowing that the goods are parked in a safe location and secondly, they will incur fewer expenses or lose potential revenues due to cargo crimes and accidents caused by roadside parking or fatigued drivers.

5. Parking infrastructure owners

Next, the system will cause several effects on parking infrastructure owners. First, providing information about the occupation will increase the satisfaction and comfort of their clients, and thus will make their parking seem more appealing. Next, attracting more customers will lead to higher utilization of products and services offered at the parking lot, and thus, increased revenues. Finally, parking infrastructure managers could use the information system themselves despite that they are not the main intended end-user. The system could help them to better estimate the expected demand and assist them with their planning, such as staffing, shifts et cetera.

6. Business-park managers

Business-park managers are the next involved stakeholder. The primary benefit for them relates to less nuisance caused by trucks that park illegally at business parks. Moreover, heavy goods vehicles cause asphalt to bend and crack more easily, and thus, the reduced number of illegally parked trucks will lead to lower associated maintenance costs.

7. The province of Overijssel and 8. Rijkswaterstaat

Next, as a governmental body controlling the project, the province of Overijssel and Rijkswaterstaat are key stakeholders. From their perspective, the benefits of implementing the system relate to increasing traffic safety and reducing the number of accidents, which leads to reducing the externalities arising from traffic accidents. Secondly, better utilization of the already existing parking infrastructure will lead to fewer required newly build truck parking areas, and thus, financial savings for the construction and maintenance of parking infrastructure. Finally, the reduced number of illegally parked trucks will cause the highway infrastructure to wear off slower, and thus, the government will incur fewer costs due to damage to the roads.

9. Road authorities

The next identified stakeholder is the road authorities of Rijkswaterstaat whose responsibilities relate to the surveillance and security along the Dutch highways. They continuously and closely monitor the situation on the Dutch highways, and thus, a reduced number of illegally parked trucks on highway access ramps and emergency lanes and a reduced number of traffic accidents will alleviate the tasks of their demanding profession.

10. Nearby communities

Furthermore, nearby communities are another group that will experience positive effects from fewer illegally parked trucks. Firstly, they will benefit from less nuisance caused by these vehicles. Secondly, the reduced number of illegally parked trucks will lead to less air pollution and thus, improved air quality and health for the nearby residents.

11. Environment

Next, the environment will also be indirectly affected by the reduced number of illegally parked vehicles and the reduced time spent searching for a parking spot. These lead to less fossil fuel consumption and hence, less associated air pollution.

12. Software application developers, 13. Service admins and 14. System support staff

Lastly, to bring the idea of the information system to life and maintain it operational, one needs a team of software application developers, system admins, and support staff. As being directly engaged with the development and maintenance of the system, they are important stakeholders to consider. The benefits for them are primarily related to job creation, as they would have the opportunity to get experience with developing and maintaining a system that is relatively new and unique.

2.3 Identified negative implications

Besides benefits, it is important to identify negative aspects associated with the system. Overall, 4 drawbacks were identified, which are as follows:

1. Malfunctioning of the system

The first issue relates to a possible malfunctioning of the system. If the intended information about the occupation status of parking locations will not be delivered to truck drivers due to a breakdown in the system, truck drivers will not be able to make informed decisions about which parking area to park at. This reflects the current situation, which causes a nuisance, traffic unsafety, and other problems. It is not expected that there will be any serious issues emerging from a temporary breakdown, however, it might harm the reputation of the service providers.

2. Wrong predictions

Secondly, the model will not always predict the occupancy of the parking areas 100% accurately. This means that occasionally, there might be situations when a truck driver arrives at a full parking lot, despite that the information system indicates that there should be free parking spots at the parking location. Following such experiences, truck drivers might get disappointed and not trust the system anymore, which is not the desired outcome.

3. A data breach

Next, like other computer systems, the system may be breached, and the model may be misused. The results from this would be the system malfunctioning or making wrong predictions, which are the situations discussed above. As this would lead to truck drivers not trusting the system and/or to the service providers harming their reputation, the system's security must be taken into account.

4. Drivers tempted to use their phones while driving

Fourth, if the system is implemented, for example, in a mobile application, truck drivers might feel tempted to use their phones to check the availability status of parking areas while driving. This increases the risk of accidents and therefore, imposes serious health risks, not only to truck drivers themselves but to all participants in traffic.

2.4 Conclusions

In this chapter, we determined which stakeholders are affected by the implementation of a system providing information to truck drivers about the availability of truck parking lots. Furthermore, we analyzed the expected benefits from the perspective of each stakeholder. Following the discussion, we can conclude that the implementation of the system is expected to affect a wide range of stakeholders in positive ways: Primarily, truck drivers will benefit from an improved work lifestyle. Road haulage companies and goods owners will encounter lower costs. Parking infrastructure owners will increase their revenues. Next, the system will create more job opportunities and finally, the province of Overijssel and Rijkswaterstaat will realize safe traffic and less nuisance, further benefiting the environment, nearby communities, business-park managers, and road users. Finally, we indicated some of the limitations of the system, namely a system breakdown, inaccurate predictions, a data breach, and an increased risk of accidents as a result of truck drivers using their mobile phones while driving to check the parking lots' occupancy.

Chapter 3 Literature review

This chapter presents an overview of contemporary machine learning methodologies, as well as their predictive potential within the truck parking domain. First, we provide a brief background about the field of machine learning in section 3.1. In section 3.2, we present several machine learning approaches, which are used for generating predictions. Subsequently, based on studying existing research, we define relevant input variables and present the most frequently used machine learning techniques for parking occupancy prediction in sections 3.3 and 3.4, respectively. Next, section 3.5 discusses different metrics for evaluation and comparison between models. Finally, section 3.6 provides a discussion about the similarities and differences between predicting car parking and truck parking occupancy.

3.1 Introduction to data science and machine learning

As the application of machine learning is the primary focus of this research, in this section we will present the machine learning fundamentals, by outlining the definitions and the different types of learning techniques.

Machine learning refers to a group of techniques used by data scientists. It is a branch of artificial intelligence that specializes in training a machine how to learn from data rather than through explicit programming. To achieve that, machine learning makes use of a range of algorithms. The algorithms are repetitively fed with training data and based on that data, more accurate models are produced. A machine learning model is the produced outcome from training a machine learning algorithm with data (Hurwitz & Kirch, 2018). It is important to note the difference between the terms *machine learning algorithm* and *machine learning model* as they are not interchangeable.

The machine learning discipline consists of *supervised*, *unsupervised,* and *reinforcement learning*. Before explaining their characteristics, we will first define the types of variables that machine learning makes use of. These are *input* and *output* variables. The input variables have some influence on the output variables. Hence, the inputs are called *independent* variables[8], and the outputs - *dependent* variables[9].

In supervised learning, the output variables are present and used in the learning process to predict the value of the output variables, whereas, in unsupervised learning, we have no measurements of the output variables. The main objective in unsupervised learning is to find patterns in the data sets, rather than to predict a value (Hastie, Tibshirani, & Friedman, 2001). Hence, since the main task of this research is to predict the occupancy rates of truck parking areas, supervised learning is the desired approach.

Furthermore, supervised learning divides into *classification* and *regression* techniques. In classification, the goal of the algorithm is to assign data into specific categories, by recognizing certain entities within the dataset and concluding on how those entities should be labeled (classified). On the contrary, regression algorithms are used to make predictions, by understanding the relationship between the dependent and independent variables (IBM Cloud Education, 2020). As the outputs of regression models are quantifiable (numerical), a regression technique will be used to predict the occupancy rates of truck parking areas.

Finally, reinforcement learning is a machine learning type based on rewarding desired actions while punishing undesired ones. A reinforcement learning agent, in general, is capable of seeing and

[8] Independent variables are also referred to as *features, predictors*, or *explanatory variables*.
[9] Dependent variables are also called as *target variables*.

interpreting its surroundings, taking actions, learning via trial and error. Examples of reinforcement learning applications are gaming and robotics.

3.2 Regression models

In section 3.1, we concluded that supervised regression is the most suitable type of machine learning for predicting parking occupancy rates. State-ot-the-art machine learning provides many such techniques. The following section will briefly introduce the most relevant ones, both mathematically and functionally. This section will answer RQ 2: *Which machine learning methods are known in the literature for making predictions of numerical outputs?*

3.2.1 Linear regression models

One of the most basic types of regression in machine learning is linear regression. The model consists of a dependent variable and one (simple linear regression) or more (multiple linear regression) independent variables and the independent variables are linearly related to the dependent variable through the equation. Due to their simplicity and straightforward approach, these models are relatively transparent and easy to interpret, compared to other machine learning models.

Simple linear regression

Simple linear regression is the simplest form of linear regression. It consists of one predictor variable and one target variable. The model has the following components:

- Output (target variable), commonly referred to as y;
- Input (predictor variable), commonly referred to as x;
- Intercept coefficient β_0, indicating the point where the estimated regression line crosses the y axis;
- Coefficient β_1, indicating the slope of the estimated regression line;
- Random error, commonly referred to as ε, indicating the random component of the linear relationship between the output and input variable, or the part of y that x is unable to explain.

Thus, we compose the following mathematical equation:

$$\hat{y} = \beta_0 + \beta_1 x + \varepsilon$$

To estimate the parameters β_0 and β_1 and fit the best possible line to predict the target variable, we use the method *Ordinary Least Squares*. The ordinary least squares linear regression aims at finding the plane that minimizes the Sum-of-Squared Errors (SSE) between the observed and predicted response:

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 ,$$

where y_i denotes the outcome and $\hat{y}_i$ denotes the model prediction of that sample's outcome.

Multiple linear regression

If the model uses more than one independent variable to predict the outcome of the dependent variable, it is called multiple linear regression. It is similar to the model described above but includes additional predictors. The equation then has the following form:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n + \varepsilon$$

3.2.2 Polynomial regression models

Polynomial regression is another type of regression analysis in which the relationship between the dependent and independent variables is represented by an nth degree polynomial. It is a special case of linear regression, in which the polynomial equation is fitted to the data with a curvilinear relationship between the dependent and independent variables. These models are usually fitted with the method of *least squares*. The general equation takes the following form:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_1^1 \ldots + \beta_n x_1^n + \varepsilon$$

Polynomial regression models do not require the relationship between the dependent and independent variables in the dataset to be linear, which is the main difference between ordinary linear regression and polynomial regression. Polynomial regression is not linear in the way that x is not linearly correlated with the function $y = f(x, \beta)$. However, the equation itself is linear in the parameters β we are trying to estimate. Since the statistical estimation problem is linear and the polynomial regression function is linear in the unknown parameters that are estimated from the data, polynomial regression is considered a special type of multiple linear regression.

3.2.3 Non-linear regression models

Non-linear regression is the third type of regression. In that case, the models are both non-linear in the way that x is not linearly correlated with $y = f(x, \beta)$ and the equation itself is not linear. Machine learning offers several of these algorithms. In this section, we will introduce the ones that appear most frequently in the literature about predictive modelling (Kohn & Johnson, 2013; Friedman, Hastie & Tibshirani, 2001), namely *Artificial Neural Networks (ANN)*, *Support Vector Machines (SVMs)*, *K-Nearest Neighbours (KNNs)*, and *Decision Trees (DT)*.

Artificial Neural Networks (ANN)

Artificial neural networks (also commonly called *neural networks*) are a powerful learning method, with successful applications in many fields (Hastie, Tibshirani, & Friedman, 2001). It is inspired by the way that the human brain processes information, as described by Haykin (2010).

The fundamental unit of neural networks is the **neuron**, also called a *node* or *unit*. It receives input either from some other neurons, or from an external source, and computes an output. Each input has a corresponding *weight*, which is assigned based on its relative importance to other inputs. Afterward, an activation function is applied to the given inputs, in order to produce the output value. Additionally, a *bias* input is added, whose main function is to provide a constant value to the function.

The first and simplest type of artificial neural network is the *feedforward neural network*. It is organized in layers, containing multiple neurons. It consists of the following three types of neurons:

- **Input neurons**, which feed data from external sources to the model and build the so-called *input layer*. No computations are performed in this layer, all neurons just transfer the information to the next layer.

- **Hidden neurons** have no direct connection with the outside world. They are responsible for calculations and transferring the information to the output neurons. All hidden neurons form the so-called *hidden layer*.
- **Output neurons** form the last layer, namely the *output layer*. They perform the final computations and pass the information to the outside world.

The connections of a typical feed-forward neural network do not form a cycle, and thus, the information flows only in a forward direction. However, there exist other model architectures, which have loops going in both directions between layers.

Support Vector Machines (SVMs)

Support vector machines are another class of powerful, highly flexible modeling techniques, whose theory originates from classification models (Kuhn & Johnson, 2013). The goal of the SVM algorithms is to find a *hyperplane* in an N-dimensional space that distinctly classifies the data points, where N is the number of features. Hyperplanes are decision boundaries that assist in the classification of data points. Different classes can be assigned to data points that lie on either side of the hyperplane. With two features, the hyperplane is a line. When the number of features increases to three, the hyperplane becomes a two-dimensional plane. It becomes more difficult to imagine when the number of features exceeds three. The position and the orientation of the hyperplane are influenced by the data points that are closest to the hyperplane. These are referred to as *support vectors*. The goal of the hyperplane is to maximize the margin between the support vectors on either side of the hyperplane, such that the support vectors form boundary lines. This way, the model can easily determine the target classes for new cases.

When the task is regression, the algorithm is commonly referred to as Support Vector Regression (SVR) and is based on a similar principle. The idea behind SVR is to find the best fit line, which is the hyperplane that has a maximum number of points. While other regression models try to minimize the error between predicted and real values, SVR fits the best line within a threshold value. The threshold value is the distance between the hyperplane and the boundary line.

Overall, the SVMs are a powerful algorithm, capable of discovering complex patterns in the dataset. A disadvantage, however, is that with the increasing number of samples, the computational time increases drastically.

K-Nearest Neighbours (KNNs)

K-nearest neighbours is another algorithm used both for regression and classification problems. The KNN method does not calculate a predictive model from a training dataset, meaning that there is no learning phase, and, thus, is categorized as a *lazy learning* method (Wettschereck, Aha & Mohri, 1997). KNN uses the entire dataset to make a prediction. For a new observation x for which we want to predict its output variable y, the algorithm will look for the K instances of the dataset closest to our observation. For regression problems, predictions are made based on the mean (or median) of the y variables of the K closest observations.

Hence, the KNN method requires the following input: a data set D, a distance function d, and an integer K. The distance function is chosen according to the types of data we are working with. For quantitative data of the same type, Euclidean distance is a good measure. When the input variables are not of the same type, then Taxicab geometry is a good candidate. Finally, to select the K value, we run the algorithm

multiple times with different values of K and subsequently, choose the K that results in the least number of errors, while maintaining the ability of the algorithm to make accurate predictions on unseen data.

Overall, the KNN method is simple and easy to implement because there is no need to build a model, tune parameters multiple parameters, and so on. However, its main disadvantage is becoming significantly slower as the volume of data and/or independent variables increases.

Decision Trees (DT)

Decision trees, which are generally applied to classification problems, utilize a flowchart-like tree structure to recursively predict the value of a target variable by learning simple decision rules inferred from prior data (training data). When the target variable is numerical, we refer to them as *regression trees*. While training a decision tree model, the dataset is split into smaller and smaller subsets and an equivalent tree structure is gradually generated at the same time. The resulting tree contains the following nodes:

- A **root node**, which represents the entire sample and gets further divided into two or more homogenous sets.
- **Decision node**, resulting from sub-nodes that further split into more sub-nodes.
- **Terminal nodes (leaves)**, representing the final subsets. They have no outgoing branches and terminate the tree structure, and thus, represent a prediction (or classification).

In a regression tree, the model searches the entire data set, including every value of every independent variable, to find the independent variable and split value that separates the data into two groups, such that the overall sums of squares error are minimized. Decision trees have the advantage of being highly interpretable and easy to compute. However, they have some noteworthy disadvantages, such as being more prone to suffer *overfitting* i.e., the tree is designed to perfectly fit all samples in the training data set, which leads to poor performance on unseen data.

3.3 Relevant variables

In the real world, many factors influence (truck) parking behavior. Within machine learning, these factors are translated into input variables which are used to predict the output variable(s). Provoost et al. (2019) point out the importance of feature selection both for optimizing the model's performance and for providing an improved understanding of the underlying processes. Thus, the following section contains a comprehensive literature study, aimed at determining the most promising input variables. We selected 9 articles, which study the parking occupancy prediction. Due to limited literature sources devoted to truck parking occupancy prediction, we consider articles researching other types of parking occupancy prediction, such as car parking. Nevertheless, we acknowledge the limited validity of these sources in the context of this research. Hence, this section is complemented with an in-depth discussion about the applicability of these features to the proposed model. In this section, we aim to answer RQ 3: *Which variables are most relevant to be used as input in the predictive model according to the literature?*

Traffic conditions, including parking behaviour, are highly dynamic over time, and therefore, time variables are among the most frequently chosen input variables. In fact, the variable *time of the day* is included in all articles, which is visible in Table 3-1. This is reasonable, as parking occupancy variates depending on the time of the day, as highlighted by Fabusuyi et al. (2014). In the context of truck parking, our preliminary research shows that truck parking occupancy also varies during the day, with peaks observed in the evening hours, when most truck drivers park for their long rest.

Table 3-1 Matrix of independent variables used in parking occupancy prediction models

Article	Time of the day	Weekday	Historical occupancy	Traffic flow	Rainfall	Temperature	Holiday	Event	Other
Provoost et al. (2019)	X	X	X	X	X	X			
Chen (2014)	X	X						X	X
Zheng et al. (2015)	X	X	X						
Reinstadler et al. (2013)	X				X	X	X	X	
Fabusuyi et al. (2014)	X	X	X		X			X	
Chawathe (2019)	X	X							
Kim & Koshizuka (2019)	X	X	X						X
Vlahodianni et al. (2016)	X	X	X						
Pflügler et al. (2016)	X	X		X	X	X	X	X	X

Another time-dependent variable, that is often cited in research, is *weekday*, ranging from Monday to Sunday. Vlahogianni et al. (2016), for instance, perform statistical testing before developing a prediction system and prove that there exist differences in the mean of parking occupancy between weekdays and weekends for all tested regions. Overall, this variable is mentioned by almost all selected authors and can therefore be regarded as an important predictor.

Next, several authors recognize the importance of *historical occupancy* as a strong predictor. After performing feature elimination, Provoost et al. (2019) observe that the preceding occupancy is the most important feature for the proposed by them model. Zheng et al. (2015) come to similar conclusions. The authors observe that including the historical occupancy yields better results than considering the time of the day and day of the week alone, with an improved performance of 30%. Hence, providing that a lookback window is possible, the historical occupancy appears to be an important input variable.

Additionally, some authors suggest adding a weather variable, such as *temperature* or *rainfall*, in the model. For instance, Reinstadler et al. (2013) specifically highlight the importance of weather data, which appears with a rather high weight in their regression model. On the contrary, Provoost et al. (2019) observe that weather variables improve their model to a lesser extent. A way to explain this is that weather conditions are country/region-specific and therefore, affect the traffic conditions differently. In the context of truck parking occupancy, it might be useful to explore whether the weather conditions affect the parking behaviour. It is reasonable to assume that, for instance, on hot days there might be more truckers wishing to park at a private truck parking area due to the availability of services, such as showers.

Two of the selected literature sources explore the importance of *traffic intensity*. While Pflügler et al. (2016) state that this variable is of secondary importance for modelling parking flows, Provoost et al. (2019) conclude that traffic flows is one of the most important features in their model. A reason why this variable is rarely cited in research might be the unavailability of data streams, making it harder to

implement in a model. In the context of truck parking occupancy prediction, it is uncertain whether the traffic intensity affects the occupation. Since the traffic variable is complex to model, we decide to omit this predictor and leave it for experimentation in the future.

In some of the relevant articles, the authors incorporate features representing external factors, such as *holidays* and *events*. Reinstadler et al. (2013) observe that adding these variables improves the model's performance. Pflügler et al. (2016), however, challenges this by stating these features are only of secondary importance. In the context of truck parking, we assume the variable event to be of no relevance, as truck parking areas are located outside cities, and thus, events happening in the cities will not impact the occupancy. On the contrary, we consider the variable holiday to be more important, as our preliminary research shows that the occupancy drastically increases on German national holidays. This is due to the truck traffic ban imposed by the German government, which does not allow trucks to enter highways on holidays. Furthermore, the Dutch national holidays might also affect truck parking occupancy, due to changes in the working hours of distribution centers and logistic companies. Hence, this seems like a promising predictive variable.

Overall, it becomes evident that time variables, such as the *time of the day* and *day of the week* seem to be the most prominent predictors of truck parking occupancy. Next, the *historical occupancy* is also positively regarded in the literature. Secondary to this, the variables *temperature*, *rainfall*, and *holiday* could increase the predictive power. On the contrary, the variable *event* is disregarded from the potential predictors. Finally, whether or not the *traffic flow intensity* is a good predictor for truck parking models remains unclear and the variable will not be further explored in this study.

3.4 Analysis of (truck) parking occupancy prediction techniques

In section 3.2, we briefly introduced the most common types of machine learning techniques, which are used to predict numerical outcomes. The following section will further elaborate on this topic, as we will analyze existing literature about parking occupancy prediction techniques. As far as we know, little previous research has focused on truck parking occupancy prediction. Therefore, we investigate articles concerning car parking occupancy and this subsection will answer RQ 4: *Which forecasting techniques are known in the literature for predicting (truck) parking occupancy rates?*

Prior work has studied the effect of various methods on the prediction of parking occupancy. Zhao and Zhang (2020) test linear regression, SVM, neural network, and ARIMA[10] and conclude that SVM outperforms the other algorithms. The authors argue that SVM offers higher complexity with more parameters compared to ARIMA and linear regression. Furthermore, they demonstrate that by adopting a structure risk minimization principle, the SVM model does not suffer the typical weaknesses of conventional models, such as overfitting problems, which allows it to obtain more stable and robust results.

Similarly, Zheng et al. (2015) evaluate the performance of various prediction mechanisms for the parking occupancy rate, namely a regression tree, support vector regression, and neural network. Their analysis reveals that the regression tree outperforms the SVM and neural network for parking availability while being the least computationally intensive algorithm. They argue that a regression tree is not only easy to build and fast to train, but also easy to interpret after fitting the data. Furthermore, Reinstadler et al.

[10] ARIMA stands for AutoRegressive Integrated Moving Average. It is a class of statistical models used for analyzing and forecasting time series data.

(2013) point out that compared to other techniques, regression trees offer more flexibility and are "often also more powerful"(p.6). This is supported by others, for example, Fabusuyi et al. (2014) compare the performance of regression trees to other prediction techniques and conclude that regression trees perform "superior" to other models (p.296). The authors argue that this is because other techniques, such as ordinary linear regression and time series techniques, assume that the input variables are independent. On the contrary, regression trees do not make that assumption and freely explore any correlations in the feature structure.

To predict parking occupancy rates based on historical data, Chawathe (2019) also studies a variety of 12 regression algorithms, including the previously mentioned regression tree, random forest, linear regression, and SVMs. The author further experiments with different feature sets and his findings show that the random forest method, which is a variation of regression tree, performs best while also being insensitive to the feature selection. Kim and Koshizuka (2019) support this, after testing a variety of tree-based regression models, such as random forest, gradient boosting regression tree, Xgboost[11] and lightGBM[12]. The authors find the random forest algorithm to achieve the highest accuracy.

Additionally, neural networks also receive a lot of attention in the literature and appear to produce promising results. For instance, Provoost et al. (2019) implement both a random forest and a feed-forward neural network model and compare their performance. The feed-forward neural network outperforms the random forest; however, the observed differences are small and both models outperform a naïve model. Next, Vlahogianni et al. (2016) develop a system for parking availability prediction in urban areas, by analyzing the performance of neural networks. The authors note that neural networks can adequately capture the temporal evolution of parking occupancy and accurately forecast the occupancy. Similarly, Pflügler et al. (2016) develop a neural network to predict the parking space availability of urban areas. The authors claim that neural networks are particularly well suited to forecast occurrences where little or nothing is known about the underlying relationships and features of the events, but there is sufficient training data or observation values. Moreover, neural networks offer continuous learning, and it is possible to continuously improve the prediction.

Nevertheless, some authors point out some disadvantages of neural networks, such as their 'black-box' concept, which prohibits stakeholders from recognizing the impact and influence of any variable (Provoost et al., 2019). Finally, Chawathe (2019) claims that while reporting good prediction accuracy, neural networks sustain high model building costs, due to the long training times. This is further supported by Chen (2014), who compares the performance of ARIMA, linear regression, SVM, and feed-forward network. The latter provides the best prediction among the tested models, however, it accumulates the longest training time.

In conclusion, many authors positively regard decision trees due to their transparency, which allows analyzing correlations between variables. Next, the random forest method receives a lot of attention in the literature and has the potential to perform even better. Finally, ANN appear to produce good results, even though they require long training times and are unable to provide clear insights about the underlying structure due to their black-box concept. Nevertheless, there is no consensus in the literature and the

[11] Xgboost stands for eXtreme Gradient Boosting, which is an implementation of gradient boosting machines, engineered for efficiency of compute time and memory resources.
[12] lightGBM stand for light Gradient Boosting Machines, which is a framework using tree-based learning algorithms.

research on truck parking occupancy is limited. Thus, to choose an algorithm for implementation, different approaches need to be evaluated.

3.5 Performance evaluation

Models predicting a numeric outcome typically use some measure of accuracy to evaluate the effectiveness of the model. There are different metrics to measure accuracy, and each has its strengths and weaknesses. Furthermore, measures of accuracy also depend on the model type. The following section will therefore discuss these techniques and their applicability in the context of this research and answer RQ 5: *Which evaluation metrics can be used to assess the model's performance?*

Firstly, the *coefficient of determination*, or R^2, is a common metric for assessing predictive models. It is interpreted as the proportion of the variance for a dependent variable that is explained by the independent variable(s) in a regression model, hence it indicates the strength of the relationship between the model and the dependent variable. It has a range of [0,1] and the higher the value, the better the model fits the data i.e., R^2 assesses the goodness-of-fit of a regression model. In the formula below $\hat{y}$ denotes the predicted value and $\bar{y}$ is the mean value of y.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2}$$

However, R^2 always increases with the number of explanatory variables in the model, regardless of whether these variables improve the predictability of the model or not (Chugh, 2020). To solve this issue, one can use a modified version of R^2, the *adjusted R^2*, which includes the number of predictor variables in the model. In the formula below, n denotes the number of observations in the data and k is the number of explanatory variables.

$$R^2_{adj} = 1 - \frac{(1 - R^2)(n - 1)}{(n - k - 1)}$$

The value of the adjusted R^2 is always less than or equal to R^2. If the value of the additional variables is not significant enough, the adjusted R^2 decreases, so intuitively, one wants the value of the adjusted R^2 to be as close as possible to R^2.

In the context of parking occupancy prediction, R^2 is mentioned by Zheng et al. (2015). However, the authors use it in combination with other metrics to provide a better understanding of the performance of the model. Kuhn and Johnson (2013) acknowledge this and point out that while being an easily interpretable statistic, one must remember that R^2 is a measure of correlation and not accuracy. The authors further demonstrate this by showing a model with a high value of R^2, which, nevertheless, tends to overpredict low values and underpredict high ones. Therefore, for characterizing the model's predictive capabilities, a more sophisticated metric is desired.

For evaluating prediction accuracy, the most used metric is the *Root Mean Squared Error*, or *RMSE* (Kuhn & Johnson, 2013). The *Mean Squared Error (MSE)* represents the average of the squared difference between the original and predicted values in the data set i.e., the variance of the residuals. Then, as the name indicates, the root mean squared error, is the arithmetic square root of the *mean squared error* i.e., the standard deviation of the residuals. Hence, the RMSE is measured in the same units as the dependent

variable, which makes it more interpretable. The RMSE is interpreted as how far on average the errors are from zero, meaning that lower values of RMSE imply higher accuracy of the model.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2}$$

The (root) mean squared error penalizes large prediction errors more than small prediction errors. Therefore, for data sets containing outliers, a more robust evaluation metric might be the *mean absolute error* (MAE), which represents the absolute difference between the actual and predicted values in the data set.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}|$$

In the context of parking occupancy prediction, authors most commonly use (R)MSE or a combination of MAE and (R)MSE to measure the model's performance. However, Provoost et al. (2019) note that these metrics have the drawback of becoming infinite or undefined when the occupancy approaches zero and propose a way to solve this, by using the *Mean Absolute Scaled Error* (MASE). It is calculated by dividing the tested model's MAE by that of an arbitrary naive[13] model. Handyman (2006) supports this claim and explains that MASE would become infinite or undefined only if all historical observations are equal. Moreover, MASE can demonstrate the added value of each model compared to a benchmark model, which provides additional insights into performance for stakeholders (Provoost et al., 2019). However, it is more time demanding and computationally expensive, compared to its simpler counterparts MAE and (R)MSE. Finally, MASE seems to provide a more valuable assessment during the inter-model comparative testing phase than during the training and validation of each model.

$$MASE = \frac{MAE}{MAE_{naive}}, \text{where } MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}|$$

Overall, R^2 and the adjusted R^2 are easy to understand and interpret but do not provide good insight into the actual performance of the model. MAE and (R)MSE are widely used to evaluate the performance of regression models but have some disadvantages, such as becoming undefined. These metrics, however, are useful during the training and validation phase of each model, as they provide a tangible performance measure for stakeholders. Furthermore, some algorithms require a loss function, which is often the MSE. In that case, there is a direct link between the performance measure and the training of the model. Finally, once the models have been trained and validated, MASE is a useful metric to compare the models between each other and to observe how they perform against a naïve model, providing a stronger understanding of the added value of the model.

[13] Naive model forecasting is a technique in which minimum amounts of effort and manipulation of data are used to generate the forecast. These models are used only for comparison with the forecasts generated by more sophisticated techniques.

3.6 Car parking versus truck parking occupancy prediction

One of the aims of the literature study was to discover whether there exist some differences in predicting the occupancy of different types of parking lots i.e., car parking and truck parking. Both types of parking lots operate in different environments, are affected by different factors, and host different customers, thus, it is logical to determine whether their occupation is predicted by the using same approach. Following this, the final question we wish to address in this chapter is RQ 6: *What differences and similarities are there between predicting car parking occupancy and truck parking occupancy?* The main differences and similarities that we wish to discover concern the input variables of the prediction models and the applied machine learning algorithms.

As demonstrated in sections 3.3 and 3.4, a closer look at the literature on truck parking occupancy prediction reveals many gaps and shortcomings. Previous research has almost exclusively focused on analyzing car parking areas. Only a few studies focus on truck parking areas or use truck parking data for testing and training prediction models. Although several authors conducted studies, this topic is still insufficiently explored. Specifically, in the context of truck parking occupancy prediction, the machine learning approach remains rarely addressed and whenever addressed, authors use different approaches and algorithms. Following these observations, we conclude that currently, there is a lack of evidence for the comparison of the differences and similarities between forecasting car and truck parking occupancy. Therefore, this remains an open question. However, we see a clear opportunity to use this research as a way to fill the identified literature gap.

3.7 Conclusions

Selecting an appropriate approach is crucial for the successive development and validation of a machine learning model. The goal of this literature review was to assist us in determining relevant input variables, choose a suitable prediction algorithm and select an appropriate evaluation metric for validation of the resulting model. Using a systematic literature review, we defined the following candidate prediction variables, namely: time of the day, day of the week, temperature, rainfall, Dutch national holidays, German national holidays, and historical occupancy. Nevertheless, we used research in car parking occupancy as a basis and it remains unclear to what extent the findings are applicable in the domain of truck parking occupancy prediction. Thus, more experimentation with the preselected variables is necessary, before determining a final subset of variables. Regarding the machine learning algorithms, the situation is similar. There is no consensus in the literature and different authors use different approaches. Following this, we conclude that experimentation with different algorithms is needed, in order to select an appropriate one. When it comes to the selection of evaluation metrics, we conclude that a balanced combination of several metrics is necessary. However, the final selection will depend on the chosen algorithm. Overall, we find truck parking occupancy prediction to be rarely addressed in the literature, and thus, drawing conclusions from the literature only regarding the addressed research questions is insufficient. However, this gives us a clear opportunity to use this research as a means to fill in the identified knowledge gaps.

Chapter 4 Data understanding

Machine learning models learn from data and, thus, require large amounts of data for training. However, to get the desired results, obtaining a lot of data is not enough, because the success of a machine learning project is also highly dependent on the quality of the data (Komarraju, 2021). Following this, Chapman et al. (2000) advise assessing the data, as well as available tools and techniques, as early as possible, as these influence the entire project. In this chapter, we start with initial data collection and description in sections 4.1 and 4.2, followed by data exploration and verification of data quality in sections 4.3 and 4.4, respectively. The goal is to get familiar with the structure of the datasets, as well as inspect them for missing values and outliers, and to discover initial insights that we could use to formulate hypotheses. The chapter addresses RQ 7: *What data is available that is relevant for predicting the occupancy rates of truck parking areas in Overijssel?*

4.1 Collection of initial data

We will start this chapter by gathering the data, which can be used to form features for the proposed prediction model. Given the fact that the intended model aims to predict the variable occupancy rate, **truck parking data** is undoubtedly the most important data source for this research. Unfortunately, historical truck parking open data sources are still limited today. However, as some truck parking areas are willing to participate in research studies, we were able to obtain historical data from the *A1 Truck Parking Deventer*. Consequently, the scope of data collection and the research becomes the A1 Truck Parking Deventer. The provided dataset contains parking transactions from the 1st of January 2020 until the 15th of June 2021. The exact occupancy numbers are not present in the data, but it is possible to calculate them based on the provided information. Following this, we can derive the historical occupancy variables as well.

Section 3.3 suggested that weather variables, such as **temperature** and **rainfall**, might influence positively the predictive performance of the model. To collect weather data, we utilized data from the Royal Netherlands Meteorological Institute KNMI (KNMI, n.d.). KNMI offers an online platform where the historical hourly data of different variables can be queried. We chose to collect the measurements of the Deelen[14] weather station due to its proximity to the A1 Truck Parking Deventer (approximately 30 km). We obtained the hourly data from the 1st of January 2020 until the 15th of June 2021.

Next, in section 3.3 we also concluded that the variable **holiday** might be a useful addition to the model, as we would like to investigate the impact of both Dutch and German holidays on truck parking behaviour. We are interested in the national holidays in the period between the 1st of January 2020 and the 15th of June 2021. Firstly, we obtained a list of public holidays in Germany when trucks are banned through the website of the Federal Office for Freight Transport (Bundesamt für Güterverkehr, n.d.). Next, we gathered a list of all national Dutch holidays through the website of the Government of the Netherlands (Ministerie van Algemene Zaken, n.d.). Table 4-1 displays the findings.

[14] The Heino weather station is the closest one to the studied area (approximately 20 km), however, the dataset contained a lot of missing values and did not provide any information about rainfall, which is the reason why we selected the Deelen weather station instead.

Table 4-1 A list of all Dutch and German public holidays in the period 01.01.2020 - 15.06.2021

German public holidays	Dutch public holidays
2020	
01.01.2020 – New Year	01.01.2020 – New year
10.04.2020 – Good Friday	10.04.2020 – Good Friday
13.04.2020 – Easter Monday	12.04.2020 – Easter Sunday
01.05.2020 – Labor Day	13.04.2020 – Easter Monday
21.05.2020 – Ascension Day	27.04.2020 – King's Day
01.06.2020 – Pentecost (Whit Monday)	05.05.2020 – Liberation Day
11.06.2020 – Corpus Christi	21.05.2020 – Ascension Day
03.10.2020 – German Reunification Day	31.05.2020 – Pentecost (Whit Sunday)
31.10.2020 – Reformation Day	01.06.2020 – Pentecost (Whit Monday)
01.11.2020 – All Saints' Day	25.12.2020 – Christmas
25.12.2020 – Christmas	26.12.2020 – Christmas
26.12.2020 – Christmas	
2021	
01.01.2021 – New Year	01.01.2021 – New Year
02.04.2021 – Good Friday	02.04.2021 – Good Friday
05.04.2021 – Easter Monday	04.04.2021 – Easter Sunday
01.05.2021 – Labor Day	05.04.2021 – Easter Monday
13.05.2021 – Ascension Day	27.04.2021 – King's Day
24.05.2021 – Pentecost (Whit Monday)	05.05.2021 – Liberation Day
03.06.2021 – Corpus Christi	13.05.2021 – Ascension Day
	31.05.2021 – Pentecost (Whit Sunday)
	01.06.2021 – Pentecost (Whit Monday)

Finally, during the literature review (section 3.3) it became evident that **time variables**, such as time of the day and weekday, are among the most prominent predictors of the model. However, there is no need to collect them from sources, as these variables are already present in the previously obtained datasets (truck parking and weather data). These datasets are characterized by a timestamp, describing the date and time of each event. We can use the corresponding timestamps to obtain other time-related variables, such as the day of the week.

4.2 Description of data

To get more familiar with the previously acquired datasets (truck parking data and weather data), we will start by examining their surface properties, such as volume and format, and evaluate whether the data satisfies our requirements.

Truck parking dataset

The truck parking historical dataset was provided as an Excel Worksheet. It consists of 46212 rows (transaction entries), where each row denotes a single parking movement, either an incoming or an outgoing truck. Hence, each truck is present in the dataset twice.

Table 4-2 Attributes of the raw truck parking dataset

Original column name	Description	Format
ID	Unique ID number	General[15]
Datumtijd	Timestamp (date and time)	General
Toegandid	Movement ID (1=Enter, 2=Exit)	General
Kaartnummer	Parking meter number	General
Lezer	Used parking meter reader (entry or exit)	General
Toegang	Movement (in or out)	General

As visible in Table 4-2, the dataset does not contain definitive information about the occupancy rates, which is the variable that we aim to predict. However, we may utilize the available data to derive the occupation ourselves. Intuitively, not all columns contain relevant information for determining the occupancy. Based on their ability to help us calculate the occupancy rates, we selected the following attributes: *Datumtijd* and *Kaartnummer*, which denote the timestamps of each event (in- or outgoing vehicle) and the unique parking meter numbers. All remaining columns were deleted from the dataset.

Weather dataset

We downloaded the weather dataset as a TXT file. From the KNMI data source, we acquired the variables *air temperature at 1.5-meter height* (measured in 0.1 °C) and *rainfall* (a binary variable indicating whether rain fell in the last hour). As we aim to predict the truck parking occupation, preprocessing, and exploring the parking dataset is a crucial task, which we will focus on in the rest of the chapter. The weather dataset will be explored for missing or erroneous values during the data preparation phase.

4.3 Exploration of data

4.3.1 Initial preprocessing

To be able to further explore the truck parking dataset, it is essential to start by calculating the occupancy rates. We start by further focusing on the structure of the data. As mentioned in section 4.2, each record represents a single parking movement, a vehicle entering or exiting the truck parking. The associated columns are a timestamp, containing the date and the time of the event, and a unique parking meter number. The data set is ordered chronologically based on the timestamps. The timestamps are unevenly spaced, meaning that we are dealing with *irregular time series*. As we aim to evaluate the occupancy at fixed points in time, the dataset needs to be transformed.

Our next goal is to reshape the dataset so that every row denotes the parking movement of a particular truck, characterized by the *in* and *out* timestamps. This requires us to start with initial verification of the quality of the data and cleaning it, if necessary. Firstly, we identify the incomplete parking movements (e.g., a truck which enters the parking area but never leaves). We regard these cases as invalid to guarantee the stability of the occupancy numbers. As the occupation of a given parking area is a function of the ingoing and outgoing vehicles, incomplete movements, in case they are not compensated by another incomplete movement in the opposite direction, could destabilize the occupation numbers over time. Hence, we aim at reshaping the dataset so that in every row we have both in and out timestamps, which will ensure the stability of the dataset.

[15] This is the default number format for all cells in a new worksheet in Excel, and it shows the data the same way as it is typed in.

To do so, we look at the unique parking meter numbers. For most of the dataset, each unique parking meter number appears twice in the column *Kaartnummer*, when the vehicle enters the parking and when it exists. If a parking meter number appears only once, this denotes an incomplete movement. This could be explained, for example, as a technical error. We find 2218 unique values in the parking meter number column (approximately 5% from the whole dataset), which are subsequently deleted from the dataset. Next, we combine the duplicated rows into one, which results in a dataset with 21968 observations.

Next, when further exploring the dataset, we find that 19 rows of the *Datumtijd* column contain more than two timestamps. Further analysis shows that for these columns the first few timestamps differ by a few seconds. This noise might be explained as a technical error. As the number of erroneous rows is relatively low, we decide to manually adjust them, by keeping the first and last timestamps. After cleaning the dataset, we divide the column *Datumtijd* into two new columns, named *In* and *Out*, each containing the associated timestamp. Subsequently, we delete the column *Kaartnummer*, as we do not need it for further analysis.

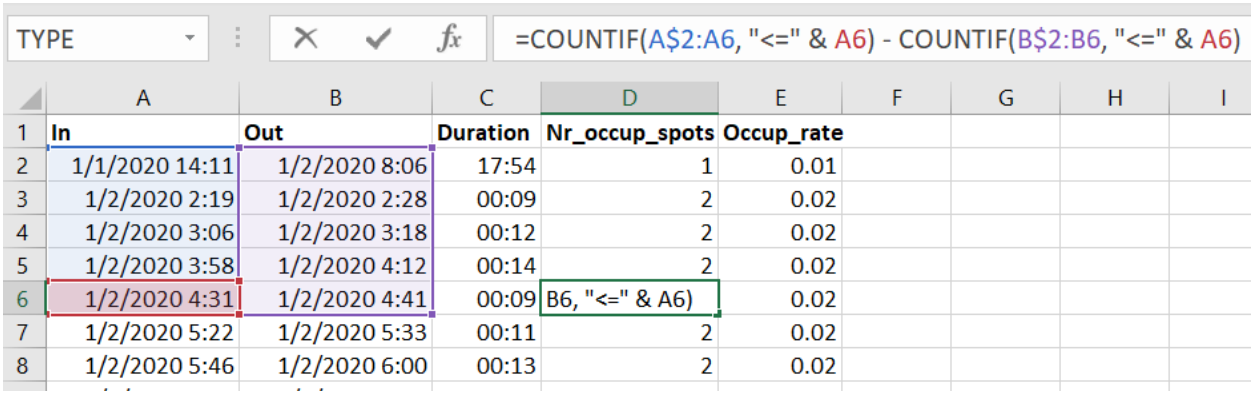
Then, we add a new column to the dataset which denotes the duration (in the format HH:MM) of each vehicle in the truck parking. To derive it, we apply the following formula:

$$Parking\ duration\ of\ truck\ x = Timestamp\ Out\ of\ truck\ x - Timestamp\ In\ of\ truck\ x$$

The next step is to calculate the number of occupied parking spaces after each event and assign them to a new column. For this purpose, we construct a formula in Excel, visualized in Figure 4-1. Finally, we determine the occupancy rate with the following formula:

$$Occupancy\ rate\ at\ time\ t\ = \frac{Number\ of\ currently\ occupied\ parking\ spots\ at\ time\ t}{Total\ number\ of\ parking\ spots}$$

The occupancy rate is formatted as a decimal number. The first few rows from the resulting dataset are displayed in Figure 4-1.



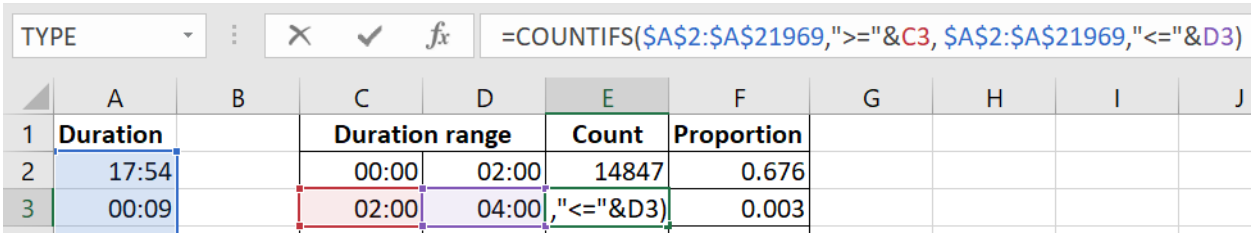
	A	B	C	D	E	F	G	H	I
1	**In**	**Out**	**Duration**	**Nr_occup_spots**	**Occup_rate**				
2	1/1/2020 14:11	1/2/2020 8:06	17:54	1	0.01				
3	1/2/2020 2:19	1/2/2020 2:28	00:09	2	0.02				
4	1/2/2020 3:06	1/2/2020 3:18	00:12	2	0.02				
5	1/2/2020 3:58	1/2/2020 4:12	00:14	2	0.02				
6	1/2/2020 4:31	1/2/2020 4:41	00:09	B6, "<=" & A6)	0.02				
7	1/2/2020 5:22	1/2/2020 5:33	00:11	2	0.02				
8	1/2/2020 5:46	1/2/2020 6:00	00:13	2	0.02				

Figure 4-1 Overview of the preprocessed truck parking dataset with derived parking duration of each vehicle and resulting occupancy rates

4.3.2 Analysis of parking durations and occupancy rates

After performing the initial preprocessing of the truck parking data, our next task is to investigate the resulting outputs. We start with further analysis of the truck parking durations, as we are interested in how long trucks spend in the parking location. We define nine duration ranges and calculate the

corresponding number of trucks that belong to this category. Figure 4-2 shows how the numbers are calculated.



	A	B	C	D	E	F	G	H	I	J
1	**Duration**		**Duration range**		**Count**	**Proportion**				
2	17:54		00:00	02:00	14847	0.676				
3	00:09		02:00	04:00	,"<="&D3)	0.003				

Figure 4-2 Excel formula for counting the numbers of trucks based on parking duration

Lastly, based on the calculated counts and the total number of vehicles, we determined the corresponding proportions. Figure 4-3 displays the findings. The left pie chart clearly shows that approximately two-thirds of the trucks visiting the parking location spend between zero and two hours there. As this is the largest number, we decided to further examine how the durations are distributed within this range. Subsequently, we created new duration ranges every 30 minutes from zero to two hours and calculated the proportions, displayed in the pie chart on the right. The outputs show that approximately 60% of all trucks are parked for a period of up to 30 minutes. This comes as no surprise considering that truck drivers are obliged to stop for a short break of 15 to 30 minutes after every 4½ hours of driving according to the European regulations (EC Regulation No 561/2006). From the pie charts, it becomes clear that truck drivers rarely stop for a period of one up to eight hours. Next, the figure illustrates that approximately 23% of the trucks were parked for a period between 8 and 14 hours, which corresponds to the truck drivers who stop for their obligatory long rest period. Finally, the pie chart shows that almost 7% of the trucks spend over 16 hours at the parking locations. This number could be explained by the trucks that arrive at the parking lot during the weekend and need to wait for a long period at the parking lot due to the truck traffic ban in Germany.

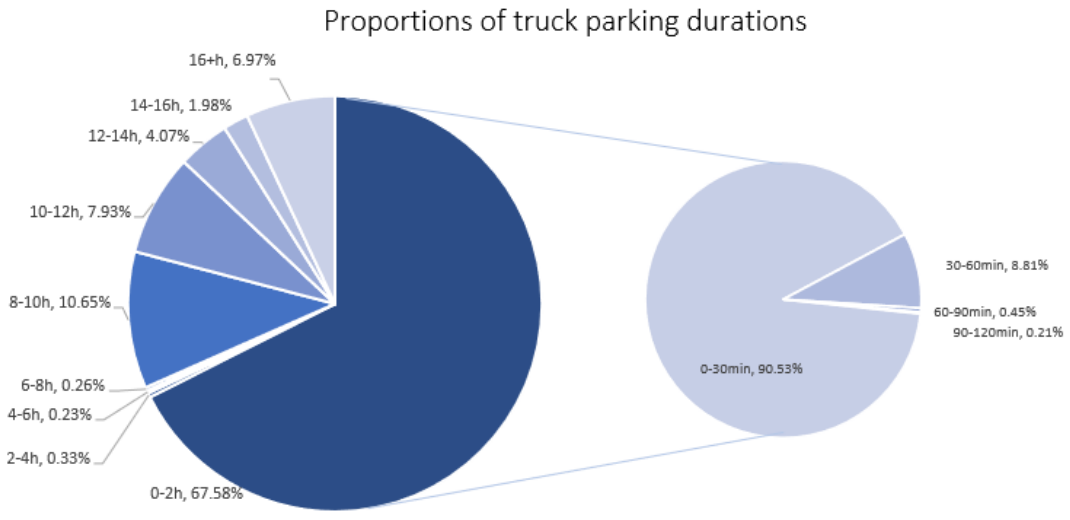


Figure 4-3 Overview of truck parking durations. On the left: the two-hour-long duration ranges from 0 to 16+ hours. On the right: the 30-minute-long duration ranges from 0 to 2 hours.

After analyzing the parking durations, the rest of this section will focus on the occupancy rates. To examine the dataset more thoroughly, we utilized the Python programming language, as its extensive range of libraries for data science provides a solid basis for this purpose. The literature review (section 3.3) suggested that time variables are among the most prominent predictors of parking occupancy. Therefore, we would like to investigate how the truck parking occupancy changes based on the time of the day, day of the week, and month of the year.

We started by examining how the truck parking occupancy fluctuates during the day. We divided the day into 24 hours and grouped the occupancy based on the corresponding hour of the day, displayed in Figure 4-4. The boxplot shows that the occupation highly fluctuates in a day. From 11:00 until 17:00, the occupation number is relatively low. It starts increasing in the afternoon (after 17:00), reaches its peak in the early morning hours (around 3:00) and then, decreases slowly. This observation is not surprising, as most truck drivers take a longer rest during the night so that they can get enough hours of sleep. As more trucks are parked for longer periods, the occupation numbers naturally increase. Finally, we conclude that the time of the day is a characteristic that should be added to the model, due to its potential to positively affect the prediction process.

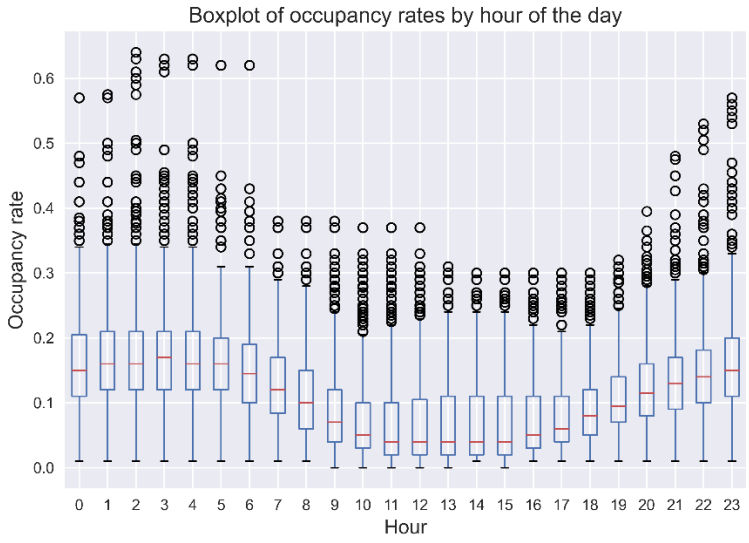


Figure 4-4 Overview of truck parking occupancy fluctuations during the day

Secondly, we analyzed how the occupancy rates vary according to the day of the week by grouping the weekdays and calculating the mean occupancy rates. The results are displayed in Figure 4-5. and the bar chart shows the fluctuations of the occupancy during the week. Firstly, when comparing the workdays and weekends, we see that the average occupancy rates on the weekend are 5-10% higher than during the business working days, with a peak on Sundays. A way to explain this is the German truck traffic ban which does not allow trucks to enter Germany on Sundays. Secondly, we look at the fluctuations during the workdays more closely. The differences are not staggering, however, the truck parking demand seems to be the highest on Wednesdays. Overall, the chart shows that truck parking demand fluctuates over the week, hence, adding the weekday to the predictors might aim the prediction model. Furthermore, adding

another variable for differentiating between the workdays and weekdays also might increase the model's capabilities.

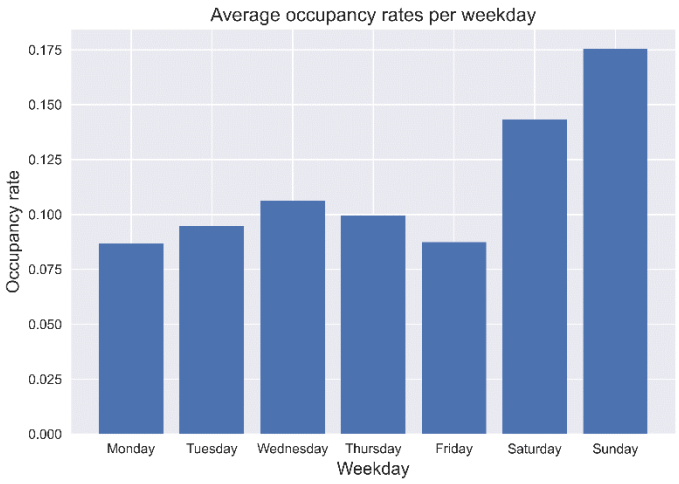


Figure 4-5 Overview of the average truck parking occupancy fluctuations during the week

Finally, we investigated whether the truck parking demand is dependent on the month of the year. As we were provided with observations from January 2020 until June 2021, the quantity of the data was enough to produce accurate results. As visible in Figure 4-6, there are no observable monthly variations of the occupancy rates, and the data does not show monthly seasonality. Thus, we conclude that adding the month as a feature of the prediction model is not necessary.

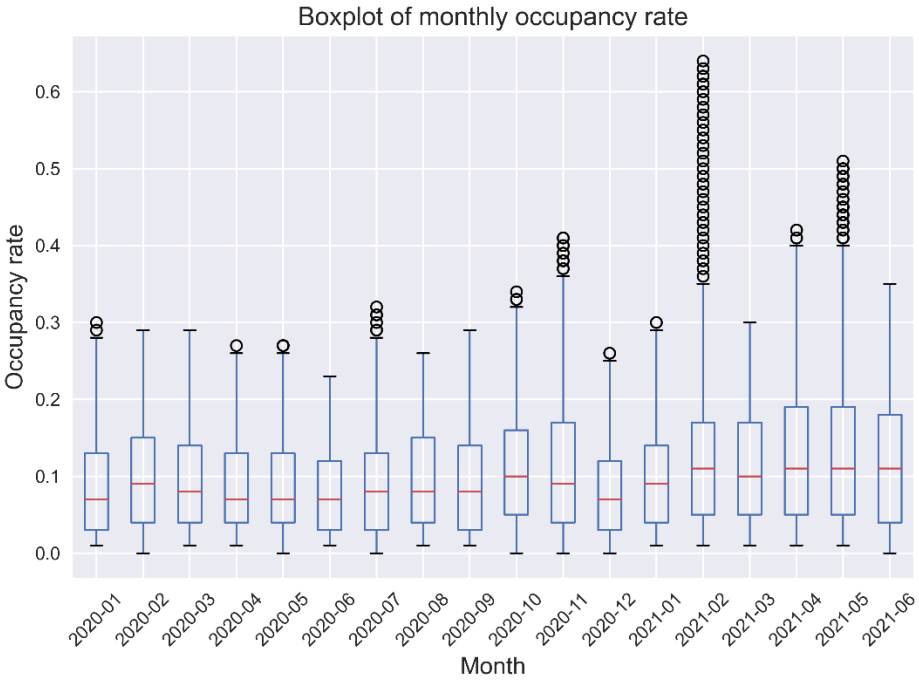


Figure 4-6 Overview of monthly truck parking fluctuations

4.4 Verification of data quality

Now that we have explored our data, we have a better overview of the truck parking behaviour. Next, we are going to measure the quality of the data and check for ambiguity and missing values, respectively.

In section 4.3.1, we performed initial preprocessing of the truck parking data and identified 2218 erroneous cases (incomplete parking movements). To perform the data exploration, we removed them from the dataset. Consequently, the resulting number of observations (complete parking movements) became 21968, meaning that we lost approximately 10% of the data. Secondly, as visible in Table 4-2, the column with the timestamps was not formatted appropriately in the raw dataset. To allow for further processing and performing operations with dates and times, we adjusted the format to datetime. We did not find other formatting inaccuracies in the raw data set. Finally, we examined the derived occupancy rates for inconsistencies by calculating the standard statistical measures, displayed in Table 4-3. The occupancy rate should always be between 0 and 1 and as visible in the table, all numbers are within the interval. Additionally, we do not observe other irregularities.

Table 4-3 Standard statistical metrics of the variable occupancy rate

Count	Mean	Std	Min	Max	Median	Kurtosis	Skewness
21968	0.103718	0.077815	0.00	0.64	0.09	2.812433	1.276927

Next, inspecting the dataset for missing values is a challenging task due to the irregularity of the data points (recorded every time a truck arrives and not in regular intervals). Hence, we examine the data for missing days or periods of data by visualizing the average daily occupancy rates in Figure 4-7. As visible, there is a missing period of data, spanning from the 16th of August until the 4th of September of 2020.

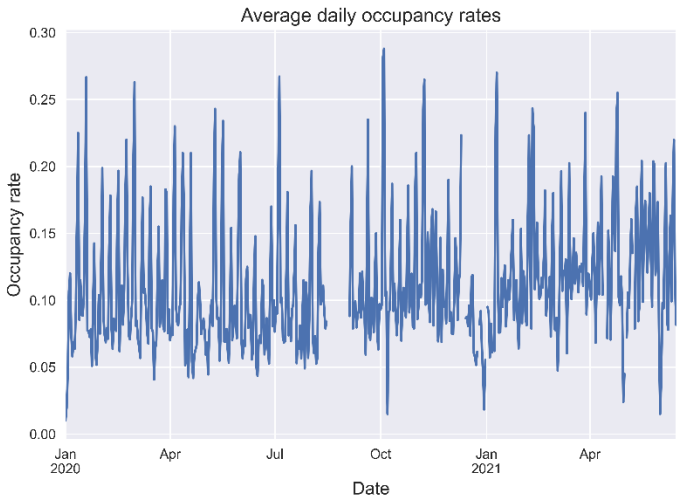


Figure 4-7 Overview of average daily occupancy rates

Then, we determine what fraction of the measurements is missing. The data set spans over 532 days and data from a period of 20 consecutive days is missing. Thus, the missing data accounts for 3.7% of the whole measured period. We further investigated the reasons for the incomplete dataset, which relate to

a temporary closure of the truck parking entrance due to construction work. During this period, the measurement system was not utilized, however, the truck parking area was operational as usual.

When dealing with real-world problems, data scientists often face missing values in the data because of technical or human errors. Traditionally, these values are either omitted or replaced by using imputation methods. While omitting the missing values might lead to temporal discontinuity, imputation methods may change the original time series. Since we already lost some observations after deleting incomplete transactions during the preprocessing of the parking data, finding an appropriate imputation method to fill in the missing values is our preferred choice. Our preceding analysis shows that the data has both daily and hourly patterns, as the average occupancy rates differ based on the day of the week and the hour of the day. Therefore, our goal is to fill in the missing values, so that these characteristics are taken into account.

4.5 Conclusions

In this chapter, we collected initial data, which we could utilize in our prediction model. Truck parking data, which is the most important part of the data-gathering phase, was obtained from the A1 Truck Parking Deventer. Subsequently, the dataset was reshaped, preprocessed and the occupancy rates were derived and analyzed. Following this, we explored the occupation and truck parking durations and discovered interesting dependencies and seasonal features of the occupancy rates, such as hourly and daily variations. Finally, we verified the quality of the observations and evaluated the missing values. We discovered approximately three weeks of missing data and choose to impute them before modelling. Additionally, we collected weather data, including temperature and rainfall, and obtained a list of all Dutch and German national holidays, which could be implemented as predictors in the forecasting model.

Chapter 5 Data preparation

Now that we have gathered more insight into the available data, we will further prepare it for modeling. We will start by providing a definitive overview of which attributes will be included in the dataset in section 5.1. As all sources have their format and data structure, we should refine the data before supplying it to the training, testing, and validation process. We will first clean the individual datasets and once the datasets are cleaned, we will merge them and establish the full dataset. These tasks are addressed in sections 5.2 and 5.3, respectively. Finally, we will address feature selection by applying pairwise correlation, which is explained in section 5.4. In this chapter, we focus on RQ 8: *How should the dataset, which will be fed to the model, be configurated?*

5.1 Choice of attributes

Based on the literature review (section 3.3) and data exploration (section 4.3.2), we came across several features that could be used as input to our model. Based on data availability and feasibility, we selected the attributes which will be used in the prediction model. Table 5-1 shows a definitive overview of the dependent and independent variables.

Table 5-1 Overview of dependent and independent variables

Dependent variable	**Independent variables**	
Occupancy rate	Time of the day	Temperature
	Day of the week	Holiday in Germany
	Weekend or not	Holiday in the Netherlands
	Rainfall	Historical occupancy

5.2 Data cleaning

After initializing which data will be used as input to the prediction model, we will start by cleaning the truck parking and weather historical datasets, respectively.

5.2.1 Truck parking data

In section 4.3.1, we performed some initial preprocessing of the truck parking data to derive the occupancy rates. For the cleaning process, we will utilize the *Python* programming language due to its extensive range of libraries for data processing, such as Pandas. We started by appending the dataset to a Pandas *DataFrame*[16]. Subsequently, we deleted the obsolete columns (i.e., *'Out'*, *'Duration'*, *'Nr_occup_spots'*), such that only the columns with the timestamps and the occupancy rates remained. Then, we renamed the column containing the dates and times to *'Date'* and set its formatting as a *datetime* type, which enables us to perform temporal operations, such as interpolation, extrapolation, or resampling. Next, we set the datetime column as an index of the DataFrame and *resampled* the dataset in one-hour intervals, by taking the average of the occupancy rates for each hour. This results in a dataset with some empty cells, as the occupancy rates do not always change every hour. To fill in these missing data points, we applied the method *forward fill*, which takes the last valid observation and copies it forward to fill in the missing values. Resampling the dataset in one-hour intervals implies that the future predictions will also be every hour. While exploring the dataset, we noticed that the occupancy rates are

[16] A DataFrame is the main multi-dimensional data structure in Pandas and one can think of it as a spreadsheet, which allows for further manipulations

not highly dynamic, so generating a new prediction in smaller intervals is unnecessary and would imply higher computational costs.

When resampling the data, the missing dates between the 16th of August and 4th of September 2020, were generated. However, as we applied the method *forward fill*, all these cells obtained the same value i.e., the last valid observation. As concluded during the data exploration, the truck parking data has both hourly and daily patterns, thus, a more sophisticated method for imputation is needed. Hence, we dropped the occupancy rate values for the missing period. To fill in the values, we created a custom function, which first calculates the average occupancy rates by day of the week and for each hour of the day and then maps these values onto the missing values in the occupancy rates column of the DataFrame. Developing this function allowed us to reflect on both the hourly and daily seasonality of the time series data. To visualize the results of the interpolation, we plotted the average daily occupancy rates in Figure 5-1.

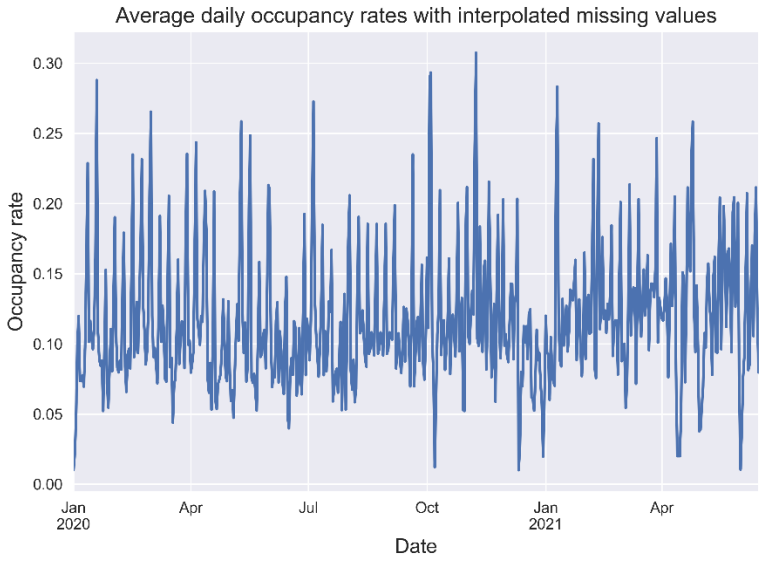


Figure 5-1 An overview of the average daily occupancy rates after interpolating the missing period of data

5.2.2 Weather data

The weather dataset requires considerably less preprocessing compared to the truck parking data. Firstly, we loaded the hourly temperature and rainfall data into a Pandas DataFrame. Then, we dropped the obsolete column *'#STN'*, denoting the number of the KNMI weather station of Deelen. Afterwards, we performed several operations to turn the columns with the date and the hour of the day into a single datetime column with the correct format. Then, we renamed the columns for more clarity as *'Timestamp'*, *'Temperature',* and *'Rainfall'*. Finally, we checked whether there were any missing values. The dataset was complete and did not need further processing.

5.3 Establishing the dataset

To create the final dataset, we started by importing the cleaned into two Pandas DataFrames, respectively. Then, we appended both datasets into a single DataFrame by using the *merge* method. Each dataset contained a timestamp column, which was used as a key to perform this operation. After merging the datasets, we engineered the temporal features i.e., we used the timestamp to determine the day of

the week, the hour of the day, and whether the day was on the weekend (Saturday or Sunday) or not. These were added as new columns to the DataFrame, named *'DOW'*, *'Hour',* and *'Is_weekend'*, respectively. The column denoting the day of the week contains an integer from 0 to 6, as 0 stands for Monday, 1 for Tuesday…, 6 for Sunday. The hours of the day are also denoted as integers, between 0 and 23. The column indicating whether the day is on the weekend takes a value of either 1 if the day is a Saturday or Sunday, or 0 if it is not.

Afterwards, we added the *holiday* features to the dataset. Based on Table 4-1, we created two DataFrames, containing the dates of the German and Dutch public holidays, respectively. Then, we turned the values from strings to datetime type, to be able to perform the necessary operations, and added two new columns to the dataset, *'Is_holiday_GE'* and *'Is_holiday_NL,* respectively. We iterated over the column containing the timestamp and assigned the value 1 if the date is contained in the DataFrames with the German and Dutch holidays, respectively. Otherwise, the assigned value is 0.

Finally, we would like to add a feature that provides our model with knowledge about occupancy in the recent past, as this approach was positively acknowledged in the literature. Hence, we created a full lookback window of eight hours. This was achieved by shifting the '*Occup_rate'* by one up to eight units and subsequently creating eight new columns: '*Hist_occup_1h'*, *'Hist_occup_2h',* etc. We choose to experiment with a long lookback window of up to eight hours, after analyzing Figure 4-4. As visible in the plot, the occupancy rates do not change a lot within a few hours, hence, providing the model with information about the occupancy from an hour or two ago alone might not provide enough insight into the historical occupation. We will provide a more in-depth analysis of this variable in section 5.4.

5.4 Feature selection

So far, we have spent a considerable amount of time engineering new features and adding them to our dataset. In this section, we will do the opposite and perform *feature selection*, which is the process of reducing the number of input variables when developing a predictive model. Feature selection is a crucial task because more attributes do not necessarily lead to a better-performing prediction model. As the number of input variables increases, the model increases dimensionality, which increases the computational cost of modelling. Furthermore, a reduced number of attributes can prevent the model from including noise and affect the predictions positively.

To choose an optimal set of attributes for our model, we will explore the measure *correlation*, a statistical term used to describe the likelihood of two variables having a linear relationship. The correlation number takes a value between 1 and -1, -1 being a perfect negative correlation and 1 being a perfect positive correlation. When two variables are positively correlated, one could say that they move in the same direction. For instance, if one of them increases, the other one also increases. On the contrary, two variables that are negatively correlated move in opposite directions from each other, such that if one of them increases, the other one decreases.

In the context of machine learning, a pairwise correlation for feature selection is about identifying groups of highly correlated variables. For example, if two variables have a correlation coefficient of 0.90, this means that 90% of the time, we can predict the values of feature 1 by just feature 2 and vice versa. Thus, identifying highly correlated features and removing one of them from the dataset allows us to increase the predictive power of our model while decreasing the computational time. To explore the correlations of the features we derived in section 5.3, we plotted a heatmap, visualized in Figure 5-2.

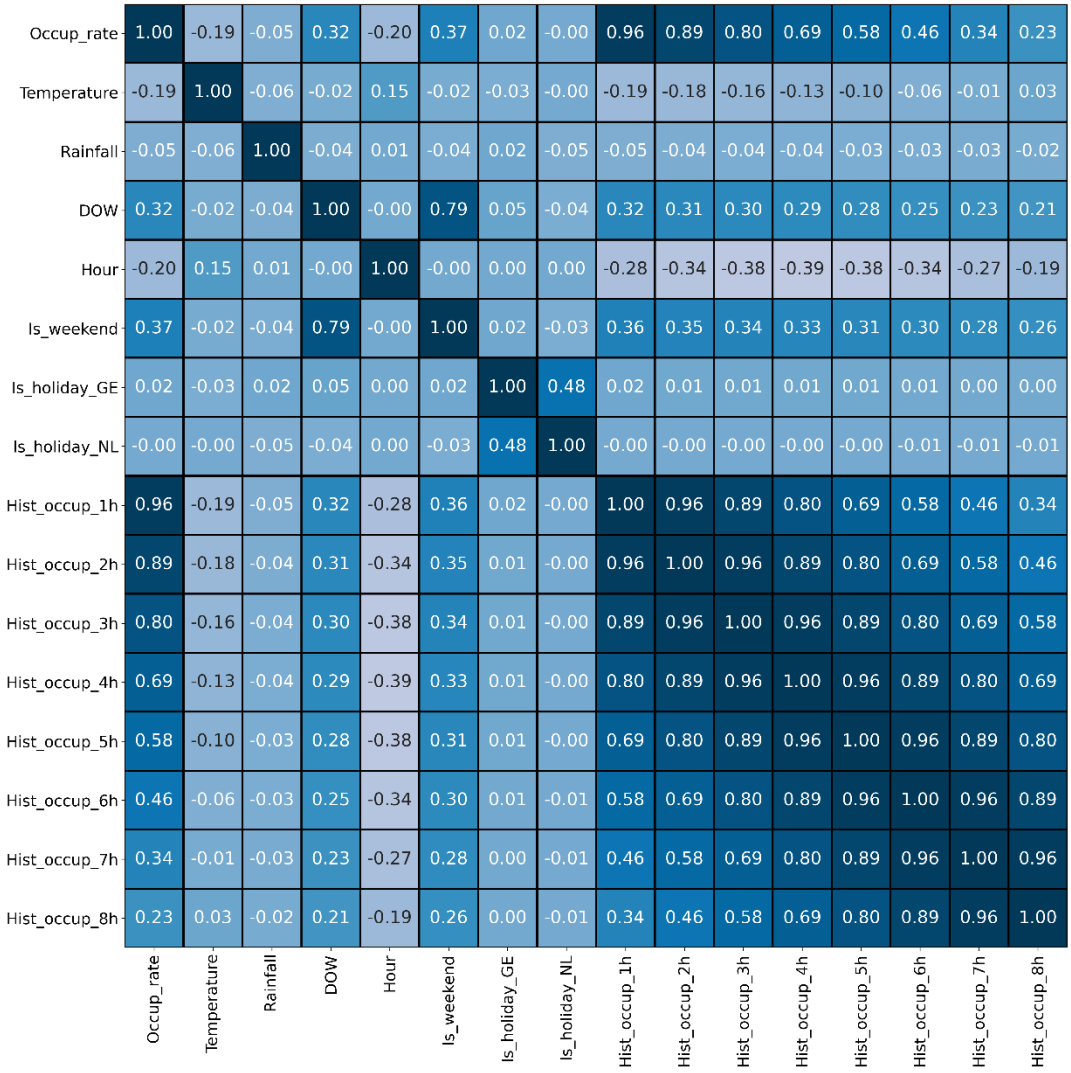


Figure 5-2 Correlation matrix indicating the strength of the relationship between each pair of variables

Taking a close look at the heatmap, we see that the historical occupancy features are closely correlated with each other. This means that the model does not obtain new information from having a full lookback window of eight hours and hence, we should delete some of these features. Based on the correlation coefficients between the historical occupancy variables, we decide to keep only the occupancy rates from the previous hour, from four hours ago, and from seven hours ago. We choose these because they have a high correlation with the dependent variable but are only moderately correlated with each other.

5.4 Conclusions

In this chapter, we prepared a dataset as input for predicting the truck parking occupancy rate. Firstly, we listed the choice of attributes that we derived from our literature study and from exploring the dataset. Secondly, we cleaned the truck parking and weather datasets, respectively, and merged them into a single DataFrame. Then, we derived the temporal attributes, such as weekday, the hour of the day, etc. Next, we added a full lookback window of eight hours to the model. We ended this chapter with a correlation-

based feature selection process and selected the optimal subset of features to be used as input of our prediction model. Table 5-2 shows a comprehensive list of the resulting variables.

Table 5-2 A list of all (independent and dependent) variables resulting from the data preparation task

Variable	**Format**	**Example: Timestamp 2020-01-08 12:00:00**
Occup_rate	float[17]	0.02
Temperature	integer	113
Rainfall	boolean[18]	0
DOW	integer	2
Hour	integer	12
Is_weekend	boolean	0
Is_holidays_GE	boolean	0
Is_holidays_NL	boolean	0
Hist_occup_1h	float	0.02
Hist_occup_4h	float	0.09
Hist_occup_7h	float	0.14

[17] A *float* is a data type composed of a number which is not an integer and includes a fraction represented in decimal format.

[18] A *boolean* is a data type with only two possible values, usually “true” and “false”.

Chapter 6 Modeling

Now that we have explored and prepared the data for modelling, we can start with building a prediction model. First, we will choose the appropriate modelling technique in section 6.1. Then, we will explain some machine learning fundamentals, which need to be followed during the training, validation, and testing of a machine learning model, and will generate an appropriate experimental design in section 6.2. Following this, in section 6.3 we will focus on the model development. We will experiment with different model configurations and following this, propose candidate models. Hence, this chapter focuses on RQ 9: *What are the characteristics of the prediction model(s)?*

6.1 The modelling technique

In section 3.2 of the literature review, we introduced the most prevalent machine learning techniques, and in section 3.4, we investigated which of them are often applied in the context of parking occupancy prediction. In this section, we will further explore which machine learning algorithm(s) have the potential to perform best on our dataset. For this purpose, we will utilize a data science software platform called *RapidMiner*. This tool enables us to compare multiple algorithms on the same dataset and consequently, compare them based on their performance and run time.

After uploading the dataset into RapidMiner and selecting the task (i.e., prediction), the tool can automatically propose suitable machine learning algorithms based on the characteristics of the input data and the desired task. Based on the input, RapidMiner recommended the following machine learning algorithms: *Generalized Linear Model*, *Deep Learning* (based on multi-layer feed-forward ANN), *Decision Tree*, *Random Forest*, *Gradient Boosted Trees,* and *Support Vector Machine*. It is worth mentioning that these machine learning techniques closely resemble the ones we described previously in our literature review. The Generalized Linear Model expands the general linear regression that we discussed in section 3.2. Deep Learning (a multi-layer feed-forward ANN), Decision Tree, and Support Vector Machine were also elaborated on in the same section. We did not specifically focus on Random Forest and Gradient Boosted Trees, as they are a variation of decision tree models, however, these techniques were also often mentioned in the literature about parking occupancy prediction. As during our literature study, we had acknowledged the relevance of all suggested algorithms, we decided to test the performance of all of them. Table 6-1 shows the results of the comparison.

Table 6-1 Results of comparing multiple machine learning algorithms

ML algorithm	RMSE	Standard deviation	Runtime
Generalized Linear Model	0.02	0.002	7 sec
Deep Learning (ANN)	0.018	0.001	30 sec
Decision Tree	0.02	0.001	4 sec
Random Forest	0.022	0.001	52 sec
Gradient Boosted Tree	0.018	0.002	1 min 3 sec
Support Vector Machine	0.02	0.002	29 min 18 sec

By analyzing the results from the experiment, we notice that no technique outperforms the rest on both criteria (error and runtime). If we consider both criteria, we see that the Decision Tree seems like a promising candidate, as it has the second-best error rate (0.02) and the lowest runtime. Deep Learning (ANN) and Gradient Boosted Trees outperform Decision Tree, when we take the error into account

(0.018), however, both models have longer computational times. As the differences in error are not significantly big (0.002), we conclude that the performance of the Decision Tree is satisfactory, and we choose to apply this modelling technique further in the research.

6.2 Experimental design

One disadvantage of decision trees is that these models tend to 'overfit' the data (Kuhn & Johnson, 2013) and this problem should be handled seriously while training the model. Hence, generating the best possible experimental design is a crucial task, which we will focus on in this section. Firstly, we will explain the concepts of overfitting and underfitting, and then, we will choose an experimental design that would best support us in tackling these problems.

6.2.1 Overfitting, underfitting, and the bias-variance trade-off

There are a few fundamental concepts in the field of supervised machine learning, namely *overfitting*, *underfitting,* and the *bias-variance trade-off*, which are important to address. In supervised learning, we assume that there is a mapping f between dependent and independent variables. This can generally be expressed as $y = f(x)$, where f is the unknown function that we want to determine. However, in real life, data is always accompanied by randomness and noise. Hence, our goal is to build a model $\hat{f}$ which best approximates f. When we train this model, we try to disregard noise as much as possible, so that in the end the model achieves a low prediction error on unseen data.

When approximating f, two difficulties of the difficulties that one can encounter are overfitting and underfitting. Overfitting occurs when models fit the noise[19] in the data too well. On the contrary, models that are not flexible enough suffer from underfitting. When a model overfits the training data, its predictive capabilities on unseen data are low. Such a model memorizes the noise present in the training set and subsequently, achieves a low training set error and a high test set error. On the contrary, models that underfit the data are not flexible enough to capture the dependencies between the dependent and independent variables and such models generally have a high error.

The *generalization error* of a model measures how accurately an algorithm can predict outcome values for previously unseen data. It can be decomposed into three terms: *bias*, *variance*, and *irreducible error*, where the irreducible error is the error contribution of noise. Bias is the difference between the actual value (f) and the average prediction of the model ($\hat{f}$). High bias causes a model to miss essential relations between the predictor variables and the target variable (underfitting). Variance describes how much the estimate of the target function ($\hat{f}$) alters when different training sets are used. When the variance is high, the model has learned the noise in the dataset too well and is perfectly capable of predicting the training data, however, it will then perform poorly on never seen before data because this data is different (overfitting).

The complexity of a model determines its flexibility to approximate the true function f, hence, the optimal model complexity corresponds to the lowest generalization error. When a model becomes more complex, the variance increases, while the bias decreases. Conversely, when the model complexity decreases, variance also decreases but the bias increases. The goal is to find the optimal model complexity, which therefore achieves the lowest generalization error. This error is the sum of three elements where the

[19] In statistics, noise is defined as unexplained variability within a data sample. It affects adversely the results of any data mining analysis.

irreducible error is constant and the variance and bias are negatively correlated, hence, we aim to find a balance between them, also known as the *bias-variance trade-off*.

6.2.2 Splitting the dataset

The first step towards developing a machine learning model and combatting overfitting is splitting the total dataset into multiple subsets. As we wish to evaluate the model's performance on unseen data, we divide the data into two parts, also referred to as the *train-test split*:

- The **training set** is the larger data subset, which is used during the learning process to build and optimize the model.
- The **testing set** is the smaller subset that is kept separately until the very end. The already trained model is then applied to the testing set, in pursuance of achieving a truly unbiased estimate of the model's performance. It gives a good indication of how the model might perform in a real-world setting.

When it comes to the proportions of the splits, there is no consensus in the literature. In general, guidelines suggest a split of 70-90% training set and 10-30% testing set. This is highly dependent on the size of the dataset and practitioners advise having a sufficiently big training set so that the model has enough data to learn from. Based on these recommendations, we tested the following splits: 70%/30%, 75%/25%, 80%/20%, 85%/15% and 90%/10%. The split yielding the best results on our data is *80%/20%,* so the total dataset was divided into 80% training data and 20% testing data. Note that for most machine learning models, this split is performed after the data is randomly shuffled. However, due to the sequential nature of the input data (i.e., time series), we must respect the temporal order in which the values were observed (Brownlee, 2019). We will discuss this topic in more detail in section 6.2.3.

6.2.3 Cross-validation

As explained in section 6.2.1, finding the optimal model complexity is a crucial task when it comes to training a machine learning model. The parameters that define the model architecture are referred to as hyperparameters. Following this, the process of searching for the best model architecture is called *hyperparameter tuning*. While exploring various model architectures, one also needs to evaluate the model's performance on unseen data. If the test set is used for this purpose and we "fit" the model to the testing data, there is still a risk of overfitting the model because we lose the ability to truly evaluate the performance on unseen data.

To solve this problem, a frequently used technique is *k-fold cross-validation*. The data is divided into k subsets (referred to as folds). Then, the model is trained on all subsets except one (k-1) and validated on the remaining part of the data. However, this technique assumes that the samples are independent and identically distributed, hence it randomly picks them out of the training set and splits them into training and validation sets. For this reason, in the case of time series data, k-fold cross-validation is not a robust method. Data gathered using a time-dependent process is characterized by a correlation between observations that are near in time (also referred to as *autocorrelation*). Thus, for proper evaluation of our model on time series data, it is very important to always test the performance on "future" observations.

To cross-validate time series data, sampled at fixed intervals, we can use a variation of k-fold called *time series split*. The concept behind time series splits is to divide the training set into two folds at each iteration, as the corresponding training set consists of observations that occurred *before* the observations that form the test set, hence, there are no future observations in constructing the predictions. Then, the

forecast accuracy is calculated by taking the average over all test sets. In the literature, this method is referred to as "evaluation on a rolling forecasting origin" (Tashman, 2000) since the "origin" used for the forecast rolls forward in time.

When it comes to the number of subsets used in cross-validation, there is no consensus in the literature. Depending on the data size, practitioners often choose *k=5* or *k=10*. Choosing the right value of *k* requires a bias-variance trade-off because higher values of *k* lead to less biased models but large variance that might result in overfitting, whereas low values of *k* might lead to underfitting if the amount of data is limited. On our dataset, we tested different values between 5 and 10. Since dividing the dataset into 5 iterations resulted in the lowest prediction errors, we choose *k=5* for our time series split cross-validation. Figure 6-1 provides an overview of how our dataset was partitioned for model development.

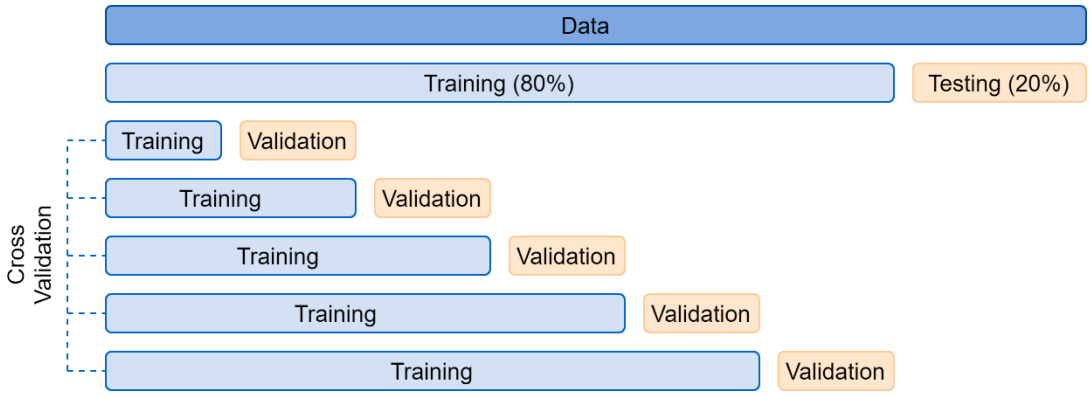


Figure 6-1 Partitioning of the dataset for model development

6.3 Model development

To build the model, we applied the aforementioned experimental design. To train, validate and test the model, the *Scikit-learn (Sklearn)* library was utilized, since it provides a wide selection of machine learning algorithms and tools for training and testing.

6.3.1 Basic model

Firstly, we loaded the prepared dataset and initialized the dependent and independent variables. Then, we split the data into training and testing subsets, as indicated in Figure 6-2, without performing a shuffle. We built a basic regression tree model, without specifying any of the parameters and using the default values, fit it on the training data, and then evaluated its performance on the testing set, to get a baseline idea of the performance. For evaluation, we chose to use RMSE, as visible in Figure 6-2.

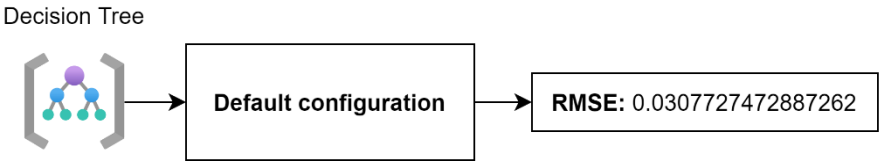


Figure 6-2 An overview of the basic decision tree performance on unseen data

Note that the RMSE is measured in the same units as the dependent variable, thus, the result is interpreted as follows: the average difference between the predicted and measured occupancy rates is 0.03. Considering that the occupancy rate can vary between 0 and 1, an average error of 0.03 is a satisfying

result. Surprisingly, the decision tree model was able to properly capture the dependencies between the dependent and independent variables, without overfitting the model too much.

6.3.2 Hyperparameter tuning

Next, to further optimize the performance of our prediction model, we will tune some of its parameters and find the best configuration. As the decision tree algorithm does not have a lot of parameters, it is relatively easy to optimize. The most important parameter to tune is *max_depth*, which indicates how deep the tree can be. As the tree grows deeper, it generates more splits and captures more information (which could lead to overfitting). To better understand the effect of the tree depth parameter on our model, we experimented with different tree depths and evaluated the model's performance on the training and testing set, respectively. For the values, we used a range of 1 to 25. Figure 6-3. shows the results, from which we conclude that when the tree depth grows over 9, the testing error begins to increase although the training error decreases continuously.

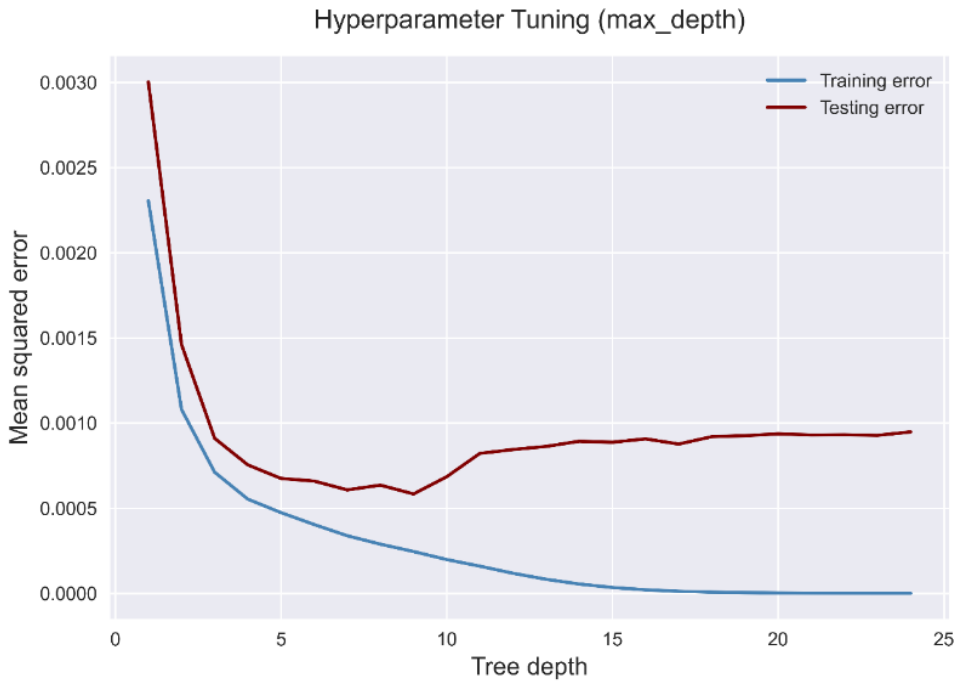


Figure 6-3 Single hyperparameter tuning (maximum tree depth)

The second parameter that we wish to optimize is called *min_samples_split*, representing the minimum number of samples required to split an internal node. The values for this parameter can vary from considering one sample to considering all samples at each node. As this number grows, the tree becomes more constrained because it has to consider more observations at each node. To see how different values of this parameter affect our model, we plotted the model's performance with different minimum samples split values as follows: 10, 20, 30, … 180, 190. The results are displayed in Figure 6-4, from which we can deduce that minimum samples split of 70 might be a good value for this parameter.

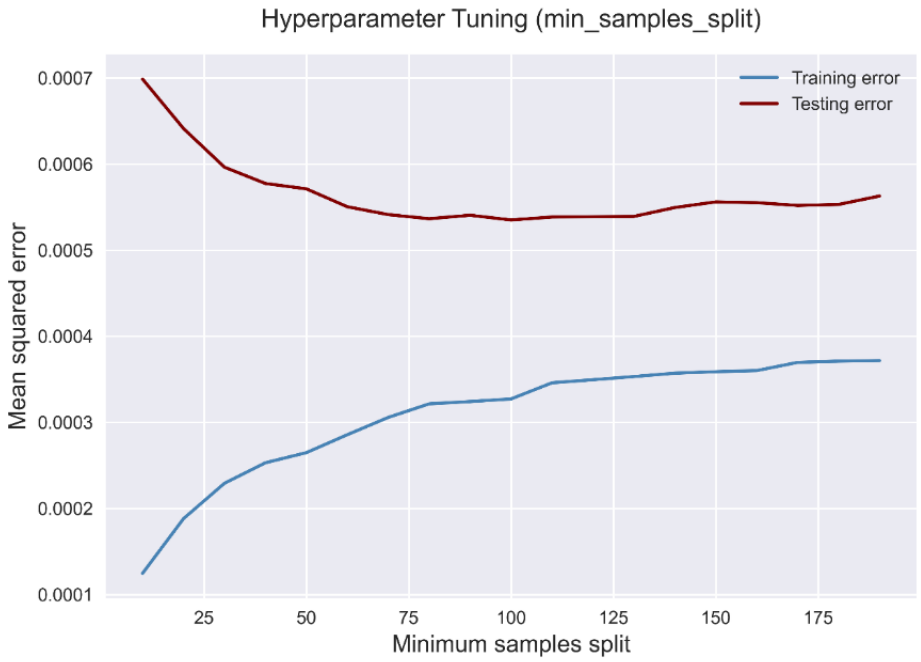


Figure 6-4 Single hyperparameter tuning (minimum samples split)

Note that so far, we have performed experiments considering one parameter at a time and used the train-test split only. Now that we have a better understanding of how the model performs, we will tune it by applying the explained in section 6.2.3 procedure *cross-validation*. This allows us to tune both parameters simultaneously, by using a **grid search** method along with the time series split cross-validation. Based on preceding experiments, we chose the following input values:

- max_depth: 3, 4, 5, 6, 7, 8, 9, 10
- min_samples_split: 40, 50, 60, 70, 80, 90, 100, 110

Figure 6-5 shows the hyperparameter space. To compare the resulting models, we used the *'neg_root_mean_squared_error'* scoring method, which is an equivalent of the RMSE. However, this explains why the values in the figure below are displayed with minuses. From the matrix, we conclude that the best combination of hyperparameters is as follows: *max_depth* = 6 and *min_samples_split = 80*. It results in the lowest average RMSE when the model is tested on the validation sets. Note that the optimal values from tuning both parameters simultaneously slightly differ from the results of our previous experiments.

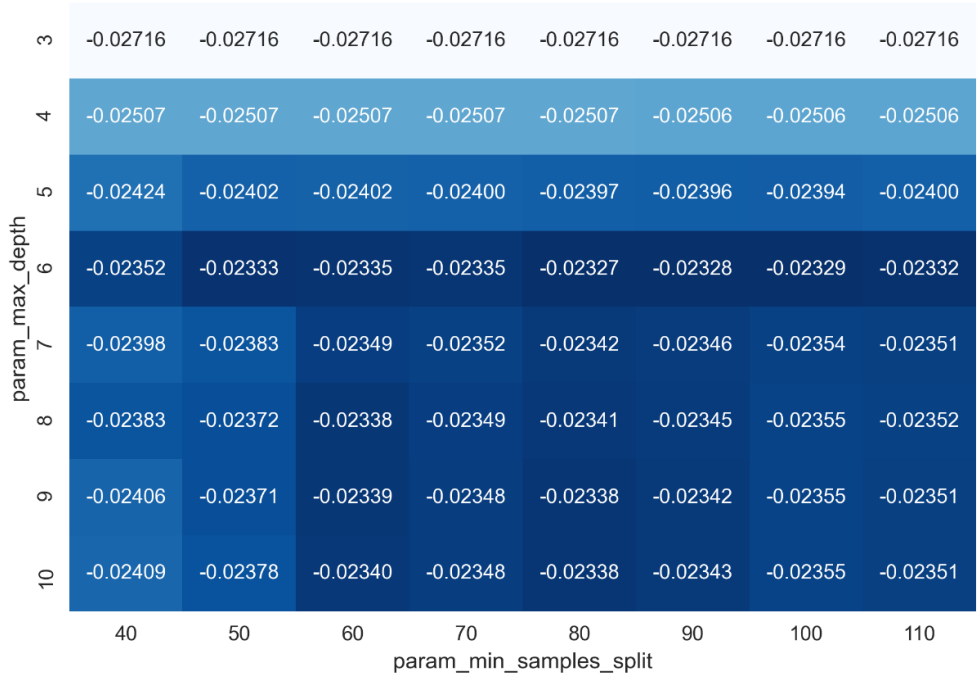


Figure 6-5 Hyperparameter tuning by applying time series split cross-validation and grid search

6.3.3 First candidate model

In this section, we will summarize what the resulting candidate model looks like. Based on the optimized parameters, derived in the previous section, the resulting decision tree consists of **a maximum depth of 6** and **minimum samples split of 80**. When the decision tree was trained with these parameters during the cross-validation phase, the associated root mean squared error yielded the lowest values. Moreover, the model includes all previously derived attributes. An overview of the resulting model with its configuration, as well as the respective validation score is shown in Figure 6-6. Note that, compared to the basic model that we built in section 6.3.1, the candidate model shows an improvement when comparing both models RMSE, which proofs the importance of applying hyperparameter tuning and finding the optimal model configuration. The values show that the model is capable of predicting the target variable quite well, with an average error of only 0.02327, considering that the occupancy rate ranges between 0 and 1.

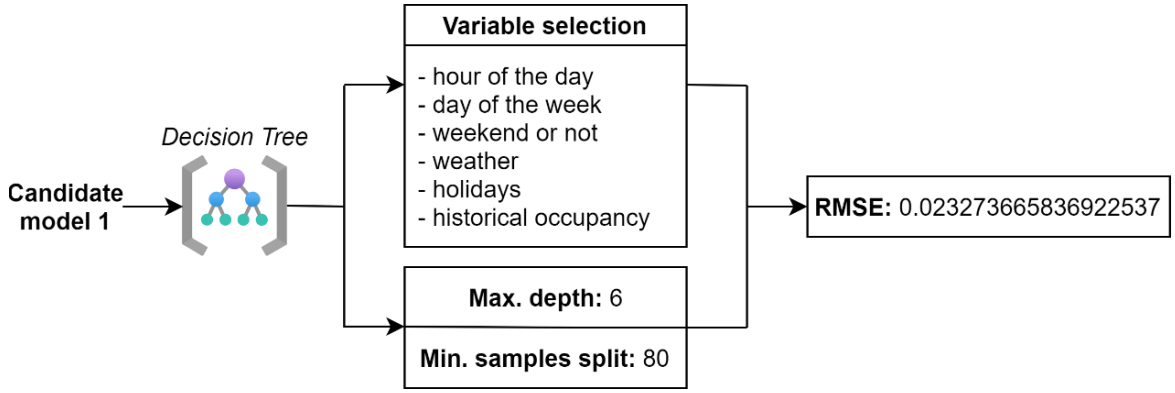


Figure 6-6 An overview of the resulting decision tree configuration (Candidate model 1)

6.3.4 Second candidate model

So far, we selected most of the model's predictors based on a literature study focused on car parking occupancy prediction (section 3.3) and developed and optimized a model including all pre-selected parameters, as listed in Figure 6-6. However, considering that we are forecasting truck parking occupancy, we have not yet investigated whether this configuration is optimal or there is another configuration to potentially develop a second candidate model from. The objective of this section is then to examine the model's performance by utilizing a *feature elimination* strategy. The experimental design is as follows:

1. We group the features into categories, based on their origins (for example, *rainfall* and *temperature* would be grouped under the same category i.e., *weather*)
2. We subsequently eliminate each category of variables or multiple categories of variables from the input dataset.
3. We train a new model with the remaining inputs. For each retrained model, we perform hyperparameter tuning, following the approach described in section 6.2.3, since the values of the parameters also affect the model's performance.
4. We evaluate the performance of the resulting models on the validation sets, by deriving the RMSE.
5. We compare the resulting RMSEs and use these values to determine the importance of each category of variables.

We decided to group the features into categories, based on what data sources they originate from, and defined them as follows:

- **Time** (*'DOW'*, *'Hour'*, *'Is_weekend'*)
- **Weather** (*'Temperature'*, *'Rainfall'*)
- **Holiday** (*'Is_holiday_GE', 'Is_holiday_NL'*)
- **Historical occupancy** ('*Hist_occup_1h'*,*'Hist_occup_4h'*, *'Hist_occup_7h'*)

The full list of experiments along with the results is available in Appendix A. We performed the experiment 11 times and discovered a new model architecture. The model includes the **time-dependent** variables i.e., the hour of the day, day of the week, and weekend or not, along with the **historical occupancy**. Surprisingly, the simpler model yields a slightly lower RMSE on the validation set compared to the first candidate model (approximately 0.0221 and 0.0233, respectively). Thus, this experiment suggests the hypothesis that when it comes to predicting truck parking occupancy, the meteorological conditions and holidays do not highly influence truck parking behaviour. Therefore, we propose the second candidate model, visualized in Figure 6-7.

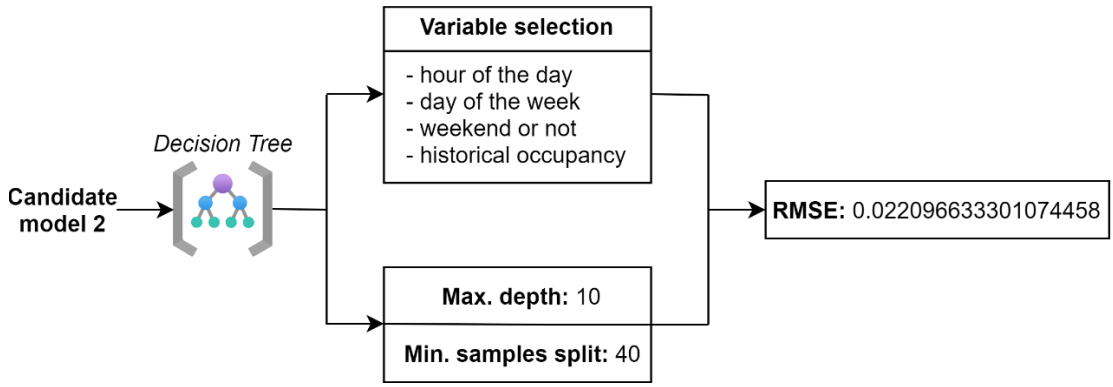


Figure 6-7 An overview of the resulting decision tree configuration (Candidate model 2)

6.4 Conclusions

In this chapter, we focused on the model development. We started with choosing the most appropriate machine learning technique, which we concluded to be a *decision tree*. Next, we treated the machine learning fundamentals *bias*, *variance,* and the *bias-variance trade-off* and emphasized the importance of generating an appropriate experimental design for the structural nature of our dataset (i.e., time series). Based on this discussion, we proposed an experimental design, which fits our data. The approach consists of splitting the data set into *training* and *testing* subsets, respectively, and then, using a *time series splits cross-validation* for hyperparameter tuning. Then, we proposed a first candidate model, including all pre-selected input variables (i.e., *hour of the day*, *day of the week*, *weekend or not*, *weather*, *holiday,* and *historical occupancy*), which generates RMSE of approximately 0.0233. Furthermore, we experimented with different variable configurations and proposed a second candidate model, which consists only of the time-dependent predictors (i.e., *hour of the day*, *day of the week*, *weekend or not*) and the *historical occupancy*. The simpler model yields a slightly lower RMSE of approximately 0.0221. The results from this chapter are of significant importance for the research since they offered new insights into the variables relevant for predicting truck parking occupancy, suggesting that some of the input variables used to predict car parking occupancy are not needed to predict truck parking occupancy.

Chapter 7 Evaluation

The next chapter will focus on the evaluation of the developed candidate models. We will start by comparing both models and selecting a model for deployment in section 7.1. Afterwards, we will further validate the performance of the chosen model in section 7.2. Thus, these sections will answer RQ 10: *What are the performance indicators of the proposed prediction model(s)?* Once the final model has been validated, section 7.3 will focus on one of the main objectives of the research, namely assessing whether the resulting approach is easily applicable towards other truck parking locations, and will answer RQ 11: *To what extent is the resulting model transferable towards other truck parking areas?*

7.1 Inter-model comparative testing

So far, we have discovered the parameters of both candidate models through cross-validation. After these have been established, we will assess the performance of both models and compare the results, in order to choose the optimal model for the predictive system. Thus, we will evaluate both models' performance on the testing set, which we have not used during the training phase. Keeping this set separately so far allows us to get an unbiased evaluation of the candidate models' fit on the training dataset.

To obtain these results, we fit both decision trees on the whole training set and derived the quantitative performance measures, visualized in Table 7-1. So far, we have only used the RMSE as a measure of performance, due to its interpretability since it is measured in the same units as the target variable. For the final evaluation of our models, we derived some additional metrics to help us get a better comprehension of how the models perform on unseen data. These metrics were previously discussed in section 3.5 of the literature study. Moreover, we used green and red colour-codings to denote which model performed better, by achieving lower errors or higher correlation coefficients, respectively.

Table 7-1 Model evaluation of both candidate models on the test set

Metric	Candidate model 1	Candidate model 2
RMSE	0.024068	0.022948
MAE	0.014649	0.013330
R^2	0.923590	0.930538
Adjusted R^2	0.923289	0.930375

Even though the differences are not large, the summarized overview of the metrics demonstrates that candidate model 2 outperforms candidate model 1 in every aspect. Candidate model 2 predicts with lower errors and has a higher correlation coefficient, meaning that it can be regarded as the stronger model configuration. Moreover, the model is simpler, and it requires fewer data sources since all input variables can be derived from a single data source (both the time-dependent features and the historical occupancy are derived from the parking occupancy dataset). Following this, the model requires less data preparation, maintenance, and thus, a generally lower cost of implementation, which is crucial for the ease of deployment of the model in practice. Considering all arguments at hand, we prove that the second decision tree configuration is the more suitable model to predict the occupancy rates of truck parking locations.

7.2 Validation of final model

Now that we have established that the **second candidate model** can be considered as the better model, we will further focus on the quality of its predictions by evaluating to what extent the model tends to over- or underfit. For validation, we will consider the results from Table 7-1, which were obtained by testing how the model performs on unseen data and thus, provide an unbiased assessment. After the validation procedure, we will assess whether or not the second candidate model is robust enough to be put in production.

First, focusing on *RMSE*, which measures the standard deviation of the residuals, we see that the error is a bit higher than during the training phase with values of approximately 0.0221 and 0.0229, respectively. This is expected since the model is tested on data that it has never seen before. However, the differences are quite small, which is a sign that the model is flexible enough to learn the dependencies in the data without overfitting. Next, *MAE* scores a value of approximately 0.0133. It measures the average of the residuals in the dataset and is interpreted as follows: on average, the forecast's distance from the actual value is 0.0133, which is a satisfying result.

Moving onto the next value, R^2 measures the strength of the relationship between the model and the target variable. A value of 0.9305 tells us the model's explanatory variables account for 93.05% of the variation in the occupancy rates. Secondly, the *adjusted* R^2 indicates the percentage of the variation explained only by the independent variables which *affect* the dependent variable. Hence, we strive towards a value that is the same as R^2 or a bit lower. Looking at the results, the adjusted R^2 is 0.9304, which is very close to R^2.

During the literature study (section 3.5), we concluded that R^2 is a measure of correlation and not of accuracy, hence, to properly assess whether the model is overfitting or underfitting, it is not enough to only compare these values. To better assess the model's performance, we will examine a visual representation of the data. Figure 7-1 shows the predicted versus the actual values, as each dot represents a value. In a model that underfits, the dots would be far from the regression line, whereas, in a model that overfits, the dots would be perfectly situated along the line. Inspecting the plot shows that neither of these occurrences is happening. From this, we can conclude that the model's performance is sufficient, and it does not overfit or underfit.

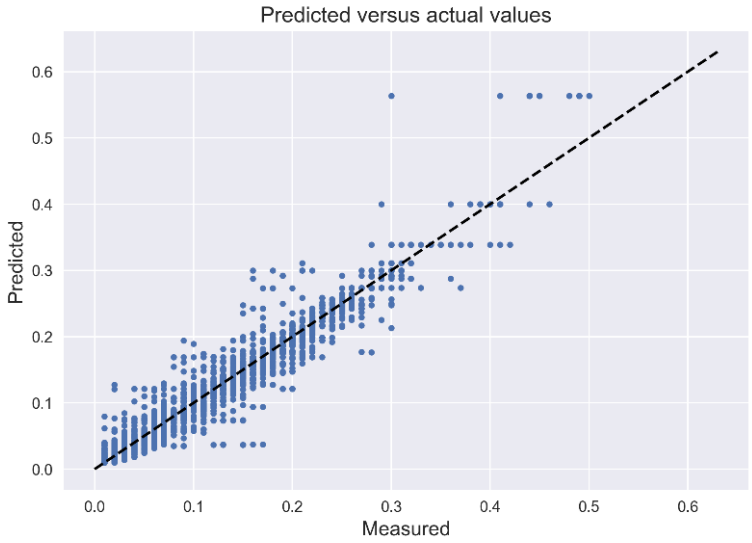


Figure 7-1 Scatterplot of predicted versus measured occupancy rates (Candidate model 2)

This concludes the validation procedure, from which becomes evident that the second candidate model is able to produce predictions of sufficient quality. After validating its performance, we can deduce the model strong enough and thus, can be further proposed for deployment.

7.3 Transferability of the system

So far, the study has explicitly focused on developing a model for the A1 Truck Parking Deventer. However, one of the main objectives of this research is to propose a model that could be adopted by other truck parking areas as well. Hence, evaluating the transferability of the model is an important topic, which we will address in the following section.

7.3.1 Impact of data volume

The first question that we will focus on considers the amount of training data and its impact on the predictive capabilities of the model, as this is a crucial factor for the transferability of the model. Currently, the number of truck parking areas that collect extensive historical datasets is limited. Nevertheless, an opportunity would be to dynamically collect historical data and gradually build training datasets, which could be exploited to develop models for more truck parking areas in the future.

Ideally, machine learning models need a considerable amount of data to learn from. For example, we utilized 1.5 years of data to build and train our prediction model. Thus, collecting an appropriate amount of data can be a challenging and time-consuming task. We recognize these time and resource constraints, and that they threaten the transferability of the model towards other truck parking areas, which is one of the main objectives of this research. To address these concerns, it is then crucial to assess how much data is needed to train and validate a model, without compromising the quality of its predictions.

We decided to experiment by recursively dividing the total dataset into halves and training a new prediction model from the resulting subset. For the training and testing procedure, we used the configuration of the second candidate model, which we concluded to be the deployment model. When dividing the datasets, we respected the natural order of the time series, thus after each iteration, the

oldest part of the dataset was deleted, whereas we kept the most recent half as input for the next iteration. The full dataset contains 12751 entries, hence, in the second experiment, we used 6376 observations, in the third one 3118, et cetera. The final dataset consisted of 0.2% of the total dataset, which is 25 observations, or approximately one day of gathered data. To assess the resulting models, we used the RMSE, measured on the test set. Considering that our dataset contains hourly observations (i.e., 24 per day), we can determine the length of period for data collection by diving the number of observations by 24. To visualize the results, we plotted a line graph in Figure 7-2.

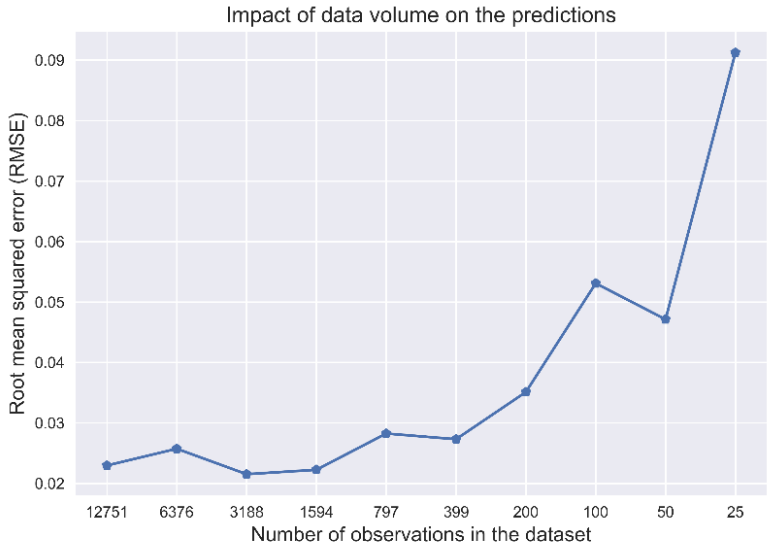


Figure 7-2 Plot of RMSE against the number of observations in the dataset

As expected, the plot shows that the RMSE increases when the number of observations in the dataset decreases. In the last experiment, when the training set has been halved 9 times, the RMSE is around 0.091, which is an increase of 295.7 % from the initial 0.023. However, surprisingly the RMSE does not drastically change up until the fifth halving of the dataset, corresponding to 0.027. While the volume of the data is decreased by 96.9%, the RMSE only increases by 18.9%. In that case, 399 observations correspond to approximately 16 days of collected data. From that, 80% are used for training, or approximately 13 days, and the rest is used for testing the performance. As the time series show seasonal patterns, both hourly and daily, it is reasonable that the model needs about 2 weeks of training data to learn these dependencies.

All in all, the experiment demonstrates that a considerable amount of data can be compromised while the level of performance is kept satisfactory. Only 16 days of data are needed for a model that performs slightly worse than a model trained with 1.5 years of data. This is a promising finding of the scalability and transferability of the system towards other truck parking areas, and especially regarding the ease of implementation for truck parking locations, which are currently not able to provide historical data. Nevertheless, it should not be overlooked, that better results are obtained when the model is trained with a higher number of observations. This is in line with the general understanding that machine learning models need larger volumes of data to generate more accurate predictions and the general goal should be to collect as much data as possible, considering the time and resources at hand. However, it should be

considered whether the data from the past is stationary or some new trends and seasonality are observed. Thus, to generate predictions with high accuracy, one should strive towards collecting as much data as possible which is still reflective of the current occupancy rates.

7.3.2 Variable importance

The second bottleneck, which we will address in the following section, concerns the availability of data sources, which proved as an obstacle of this research. Note, that the model we choose for deployment consists only of *time-dependent variables* (hour of the day, day of the week, and weekend or not) and *historical occupancy*, both of which can be derived from a truck parking data source. Furthermore, the time-dependent variables could be considered as independent of any data source, as they could also be derived from the system's clock. Thus, the availability of these variables is guaranteed and cannot compromise the transferability of the model to other locations.

On the contrary, adding the historical occupancy to the model requires the model to be fed with data regularly (i.e., every hour). Currently, most truck parking locations in the Netherlands do not provide such data streams, which imposes a limitation on the transferability of the system towards other parking areas. Thus, an understanding of the influence of this variable on the model's predictions will provide a better indication of the model's flexibility. To examine the importance of the historical occupancy, we will first test how the model performs when each of the lookback windows is excluded from the input data set. Consequently, we will exclude the full lookback window and evaluate the model's performance again. For the experiment, we will use the final model which was chosen for deployment, and we will follow the same experimental design (described in section 6.2), meaning that we will first tune the parameters of the resulting models, train them on the training set and finally, evaluate the performance on the testing set. Detailed results of the experiment are available in Appendix B.

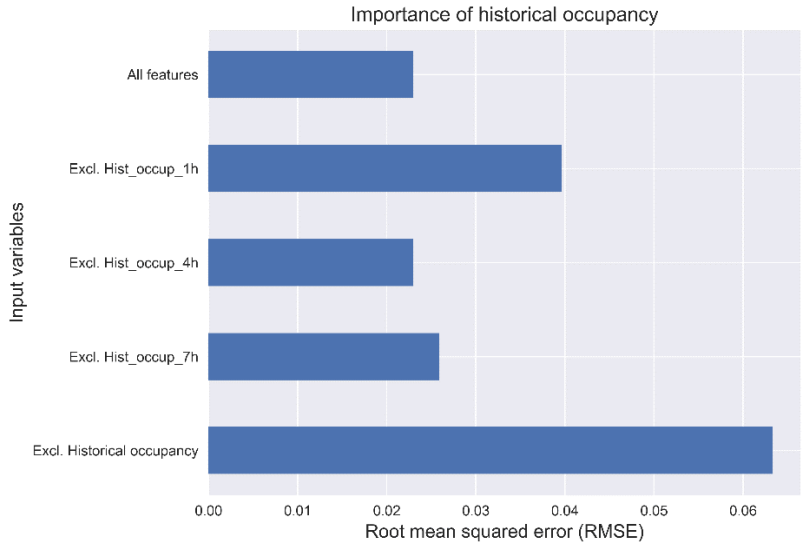


Figure 7-3 The impact of a lookback window on the quality of predictions

To visualize the results, we used a horizontal bar chart (Figure 7-3). From the experiment, the significance of the historical occupancy immediately stands out. When the model is trained without providing it any information about the previous occupation, the RMSE increases to approximately 0.063, which is an increase of 175% compared to the initial RMSE of 0.023. The experiment indicates that the model is highly

driven by its knowledge about the occupation of the previous hours. As a result, feeding the model every hour with new data about the occupancy rates is desirable for the model's predictive accuracy.

Furthermore, when we observe the individual lookback windows (i.e., the occupation from the previous hour, four hours ago, and seven hours ago), we notice that the model is mostly driven by its knowledge about the past hour and the absence of this variable alone causes the RMSE to almost double. Surprisingly, the occupation from 7 hours ago provides the models with more valuable information compared to the occupation from 4 hours ago. The plot shows that when we eliminate the occupation from 4 hours ago, the model's accuracy is only slightly affected, whereas the elimination of the occupation from 7 hours ago, causes the RMSE to increase by about 11%.

Overall, this experiment proves the significance of having a data source, which provides the model with information about the most recent occupation. Considering that such data streams of truck parking occupation are currently limited and developing a prediction system of this kind would cause a higher cost of implementation, it should be noted that even though the model performs worse, an RMSE of 0.063 is still an acceptable error, considering that the occupation varies between 0 and 1.

Finally, we would like to note some advantages of this model compared to a model that uses the latest occupancy as input. The latter provides higher accuracy of the predictions, as demonstrated in this section, however, it is only able to predict the occupation of the next hour. On the contrary, a model trained with time-dependent features only can provide an insight into the short-term occupation of the parking lots, for example, for a whole week. Although these predictions are less accurate, they can provide valuable information to truck drivers and parking infrastructure owners by informing them about the expected occupation of the parking lot at the time of their arrival. We will further discuss this model in section 8.2.

7.4 Conclusions

The results obtained from this chapter are highly valuable both for the practical implementation of the model and for filling in the identified knowledge gaps in the literature about truck parking occupancy prediction. Firstly, by comparing the proposed candidate models and selecting the second one, we concluded that a relatively simpler model configuration is needed to predict truck parking occupation compared to car parking occupation. The findings show that the model does not suffer from excluding the *weather* and *holidays* variables. On the contrary, the performance slightly improves. Following this observation, we conclude that for predicting the occupation of truck parking lots, using only time-dependent features and the previous occupancy is sufficient. This model is not only slightly more accurate but also easier to implement in practice since all features can be derived from the same data source. From the perspective of stakeholders, that is a valuable finding because using a single source significantly decreases the costs of implementation.

Secondly, the results about the extent to which the same approach applies to other truck parking lots also seem promising. Based on experimentation, we proved that the system can be scaled fast towards other truck parking locations, as sufficient results are obtained from training a model with 16 days of data. When it comes to the importance of supplying the model with information about the latest occupancy, the added value of this variable was emphasized. The performance decreased by 175% when the variable was excluded from the model, however, the resulting RMSE of 0.063 is still acceptable and indicates that the model can learn the main dependencies in the data set based on time features only.

Chapter 8 Deployment

The last phase of the CRISP-DM cycle concerns the *deployment* of the machine learning model. This is the process of taking a trained machine learning model and integrating it into an environment, such that its predictions are available to the intended end-users. Based on this, the following chapter will address RQ 12: *How can the outputs from the model be communicated to the relevant stakeholders?* To answer the question, in section 8.1 we will propose a conceptual model for the integration of a system that predicts the truck parking occupation of the next hour. In section 8.2, we will show how the system can be further complemented by the integration of a second model, which depends on time-dependent features only. Finally, in section 8.3 we will discuss how to proceed with the implementation of the system, by specifying the main steps and responsible parties.

8.1 Conceptual design of an integrated predictive system

The main objective of this section is to propose a concept for the implementation of the machine learning model into a comprehensive system, which can continuously predict the occupancy rates of truck parking locations. While designing the system architecture, the main elements of the chain must be considered i.e., gathering and preprocessing the data, generating predictions, evaluating the system's performance, and communicating the outputs to the main intended end-users i.e., truck drivers. Figure 8-1 shows an overview of the proposed system design.

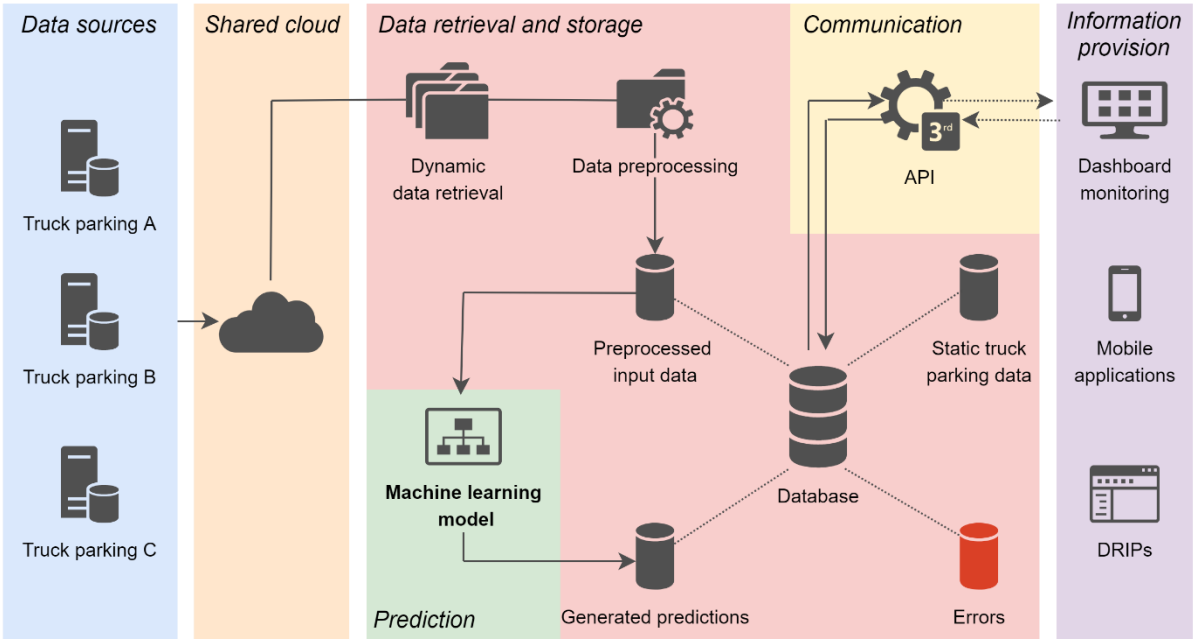


Figure 8-1 Conceptual design of the resulting predictive system

The key component of the system is the machine learning model, which requires to be fed with data to generate predictions. So far, data collection has proved to be a challenge for this research because truck parking locations do not continuously provide data about their occupation. To solve this, we propose implementing a *shared cloud*, that is fed with data from each truck parking location. For computing the occupancy rates of the A1 Truck Parking Deventer at a given point in time, we utilized the timestamps of the incoming and outgoing vehicles, thus the proposed system assumes the electronic toll collection data as the main source of data for the prediction model. The model generates a new prediction every hour

and, thus, we propose that the data from the electronic toll systems are pushed to the cloud on an hourly basis as well. Since truck parking locations provide the data, parking infrastructure owners are the envisioned *data owner*. As such, they make decisions regarding who has the right to access the data and regarding its usage.

When it comes to deploying an information system including a cloud service, some points must be taken into account, such as cloud administration and security. In that sense, *Identity and Access Management (IAM)*[20] plays an important role. It helps to implement adequate security policies across all systems, platforms, applications, and devices. This makes it easier to detect security violations, revoke access when necessary, and eliminate inappropriate access privileges. The IAM systems have three key tasks, namely *identification*, *authentication*, and *authorization*. This ensures that only the right persons will have access to the cloud and can perform certain tasks. The proper organization of the administration and protection of the cloud is essential because the machine learning model is trained based on data retrieved from the cloud. Thus, a cyberattack, for example, where the truck parking data is altered or deleted, will seriously threaten the overall functionality and accuracy of the information system. Finally, due to the high importance of historical parking data for the model, it is advisable to periodically create back-ups, such that in case of a data breach, the data can be easily restored.

The next system elements, which we will address, are *data retrieval* and *preprocessing*, as these are crucial for the machine learning model. The system should dynamically retrieve the most recent information from the cloud on an hourly basis and consequently, preprocess it. In the preprocessing step, the occupation is first computed based on the incoming and outgoing vehicles, and the time-dependent and historical occupancy features are generated. Finally, the data is stored in a *database of input data*. This database is a part of the main database of the system. Like the shared cloud, the security of the system's database is also a priority and should be taken into account.

Following this, the *machine learning model* retrieves the most recent input data and generates a prediction for the occupancy of the next hour. A new prediction is generated every hour, thus, this is a recurrent process on an hourly basis. All predictions are stored in a *database of generated predictions*, which is also a part of the main database of the system. Including databases that store all input data and predictions is essential for the evaluation of the system.

Every new batch of input data contains the occupancy of the previous hour, as this is one of the variables used by the machine learning model. Comparing this value with the generated prediction of the previous hour allows the system to simultaneously store all *prediction errors*. In this process, both databases with the input data and generated predictions are utilized and the errors are stored in a separate database, which is also a part of the main system's database. Finally, by analyzing these errors the system's performance can be continuously assessed. Setting benchmark performance metrics can indicate how well the model predicts and if the performance is better or worse than usual. Thus, the errors can be used as a measure of reliability, and based on their magnitude, one can decide when the system needs further

[20] In enterprise IT, IAM refers to the process of establishing and managing the roles and privileges of individual network entities (users and devices) in relation to a number of cloud and on-premises applications. Examples of users include customers, partners, and employees, while devices refer to computers, smartphones, routers, servers, controllers and sensors.

maintenance, such as retraining the model or deleting some of the historical data, which might have become obsolete.

So far, we have introduced three parts of the system's database i.e., the input data, predictions, and predictions errors. As visible in Figure 8-1, the database contains another part, namely the *static truck parking data*. This database contains information about each truck parking area, such as the location, total number of parking spaces, services et cetera, such that these can be easily queried by third parties when the model is implemented into an application or a website, for example. To illustrate which data is collected and stored in the resulting database, we created an Entity Relationship Diagram, displayed in Figure 8-2.

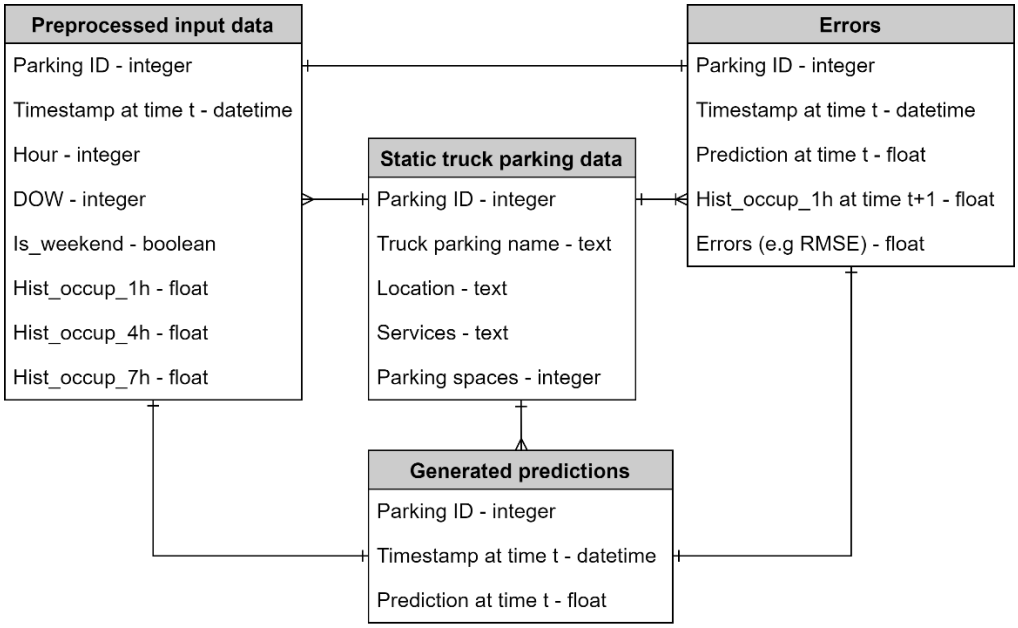


Figure 8-2 Entity Relationship Diagram depicting the relationships of the entity sets stored in the database

Finally, the system contains an *application programming interface (API)*, which enables the functionality of the system to be linked to third parties. The API acts as an intermediary layer that processes the data transfer between systems and thus, belongs to the communication component of the system. This will enable the information about the occupancy to be provided to the relevant stakeholders. Communication options include mobile applications, Dynamic Route Information Panels (DRIPs), and dashboard monitoring.

8.2 Deploying a short-term forecast

At the end of section 7.3.2, we discussed the advantages of implementing a model, trained with time-dependent features only. Such a model can be used to deploy a short-term forecast, for example, the expected occupation for every hour of the next week. Providing truck drivers with information about the expected occupation of the parking lot in the near future will allow them to make better-informed decisions when planning their trips. The second model could be implemented into the proposed system architecture from section 8.1. However, some adjustments to the machine learning pipeline[21] are needed

[21] A machine learning pipeline is a method for automating the workflow required to create a machine learning model. It consists of multiple sequential steps that handle all processes from data extraction to deployment.

to ensure that the system can support both models in production. The main modifications concern expanding the system's database, so that the inputs, generated predictions, and errors of both models are stored separately.

Next, we propose a clear and insightful way to present the generated forecast to the relevant stakeholders. For this, we created a prototype of a dashboard, which allows for viewing multiple metrics simultaneously. To build the prototype, we generated predictions for a week by training a model consisting of time variables only (hour of the day, day of the week, and weekend or not). Then, to make the dashboard more interactive, we added filters, which allow the user to specify the location of the truck parking area and the day and time they are interested in. Finally, we implemented the forecast into four visualizations. An overview of the resulting prototype is visible in Figure 8-3.

First of all, the dashboard includes a *gauge chart* (in the top right), which gives an instant overview of the expected occupancy rate of the chosen parking location for the specified day and time. Secondly, to visualize the expected occupancy rates over time, we developed a *bar chart* (in the middle), which shows the expected occupation for every hour of the day specified by the user. Then, we developed a type of *heat map* (in the bottom left), which shows an overview of the expected occupation for the whole week by the hour. The varying size of the squares resembles the variability of the expected occupancy rates. Finally, the dashboard includes another *bar chart* (in the bottom right), which displays the expected averages of the occupancy rates for each day of the week.

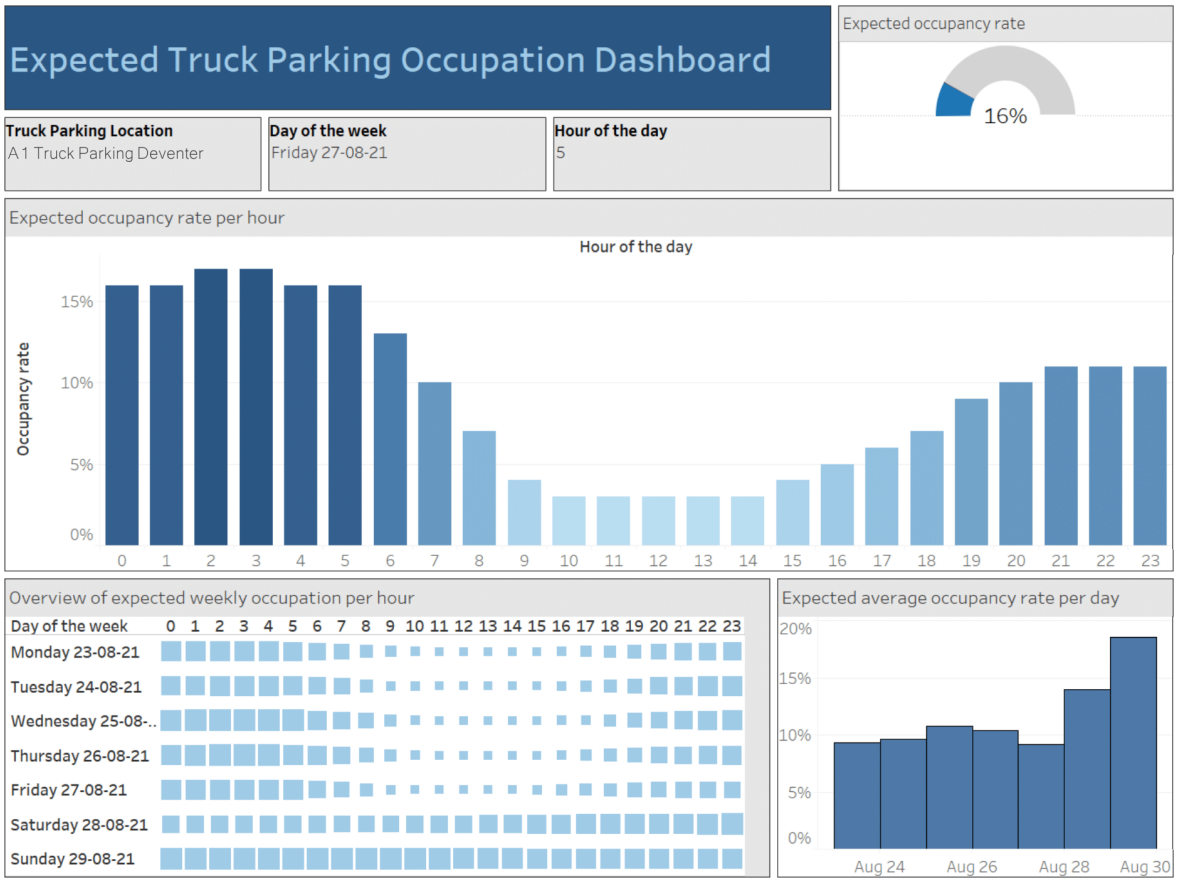


Figure 8-3 Prototype of an interactive dashboard about the expected occupation of truck parking lots

The proposed dashboard design provides valuable insight into the expected parking lot occupation not only to truck drivers but also to parking infrastructure owners. The design was validated with representatives from the A1 Truck Parking Deventer, which regarded it as useful for their planning and analysis of the occupation.

8.3 Implementation plan

Now that we have explained the concept of the prediction system, we will proceed with a plan of implementation, consisting of the most important steps that need to be undertaken in the future to deploy the system in practice. We will start by determining who the envisioned *system owner* is. This is an important first step, as the system owner will also be the main party responsible for the management of the project, including all operations and maintenance. The two possibilities we consider are a *private company* and a *governmental organization*. Considering the situation, we believe that a governmental body, for example, the province of Overijssel itself, is the more suitable option. The reasons for this are as follow:

- The government, represented by the province of Overijssel, is the main problem owner and responsible party for ensuring traffic safety and mitigating other problems resulting from illegal truck parking.
- The province of Overijssel is more likely to apply and be awarded a grant and be sponsored by the government for the deployment and maintenance of the system. Thus, it is more likely that the resulting service will be free of charge or with a low service fee. If the system is owned by a company from the private sector, this company might prioritize its interests and insist on high fees from the users. This might not be economically sustainable, as truck drivers from low-income countries might not be able to afford the service. Thus, the long-term existence of the system is threatened.

After specifying the project owner, we will continue with proposing an implementation plan. The steps are as follow:

Step 1: Continue working on public-private partnerships with parking infrastructure owners
For the successful implementation of the project, truck parking locations must install the corresponding technology which allows them to collect, store and share data. Thus, the province of Overijssel and Rijkswaterstaat must continue engaging in discussions and build mutually beneficial relationships between the public and private sectors.

Step 2: Install the appropriate technology at truck parking locations
The proposed system calculates the occupancy rates by utilizing data generated by the electronic toll systems at the entrance and exit of parking lots i.e., the timestamps of the events. This data is crucial for the functionality of the system, and thus, all included truck parking lots must be able to collect and share it automatically. This requires the installation of appropriate technology. For the organization and execution of this task, the responsibility is shared between the project owner i.e., the province, and parking infrastructure owners. The executions of the task might also involve third companies, specialized in data acquisition and control systems.

Step 3: Select a cloud service provider and plan the implementation of the shared cloud
This step relates to developing the proposed shared cloud, which allows all data from the separate truck parking lots to be stored at the same place. The province must select a cloud service provider and together

both parties must produce an efficient strategy for the implementation of the platform, as well as its security.

Note: Steps 2 and 3 should be performed simultaneously to ensure that the data from all truck parking locations can be automatically uploaded to the cloud and there are no technological inconsistencies. Thus, proper communication between the teams is essential.

Step 4: Implement the shared cloud
Next, the shared cloud must be implemented based on the determined strategy. The cloud service providers are responsible for the execution of this task.

Step 5: Select a team of machine learning engineers and plan the development of the main part of the system
Once the shared cloud is functioning, the development of the main part of the system can begin. For this, the province must select a team of machine learning engineers, whose responsibility will be the design of a self-running software for the automation of the predictive model. Then, the province, together with the machine learning engineers, must produce a plan for the development of the system.

Step 6: Develop the machine learning pipelines
Next, the machine learning engineers must convert the proposed conceptual model into a functioning system. Thus, based on the developed strategy, the team should fine-tune the proposed model and put it in production, by taking care of all tasks from data retrieval to deployment of the model. Future responsibilities of the team include the maintenance of the machine learning model.

Step 7: Select a team of mobile developers and/or web developers and plan the development of the mobile application and/or website
After the predictive part of the system is functioning, the predictions must be communicated to truck drivers. We propose incorporating the model into a mobile app and/or a website. For this, the province must select a team to develop the mobile application and/or website respectively. Together, all parties should select a development plan. Furthermore, they should also agree on a user-friendly design.

Step 8: Develop the mobile application and/or website
Based on the predefined plan, the mobile developers and/or web developers should produce a mobile app or a website, which informs truck drivers about the expected truck parking occupation. Furthermore, they will also be responsible for future maintenance and regular updates of the product(s).

Step 9: Display the predictions on DRIPs
To display the generated predictions on DRIPs, Rijkswaterstaat is the party responsible for the organization and execution of this task.

Step 10: System launch and marketing
After all aforementioned steps have been executed, the information provision system can finally be launched. The province is responsible for this task, along with organizing marketing campaigns to inform truck drivers about the newly functioning system, such that it reaches the end-users it is intended for.

Overall, the proposed plan consists of the main steps, required to implement the proposed information system in practice. The sequence of most steps is not rigid and the output from completing one step affects the input of the following step, however, shifting back and forth between the steps might often be

necessary. Thus, proper communication between all involved teams should be established and maintained.

8.4 Conclusions

In this chapter, we proposed a conceptual model for the development of an information system, which predicts the truck parking occupation for the next hour. The proposed architecture includes all main components i.e., data gathering, preprocessing and storage, prediction, evaluation, and communication. Furthermore, we suggested implementing a second model into the system, which generates an additional short-term forecast of the expected occupancy for a week ahead. This model forecasts a longer period in the future, and thus, its predictions are not as accurate, however, they could provide valuable information to truck drivers and aim them when planning their stops. Then, we developed a prototype and showed how the short-term predictions could be integrated into a single interactive dashboard. Finally, we produced an implementation plan, including the main future steps that need to be undertaken and actors responsible for them, such that the proposed information system could be implemented in practice. Furthermore, we distinguished between the *data owner* and the *system owner*. In our vision, the data is owned by parking infrastructure owners, as they are the party responsible for its collection and sharing. When it comes to who owns the system, we believe that the most adequate choice is a governmental organization, such as the province of Overijssel itself. As such, the province has the responsibility to organize the development and maintenance of the system.

Chapter 9 Discussion and conclusions

This research was realized as a part of the province of Overijssel's multi-year program, aimed at improving the issues emerging from truck parking in the region. The main objective was to determine how a system providing information about the availability of truck parking lots situated in the province of Overijssel can be developed, by employing a machine learning approach. Truck drivers are the main intended end-user, and it is hoped that the implementation of the system will significantly alleviate the associated stress from searching for a parking spot by providing them with information about the expected occupation of the parking lot at the time of their arrival. Nevertheless, an in-depth stakeholder analysis showed that the implementation of the system is beneficial not only to truck drivers but to a wide range of stakeholders, including road haulage companies, goods owners, business-park managers, governmental bodies, road users, and others.

Due to a lack of consistent research into forecasting the occupancy of truck parking areas, we used literature about car parking occupancy prediction as inspiration. Some of the research questions could be answered through the literature review, however, the main questions concerning building the model remained unclear. The literature research did not provide a comprehensive overview of which input variables are relevant for truck parking occupancy prediction and which type of algorithms have the potential to produce the most prominent predictions.

Next to the literature review, to determine the most significant predictors for the model we explored the given truck parking dataset, containing the timestamps of each ingoing and outgoing vehicle. We utilized these to calculate the occupancy rates of the parking lot on an hourly basis and explored the results, which provided some valuable insights about the factors affecting truck parking occupancy. The literature review and data exploration revealed eight candidates for predictors, namely: the hour of the day, day of the week, weekend or not, temperature, rainfall, previous occupancy, Dutch national holidays, and German national holidays. Based on a series of experiments, we concluded that a combination between **time-dependent variables** (the hour of the day, day of the week, and weekend or not), and the **previous occupation** (from an hour, four hours, and seven hours ago) produce the highest accuracy.

Regarding the machine learning algorithms, we used RapidMiner, a data science software platform, to test six different types. Based on the results, we selected the **decision tree** algorithm. In addition, we used cross-validation and determined that a decision tree with a maximum depth of 10 and minimum samples split of 40 is the optimal configuration for the proposed model. Following this, the resulting model predicts the next-hour truck parking occupation with RMSE of 0.0229 This result is interpreted in the following way: On average, the model predicts with an error of 0.0229. This means that if, for example, the real occupancy rate is 86%, most of the time the model will predict a value between 84% and 88%.

Overall, the results indicate some differences between predicting car parking and truck parking occupation. While the literature about car parking prediction suggests adding more input variables, for example, holidays and weather, this study shows that adding these variables to the truck parking prediction model is not necessary and that they even worsen the model's forecast. This indicates that while car parking behaviour is affected by weather conditions and national holidays, these factors are less significant in the context of freight transportation and do not influence truck parking behaviour.

When it comes to the machine learning algorithms, the findings demonstrate some correlation between predicting truck and car parking occupancy, as the decision tree model was a frequently chosen algorithm.

Nevertheless, literature about car parking occupancy prediction often suggests using more complex models, such as ensemble models or neural networks, since they produce more accurate predictions. RapidMiner's results also suggested that the gradient boosted trees and artificial neural network models generated lower errors, however, the differences were not very significant. A reason to explain this might be that truck parking occupation does not vary as much as car parking occupation, and thus, the model can learn the patterns more easily.

Overall, the research provides new academically relevant insights into parking occupancy prediction using machine learning, as it specifically focuses on truck parking lots. It suggests a relatively simpler configuration, compared to car parking availability prediction. The simplicity of the model has valuable implications as it makes the model easier to integrate in practice. Deploying the information system is a complex task. It involves installing the corresponding technology at truck parking locations, storing, and securing the parking data on a cloud platform, developing software of a self-running prediction model, and communicating the predictions through different platforms, like mobile applications or displaying them on DRIPs. The project includes the continuous involvement of teams with different professional expertise, such that all systems are well-integrated together and continuously maintained. These tasks are costly: they require large amounts of money and manpower. Hence, a system that utilizes fewer data sources and a simpler machine learning algorithm will generally require fewer resources, and thus, its implementation will incur lower costs. This is an indication of the significance of this research, not only for the scientific community but also in practice.

Due to data limitations, we were unable to validate whether the proposed approach applies to other truck parking locations, and thus, more research is required. However, an upward trend is expected in the availability of truck parking data in the future. Hence, it can be assumed that the potential for implementation of the system will increase over time.

To address this limitation, we tested the transferability of the system towards other parking areas and the overall results seem promising. The main concerns refer to supplying the model with information about the occupancy from earlier in the day (one, four, and seven hours ago). Compromising the previous occupancy variables causes the accuracy to decrease by 175%, however, the model is still able to learn the patterns of the dataset and can be used to provide valuable insights to truck drivers about the expected occupation of the parking lot up to a week ahead. To demonstrate this, we developed a prototype of an interactive dashboard that shows the expected occupation of the parking lot for each hour of the week. Overall, we recommend implementing both model configurations into a single predictive system, such that the highest value for the end-user is created.

Furthermore, we tested how data volume affects the accuracy of the model. The results demonstrate that a relatively small amount of data is required to train a model with satisfactory performance, which is highly promising regarding the ease of implementation in the future. Our experiments show that when training a model with only 0.2% of the original dataset, or 16 days of data, the model scores an accuracy of 0.0273. Thus, if the real occupancy is 86%, most of the time the model will generate a prediction between 83% and 89%. Thus, the accuracy is slightly worse than the accuracy of the model trained with 1.5 years of data.

Other limitations of the research refer to the presence of missing values in the dataset, which impacted the reliability of the data. Due to this, the generalizability of the model and the research's credibility is limited. The third limitation concerns the proposed conceptual model of a predictive system. Testing the

performance of such a system was beyond the scope of this study and it remains unclear whether this is the best implementation of the prediction model in practice. Finally, while testing inputs for the model, we mainly utilized publicly available data sources (such as holidays and weather data). However, it is possible that implementing other variables from private sources, such as information about truck drivers' schedules, might enhance the performance of the model. However, it should be noted that adding such data will drastically affect the complexity and scalability of the system.

Despite the mentioned limitations, this study has laid the foundations for further research in the field of truck parking occupancy prediction through machine learning. Future work should primarily focus on validating the model's deployment towards other truck parking lots. As mentioned before, it is anticipated that more parking locations are included in the system over time. Thus, future work should mainly emphasize researching structural and computational complexities associated with scaling up the system. This means that the proposed model configuration should be further validated once more truck parking locations provide the necessary data. Furthermore, the proposed concept of the information system should be further extended. The main objective should be developing a prototype and performing experiments, such that the performance of the system in terms of accuracy and reliability can be evaluated. Lastly, future work should focus on establishing a thorough business model, providing a detailed cost estimation, and improving the proposed implementation plan, such that the system can be put in production and the main goals are achieved in practice.

References

ACEA - European Automobile Manufacturers' Association. (2017, August 3). *Fact sheet: Trucks.* From ACEA: https://www.acea.be/publications/article/factsheet-trucks

Air Resources Board. (2017). *Staff report: Initial statement of reasons for proposed rulemaking.*

Brownlee, J. (2019, August 28). *How To Backtest Machine Learning Models for Time Series Forecasting*. Machine Learning Mastery. https://machinelearningmastery.com/backtest-machine-learning-models-time-series-forecasting/

Bundesamt für Güterverkehr (BAG). (n.d.). *Feiertags-Fahrverbote für Lkw in Deutschland im Jahr 2021. Bundesamt für Güterverkehr.* https://www.bag.bund.de/DE/Themen/RechtsentwicklungRechtsvorschriften/Rechtsvorschriften/Strassenverkehrsrecht/LKW-Fahrverbote/Feiertagsfahrverbote2021/Feiertagsfahrverbote2021neu.html

Burgess, E., Pfeffers, M., & Silverman, I. (2009). *Idling gets you nowhere: The health, environmental and economic impacts of engine idling in New York City. February*.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Rudiger, W. (2000). Crisp-Dm 1.0. *CRISP-DM Consortium*, 76

Chawathe, S. S. (2019, October). Using Historical Data to Predict Parking Occupancy. In *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)* (pp. 0534-0540). IEEE.

Chen, X. (2014). Parking occupancy prediction and pattern analysis. *Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, Tech. Rep. CS229-2014.*

Chugh, A. (2020, December 11). *MAE, MSE, RMSE, Coefficient of Determination, Adjusted R Squared — Which Metric is Better?* Medium. https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e

de Almeida Araujo Vital, F., Ioannou, P., & Gupta, A. (2020). Survey on Intelligent Truck Parking: Issues and Approaches. *IEEE Intelligent Transportation Systems Magazine*, *January 2020*, 2–16. https://doi.org/10.1109/MITS.2019.2926259

Directive 2010/40/EU. (2010). The framework for the deployment of Intelligent Transport Systems in the field of road. *Official Journal of the European Union*, p. 1–13. From http://data.europe.eu/eli/dir/2010/40/oj

European Commission. (2015). Fatigue 2015. *European Commission, Directorate General for Transport*.

European Commission. (2019). *Study on Safe and Secure Parking Places for Trucks* (Issue February).

Eurostat. (2021). *Freight transport statistics - modal split - Statistics Explained*. https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Freight_transport_statistics_-_modal_split#Inland_freight_transport_performance_-_the_need_to_adjust_road_transport

Eurostat. (2021). *Road freight transport statistics - Statistics Explained*. *April*, 1–10. https://ec.europa.eu/eurostat/statistics-explained/pdfscache/9217.pdf

Fabusuyi, T., Hampshire, R. C., Hill, V. A., & Sasanuma, K. (2014). Decision analytics for parking availability in downtown Pittsburgh. *Interfaces*, 44(3), 286-299.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The Elements of Statistical Learning*. New York, NY: Springer New York Inc

Haykin, S. (2010). *Neural networks and learning machines, 3/E.* Pearson Education India.

Hurwitz, J., & Kirch, D. (2018). *Machine Learning For Dummies*. John Wiley & Sons, Inc

Hyndman, R. J. (2006). Another look at forecast-accuracy metrics for intermittent demand. *Foresight: The International Journal of Applied Forecasting*, 4(4), 43-46.

IBM Cloud Education. (2020, August 19). *Supervised Learning*. From IBM: https://www.ibm.com/cloud/learn/supervised-learning

Kim, K., & Koshizuka, N. (2019, October). Data-driven parking decisions: Proposal of parking availability prediction model. In *2019 IEEE 16th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT and AI (HONET-ICT)* (pp. 161-165). IEEE.

Komarraju, A. (2021, February 22). *How Important Is Data Quality In Machine Learning?* Analytics Insight. https://www.analyticsinsight.net/how-important-is-data-quality-in-machine-learning/

Koninklijk Nederlands Meteorologisch Instituut (KNMI). (n.d.). *Uurwaarden van weerstations.,* from https://www.daggegevens.knmi.nl/klimatologie/uurgegevens

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling* (Vol. 26, p. 13). New York: Springer.

Lutsey, N., Brodrick, C. J., Sperling, D., & Oglesby, C. (2004). Heavy-duty truck idling characteristics: results from a nationwide truck survey. *Transportation Research Record*, *1880*(1), 29-38.

Ministerie van Algemene Zaken. (n.d.). *Which days are official public holidays in the Netherlands?* Government.Nl., from https://www.government.nl/topics/working-hours/question-and-answer/public-holidays-in-the-netherlands

Murray, D., & Glidewell, S. (2019). *An analysis of the operational costs of truck driving: 2019 update*. *November*.

Nagy, E., & Sandor, Z. (2012). Intelligent truck parking on the Hungarian motorway network. *Pollack Periodica*, *7*(2), 91–101. https://doi.org/10.1556/Pollack.7.2012.2.8

Olson, D. (2013). *Stakeholder Onion Diagramm*. From BAwiki | A Reference and Blog for Business Analysts: http://www.bawiki.com/wiki/Stakeholder-Onion-Diagram.html

Palaniappan, M., Wu, D., & Kohleriter, J. (2003). Clearing the Air: Reducing Diesel Pollution in West Oakland. *Pacific Institute. November*.

Pflügler, C., Köhn, T., Schreieck, M., Wiesche, M., & Krcmar, H. (2016). Predicting the availability of parking spaces with publicly available data. *Informatik 2016*.

Poliak, M., Poliaková, A., & Čulík, K. (2020). Impact of the social law on truck parking sustainability in the EU. *Sustainability*, *12*(22), 9430.

Provincie Overijssel. (2020). *Aanpak vrachtwagenparkeren. Meerjarenprogramma (EDO-registratiekenmerk 2020/0146291).*

Provoost, J., Wismans, L., Van der Drift, S., Kamilaris, A., & Van Keulen, M. (2019). Short Term Prediction of Parking Area states Using Real Time Data and Machine Learning Techniques. *arXiv preprint arXiv:1911.13178*.

Regulation EC No 561/2006. (2006). on the harmonisation of certain social legislation relating to road transport. *Official Journal of the European Union*.

Reinstadler, M., Braunhofer, M., Elahi, M., & Ricci, F. (2013). Predicting parking lots occupancy in Bolzano. *Academic Project, Computer Science, Free University of Bolzano Italy, Bolzano*.

Sadek, B. A., Martin, E. W., & Shaheen, S. A. (2020). Forecasting Truck Parking Using Fourier Transformations. *Journal of Transportation Engineering, Part A: Systems, 146(8), 05020006.*

Saltz, J. (2020, November 30). *CRISP-DM is Still the Most Popular Framework for Executing Data Science Projects*. From Data Science Project Management: https://www.datascience-pm.com/crisp-dm-still-most-popular/

Tashman, L. J. (2000). Out-of-sample tests of forecasting accuracy: an analysis and review. *International journal of forecasting*, *16*(4), 437-450.

Tefft, B. C. (2014). *Prevalence of motor vehicle crashes involving drowsy drivers, United States, 2009-2013*. Washington, DC: AAA Foundation for Traffic Safety.

Vlahogianni et al. (2016): Vlahogianni, E. I., Kepaptsoglou, K., Tsetsos, V., & Karlaftis, M. G. (2016). A real-time parking prediction system for smart cities. *Journal of Intelligent Transportation Systems*, 20(2), 192-204.

Wettschereck, D., Aha, D. W., & Mohri, T. (1997). A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review, 11(1),* 273-314.

Zhao, Z., & Zhang, Y. (2020). A comparative study of parking occupancy prediction methods considering parking type and parking scale. *Journal of Advanced Transportation*, 2020.

Zheng, Y., Rajasegarar, S., & Leckie, C. (2015, April). Parking availability prediction for sensor-enabled car parks in smart cities. In *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)* (pp. 1-6). IEEE.

Appendix A

Exp. №	Excluded category of variables	Max depth	Min samples split	RMSE (cross-validation)
1	Weather	10	40	0.022117703187418368
2	Time	8	80	0.02373195840842195
3	Holiday	6	80	0.023273696781025352
4	Historical occupancy	3	40	0.05325624598749133
5	Weather, Time	8	40	0.022555235621992544
6	**Weather, Holiday**	**10**	**40**	**0.022096633301074458**
7	Weather, Historical occupancy	11	90	0.051618629686806326
8	Time, Holiday	8	80	0.023736665016488363
9	Time, Historical occupancy	4	10	0.079482590607823
10	Holiday, Historical occupancy	3	20	0.05302282464812034
11	Weather, Holiday, Historical occupancy	10	110	0.051119516765595695

Appendix B

Exp. №	Excluded variables	Max. depth	Min. samples split	RMSE (test set)
1	'Hist_occup_1h'	9	30	0.03966840311650633
2	'Hist_occup_4h'	11	40	0.02294445133367039
3	'Hist_occup_7h'	9	30	0.02592208080236317
4	'Hist_occup_1h', 'Hist_occup_4h', 'Hist_occup_7h'	10	110	0.06329701848059695