

Timo Torvela

## **WEATHER IMPACT ON PUBLIC TRANSPORT RIDERSHIP**

Predicting passenger load using machine learning

# **WEATHER IMPACT ON PUBLIC TRANSPORT RIDERSHIP**

Predicting passenger load using machine learning

Timo Torvela  
Master's Thesis  
Spring 2024  
Data Analysis and Project Management  
Oulu University of Applied Sciences

## ABSTRACT

Oulu University of Applied Sciences  
Degree Programme in Data Analysis and Project Management

---

Author: Timo Torvela  
Title of thesis: Weather impact on public transport ridership  
Supervisor: Ilpo Virtanen  
Term and year when the thesis was submitted: Spring 2024  
Number of pages: 66 + 5 appendices

---

The purpose of this thesis is to investigate the correlation between the weather and number of bus passengers in a mid-sized city in the Nordics. The target was to build a model that could be used to predict the bus ridership based on weather forecasts. This data could be used to optimize the bus capacity.

The passenger data was collected using Automatic Passenger Counting solution provided by the company FARA. Weather data was collected from public sources. The data for the study was collected between 20 July 2022 to 20 August 2023. The main weather features this thesis focused on were temperature, snow, and rain. The correlation between weather and the number of passengers was analyzed using linear regression, gradient boosting, and machine learning models. Python library Scikit-learn was used for linear regression and machine learning. For gradient boosting, the Python library XGBoost was used.

Linear regression, both single-variable and multivariable, turned out to be inaccurate in predicting the number of passengers. Linear regression models had accuracies below 30%. Using machine learning, predicting the exact, or even close to exact number of passengers turned out to be difficult. However, when the number of passengers was divided into four or five different categories, the results were reasonably accurate. Especially, the extreme ends of the scale were categorized well. The results showed that on colder days the number of passengers grew. Also, the rain has a slight increasing effect on the bus ridership. When predicting the number of passengers using XGBoost, temperature was the most significant feature, followed by snow depth. Model accuracy was below 70%, which is average.

The impact of temperature was more severe on weekdays (from Monday to Friday) than it was on weekends. It was not clear if there is causality between weather and the number of passengers. The changes in the number of passengers were small. Using the model created to optimize bus capacity might not be reasonable. Without a further study on the data the current machine learning model is not suitable for commercial use, but the model provides a good base for fine tuning and improvements.

---

Keywords:

Machine learning, data analysis, public transport, XGBoost, Scikit-learn

# CONTENTS

1	INTRODUCTION .....	6
1.1	The idea of the thesis .....	6
1.2	How the study was conducted .....	6
1.3	Expectations .....	7
2	LITERATURE OVERVIEW .....	9
3	DATA AND METHODS .....	12
3.1	Passenger data .....	12
3.2	Weather data .....	12
3.3	Automatic passenger counting .....	13
3.4	FARA SmarHUB .....	13
3.5	SQL and relational databases .....	14
3.6	Data analysis methods .....	15
3.6.1	Linear regression .....	15
3.6.2	Lasso .....	16
3.6.3	Machine Learning and deep learning .....	16
3.6.4	Scikit-learn .....	19
3.6.5	MLPClassifier .....	19
3.6.6	XGBoost .....	20
3.7	Model accuracy metrics .....	21
3.7.1	Coefficient of determination (R-squared) .....	21
3.7.2	Accuracy score .....	22
3.7.3	ELI5 .....	22
3.7.4	SHAP .....	23
3.7.5	Confusion matrix .....	23
3.7.6	K-fold cross-validation .....	24
4	DATA PRE-HANDLING .....	25
4.1	Passenger data pre-handling .....	25
4.2	Weather data pre-handling .....	27
4.3	General view of the data .....	27
4.4	Preparing data for linear regression .....	30
4.5	Preparing data for machine learning models .....	35

4.5.1	One-hot-encoding .....	35
4.5.2	Standard scaling .....	36
4.5.3	Preparing the data .....	36
5	ANALYSIS AND RESULTS .....	40
5.1	Visual observations from the data .....	40
5.2	Linear regression.....	42
5.3	Multivariable regression.....	44
5.3.1	Using LinearRegression function .....	44
5.3.2	Using Lasso function.....	48
5.4	Creating a machine learning model.....	50
5.4.1	Classification .....	50
5.4.2	Finding the best hyperparameters using GridSearchCV .....	50
5.4.3	Classification using MLPClassifier .....	51
5.4.4	Classification using XGBoost .....	53
5.4.5	XGBoost with filtered data.....	56
6	DISCUSSION AND CONCLUTIONS.....	61
	REFERENCES .....	63
	APPENDIXES.....	65

# **1 INTRODUCTION**

## **1.1 The idea of the thesis**

FARA is a public transport company based in Norway that provides information technology services to public transport operators. FARA has hundreds of customers in 14 different countries. In total, about 34 000 buses run FARA services. FARA is a part of Modaxo Group. (FARA)

The idea for the thesis came up in a meeting between colleagues. In the conversation it was brought up that it would be interesting to investigate the correlation between the weather and the number of passengers. At this stage it was considered that simple linear regression would be enough to analyze the correlation. At later stages, when the topic was given more thought, the possibility of using machine learning in predicting the number of passengers rose up.

One of the real time information services offered by FARA is the Automatic Passenger Counting (APC) that counts the number of passengers on a vehicle. This provides an opportunity to collect data about public transport ridership. After some research, it was evident that in the existing literature on the topic the APC was not that much used for passenger data collection.

Selecting a suitable city and a bus operator required careful consideration. FARA has several customers in the Nordics, so it would be natural to select a city there. In addition, there are also several customers that have APC installed on the entire fleet. Also, Nordic weather can be harsh at times, so it could have interesting effects on the number of passengers. After these considerations, a mid-sized city was selected. The bus operator was the public operator in the selected city. To protect the customer, it is not possible to disclose which city or operator is in question.

## **1.2 How the study was conducted**

The purpose of this study is to investigate the possible correlation between weather conditions and number of passengers in bus transportation. Especially, the impacts of temperature, rain and snow are under inspection.

One of the most fundamental, and most practical, techniques in data analysis is to draw a line through the observed data points to show a correlation between two or more variables. A regression attempts to fit a function to the data to predict new values. Linear regression fits a straight line to highlight a linear relationship between variables and to make predictions. (Nield, 2022)

Artificial intelligence (AI) has gained a lot of attention in recent years. It already plays a significant role in our lives. The hype has been intense, and it shows no sign of slowing down. Despite the recent hype, artificial intelligence is not a new concept. It can be traced back all the way to the 1950s. Back then, a handful of computer science pioneers started to think if machines could be made to “think”. The underlying ideas were even older, but in the 1950s the field started to be researched. (Nield, 2022; Chollet, 2021)

When it became apparent that regression models would not be sufficient, machine learning was brought in the picture. Machine learning is a subfield of artificial intelligence. The traditional way computers have been programmed is that a programmer gives the program a set of instructions what to do with data. Then the program will follow those instructions and come up with answers. The basic idea of machine learning is that a model learns about the data it is fed. A machine learning model is given the data and the answers. Then the model will then learn from the data and come up with rules. These rules can then be used to make predictions based on new data. FIGURE 1 illustrates the difference between classical programming and machine learning. (Chollet, 2021)

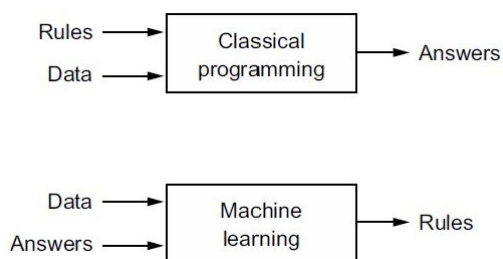


FIGURE 1. Classical programming vs. machine learning. (Chollet, 2021)

### 1.3 Expectations

Before inspecting any of the literature on the topic, there was no clear hypothesis on how the number of passengers would behave in different weather conditions. It was thought that when the temperature drops, it would make people switch from bicycles to public transport. On the other

hand, the dropping temperature could make someone switch to a car or stay at home working remotely, if those are options.

The impact of rain was also difficult to estimate. Rain might make people switch from bicycles and walking to public transport, or to a car. If the bus stop where a person is supposed to start the journey does not have shelter, he or she might decide to skip the bus journey altogether. Chapter 2 provides an overview of the literature on the topic.

The first idea was to collect data directly from the buses on production using scripts, but it quickly turned out to be an inefficient way. Instead, the data was collected from a database that contained the passenger data for more than a year.

If there was a correlation between weather and number of passengers, then the thesis should provide a good basis for future development. The ability to predict the number of passengers based on weather forecast could help public transport operators to plan the need for vehicles.



## 2 LITERATURE OVERVIEW

This chapter will provide an overview of selected studies on the topic. A lot of studies have been conducted on the effect of weather on the number of public transport passengers. It was not necessary to list them all here but to give some examples of studies and their results. Also, basics of the used technologies and tools are presented.

There have been some studies on the impact of weather on public transport ridership. This chapter gives an overview of some of the other studies. Most of the studies mentioned here are based on big cities. Also, in most of the cities studied, the weather is generally warm, or mild. There is not a clear consensus of how the weather affects passenger load.

Guo et al., (2007) noted that there has been very little literature on the topic before their study. They assumed one reason was the lack of data. Before that most of the data (trip times, passenger loads, etc.) was collected manually. Due to this there were only a few observations at any given time. Other reasons were that the impact of weather on public transport can be complex, and that the significance of weather has not been recognized.

Guo et al., (2007) conducted their study on data collected in Chicago, Illinois. They determined that generally good weather causes an increase in public transport ridership and bad weather causes a decrease. They also highlighted that on weekends the number of passengers was affected more drastically than during weekdays. It is possible that extremely bad weather can increase the number of passengers.

Nissen et al., (2020) studied the effects of weather on public transport in Berlin. The number of passengers was calculated based on ticket sales. The study concluded that on weekdays the number of passengers increased by 5% if it was raining. Cold weather caused an increase in the number of passengers as well, up to 30% when the temperature dropped below -5 °C. On Sundays cold or wet weather caused the lowest ticket sales. It was noted that on some routes the effect of weather could differ from the district average. For example, during warm days the routes leading to a public beach had an increase in the number of passengers.

Zhou et al., (2017) conducted a study that took place in Shenzhen, China. The study used smart card records to calculate the number of passengers and weather data was utilized on an hourly level. In addition to bus passengers, also metro ridership was in the study. The study applied multivariate regression approach to analyze the impact of weather on the ridership. The population of Shenzhen is more than 10 million residents. The population density is over 5 000 people per square kilometer. The yearly average temperature is about 23 °C and the yearly cumulative rainfall is more than 1900 mm.

Miao et al., (2019) studied the effects of extreme weather on bus ridership in the Salt Lake City metropolitan area and the moderating effect of bus shelters. Extreme weather in this case was either extreme cold, extreme hot or heavy rain. Their conclusion was that extreme weather conditions have a decreasing effect on the bus ridership.

According to Wei, (2022) the senior passengers are a more sensitive group to changes in weather conditions. Wei's study was on Brisbane, Australia. Smart card data and weather station data for a 12-month period was utilized in the study. The study suggested that "Morning peak hours are the period when passengers have the strongest weather tolerance.". In this study the weather impacts and passenger numbers were investigated on a daily base.

Singhal et al., (2014) focused their study on New York City. They concluded that "the PM time period is most affected, followed by midday period and least affected during AM period". The minimum square method was used to analyze the data in the study.

Machine learning and Automatic Passenger Counting (APC) were used by Thiagarajan & Prakashkumar, (2021) when they studied passenger demand on public transport. Their study was conducted on data collected from a big Swedish bus operator. Thiagarajan & Prakashkumar utilized the Python Scikit-learn libraries that were also researched to be used in this thesis. Their focus was on finding optimal timetables for different scenarios (summer vs. winter, weekdays vs. weekends, etc.). The study did not involve weather attributes.

Fontes et al., (2020) used deep learning to predict the passenger loads based on weather conditions. Their data were collected from a medium-size European metropolitan area, Porto in Portugal. The weather conditions were collected from three different times: one hour, two hours and three hours before departure. For deep learning they used the Python Keras library. For

measuring the model accuracy, the methods of Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and coefficient of determination (R<sup>2</sup>) were used. The average error of the model, when compared to actual demand, was between 5.45% and 6.30% (apart from students, for whom it was 14.72%). They determined that the model was more accurate on weekdays than on weekends. On holidays or strike days the model accuracy decreased as well. The model was the most accurate when weather conditions two hours prior departure were used.

Li et al., (2015) used cluster analysis and multiple linear regression models when studying the topic on Fengxian District in Shanghai. The study categorized different bus routes into three clusters based on different variables (route length, number of stops, etc.). They found that humidity, wind speed, rainfall and temperature had negative impact on bus ridership. They also mention that the impact of weather should be investigated further in different geographical contexts. The daily number of passengers in Fengxian is about 232 000, which is in the same region as the city in this thesis. Also, the timespan of data collection was one year, as was in this thesis.

Wei et al., (2019) studied the effects of weather on a daily and half-hourly scale in the sub-tropical Brisbane, Australia. The study included three different ride types: buses, trains, and ferries. Using regression models, and including three weather conditions (wind, temperature, and heat index), they noticed that the number of passengers is less likely to drop on morning and afternoon peak hours. The number of ferry passengers was impacted more by weather than other transit types. Trains were the least affected.

Brisbane was also the location of the study by Tao et al., (2016). Their study focused on finding geographic patterns of the correlation between weather and number of passengers. Based on Tao et al., some locations were affected by weather more than others. The direction of the effect also varied in different geographic locations, depending on the surroundings of the area such as the possibility to take shelter when waiting for a bus.

In another study by Tao et al., (2018) impact of hourly based weather data was researched in Brisbane, Australia. Implementation of time-series regression highlighted that there were significant hourly changes in transit ridership, especially in the cases of rainfall or temperature change. As other studies, Tao et al. noted that the effects of weather were more dramatic on weekends rather than on weekdays. The destination of travel had no influence on the effect of weather as well (e.g., a shopping center vs. the university).

## **3 DATA AND METHODS**

### **3.1 Passenger data**

The passenger data is collected automatically using Automatic Passenger Counting (APC) sensors. The method is described in more detail in Chapter 3.3.

Passenger data is collected at each stop. The passenger counts from each stop are sent to the backend system. The backend system is responsible for storing the passenger counting data and combining it with details of the bus journey. Data includes information about the line a bus is travelling on, bus stop id, bus stop coordinates, report quality, door ID, etc. This data is stored in an SQL database.

To minimize the risk of turbulence on production systems, a backup of the SQL database was used to collect the passenger data for this thesis. The data consisted of data between 20 July 2022 to 20 August 2023. The database backup was restored on a local SQL server running on a PC. SQL language was used to query the database tables, collect, and combine all the data that could be beneficial for the study. The data was stored in a comma-separated values (CSV) file.

There were two tables in the database including data that was interesting for the study. One of the tables included passenger counts, one row for each door on each bus stop. The other table included data about the location of the stop, for example which bus was in question. In total, there were 30 data features in these two tables. In the end, only date, number of boarded passengers, and number of alighted passengers were used in the study.

### **3.2 Weather data**

The weather data was collected using the web site [visualcrossing.com](https://visualcrossing.com). Data was collected as a CSV for the same period as the passenger data (between 20 July 2022 to 20 August 2023). The data there is collected from weather stations in 50 miles radius of the requested spot. The requested spot was set at the center of the city. The weather engine forms the data by analyzing the data from the weather stations. Data received from the closest weather stations are given more weight

than the stations farther away. The data used in this thesis was aggregated on a daily basis. (*Visualcrossing.com*, 2023)

### **3.3 Automatic passenger counting**

The passenger data for this thesis was collected using the Automatic Passenger Counting (APC) solution. The solution includes computer vision cameras (APC sensors) for each of the doors in the vehicle. Sensors count boarding and alighting passengers at each stop. Counts are sent to an APC service that calculates the total numbers for the vehicle. The APC service publishes passenger count reports on door level and vehicle level. The values are sent to a backend system using TCP protocol. (ITxPT, 2019)

The APC sensors used in this thesis are based on 3D stereo vision technology. The exact technologies used by the sensor manufacturer is not known. However, similar machine vision cameras commonly use binocular vision, i.e., two adjacent cameras to form a 3D image. One of the key aspects in computer vision is the shading. Based on the shades in the images or video, the sensor can identify different objects. (Davies, 2004, Chapter 16.2)

The APC sensors used in this thesis are manufactured by Dilax. The sensors can separate the counts for adults, children, and bicycles. The manufacturer promises accuracy up to 99% for their APC solutions. (Dilax Automatic Passenger Counting)

### **3.4 FARA SmarthUB**

FARA SmarthUB is a communication gateway that integrates devices on a vehicle to each other and connects them to the backend systems. SmarthUB connects to the automatic passenger counting sensors, collecting the passenger data, and sending it to the backend system for storage and processing. (FARA, 2023)

FIGURE 2 displays an example of a SmarthUB installation. The public transport operator of this thesis does not necessarily have all the components installed. The parts important to this thesis are the SmarthUB and ITxPT passenger counting.

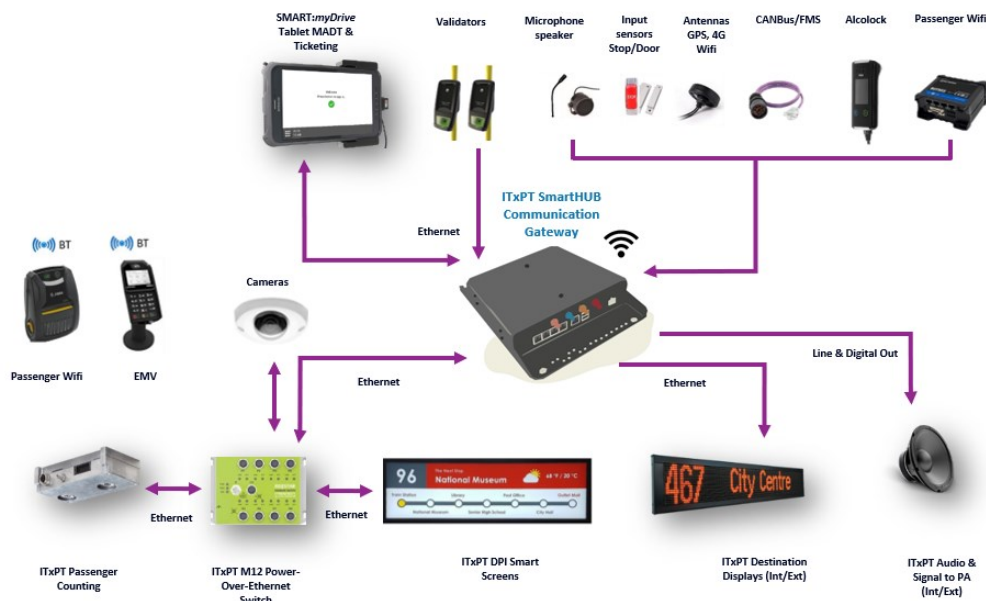


FIGURE 2. Diagram of a SmartHUB system. (FARA, 2023)

### 3.5 SQL and relational databases

Data in this thesis was stored in a relational database and it was queried using SQL.

SQL is a query language for fetching data from a relational database. A relational database is a data warehouse where large amounts of data can be stored in several data tables. The data records in different tables can be related to each other. The tables can be connected to each other by key values, like cust\_id, product\_id or account id in FIGURE 3. (Beaulieu, 5-6.)

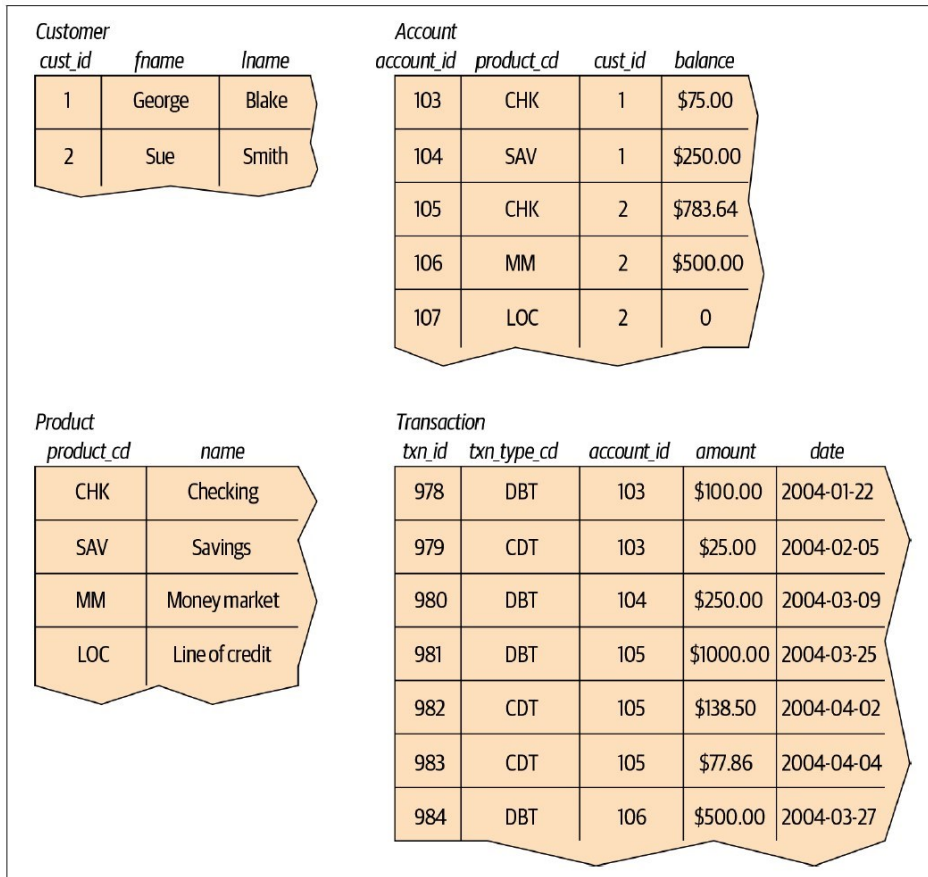


FIGURE 3. Example of a relational database. (Beaulieu, 2020)

### 3.6 Data analysis methods

#### 3.6.1 Linear regression

Linear regression is a method that was created already before the age of computers. The functions are simple and sometimes provide an adequate description of how inputs affect the output. In many cases linear regression can provide more accurate predictions than more complicated models. (Hastie et al., 2008, 43)

Let's assume we have an input vector  $X^T = (X_1, X_2, \dots, X_p)$ . To predict a real-valued output of  $Y$ , the linear regression has the form

$$f(X) = \beta_0 + \sum_{j=1}^p X_j \beta_j$$

Where  $\beta_j$  are unknown parameters of coefficients. The linear model assumes that function  $E(X|Y)$  is linear or that the linear model provides an approximation that is reasonable. Typically there is a

training data set that provides inputs and outputs in the form of  $(x_1, y_1) \dots (x_n, y_n)$  from which the parameters  $\beta$  can be estimated. (Hastie et al., 2008, 43-44)

One solution for determining the coefficients for linear model is the Numpy polyfit function. The function minimizes the squared error (or mean squared error, MSE). (Numpy Documentation)

$$E = \sum_{j=0}^k |p(x_j) - y_j|^2$$

This means that the Numpy polyfit function finds the coefficients of the polynomial that have the lowest squared error between polynomial's predictions and actual data points. In other words, it finds a polynomial that fits the best to the data. (Numpy Documentation)

### 3.6.2 Lasso

Lasso (least absolute shrinkage and selection operator) is a regression analysis method that is based on shrinkage methods. It performs variable selection and regularization to enhance the prediction accuracy. An important characteristic of lasso is that it eliminates the least important features by setting them to zero. (Géron, 2022)

Lasso is defined by

$$\hat{\beta}^{lasso} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^N (y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j)^2$$

that is subject to

$$\sum_{j=1}^p |\beta_j| \leq t$$

Where  $\beta_0$  is the constant coefficient,  $\beta := (\beta_1, \beta_2, \dots, \beta_j)$  is the coefficient vector and  $t$  is a prespecified free parameter to determine the degree of regularization. (Hastie et al., 2008, 68-69.) (Wikipedia)

### 3.6.3 Machine Learning and deep learning

Machine learning is a branch of artificial intelligence. They are systems that are designed to use data for learning. The systems should learn and improve with experience. Machine learning models



can be used to predict outcomes based on what they have learned. One category of machine learning is supervised learning. In supervised learning every data point in the training data has an input object and an output object, also known as labels. (Bell, 2020, 3.)

Evaluating the model requires splitting the data into training data, validation data and test data. Sometimes the splitting is done into two parts: training data and test data. Training data is used to train the model. After training, the model is evaluated using validation data. Finally, before the model is ready to face the real world, it is tested using the test data. The split ratio depends on the amount of data, common splits include 80/20% for training and test sets and 60/20/20% for training, validation, and test sets. (Chollet, 2021, Saleh, 2018)

Deep learning is a subfield of machine learning. The “deep” in deep learning refers to successive layers of learning neurons, like the layers of neurons in FIGURE 4. Modern deep learning networks can consist of tens or hundreds of layers. These layered representations are called neural networks. (Chollet, 2021)

FIGURE 4 shows a typical depiction of a fully connected feed-forward neural network. In a fully connected neural network each neuron (blue circles) in a layer is connected to every neuron in the next layer. In a feed-forward network, each discrete layer of sensors is connected to the next one. One layer is the input layer (small black dots) that receives the input values and feeds them forward to the next layer. The outputs from that layer are fed to the next layer until the output layer is reached. The layers in between the input layer and output layer are called hidden layers. (Nelson, 2023)

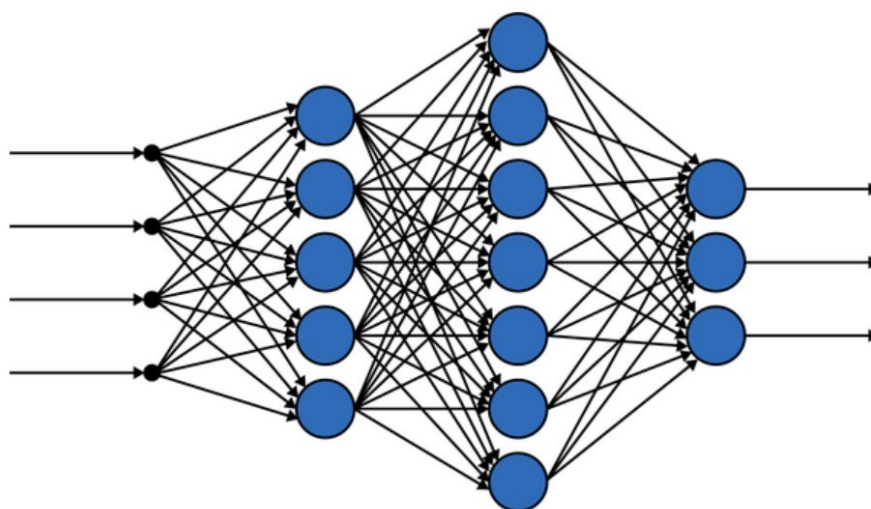


FIGURE 4. A fully connected feed-forward neural network. (Nelson, 2023)

The neural networks are modelled after the brain cortex. Our brain learns by reinforcing the connections between neurons when facing a familiar concept. When facing information that contradicts the previously learned concepts, the connections between neurons are weakened. Neural networks learn in a similar manner. The difference is that machines understand only numbers. (Nelson, 2023)

A common problem in machine learning is overfitting. Overfitting means that the model performs well on testing data, but it generalizes poorly to any new data. The most fundamental way to avoid overfitting is to use different data for training the model and another set of data to test it. Test data can be created by simply splitting the full data set into training and test data. At another end of the scale there is underfitting. (Grus et al., 2019, 149-150.)

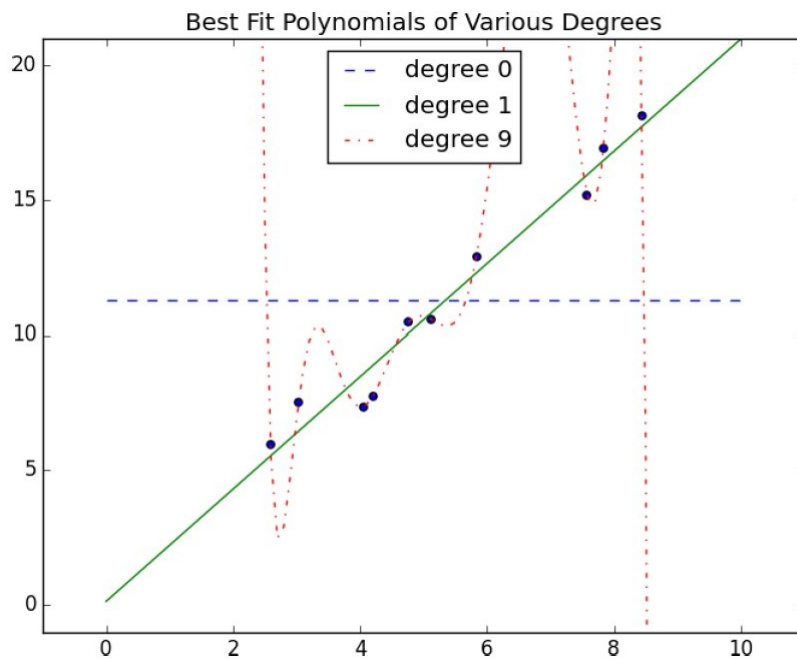


FIGURE 5. Overfitting and underfitting. (Grus et al., 2019)

FIGURE 5 visualizes underfitting and overfitting. The black dots represent the data points in training data. The dotted blue line is the degree 0 polynomial fit (i.e. a constant). The solid green line is degree 1 polynomial fit. The dotted red line is a degree 9 polynomial (i.e. a 10-parameter). The horizontal line shows the best fit for degree 0 polynomial. It severely underfits the training data. Degree 9 polynomial cuts through every single data point, but it seriously overfits. If we were to select some additional data points, the polynomial would likely miss them by a lot. Degree 1 is balanced in between. Additional data points would likely hit quite close to the degree 1 fit. (Grus et al., 2019, 149)

### 3.6.4 Scikit-learn

Scikit-learn is an open-source library that provides machine learning methods for different kinds of problems. Scikit-learn was designed to be easy to use and still have good performance. (Pedregosa et al., 2011)

Estimators are functions that can estimate some parameters based on input data. Unsupervised learning estimators take one dataset as a parameter. Supervised learning estimators take two datasets, the second one contains the labels (i.e. outputs). Any other parameters for the estimation process are called hyperparameters. Some of the estimators can transform a dataset. These estimators are called transformers. (Géron, 2022)

### 3.6.5 MLPClassifier

MLPClassifier, multi-layer perceptron, is an estimator of scikit-learn. Multi-layer perceptron is a form of feedforward neural networks that can distinguish data that is not linearly separable. Modern feedforward networks are trained using the backpropagation method. (Scikit-learn Documentation)

Training a neural network is a process of finding the parameter values, or weights, that minimize the loss function. At its base it is like the way that linear functions work. In neural networks, the linear combination of features, adding bias and passing the result through a nonlinear function – are called an activation function. All this takes place in only one of the neurons. This process happens again in dozens, hundreds, or thousands of times, depending on the complexity of the neural network. (Nelson, 2023)

When training a neural network, the desired output is known. So, when an input is fed to the network, the result will be compared to the target output and the loss is calculated. Loss is the sum of squared errors. The gradient of this loss is computed as a function of the output neuron's weights. The gradients and errors are "propagated" backwards to calculate the gradients with respect to the weights of the hidden neurons. Then a gradient descent step is taken. This is backpropagation in a nutshell. This procedure is run many times until the neural network converges. (Grus et al., 2019. 227.)

### 3.6.6 XGBoost

XGBoost stands for Extreme Gradient Boosting. (XGBoost Documentation, 2022) XGBoost is a collection of different kinds of machine learning models working together. The individual models are called base learners. Decision trees are one of the most used base learners in XGBoost. XGBoost has a robust and quick implementation of gradient boosted decision trees, which makes it commonly used in production environments. (Wade, 2020; Ameisen, 2020, 98.)

Decision tree is a supervised learning method that is used for classification and regression tasks. The idea is that a model uses data to learn to make decisions based on simple decision rules. Decision trees are simple and easy to visualize. They require only a little data preparation. On the other hand, decision trees can easily overfit, i.e., create complicated trees that do not generalize well. With decision trees a small change in data can lead to a totally different tree to be created. (Scikit-learn Documentation)

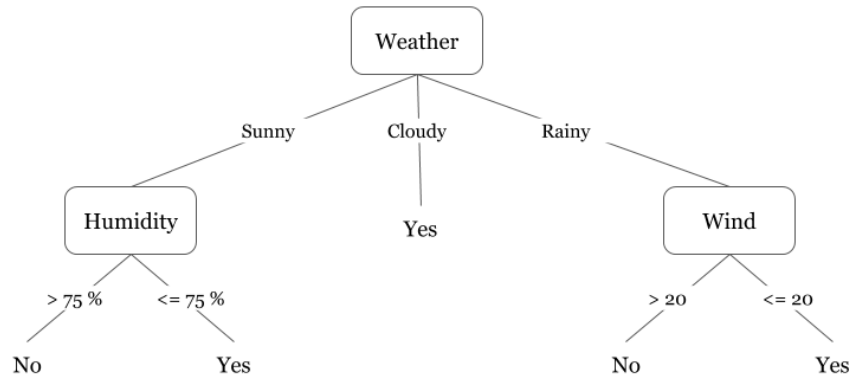


FIGURE 6. Decision tree – whether to go outside to play badminton or not. (Hacker Earth)

FIGURE 6 is an illustration of a decision tree that can be used to decide whether to go outside to play badminton or not. At the top is the root node (Weather) where the decision process starts. The first decision here is if the weather is sunny, cloudy, or rainy. Based on the decision the branch to follow is chosen. The process continues downwards going through the nodes until the final node at

the bottom of the picture is reached (here either “No” or “Yes”). The last node is called a leaf. (Joly, 2017)

Tree boosting is an effective machine learning method. XGBoost is a scalable end-to-end gradient tree boosting method. Gradient tree boosting is found to be an effective tool in many applications (Chen & Guestrin, 2016). XGBoost has been successfully used in many machine learning competitions. It can be used for a wide range of tasks like classification or ranking (Huyen, 2022). It has also been used in solving Higgs Boson classification problem. (Chen & He, 2015).

XGboost utilizes a special form of gradient boosting method. The basic method behind gradient boosting is that it learns from the errors in the previous individual decision trees. Each new tree is built based on the mistakes in the previous tree. “Gradient boosting computes the residuals of each tree's predictions and sums all the residuals to score the model.” (Wade, 2020)

### **3.7 Model accuracy metrics**

#### **3.7.1 Coefficient of determination (R-squared)**

Coefficient of determination, R-squared ( $R^2$ ), is an equation to measure how well a statistical model predicts the actual outcome. It is a commonly used metric in statistics and machine learning regressions. The  $R^2$  explains how much of the variation of one variable is explained by the variance of another variable. As R approaches the perfect correlation, -1 or 1,  $R^2$  is approaching 1. In other words, if the  $R^2$  is 0, the model does not predict the outcome any better than using the mean number of the target values would. If the  $R^2$  is 1, the model would perfectly predict the target. (Grus et al., 2019, pp. 182-183;Nield, 2022)

Coefficient of determination is defined by:

$$R^2 = \frac{\text{Variance explained by the model}}{\text{Total variance}}$$

It is still important to note that correlation does not necessarily mean causation. If a correlation is observed between x and y, it does not always mean that x causes y. Actually, it could be that y causes x. (Grus et al., 2019, pp. 182-183;Nield, 2022)

### 3.7.2 Accuracy score

Accuracy score is a function of Scikit-learn library. It determines the accuracy of a model by either calculating a fraction of correct predictions or by counting the correct predictions. In this thesis the default of a fraction of correct predictions was used. In multilabel classification the subset accuracy is returned by the function. (Scikit-learn Documentation)

Fraction of correct predictions is calculated with

$$accuracy(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} 1(\hat{y}_i = y_i)$$

Where  $\hat{y}_i$  is the predicted value of the  $i$ -th sample and  $y_i$  is the corresponding true value. (Scikit-learn Documentation)

Accuracy score was used to measure the performance of MLPClassifier and XGBoost models.

### 3.7.3 ELI5

ELI5 is a Python library to visualize and debug machine learning models. ELI5 provides interfaces to all machine learning libraries supported by Scikit-learn, and it has many features. ELI5 is used to show which features are the most important for good predictions. (ELI5 Documentation)

TABLE 1. An example of ELI5 weight values.

Weight	Feature
+309667.478	<BIAS>
+4701.352	snow
+841.742	precip
+52.609	windspeed
-20.185	cloudcover
-1219.007	temp

The features are listed in a descending order of importance. The BIAS values can be interpreted as an intercept term for a linear regression model. (Mishra, 2023)

### 3.7.4 SHAP

SHAP (Shapley Addictive Explanations) is a Python library, that uses game theoretic approach to explain the output of a machine learning model. (Lundberg et al., 2020)

SHAP TreeExplainer function was used to evaluate the importance of different features when using XGBoost to analyze the data.

SHAP function TreeExplainer uses three main contributions to improve the interpretability of tree-based models (Lundberg et al., 2020):

- A polynomial time algorithm to compute optimal explanations based on game theory.
- A new type of explanation that directly measures local feature interaction effects.
- A new set of tools for understanding global model structure based on combining many local explanations of each prediction.

### 3.7.5 Confusion matrix

Confusion matrix shows the actual and predicted values and tells how many of the predictions were classified correctly and how many times an instance was classified incorrectly. (Géron, 2022)

FIGURE 7 shows the illustration of the purpose of a confusion matrix. In the picture the first row tells how many of the values were actually “Yes”. First cell on the row shows that 732 times the prediction was correct.

Actual Labels	Yes	732	93
	No	137	38
		Yes	No
		Predicted Labels	

FIGURE 7. Illustration of a confusion matrix. (Trautmann)

### 3.7.6 K-fold cross-validation

The main idea in k-fold cross-validation is that the data set is shuffled and then divided into  $k$  same-sized segments. Then, the data is iterated so that one segment is used for testing and rest of the data is used for training. In the next iteration, another segment is used for testing, then a third segment is used for testing, and so on. K-fold cross-validation gives a good overview of the accuracy of the model because of its randomness. (Refaeilzadeh et al., 2016)



## 4 DATA PRE-HANDLING

### 4.1 Passenger data pre-handling

Passenger data was delivered in a database backup from the production backend system. Microsoft SQL Server 2022 Developer edition was downloaded and installed on a local computer for running the database. Microsoft SQL Management Studio was used for accessing the database.

The database backup consisted only of passenger data collected by Automatic Passenger Counting. The database included eight different tables and two of them were of interest for this study. The passenger data covered the period between 20 July 2022 to 20 August 2023.

First database was called VehicleData. It consisted of several columns. Columns that were not interesting for this thesis were dropped. The selected columns are listed in TABLE 2.

TABLE 2. Selected columns of database table "VehicleData".

ID	ID of the event (row in the table)
Vehicle ID	Identification number of the bus
Line	Identification number of the line for the information systems. Different than "Public Line"
Public Line	Public line number that is used for passengers. This is the one shown at the front display of the bus and on the timetables
Latitude and longitude	Positioning coordinates of a bus stop
Timestamp	Date and time of the event.
Total count	Current number of passengers in the bus
Occupancy ratio	How full the bus is (percentage)

The second database was called VehicleDoor. It showed boarded and alighted passengers for each door of the vehicle on each stop. VehicleDoor had noticeably less columns compared to VehicleData table. Again, the uninteresting columns were dropped, the selected columns are listed in TABLE 3.

TABLE 3. Selected columns of database table "VehicleDoor".

Vehicle Data ID	ID of the event (row in the table)
Door ID	Door number in the bus (1-4)
Boarded	Number of passengers that got on the bus at a certain stop.
Alighted	Number of passengers that left the bus at a certain stop.
Passenger Type ID	Number indicating the type of passenger alighted/boarded. (1-5). Different types: <ol style="list-style-type: none"> <li>1. Absent (not known/not available)</li> <li>2. Adult</li> <li>3. Child</li> <li>4. Other</li> <li>5. Bike</li> </ol>

VehicleDoor table included the passenger counts and VehicleData table had other important data, like date and time of the counts. The data between the two tables had to be combined.

The necessary data was collected from the database tables using SQL. As important data was spread into two different tables, the data had to be combined using SQL joins. The primary key in VehicleData table was ID and in VehicleDoor table it was Vehicle Data ID. Using these it was possible to combine the data. (Beaulieu, 2020, pp. 87–92)

Data was then saved in text format (CSV) for further analysis. At this stage, the file included more than 81 million lines. One line represented the counts from one door at a certain bus stop. Each line also included the date of the occurrence. The next step was to sum all the boarded and alighted passengers on daily level. After that calculation, the data consisted of only the most important data, that is the number of passengers boarding and leaving the bus and the date. Now there were 397 data points, one for each day. The data was saved in a CSV file.

## 4.2 Weather data pre-handling

The weather data had over 30 different variables for each day. The most relevant variables for this study were selected. The selected variables are listed in TABLE 4. The data was saved in a CSV file.

TABLE 4. Selected weather attributes. (Visual Crossing Documentation, 2023)

Variable	Comment
Cloudcover	Cloud cover
Freezing rain	Expected precipitation type is freezing rain
Humidity	Relative humidity
Ice	Expected precipitation is ice
Precip	Precipitation
Rain	Expected precipitation is rain
Snow	Expected precipitation is snow
Snowdepth	Snow depth
Temp	Temperature (mean temperature)
Tempmax	Maximum temperature
Tempmin	Minimum temperature
Wind speed	Wind speed
Winddir	Wind direction

## 4.3 General view of the data

The passenger data and weather information were stored in CSV format. Passenger data was stored in one CSV file and weather information in another CSV file. To make analysis easier, the two files were combined into one dataset.

Several analysis methods were studied in this thesis. Different methods have different limitations about data, and therefore separate datasets were created. For example, in some cases including weekends and public holidays in the data would have caused unreliable results. For example, when using linear regression, the weekends and holidays would make the analysis useless.

To start with, there was a decision on the passenger data to be made. The data contained two values for number of passengers: boarded and alighted. Which one should be used for the analysis? Interestingly, there was a difference in the number of passengers boarding the buses and number of passengers alighted from the bus.

TABLE 5. Total number of boarded and alighted passengers and their difference.

	Number of passengers
Passengers boarded	87 438 249
Passengers alighted	84 809 232
Difference	2 629 017

The average number of passengers each day was 220 247 passengers. On average the difference between boarded and alighted passengers was 6696 passengers per day. That means 3 % of the total number of boarded passengers. If we filter Saturdays and Sundays out of the data, the percentage increases to 3,26 %. Considering the total number of passengers, the difference between boarded and alighted passengers was insignificant to this study.

TABLE 6. Daily averages of boarded and alighted passengers and differences between them.

Year	Month	Boarded	Alighted	Difference
2022	September	253740.33	248521.77	5218.57
2022	October	235398.81	229383.55	6015.26
2022	November	258144.70	248335.30	9809.40
2022	December	204667.42	197102.10	7565.32
2023	January	248741.65	239679.45	9062.19
2023	February	265582.04	257212.04	8370.00
2023	March	280782.32	272004.58	8777.74
2023	April	208663.40	201619.93	7043.47
2023	May	232145.45	224817.77	7327.68
Total		220247.48	213625.27	6622.21

A possible explanation could be that when boarding a bus, people usually come in one by one and in the Nordics usually from only one door – the front door. When people leave the bus, they are

much closer to each other. In that case the possibility for miscalculating the number of alighting passengers is larger.

When looking at the boarded and the alighted passengers at a plot, in FIGURE 8, the difference can be seen. Also, the decreases in the number of passengers during autumn break, christmas and new year, winter break and summer holidays (July-August) are clearly visible. There is a drop in the number of passengers also every weekend. FIGURE 9 displays the same difference in September 2022.

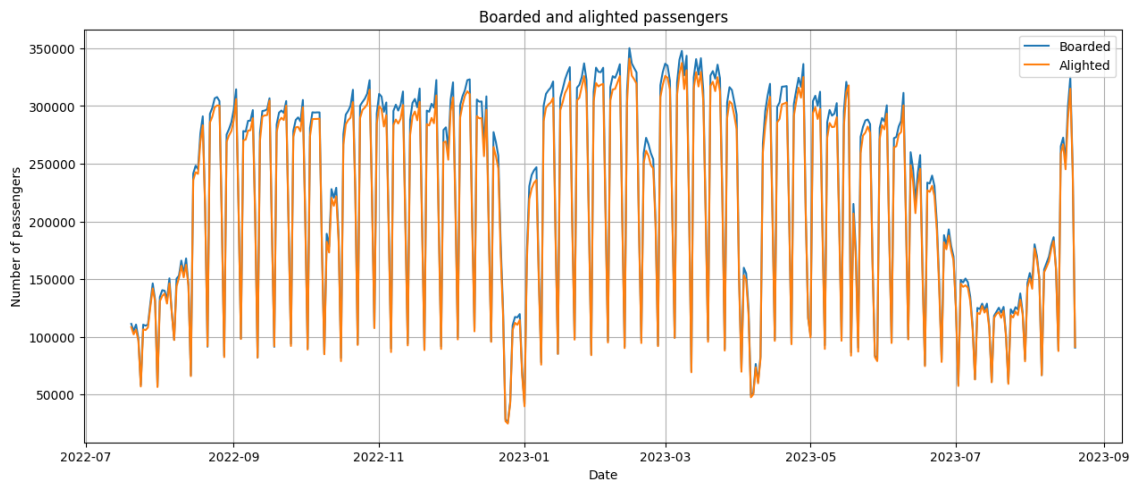


FIGURE 8. Boarded vs. Alighted passengers, all data.

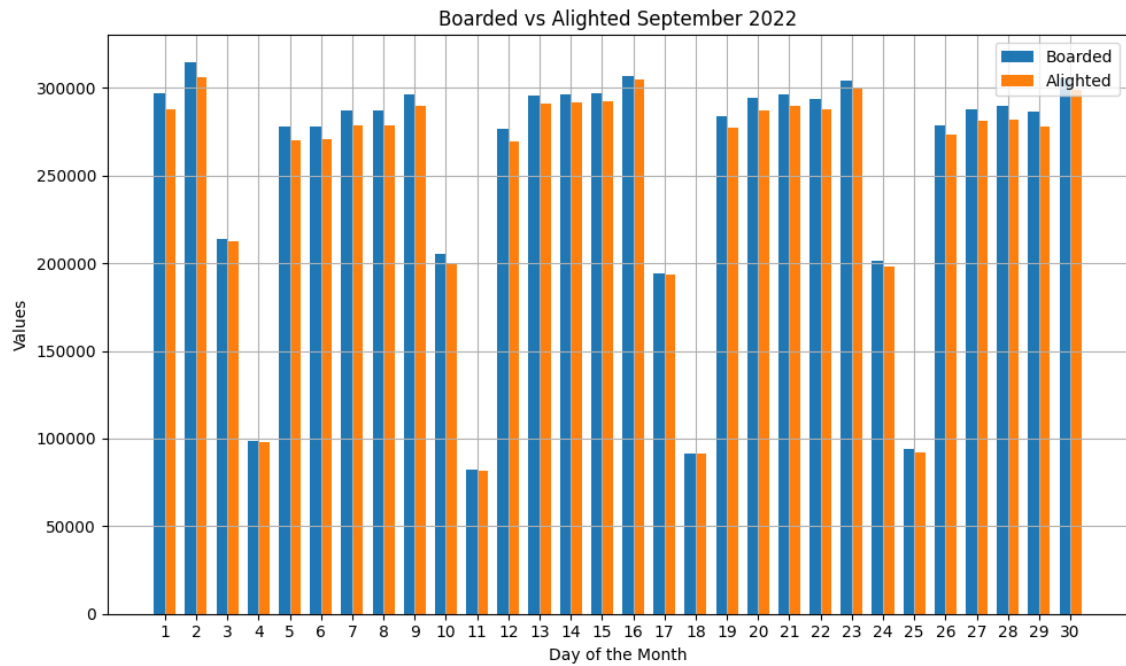


FIGURE 9. Boarded vs. alighted passengers, September 2022.

Based on these considerations, the number of boarded passengers was used in this study instead of the alighted passengers.

When Saturdays and Sundays were removed from the data, the vacation periods and public holidays can be clearly seen in the line plot shown in FIGURE 10.

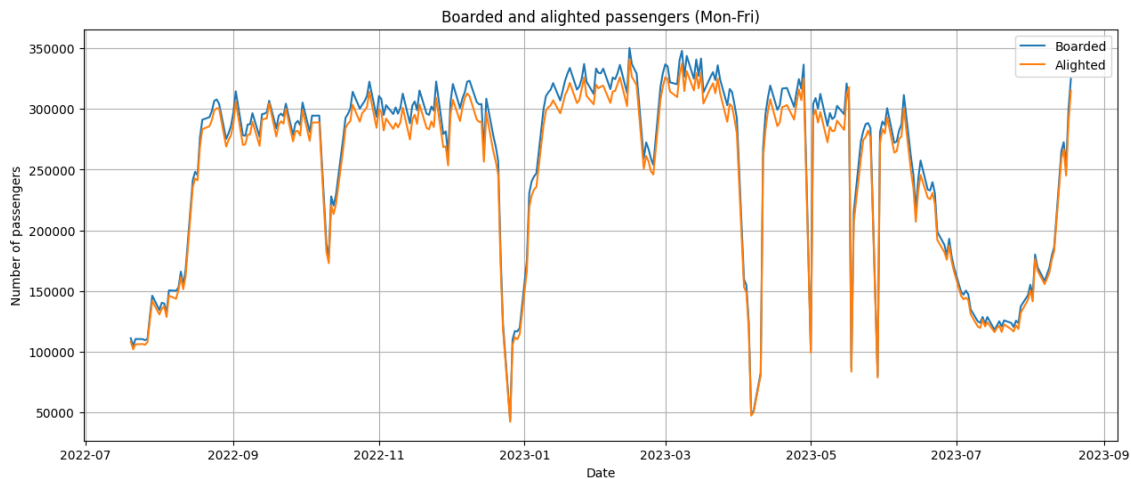


FIGURE 10. Boarded vs. Alighted passengers, only weekdays (Mon-Fri).

The dip in April was probably because of Easter. In Norway, in addition to the weekend, three days around Easter are public holidays and many Norwegians have additional days off work during Easter. In May there were several public holidays that also show in the number of passengers. The 1<sup>st</sup> of May was Labour Day, the 17<sup>th</sup> of May was Constitution Day, the 18<sup>th</sup> of May was Ascension Day and 29<sup>th</sup> of May was Pinse (also known as Whit Monday or Pentecost). These were all public holidays. (Norway Public Holidays, 2023)

#### 4.4 Preparing data for linear regression

When preparing a dataset for linear regression, it was necessary to remove weekends, public holidays, and common holiday periods from the data. Linear regression expects that a straight line can be drawn to illustrate the correlation of two variables (Nield, 2022). When people do not have to use buses to commute, this linear correspondence is broken, and linear regression would not fit on the data. FIGURE 23 shows how badly linear regression fits to data that includes all days.

Public holidays in May are clearly visible in FIGURE 11. The figure shows the boarded passengers from May 2023, Saturdays and Sundays are included.

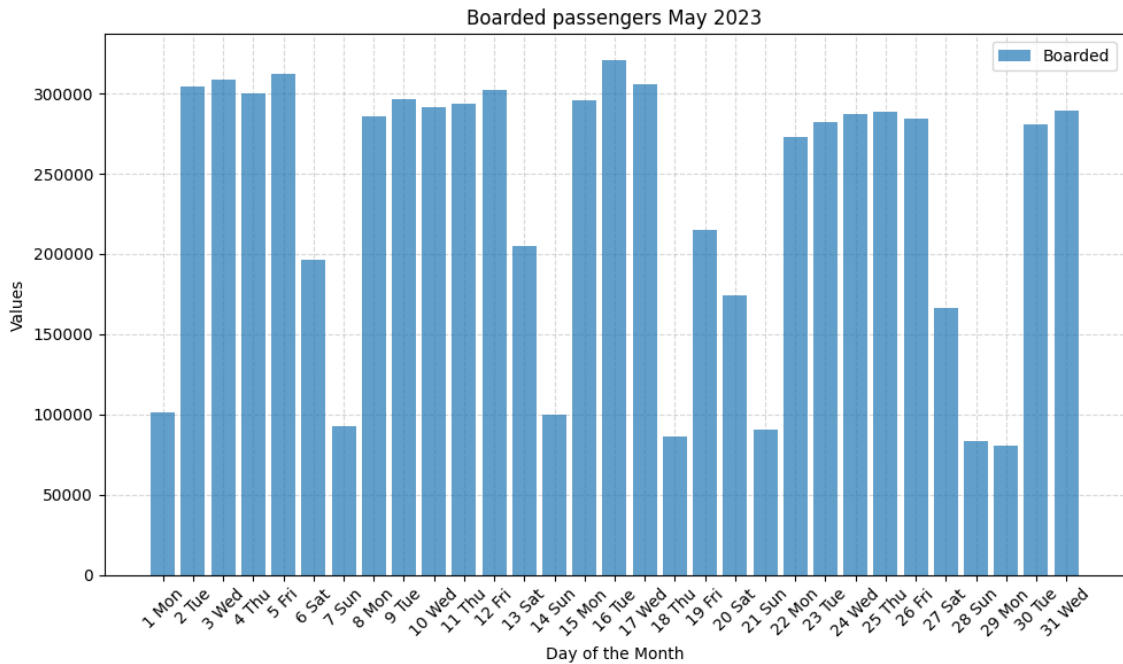


FIGURE 11. Boarded passengers in May 2023.

Information about the day of the week and public holidays was added to the dataset using one-hot-encoding. Both the public holidays and weekends were excluded from the data when applying linear regression to the data. For Sklearn classifiers and XGBoost the entire data was used.

When visually inspecting number of passengers sorted by the day of the week, weekdays (from Monday to Friday) are quite close to each other. There are around 300 000 passengers each day. On Saturdays the number of passengers drops to 200 000 and on Sundays to 100 000. When looking at weekdays, there might be a small increase in number of passengers during the winter months, from January to March.

FIGURE 12 shows the number of passengers in August. Presumably summer vacations have affected the numbers until August the 22<sup>nd</sup>. The data before that was discarded from data used for linear regression.

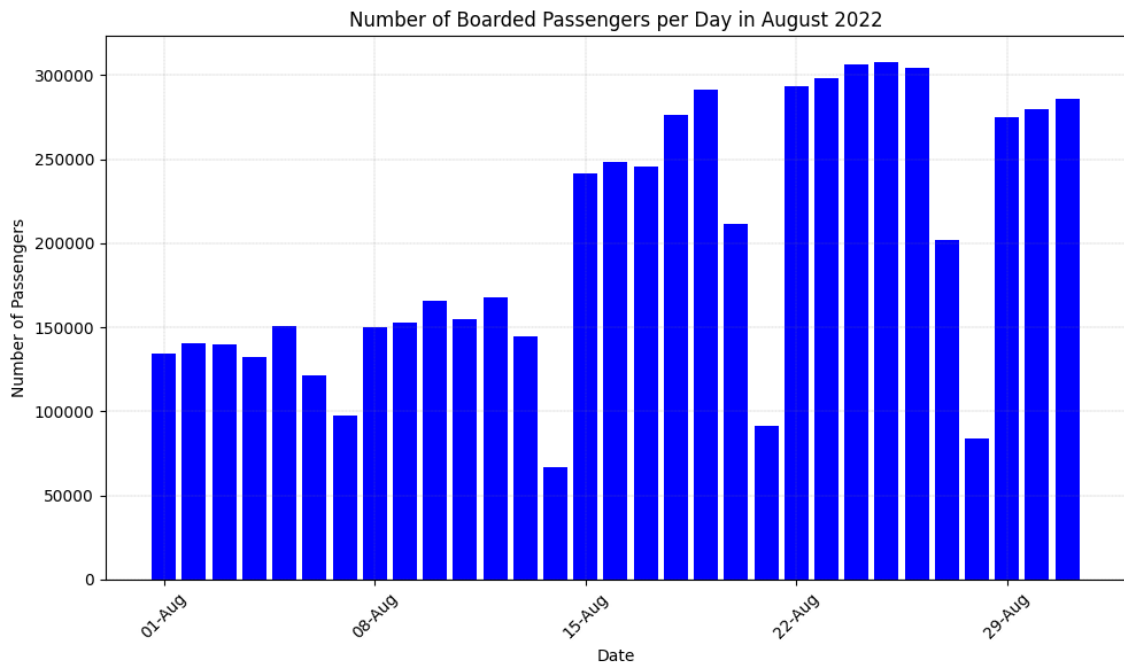


FIGURE 12. Number of passengers in August 2022.

In FIGURE 13 we can see the number of passengers in the month of October in 2022. On Saturdays there is a clear drop in the number of passengers, that is clearly visible every month. On Sundays the number of passengers is even lower. During the week starting from Monday October the 10<sup>th</sup>, the number of passengers is lower compared to other weeks. It was decided that week should be dropped from data used for linear regression.

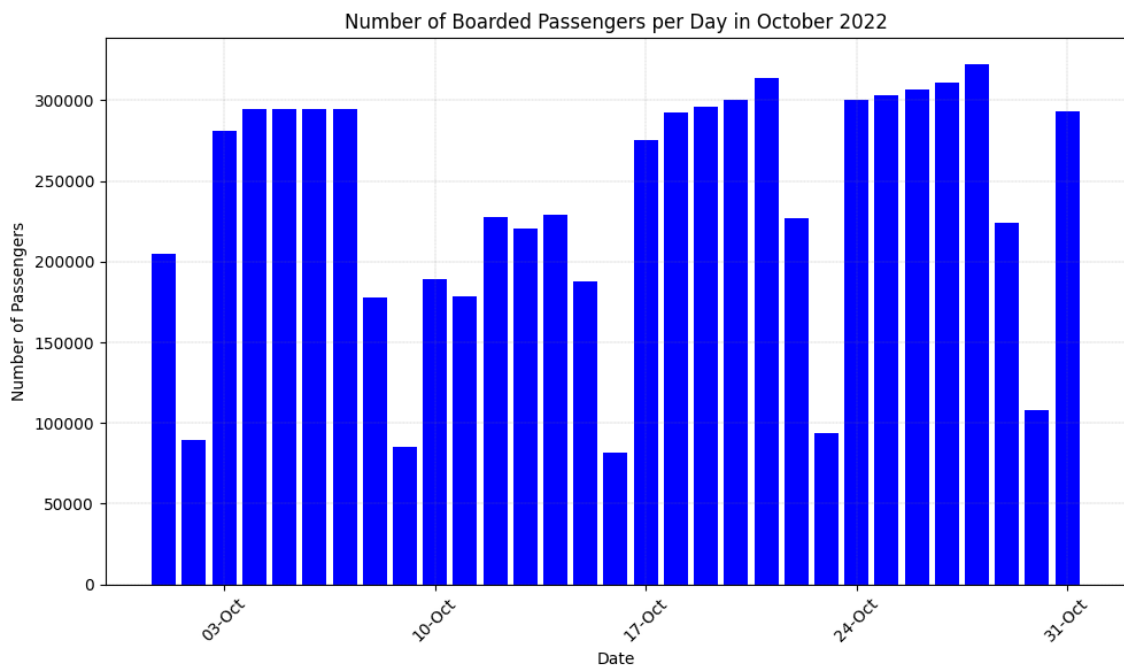


FIGURE 13. Number of passengers in October 2022.



It was noticeable that from October the 4<sup>th</sup> to October the 7<sup>th</sup> the bars visually looked the same. In practice they are the same, but there are small differences. The number of passengers for each of those days is visible in TABLE 7.

TABLE 7. Number of passengers Oct 4-7.

Day	Boarded passengers
October 4 <sup>th</sup>	294 542
October 5 <sup>th</sup>	294 329
October 6 <sup>th</sup>	294 488
October 7 <sup>th</sup>	294 454

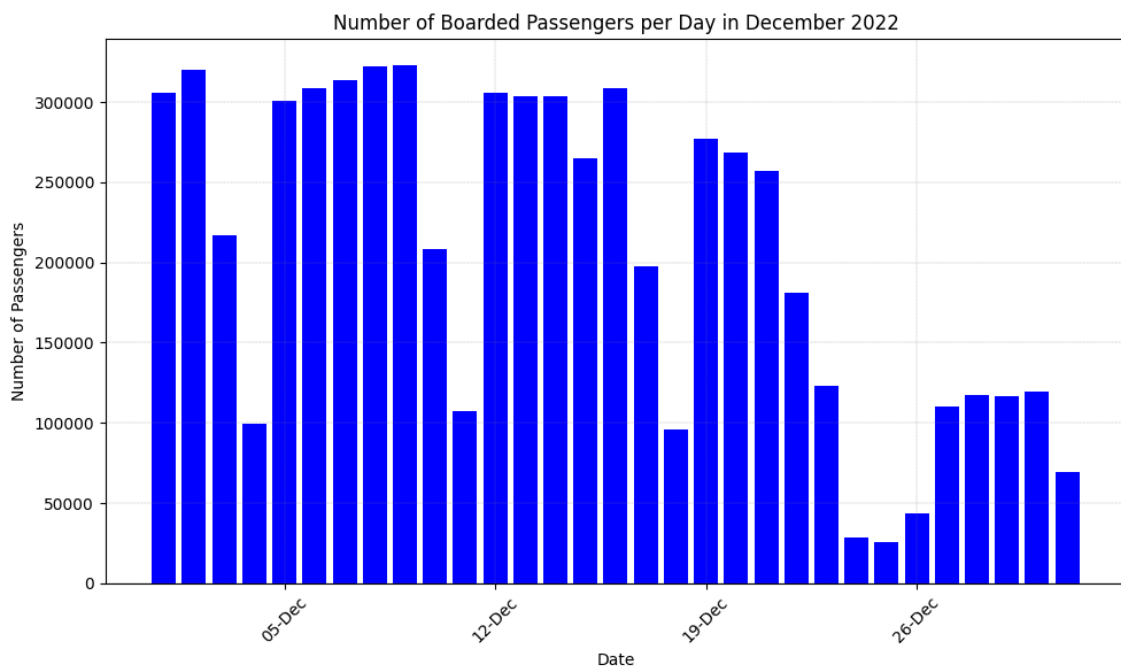


FIGURE 14. Number of passengers in December 2022.

FIGURE 14 shows the number of passengers in December 2022. Here we can see that the number of passengers starts to decrease starting from December the 19<sup>th</sup>. The Christmas vacations continued to the following year as can be seen in FIGURE 15. The numbers began to normalize starting from January the 9<sup>th</sup>. The data between December the 19<sup>th</sup> and January the 9<sup>th</sup> was discarded from the data used for linear regression.

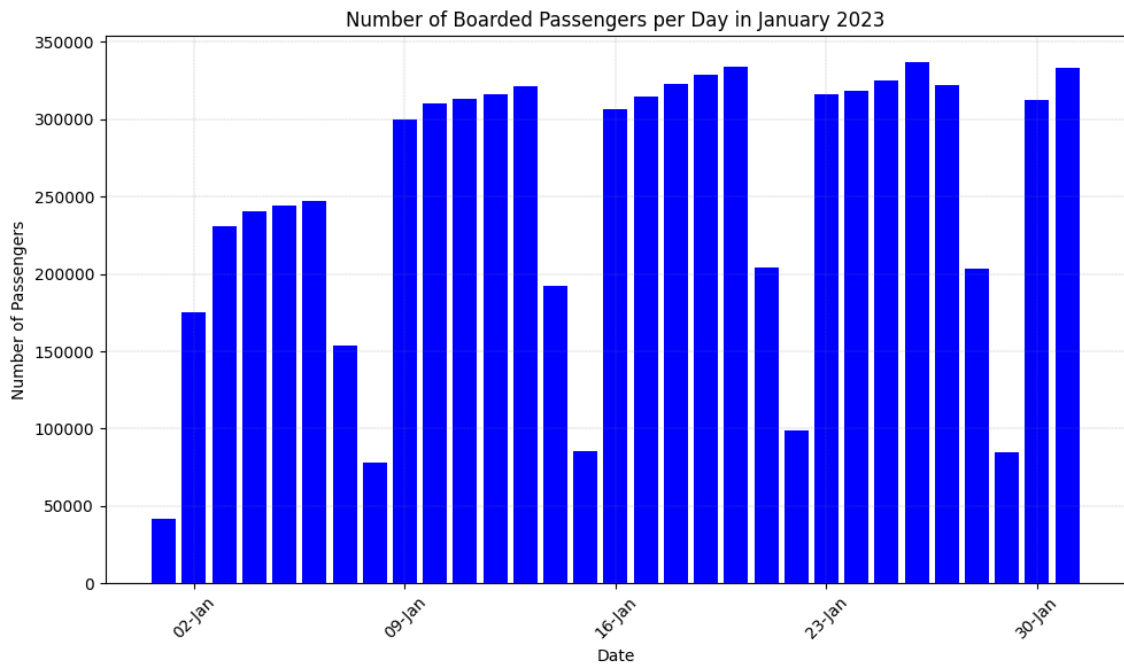


FIGURE 15. Number of passengers in January 2023.

In February there was a slight drop in number of passengers between 20<sup>th</sup> and 24<sup>th</sup> as shown in FIGURE 16. The drop was not that significant, but this week was discarded because counts were abnormal. The drop was caused by winter vacation in schools. (Trøndelag School Holidays 2023 and 2024)

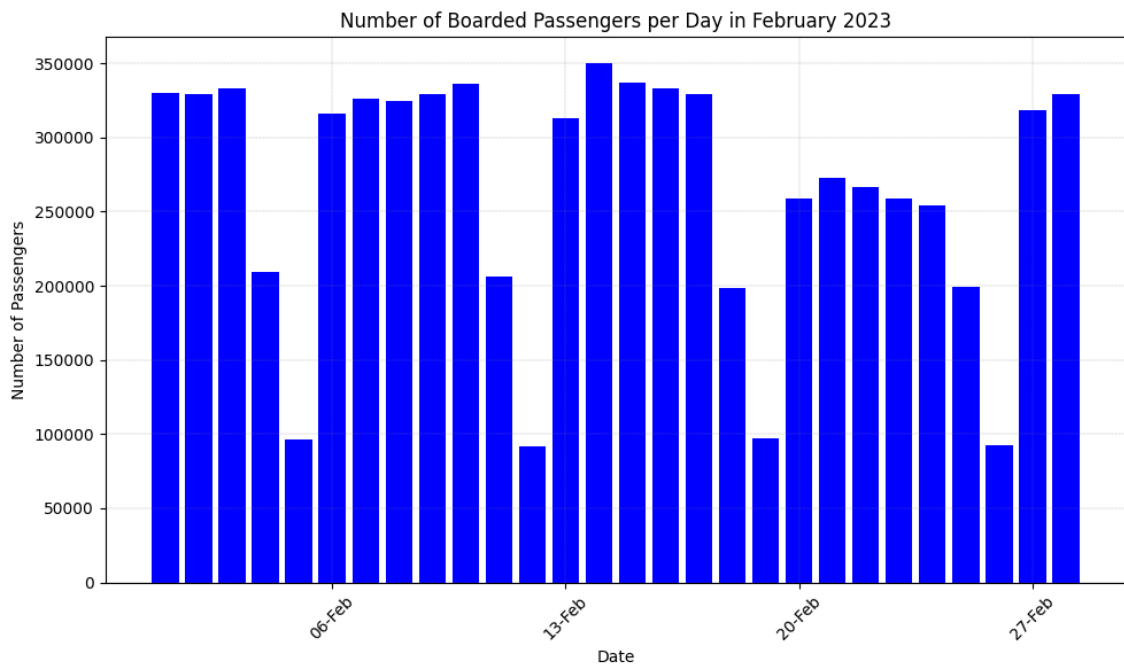


FIGURE 16. Number of passengers in February 2023.

Easter 2023 was at the beginning of April. In addition to the public holidays the Norwegians often take additional days of adjacent to Easter. That period is clearly present in the passenger numbers from April (FIGURE 17). The counts from April the 1<sup>st</sup> to April the 10<sup>th</sup> was discarded.

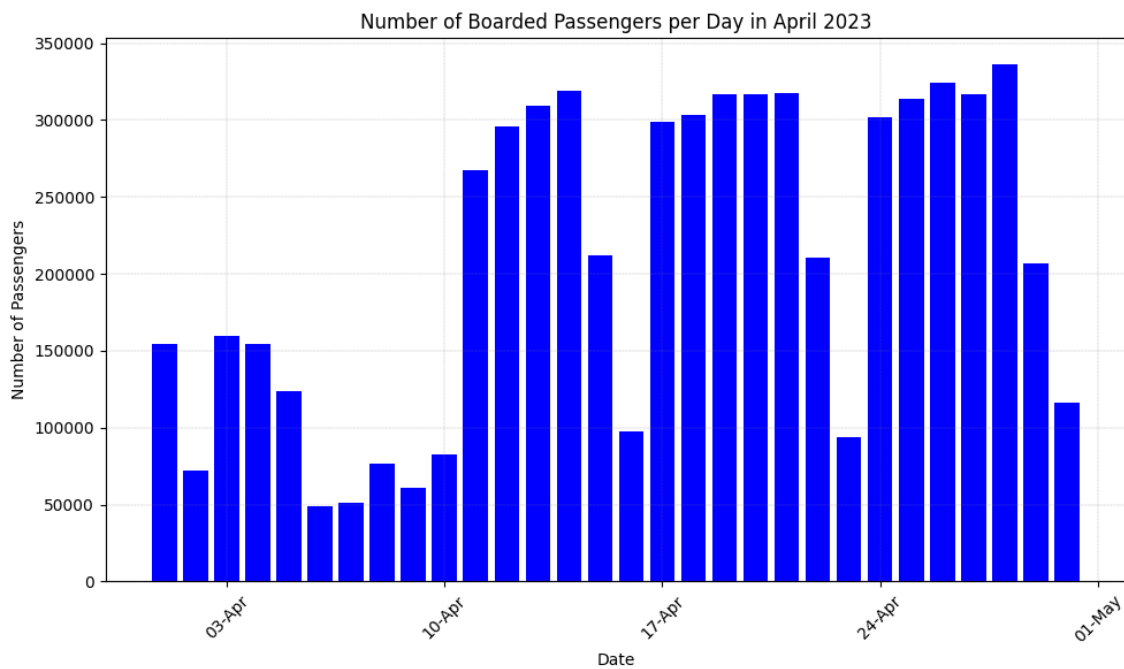


FIGURE 17. Number of passengers in April 2023.

The passenger graphs for all months can be found in appendices.

#### 4.5 Preparing data for machine learning models

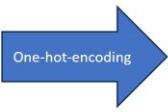
If the lower number of passengers on weekends, holidays and public holidays cause problems for linear regression models, machine learning models do not suffer from such issues. Machine learning models can learn that on Saturdays the number of passengers is different than on any given weekday, for example. The data had to be encoded to fit the machine learning models.

When applying machine learning, data must be prepared accordingly to fit the model in question. This chapter explains how data was prepared for machine learning models used in this thesis.

##### 4.5.1 One-hot-encoding

One-hot-encoding is an often-used method when processing categorical data. In one-hot-encoding, categorical variables are turned into binary values (0 or 1). This enables machine learning

algorithms to do a better job in classification tasks. In one-hot-encoding each categorical value has its own column. Each column has either a value of 0 (false) or 1 (true) depending on the categorical value of the datapoint. (Al-shehari & Alsowail, 2021)



Date	Precipitation
2023-04-01	snow
2023-04-02	rain, snow
2023-04-03	
2023-04-04	rain

Date	Rain	Snow	Freezing rain
2023-04-01	0	1	0
2023-04-02	1	1	0
2023-04-03	0	0	0
2023-04-04	1	0	0

FIGURE 18. Example of one-hot-encoding.

#### 4.5.2 Standard scaling

The nature of the different features can vary a lot. For example, the temperature values can have negative or positive values like -15 °C during winter, or +25 °C during summer. For example wind speed is often expressed within a single digit and humidity is presented in percentage.

The difference in scales can cause unreliability with many machine learning models. If some feature has an order of magnitude higher variance compared to other features, it can dominate the learning process and cause unreliable results. Scikit-learn transformer called StandardScaling solves this problem. (Scikit-learn documentation)

The standard score of a sample  $x$  is calculated with the following equation:

$$z = \frac{(x - u)}{s}$$

Where  $u$  is the mean of training samples and  $s$  is the standard deviation of the training samples. (Scikit-learn documentation)

#### 4.5.3 Preparing the data

Weekends, public holidays, and vacations were included in the data. Days of the week were one-hot-encoded to the data. One-hot-encoding is a process that is often used to make categorical data suitable for machine learning algorithms. In the final data frame, each day of the week got their own column. Value of the column would be 1 or 0. See chapter 4.5.1 for details of one-hot-encoding.

Considering precipitation there has been rain on significantly many days as can be seen in TABLE 8.

TABLE 8. *Precipitation types and their occurrence.*

Type	Number of days
Rain	184
Snow	62
Rain, snow	52
Rain, ice	1
Rain, freezingrain, snow	1

For the precipitation data to be useful for machine learning, it had to be modified. One-hot-encoding was used as shown in TABLE 9.

TABLE 9. *A snippet of the one-hot-encoding of precipitation type.*

Date	Rain	Snow	Ice	Freezing rain
2022-11-22	1	0	0	0
2022-11-23	0	1	0	0
2022-11-24	1	1	0	0
2022-11-25	1	1	0	1
2022-11-26	1	0	1	0
2022-11-27	0	0	0	0

Public holidays were encoded by simply adding a column that either has value 1 if it is a public holiday, and value 0 if it is not a public holiday. Common holidays were not encoded in the data.

For classification models, the category column was created. The category was created based on the number of passengers. Two different scales for categorizing the number of passengers were used. In the first one, the number of passengers was split into four categories (TABLE 10). In the second one, five categories (TABLE 11). Using eight categories (TABLE 12) was also tested, but it was quickly abandoned because it was not accurate enough. All the categories were tested using

MLPClassifier. With MLPClassifier, using four categories was the most accurate. Therefore, only four categories were tested using XGBoost.

TABLE 10. Four categories. With XGBoost categories had to be labeled ranging from 0 to 3. XGBoost categories are in brackets.

Category (in XGBoost)	Number of passengers	Occurrences in data
1 (0)	<150 000	116
2 (1)	150 000-250 000	89
3 (2)	250 000-300 000	81
4 (3)	>300 000	111

TABLE 11. Five categories.

Category	Passengers	Occurrences in data
1	<100 000	63
2	100 000-150 000	53
3	150 000-200 000	44
4	200 000-300 000	126
5	>300 000	111

TABLE 12. Eight categories.

Category	Passengers	Occurrences in data
1	<100 000	63
2	100 000-150 000	53
3	150 000-200 000	44
4	200 000-250 000	45
5	250 000-280 000	28
6	280 000-300 000	53
7	300 000-320 000	65
8	>320 000	46

To minimize the trouble caused by the different scales of the data features like temperature, wind speed and snow depth, standard scaling was used. More details about standard scaling in chapter 4.5.2.

The data for machine learning ranged from July the 20<sup>th</sup> 2022 to August the 16<sup>th</sup> 2023.

## 5 ANALYSIS AND RESULTS

The objective of the thesis was to investigate if there was a correlation between the weather and the number of public transport passengers. The idea was to analyze the collected data first by simple linear regression models and if necessary, use machine learning models to predict the number of passengers with reasonable precision.

Because of the different characteristics of the models, different datasets were used for regression analysis and machine learning models.

### 5.1 Visual observations from the data

FIGURE 19 shows the correlation between temperature and number of passengers. Weekends (Saturday and Sunday) and public holidays have been filtered out. There are still visible decreases in the number of passengers that are most likely caused by popular vacation periods during autumn, Christmas, winter, and summer.

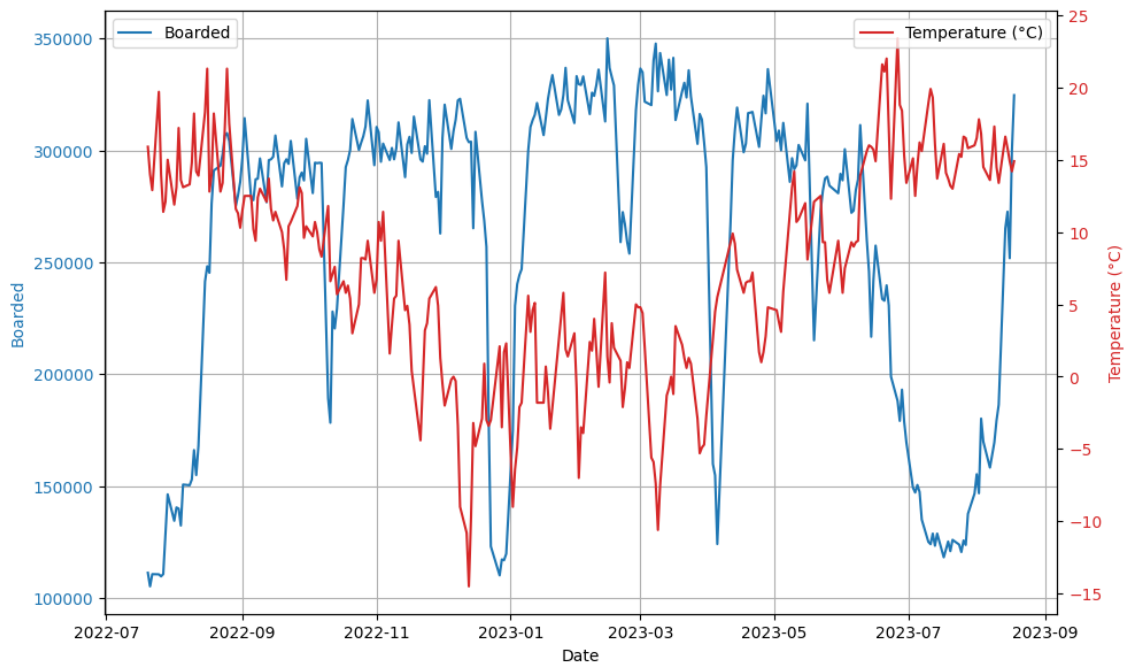
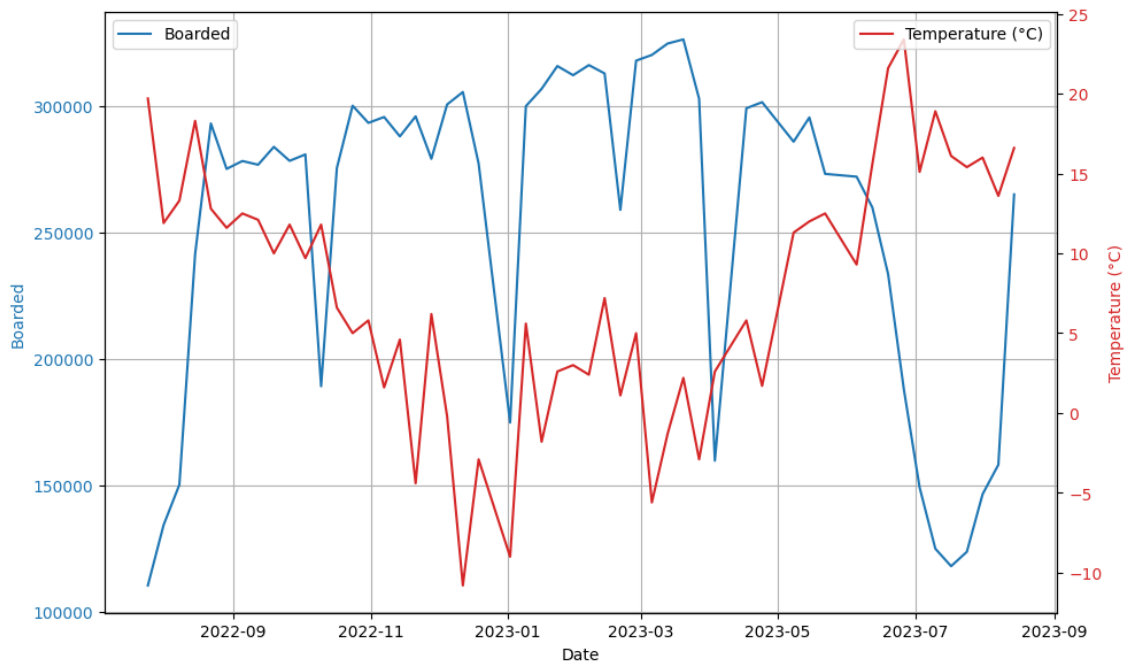


FIGURE 19. Number of boarded passengers and temperatures for the entire period.

With visual inspection there can be seen an increase in passenger numbers to some degree during the colder months.



When looking at passenger and temperature data for each day of the week separately, there was a similar correlation to be found. From Monday to Friday, the number of passengers appeared to be higher during the colder months. For Saturday and Sunday, the effect of the temperature appeared to be less significant. *FIGURE 20* shows the correlation between temperature and number of passengers for Mondays. Same graphs can be found for all days in appendices.



*FIGURE 20. Correlation between number of passengers and temperature on Mondays.*

*FIGURE 21* shows the correlation between precipitation and number of passengers. Temperature is color coded in the graph. There seems to be some correlation between the amount of rain and the number of passengers. When it rains, the number of passengers goes up, especially, when it is cold. It can be assumed that when the temperature is low, the precipitation is snow. When the temperature is close to zero, precipitation might be wet snow (slush). By visual inspection the correlation is not strong.

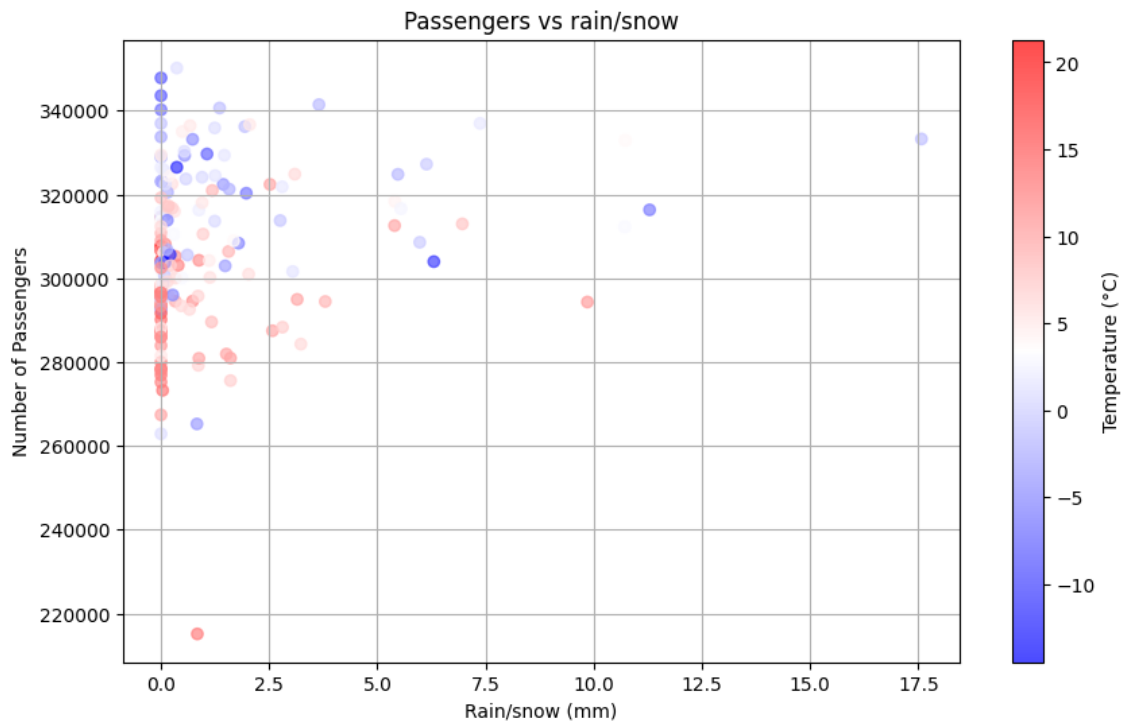


FIGURE 21. Correlation between number of passengers and precipitation. Temperature shown with color code.

## 5.2 Linear regression

A simple linear regression was used to illustrate the correlation between temperature and number of passengers. It is visualized in the scatter plot in FIGURE 22. NumPy polyfit function was used to create a trendline using 1<sup>st</sup> degree polynomial function. Weekends, public holidays, and common holiday periods were discarded from the data used for this analysis. Standard-scaling was applied to the data.

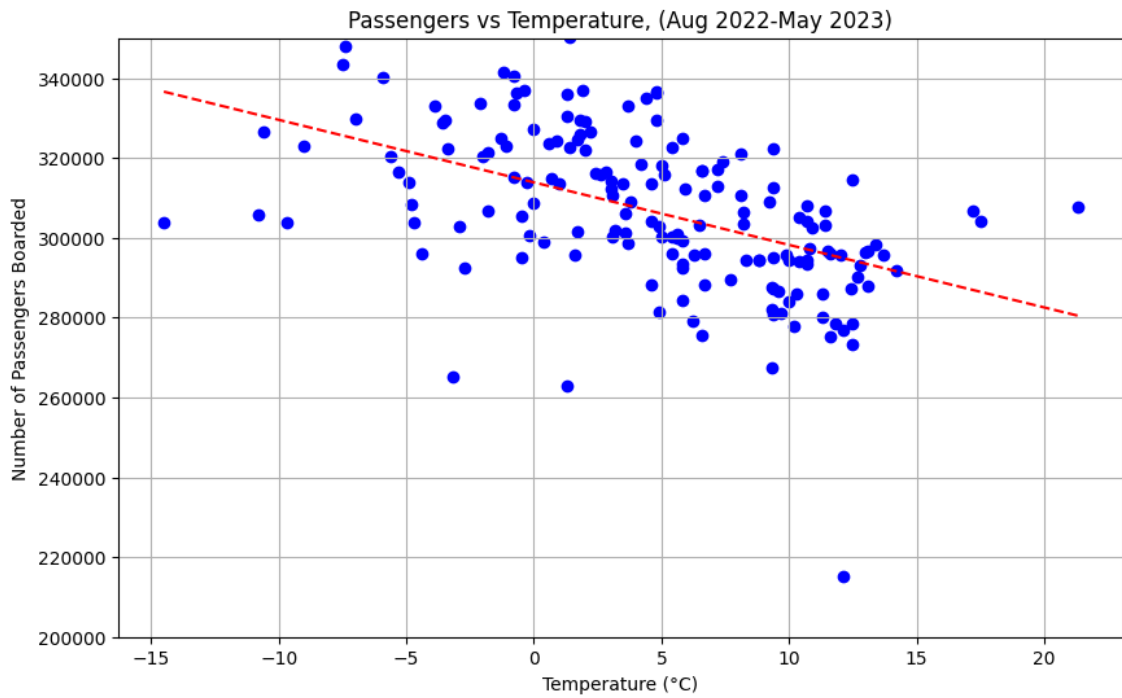


FIGURE 22. Correlation between temperature and number of passengers. Data are from August to May. Weekends, public holidays, and common holidays are removed.

Estimating the goodness of the fit for the first degree polynomial linear regression was done using coefficient of determination,  $R^2$ . The mean of  $R^2$  values for the linear model was 0,25. Such a low value implies that linear regression is not a valid method for predicting the number of passengers based on temperature.

If weekends, public holidays, and vacations are included, the data are scattered in a way that makes linear regression unsuitable for analysis. This is illustrated in FIGURE 23. The  $R^2$  for the entire data was only 0,08. This highlights why it was necessary to remove outliers from the data before using linear regression.

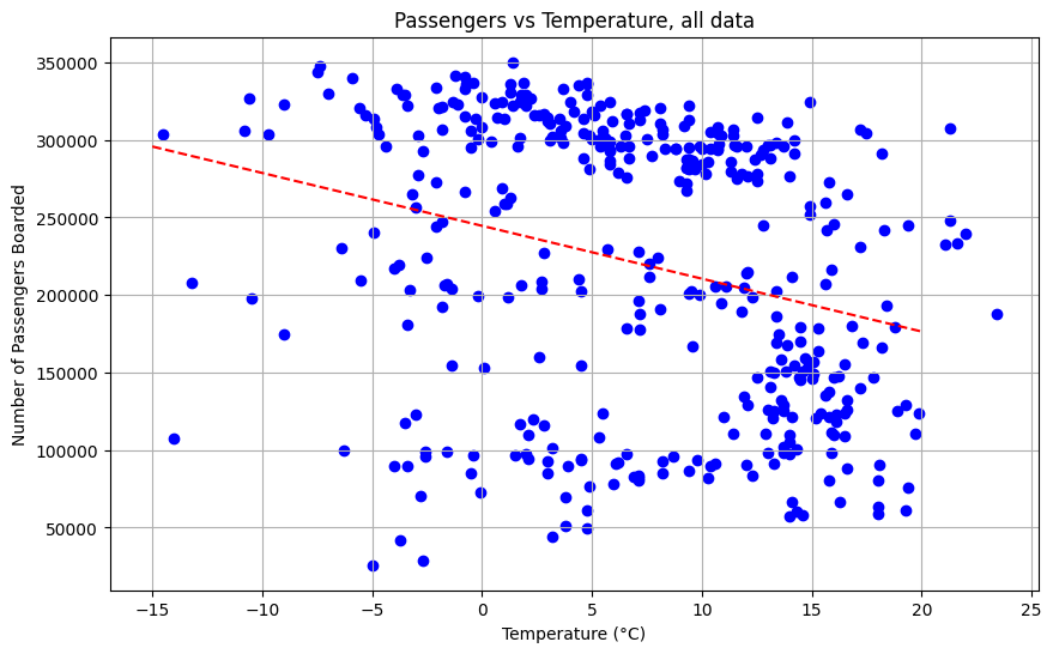


FIGURE 23. Correlation between temperature and number of passengers, all data.

### 5.3 Multivariable regression

#### 5.3.1 Using LinearRegression function

As it seems that simple linear regression with only one variable (temperature) was not a good fit, a multivariable regression was the next logical step to try. For that Python library Scikit-learn and its function LinearRegression was used.

The same data as with single-variable regression was used. Meaning that weekends, public holidays, and common holiday periods were left out. The data had 168 entries, one entry for a day. Multivariable regression is a relatively simple model and based on testing adding weekends and holidays would make the model unreliable. Standard-scaling was applied to the data.

When creating a model, it is important to divide the data into training data and test data. As the names suggest, the training data is used to train the model and testing data is used to test if the model is working as expected. Commonly 20-30% of the data is separated for testing (Tokuç, 2023). It is important to keep the test data separate from the training data. In the multivariable regression test in this thesis, 20% of the total data was separated for test data.

FIGURE 24 shows an example of results got from using the multivariable regression. In this example, the values do not match very well. The differences between the data and prediction can be measured in tens of thousands of passengers at worst. On the other hand, the predictions appear to be in the right direction. In FIGURE 25 is an example of another training and test run, where the results seem to be better than in FIGURE 24. There are big differences in prediction and data in FIGURE 25 as well, but not as many as in FIGURE 24.

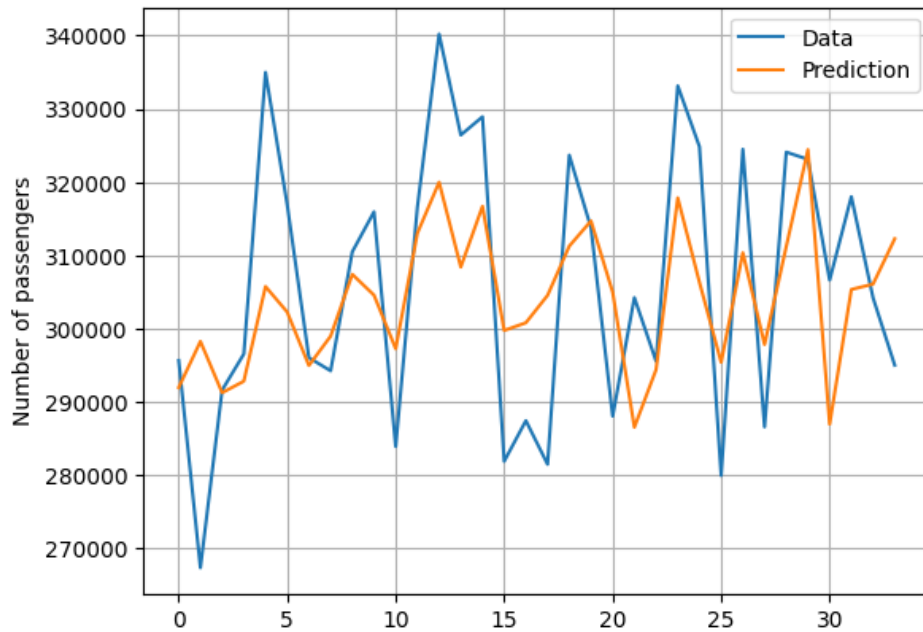


FIGURE 24. Multivariable regression test results, example 1.

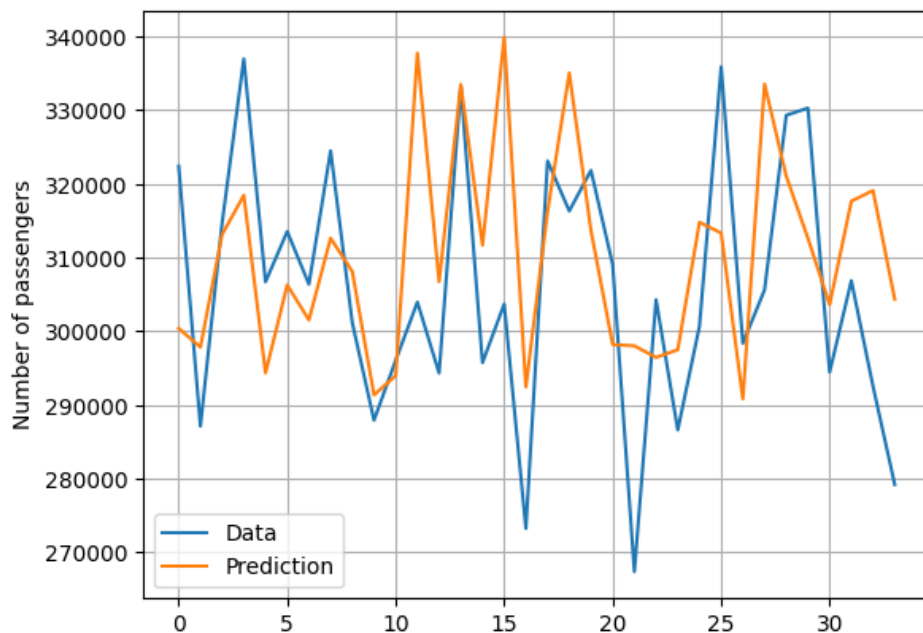


FIGURE 25. Multivariable regression test results, example 2.

Calculating  $R^2$  seems to confirm that the model is performing poorly. Training and testing the model was repeated 10 000 times, using temperature and precipitation as features for the model.  $R^2$  was calculated for each iteration and at the end the mean value was 0.21 that was not any better than with simple linear regression with one variable (temperature). Repeating the cycle several times proved that the average  $R^2$  value remained at around 0.2-0.3.

If more features are added to the model, then the value of  $R^2$  becomes a little better but not significantly. When using the following features: average, minimum and maximum temperatures of the day, precipitation, snow depth, and amount of snowfall, the  $R^2$  rose to the value of 0.30. That is still nowhere near good value, though. FIGURE 26 illustrates the results of a test that had a  $R^2$  of 0.29.

Interestingly, if even more features were added to the model, like cloud cover, wind speed, wind direction, and humidity, there was a decrease in  $R^2$ . After adding those values, the  $R^2$  was 0.26.

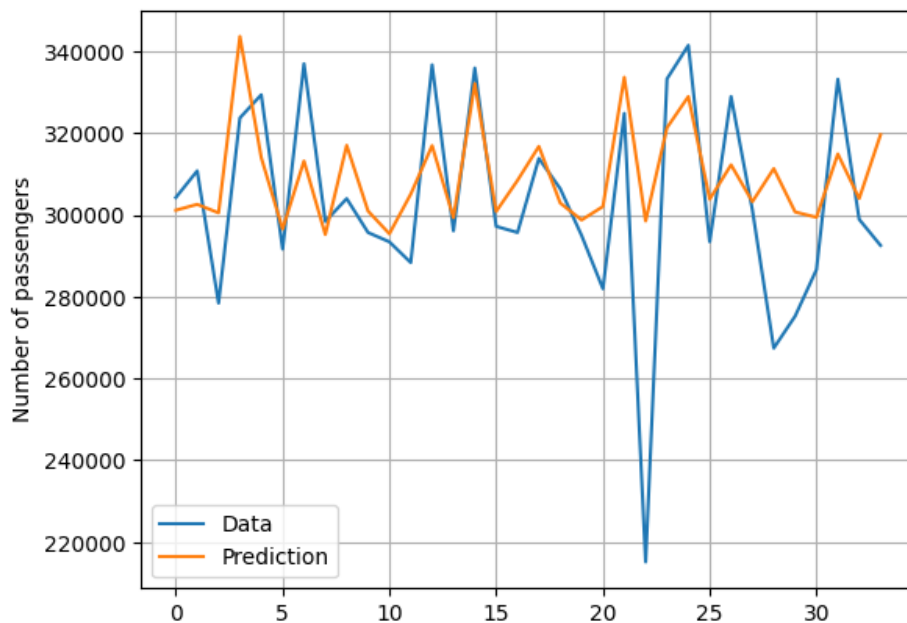


FIGURE 26. Multivariable regression test results, example 3.

TABLE 13 shows the weights of a linear regression model. Weights are calculated by Python ELI5 library. Temperature is the most important value for the predictions. Maximum temperature is the least useful feature for the model.

TABLE 13. ELI5 weights of the linear regression model with 10 features.

Weight	Feature
+307612.939	<BIAS>
+32091.813	temp
+6833.697	snowdepth
+2282.644	humidity
+2200.591	snow
+1890.403	precip
+727.108	windspeed
-719.445	winddir
-3417.354	cloudcover
-9864.841	tempmin
-26257.105	tempmax

If maximum temperature is removed from the model, the weights change dramatically. Now the temperature is the most useless feature, as seen in TABLE 14. This kind of behavior makes the performance of these models questionable and highlight that linear regression does not work well for this data.

TABLE 14. ELI5 weights of the linear regression model with 9 features.

Weight	Feature
+306142.594	<BIAS>
+9390.381	tempmin
+6839.881	snowdepth
+2339.578	snow
+1076.987	precip
+548.649	humidity
+451.507	windspeed
-193.822	cloudcover
-392.796	winddir
-13352.560	temp

TABLE 15 shows the weights of the model used when a smaller set of features was selected. Interestingly, here temperature is also rated to be the least useful feature to the model. Although, it must be noted that  $R^2$  was below 0,30 for all tests made using LinearRegression function, with any number of features used.

TABLE 15. ELI5 weights of the linear regression model with 5 features.

Weight	Feature
+306579.656	<BIAS>
+3691.540	snow
+1542.757	precip
+1048.058	cloudcover
+79.260	windspeed
-8658.768	temp

### 5.3.2 Using Lasso function

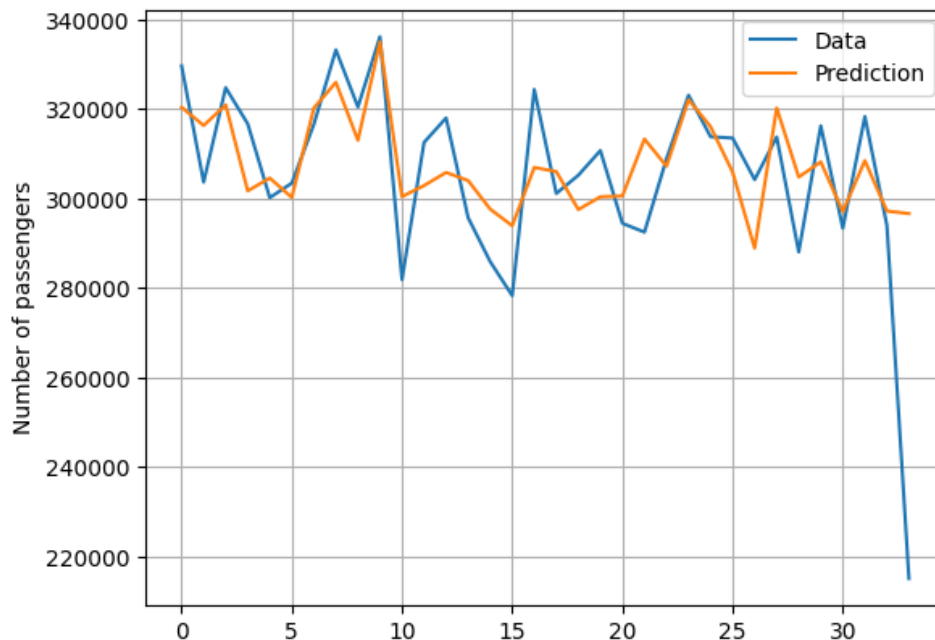


FIGURE 27. Example of predictions using Lasso function.



Using the Scikit-learn Lasso estimator the situation was not any better. When using the same ten features listed in TABLE 13, the  $R^2$  was 0,27. TABLE 16 shows the weights of the model. The weights behaved in a similar way as they did with linear regression. If maximum temperature was removed from the model, then the weight of the temperature would drop dramatically. Here, the snow depth was the most influential feature, temperature was only fourth on the list. Otherwise, the order of the weights was nearly the same as it was with linear regression. FIGURE 27 shows that the predictions were not completely rubbish with Lasso function either.

TABLE 16. *ELI5 weights of the Lasso function with 10 features.*

Weight	Feature
+7191.786	snowdepth
+4427.613	tempmin
+2847.719	snow
+2838.010	temp
+1990.842	precip
+1326.153	humidity
-208.429	windspeed
-630.109	winddir
-2183.129	cloudcover
-10481.289	tempmax

When using the same five features listed in TABLE 15,  $R^2$  was 0,22. Weights were like the ones when using the LinearRegression estimator. Weights are listed in TABLE 17.

TABLE 17. *ELI5 weights of the Lasso function with five features.*

Weight	Feature
+305882.779	<BIAS>
+2993.338	snow
+1866.090	precip
+1376.497	cloudcover
+891.528	windspeed
-8965.389	temp

## 5.4 Creating a machine learning model

The common purpose of a model is to predict values for a specific variable based on input variables. A variable that the model should predict is referred to as a y-variable. The variables that will be fed to the model and used in predicting the y-variable, are called x-variables. (Myatt & Johnson, 2014)

A good way to build a model would be to use all the original data as training data set for the model. For a test set, a new set of data should be collected. They also point out that often that is not possible, but the same data set is used to iteratively train the model. The method is called cross-validation. (Myatt & Johnson, 2014)

### 5.4.1 Classification

Predicting the exact number of passengers based on weather forecasts did not provide very good results. It probably was not realistic to expect it to be possible to start with. Perhaps multilabel classification would provide better results. The number of passengers should be divided into classes or categories. Then classification methods could be used.

Splitting the number of passengers into categories improved the results. Still the accuracy was only average, but the results would provide a good base for further development of the mode.

### 5.4.2 Finding the best hyperparameters using GridSearchCV

The machine learning mode to be tested first was MLPClassifier. A key problem when building a machine learning model is what hyperparameters would get the best results. Scikit-learn provides a tool for it called GridSearchCV. GridSearchCV is a part of the Scikit-learn library. The user provides the parameters that GridSearchCV will test as well as value ranges it will test.

For example, to find out the best structure for the hidden layers, the following code was provided to the GridSearchCV:

```
'hidden_layer_sizes': [(n_neurons,) for n_neurons in range(5, 16, 5)] +
    [(n_neurons_layer1, n_neurons_layer2)
     for n_neurons_layer1 in range(5, 16, 5)
     for n_neurons_layer2 in range(5, 16, 5)] +
    [(n_neurons_layer1, n_neurons_layer2, n_neurons_layer3)
     for n_neurons_layer1 in range(5, 26, 5)
     for n_neurons_layer2 in range(5, 26, 5)
     for n_neurons_layer3 in range(5, 26, 5)],
```

The code will first try a simple neural network with one hidden layer. The GridSearchCV function will first try with 5 neurons, then with 10 and finally with 15. Then GridSearchCV will move on to test the second hidden layer. At first it will use 5 neurons in the first hidden layer and change the neurons in the second hidden layer: 5, 10 and 15. Then 10 neurons in the first hidden layer and change the neurons in the second hidden layer: 5, 10, 15. GridSearchCV will go through all the possible combinations where the first hidden layer consists of 5, 10 or 15 neurons and the second hidden layer consists of 5, 10 or 15 neurons.

When testing the combinations for a model with three hidden layers, GridSearchCV was programmed to use 5, 10, 15, 20 and 25 neurons for each layer.

The final model consisted of three layers that had 5, 15 and 5 layers, respectively. Activation function used was relu (Rectified Linear Unit) and maximum iterations value was 10 000. Rectified Linear Unit is an activation function that will zero out all negative values.

### 5.4.3 Classification using MLPClassifier

MLPClassifier was the first machine learning model to be tested. First, MLPClassifier was tested with five categories that are presented in TABLE 11. After running tens of test rounds, the MLPClassifier model seemed to struggle to predict values for one of the categories, usually the middle one (third category). Also, between different test runs there was quite a lot of variation in the results. With training data, the accuracy was 0,67 and with test data 0,50, which would indicate that the model is slightly overfitting. Accuracy was measured using Scikit-learn accuracy\_score function.

Even if the accuracy was average, the extremes were caught reasonably well by the model, as seen in the confusion matrix in FIGURE 28. In that figure, the dataset with five categories (TABLE

11) was used. The figure shows that categories 1, 4 and 5 are predicted with the best accuracy. Although, with categories 4 and 5 there are some mistakes, the offset is only one category.

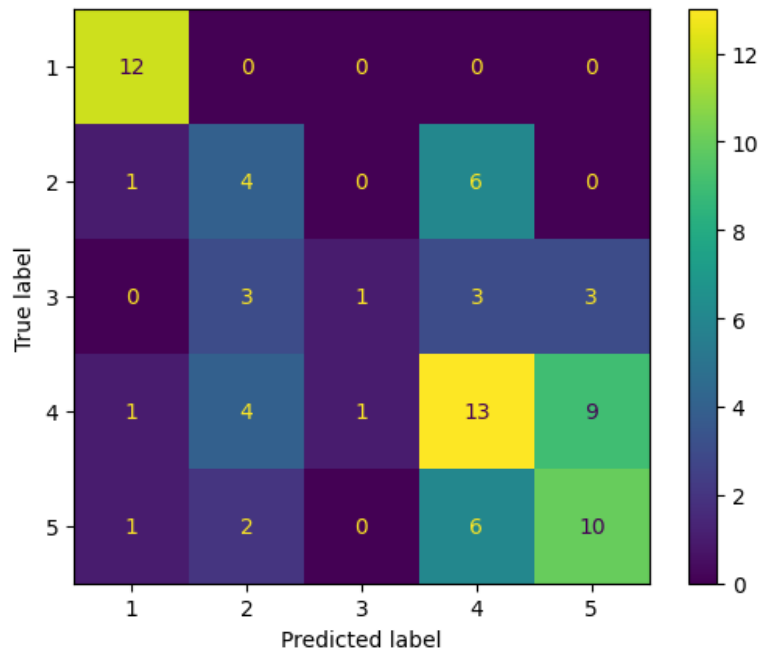


FIGURE 28. Confusion matrix when using five categories.

Using MLPClassifier, even with eight different categories, the extreme ends of the confusion matrix were clear. The samples of each category were small when using eight categories. Also here, the extremes are predicted well. With the middle categories (3-6), the model struggles, as seen in FIGURE 29. Accuracy with training data was 0,69 and with test data 0,48, indicating overfitting.

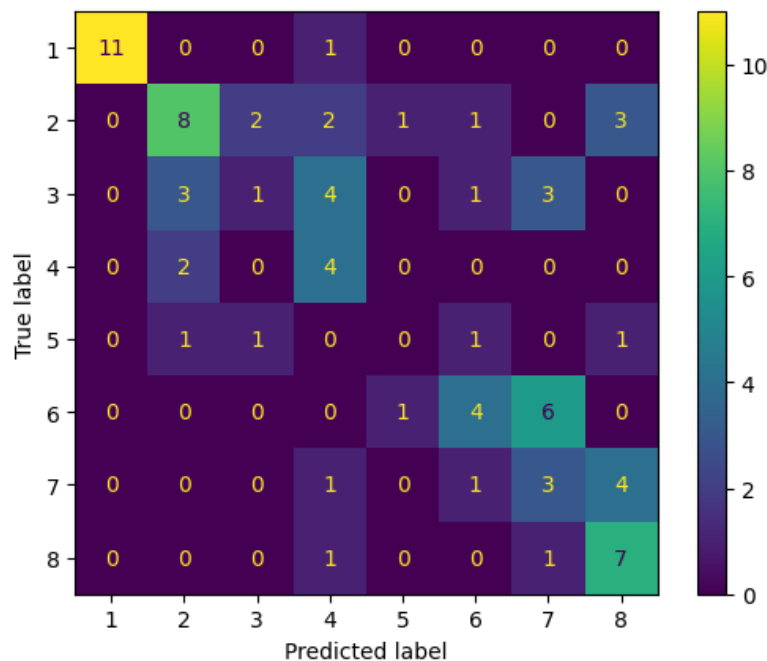


FIGURE 29. Confusion matrix when using eight categories.

When only four categories were used, the problem with overfitting was not an issue. Accuracy with the training data was 0,70 and with test data 0,78.

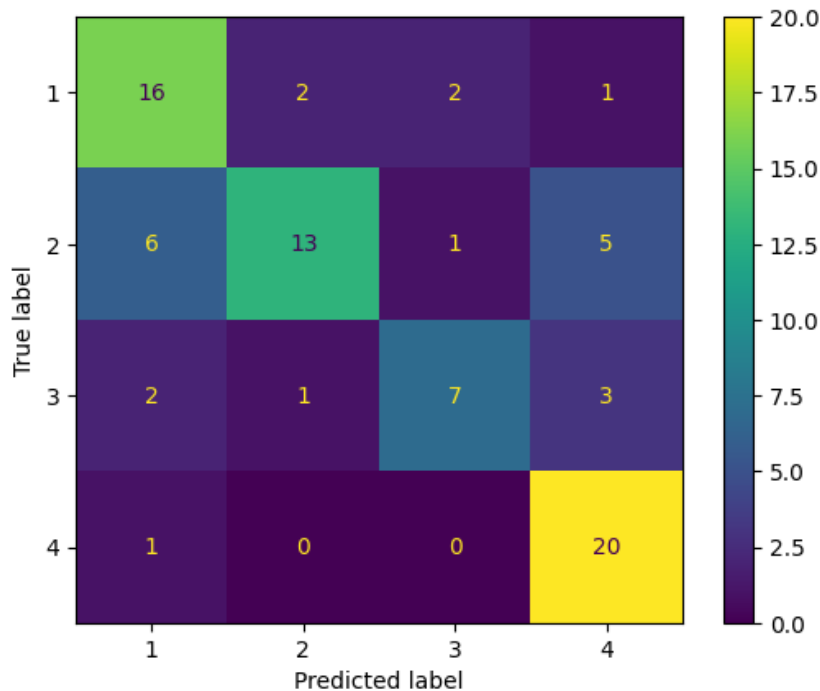


FIGURE 30. Confusion matrix when using four categories and MLPClassifier.

FIGURE 30 displays the confusion matrix when four categories were used. Here the predictions are clearly getting better as the sample sizes in each category are getting bigger.

#### 5.4.4 Classification using XGBoost

For XGBoost, the GridSearchCV was used to find the best hyperparameters.

With XGBoost first the same four categories were used as in TABLE 10. When using XGBoost and four categories, the accuracy of the model on training data was 0,67 and on test data 0,63. The decision tree is displayed in FIGURE 31. The tree is very simple, which indicates that there are only a couple of important parameters in the data. The SHAP values displayed in FIGURE 32 highlight this as well. The labeling of the categories had to be changed to range from 0 to 3 instead of 1 to 4, because of technical differences in XGBoost compared to MLPClassifier.

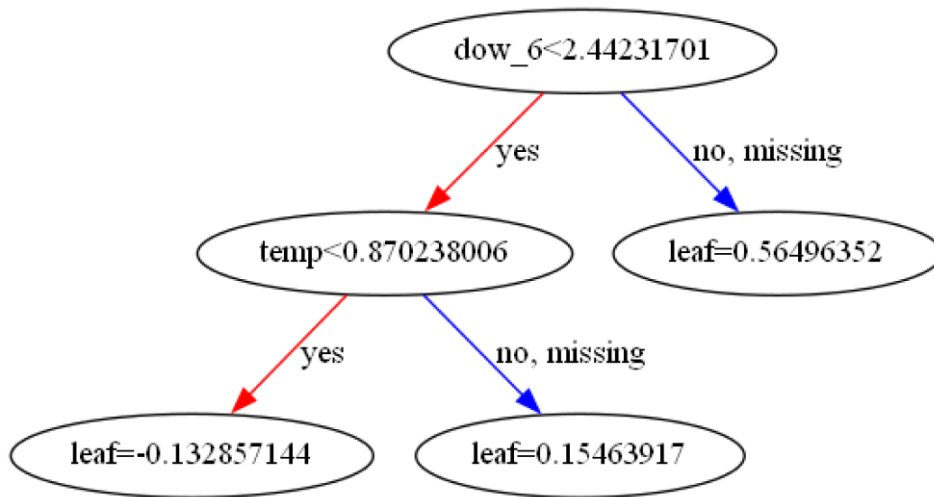


FIGURE 31. XGBoost decision tree with four categories.

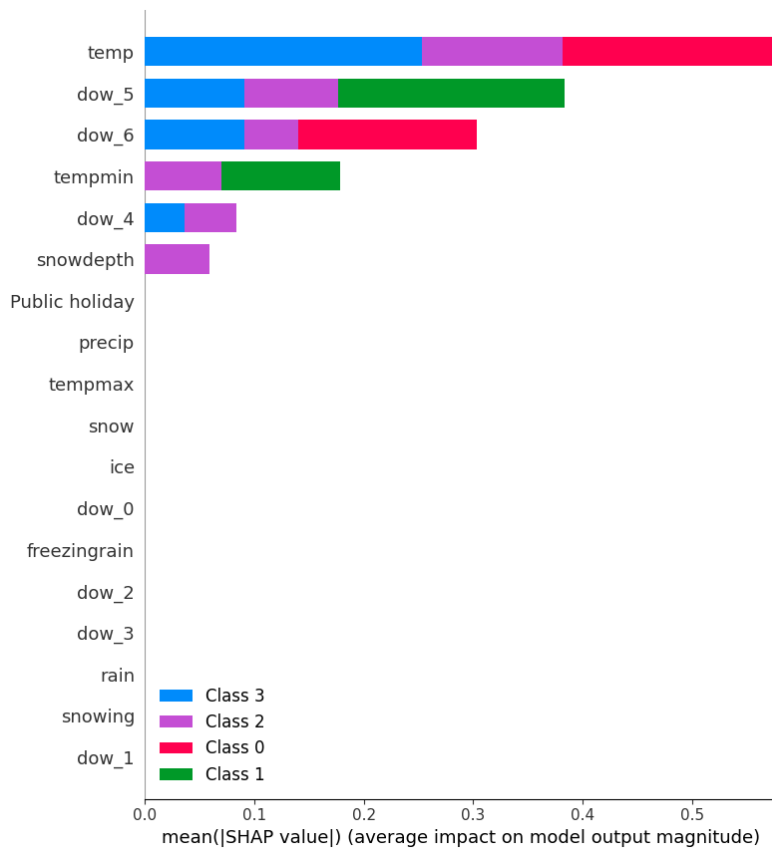


FIGURE 32. XGBoost SHAP values with four categories.

K-fold CV average score was calculated when using four categories and it was 0.62.

If the five categories described in TABLE 11 were used, the decision tree got much more complicated. It, and the SHAP values indicate that there were more meaningful features than when using the data split into four categories. Here the model was overfitting more than when using data

with four categories. Accuracy on training data was 0,79 and on test data 0,58. The first decision was to see if dow\_6 was true or not.

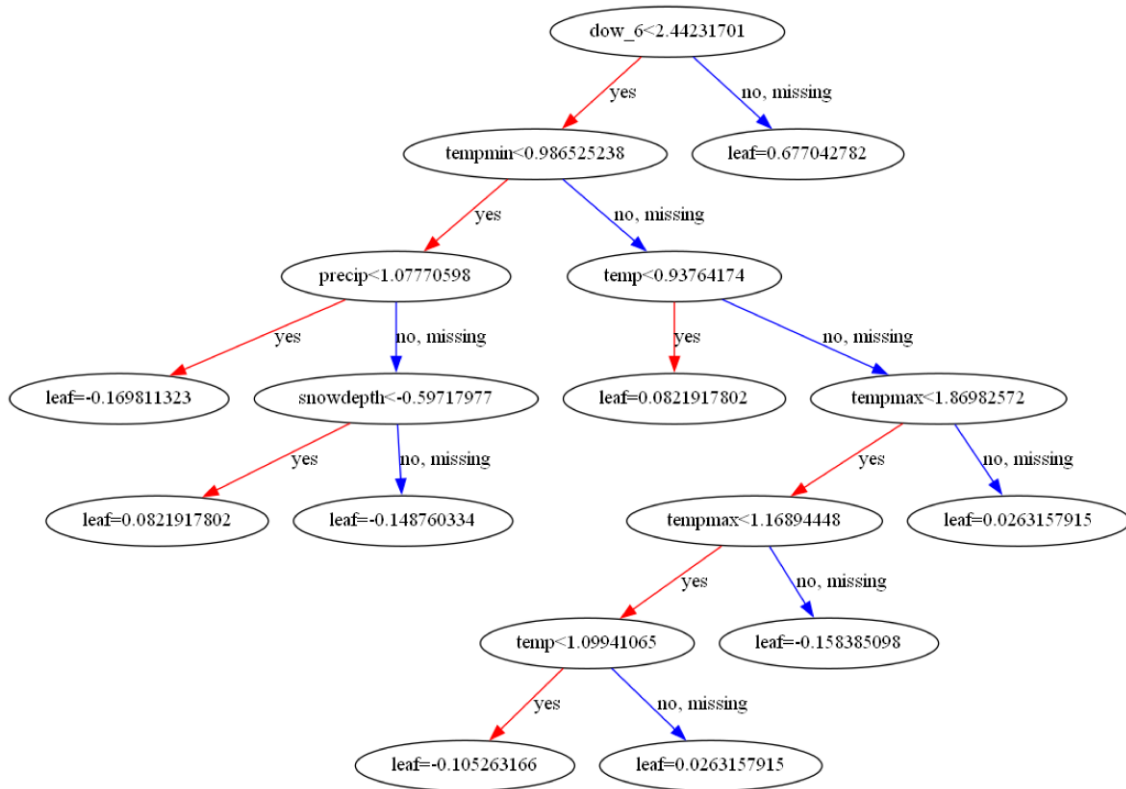


FIGURE 33. Decision tree when using five categories.

When looking at the SHAP values, shown in FIGURE 34, the snow depth was more important than any temperature values. When using four categories, snow depth was only the sixth most important value and it was not in the decision tree at all. The K-fold average score was 0,58.

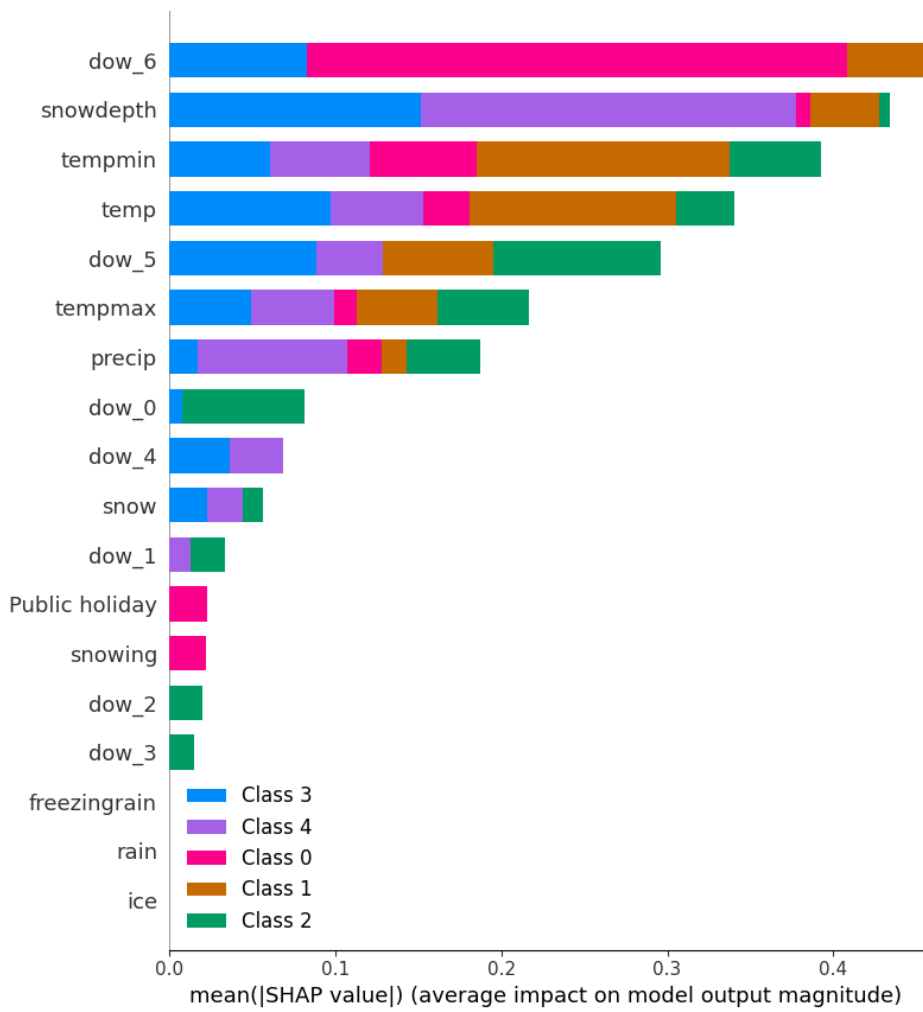


FIGURE 34. SHAP values when using five categories.

### 5.4.5 XGBoost with filtered data

The data used for the machine learning models above contained all the public holidays, weekends, and common holidays, like summer holidays. Especially the summer holidays can affect the results, because during the summer weather is warm and people are on holiday. This obviously reinforces the cause-and-effect relationship between temperature and number of passengers.

To mitigate this, XGBoost was ran on the filtered data as well. The filtered data was the same used for linear regression. The process of data filtering is explained in Chapter 4.4. One-hot-encoding was used for days of the week and different types of rain, as explained in chapter 4.5.



Problem with the data is that the number of data points is low. There are only 168 rows in the database when public holidays, common holidays and weekends are removed. This might cause unreliability in the model.

The test was conducted on four categories and five categories. The same categories as before were not suitable, because the sizes of the categories would be unbalanced. For example, the categories with the least passengers would not have any occurrences in the data. Therefore, the categories had to be altered as shown in TABLE 18 and TABLE 19.

TABLE 18. Four categories, for filtered data.

Category	Passengers	Occurrences in data
0	<295 000	43
1	295 000-305 000	39
2	305 000-320 000	41
3	>320 000	45

TABLE 19. Five categories, for filtered data.

Category	Passengers	Occurrences in data
0	<280 000	13
1	280 000-295 000	30
2	295 000-305 000	39
3	305 000-320 000	41
4	>320 000	45

As before, standard scaling and K-fold were used for the model also in this case.

With four categories, the accuracy on training set was 0,53 and on test set 0,559. K-fold average score was 0,49. Here, the model didn't perform well, but it didn't overfit either.

FIGURE 35 shows the decision tree for four categories. The first decision on the tree is if the temperature is above or below a certain number. This indicates that there might be a correlation

between temperature and number of passengers, even if the warm summertime is excluded from the study. Also, further down in the tree, different temperature values are repeated. Also, the variable `dow_4`, which is Friday (`dow_0` is Monday), plays a role in the decisions.

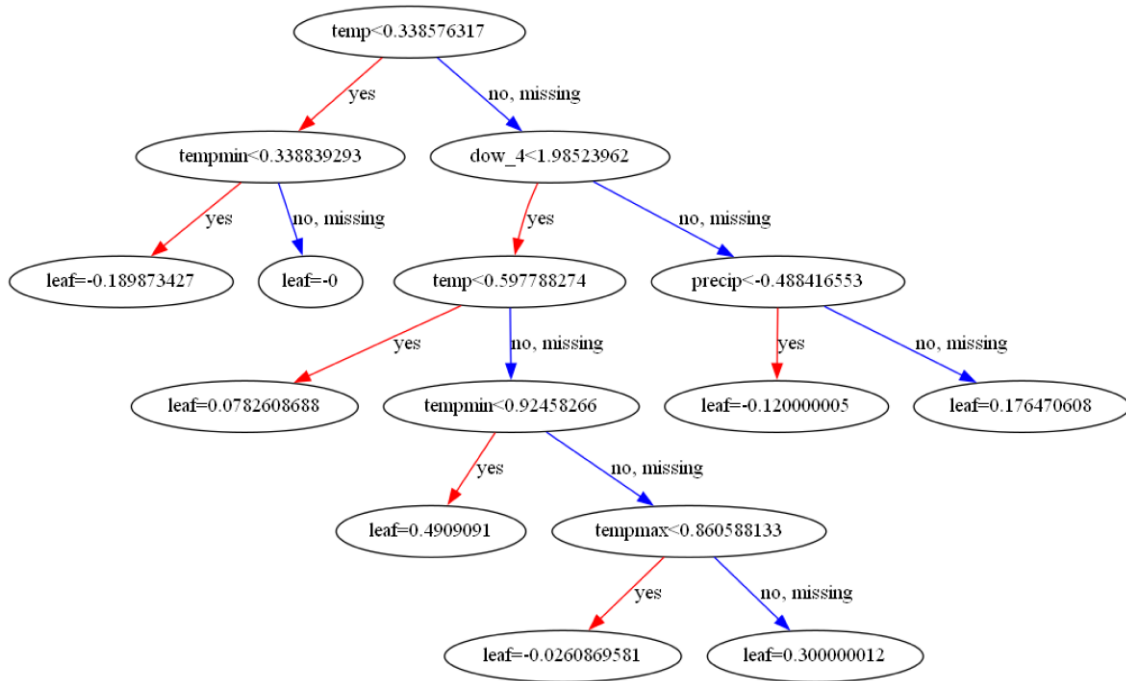


FIGURE 35. XGBoost decision tree, four categories, filtered data.

FIGURE 36 shows the SHAP values with four categories. Here, it is interesting that snow depth plays such a big part when the decision is category 3 (more than 320 000 passengers per day). Maybe this means that during the winter, when the snow depth is at highest, there are more passengers than during other times of the year.

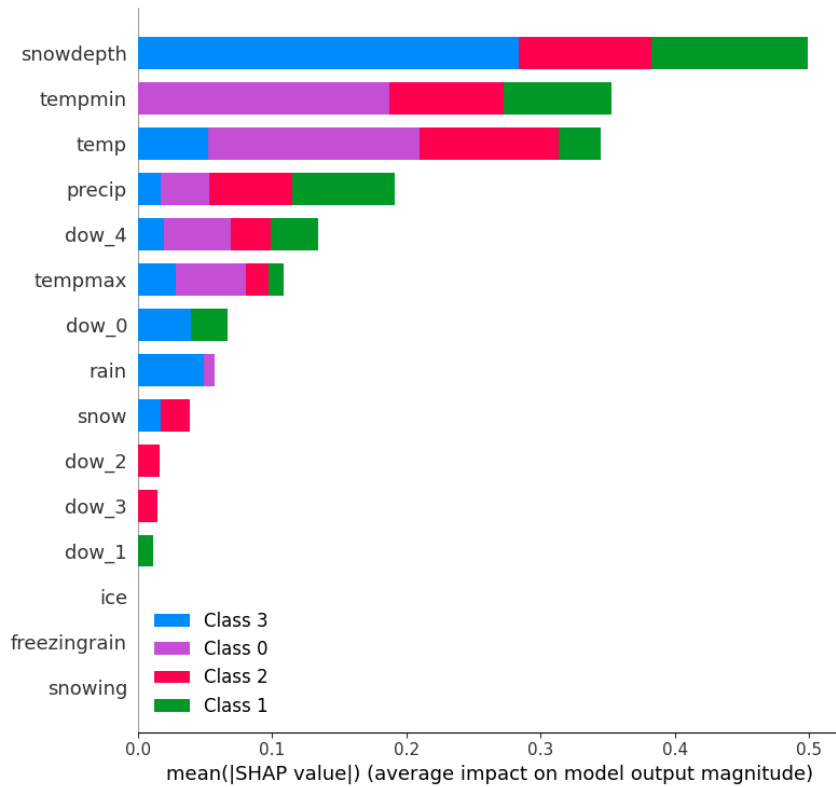


FIGURE 36. XGBoost SHAP values, four categories, filtered data.

With five categories, overfitting was a problem. Accuracy on the training set was 0,716 and on test set 0,529. K-fold average score was 0,54.

FIGURE 37 shows the decision tree when using five categories. Here, the tree is more versatile than it was with four categories. Maximum temperature would be the starting point. Then, Mondays and Fridays also have their nodes. As do precipitation, snow, and temperature.

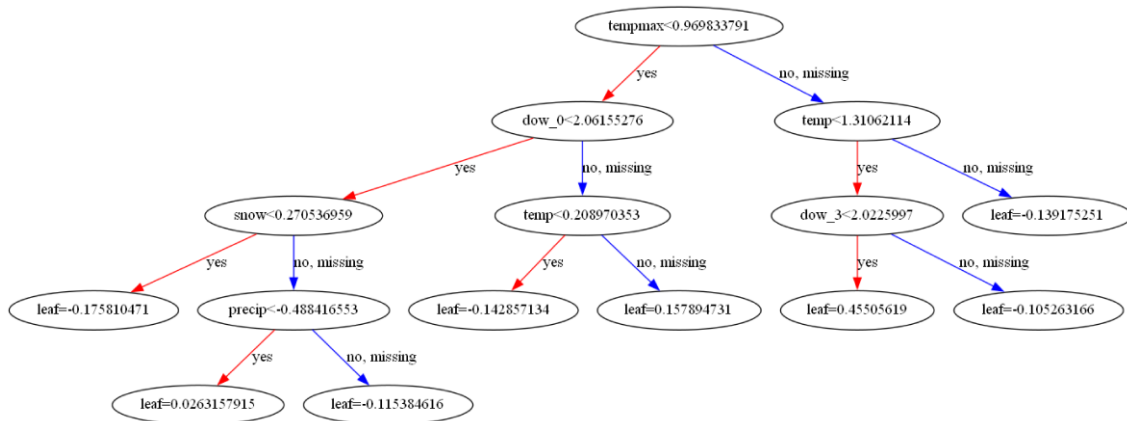


FIGURE 37. XGBoost decision tree, five categories, filtered data.

FIGURE 38 displays the SHAP values when five categories were used. Also here, the snow depth is a major factor when category is 4 (more than 320 000 passengers per day). Minimum temperature seems to be major key when the decision ends up being category 1 (280 000-295 000 passengers per day).

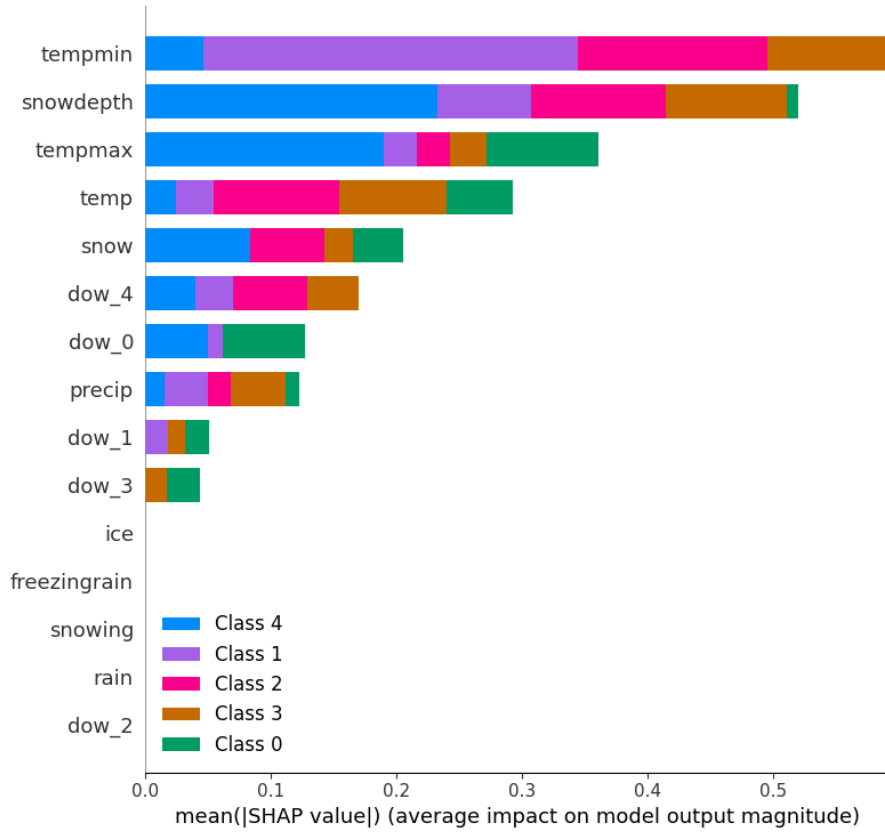


FIGURE 38. XGBoost SHAP values, five categories, filtered data.

## 6 DISCUSSION AND CONCLUSION

The purpose of this thesis was to investigate if there is a correlation between weather and number of passengers in buses in a Nordic city. Passenger data was collected using automatic passenger counting systems integrated into the buses. Weather data was collected from public sources. Target was to create a machine learning model to predict the number of passengers based on weather input.

This study found that there is a correlation between temperature and number of passengers. However, it is not clear if causality exists between them.

The significance of weather seemed to be higher during the weekdays (from Monday to Friday) than on weekends. On Saturdays and Sundays, the impact was not significant. This is in line with the other literature on the topic.

Machine Learning can be a useful and powerful tool for building models for predicting values based on certain conditions. In this study it was seen that it is possible to predict the number of passengers based on weather, at some level. It is debatable if the accuracy of the predictions is good enough for production use. The problem with the results is that the machine learning data included the public holidays, weekends, and common holiday periods. During the summer people often are on vacation and the weather is warm. This might have impacted the results heavily.

If the summer holidays, public holidays, and weekends were removed from the data used for machine learning models, the number of datapoints is low. This makes training of a reliable machine learning model a difficult task. The linear regression showed that there might be a correlation between temperature and number of passengers.

Snow depth was also a remarkable factor in predicting the number of passengers. It might be that when there is a lot of snow on the ground, people choose to go by bus instead of a bicycle or by foot. On the other hand, it might be that people just use buses more during the winter, when the snow depth is bigger.

With more development, and with further fine-tuning of the hyperparameters, the model built in this study could possibly be improved, and the accuracy of the predictions could be improved. Improved model could be integrated into an application that would predict the number of passengers based on weather forecasts. Before doing more development to the model, the causality between different weather conditions and number of passengers should be studied more carefully.

Future continuation for this study could be to collect data for a longer period, preferably for several of years. The data collected for this study was only a bit more than a year. Small dataset might be one of the reasons for inaccurate models. On the other hand, 397 data points should be enough to get good results if data is suitable for such a model. It is worth noting that when the data was split into training and test data, the training data included 80 % of the data. That means that occurrences in the data used for training the model were even lower than in the entire dataset.

The number of passengers in the data varied a lot. At lowest the number of passengers was 25 802 and at the highest it was 350 161. The data was categorized in four or five different categories based on number of passengers on that day as shown in TABLE 10 and TABLE 11. The different categories had 50 000-100 000 passengers each. Most likely machine learning is not necessary to predict the number of passengers on that kind of accuracy.

Another possible explanation for small accuracy could be that the passenger volumes were too low for reliable predictions. The changes caused by weather might be so small that it is not feasible to create a model that would reliably predict the number of passengers at this resolution.

Creating a reliable model may be difficult also because some people must travel by bus whatever the weather. On the other hand, in a small Nordic city, there can be only so many passengers. The model might be “right” in predicting that because of cold weather there should be 420 000 passengers but there simply cannot be that many passengers. Or, when the model predicts 155 000 passengers, there might be 260 000 people for whom it is necessary to take the bus to school or work, for example.

The impact of rain on the number of passengers was surprisingly low. This may be because the cumulative yearly rainfall in the city was lower than in the rest of Norway. To maintain the anonymity of the bus operator it is not possible to declare the exact numbers and references, but the yearly cumulative rainfall is well below 1000 mm. The rainfall varies a lot around the country. In the Western Coast the yearly rainfall can be 2 500 mm. On the other hand, in the north it can be even below 500 mm per year. (Climates to Travel)

As the yearly rainfall is quite small, the number of properly rainy days is most likely low. Therefore, it would be interesting to study if collecting more data would change the significance of rain.

Another topic for future research could be collecting passenger data on different geographic locations and to investigate how changes in weather affect the passenger load in different locations. Adding data about other means of travel, like car riders, pedestrians and cyclists would add depth to the study and it might offer interesting insights on how weather affects traffic in general.

## REFERENCES

- Al-shehari, T., & Alsowail, R. A. (2021). An insider data leakage detection using one-hot encoding, synthetic minority oversampling and machine learning techniques. *Entropy*, 23(10). <https://doi.org/10.3390/e23101258>
- Ameisen, E. (2020). *Building Machine Learning Powered Applications Going from Idea to Product*.
- Beaulieu, A. (2020). *Learning SQL Generate, Manipulate, and Retrieve Data*.
- Bell, J. (2020). *Machine Learning - Hands-on for Developers and Technical Professionals*.
- Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. <https://doi.org/10.1145/2939672.2939785>
- Chen, T., & He, T. (2015). *Higgs Boson Discovery with Boosted Trees* (Vol. 42). <https://github.com/TimSalimans/HiggsML>
- Chollet, F. (2021). *Deep Learning with Python*.
- Climates to Travel. *Climate - Norway*. Climate-Norway. Retrieved December 16, 2023, from <https://www.climatestotravel.com/climate/norway>
- Davies, E. R. (2004). *Machine Vision: Theory, Algorithms, Practicalities*.
- Dilax Automatic Passenger Counting. Retrieved September 18, 2023, from <https://www.dilax.com/en/products/automatic-passenger-counting>
- ELI5 Documentation. *ELI5.show\_weights*. Retrieved February 16, 2024, from [https://eli5.readthedocs.io/en/latest/autodocs/eli5.html?highlight=show\\_weights#eli5.show\\_weights](https://eli5.readthedocs.io/en/latest/autodocs/eli5.html?highlight=show_weights#eli5.show_weights)
- FARA. (2023) *FARA Webpage - about us*. Retrieved February 27, 2024, from <https://fara.no/about-us/>
- FARA. (2023). *SmartHUB Vehicle Gateway*. <https://fara.no/transport-solutions/fara-smarthub-vehicle-gateway-hub/>
- Fontes, T., Correia, R., Ribeiro, J., & Borges, J. L. (2020). A Deep Learning Approach for Predicting Bus Passenger Demand Based on Weather Conditions. *Transport and Telecommunication*, 21(4), 255–264. <https://doi.org/10.2478/ttj-2020-0020>
- Géron, A. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (3rd ed.).
- Grus, J., Farnham, B., Tokyo, S., Boston, B., Sebastopol, F., & Beijing, T. (2019). *Data Science from Scratch First Principles with Python SECOND EDITION*.
- Guo, Z., Wilson, N. H. M., & Rahbee, A. (2007). Impact of weather on transit ridership in Chicago, Illinois. *Transportation Research Record*, 2034, 3–10. <https://doi.org/10.3141/2034-01>

- Hacker Earth. *Hacker Earth*. Decision Tree. Retrieved December 18, 2023, from <https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/ml-decision-tree/tutorial/>
- Hastie, T., Tibshirani, R., & Friedman, J. (2008). *The Elements of Statistical Learning Data Mining, Inference, and Prediction*.
- Huyen, C. (2022). *Designing Machine Learning Systems An Iterative Process for Production-Ready Applications*.
- ITxPT. (2019). *S02-Onboard Architecture specification Part07-APC*. <https://wiki.itxpt.org/index.php?title=Releases>.
- Joly, A. (2017). *Exploiting random projections and sparsity with random forests and gradient boosting methods -- Application to multi-label and multi-output learning, random forest model compression and leveraging input sparsity*. <http://arxiv.org/abs/1704.08067>
- Li, L., Wang, J., Song, Z., Dong, Z., & Wu, B. (2015). Analysing the impact of weather on bus ridership using smart card data. *IET Intelligent Transport Systems*, 9(2), 221–229. <https://doi.org/10.1049/iet-its.2014.0062>
- Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmerfarb, J., Bansal, N., & Lee, S.-I. (2020). *From local explanations to global understanding with explainable AI for trees*.
- Miao, Q., Welch, E. W., & Sriraj, P. S. (2019). Extreme weather, public transport ridership and moderating effect of bus stop shelters. *Journal of Transport Geography*, 74, 125–133. <https://doi.org/10.1016/j.jtrangeo.2018.11.007>
- Mishra, P. (2023). *Explainable AI Recipes: Implement Solutions to Model Explainability and Interpretability with Python, Chapter 4*. Apress.
- Myatt, G. J., & Johnson, W. P. (2014). *Making Sense of Data I: A Practical Guide to Exploratory Data Analysis and Data Mining, 2nd Edition*. Wiley.
- Nelson, H. (2023). *Essential Math for AI*. O'Reilly.
- Nield, T. (2022). *Essential Math for Data Science*.
- Nissen, K. M., Becker, N., Dahne, O., Rabe, M., Scheffler, J., Solle, M., & Ulbrich, U. (2020). How does weather affect the use of public transport in Berlin? *Environmental Research Letters*, 15(8). <https://doi.org/10.1088/1748-9326/ab8ec3>
- Norway Public Holidays. (2023). Publi holidays.No. <https://publi holidays.no/2023-dates/>
- Numpy Documentation. *Numpy.polyfit*. Retrieved January 30, 2024, from <https://numpy.org/doc/stable/reference/generated/numpy.polyfit.html>



- Pedregosa, F., Michel, V., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Vanderplas, J., Cournapeau, D., Pedregosa, F., Varoquaux, G., Gramfort, A., Thirion, B., Grisel, O., Dubourg, V., Passos, A., Brucher, M., Perrot and Édouard, M., Duchesnay, and Édouard, & Duchesnay, Fré. (2011). Scikit-learn: Machine Learning in Python. In *Journal of Machine Learning Research* (Vol. 12). <http://scikit-learn.sourceforge.net>.
- Refaeilzadeh, P., Tang, L., & Liu, H. (2016, December 14). *Cross-Validation*. Liu, L., Özsu, M. (Eds) Encyclopedia of Database Systems. Springer, New York, NY. [https://doi.org/10.1007/978-1-4899-7993-3\\_565-2](https://doi.org/10.1007/978-1-4899-7993-3_565-2)
- Saleh, H. (2018). *Machine Learning Fundamentals* (1st edition). Packt Publishing.
- Scikit-learn Documentation. *Decision Trees*. Scikit-Learn Documentation. Retrieved December 18, 2023, from <https://scikit-learn.org/stable/modules/tree.html>
- Scikit-learn Documentation. *Scikit-learn accuracy score*. Retrieved February 15, 2024, from [https://scikit-learn.org/stable/modules/model\\_evaluation.html#accuracy-score](https://scikit-learn.org/stable/modules/model_evaluation.html#accuracy-score)
- Scikit-learn documentation. *Sklearn StandarScaler documentation*. <https://Scikit-Learn.Org/Stable/Modules/Generated/Sklearn.Preprocessing.StandardScaler.Html>. Retrieved February 2, 2024, from <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>
- Singhal, A., Kamga, C., & Yazici, A. (2014). Impact of weather on urban transit ridership. *Transportation Research Part A: Policy and Practice*, 69, 379–391. <https://doi.org/10.1016/j.tra.2014.09.008>
- Tao, S., Corcoran, J., Hickman, M., & Stimson, R. (2016). The influence of weather on local geographical patterns of bus usage. *Journal of Transport Geography*, 54, 66–80. <https://doi.org/10.1016/j.jtrangeo.2016.05.009>
- Tao, S., Corcoran, J., Rowe, F., & Hickman, M. (2018). To travel or not to travel: 'Weather' is the question. Modelling the effect of local weather conditions on bus ridership. *Transportation Research Part C: Emerging Technologies*, 86, 147–167. <https://doi.org/10.1016/j.trc.2017.11.005>
- Thiagarajan, R., & Prakashkumar, D. S. (2021). Identification of Passenger Demand in Public Transport Using Machine Learning. *Webology*, 18(SpecialIssue2), 223–236. <https://doi.org/10.14704/WEB/V18SI02/WEB18068>
- Tokuç, A. (2023, May 12). *Splitting a Dataset into Train and Test Sets*. <https://www.baeldung.com/cs/train-test-datasets-ratio>

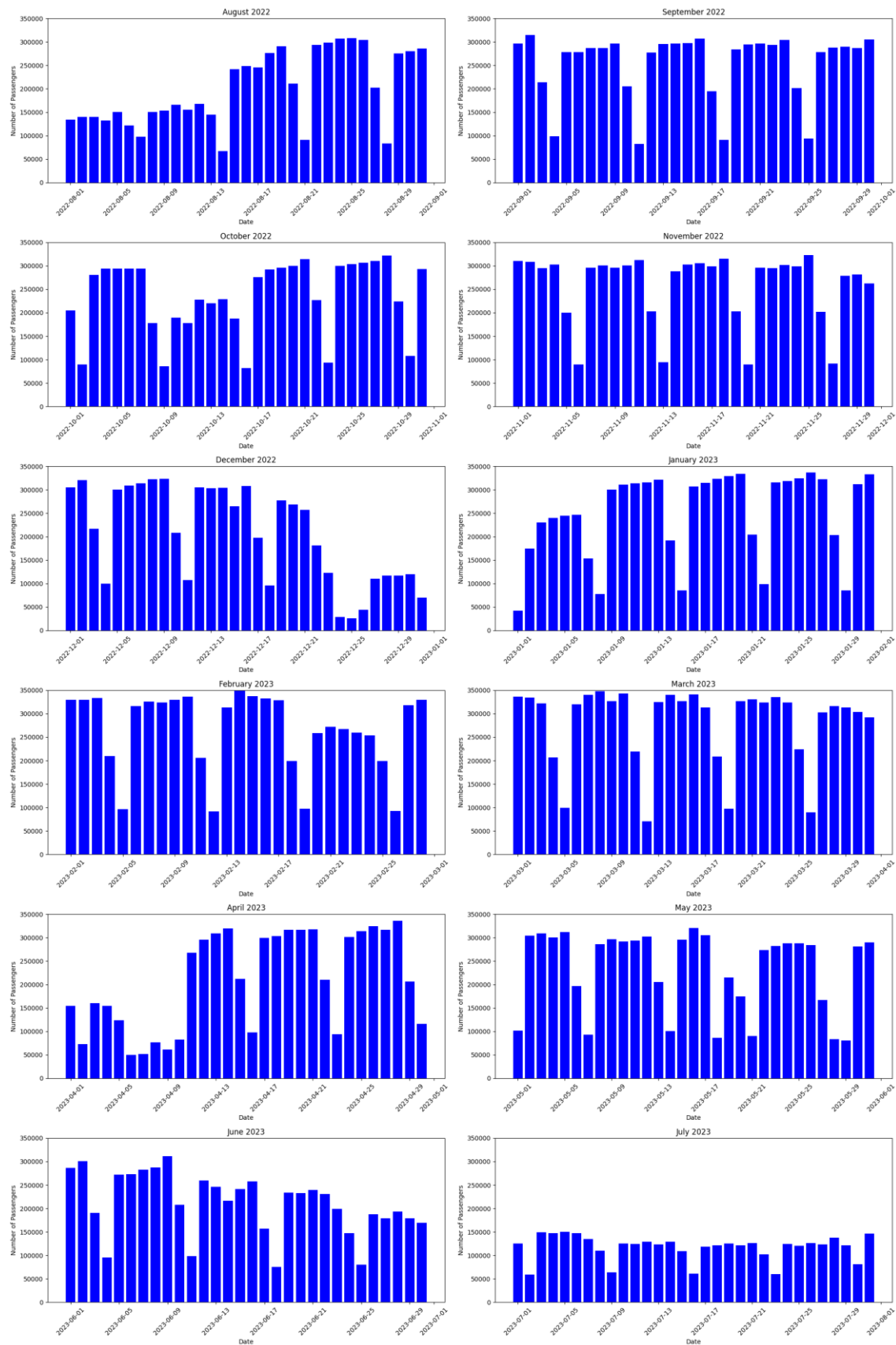
- Trautmann, D. *Confusion Matrix*. Retrieved November 28, 2023, from [https://www.researchgate.net/figure/Confusion-matrix-for-the-dev-set-of-ECHR\\_fig1\\_372989679](https://www.researchgate.net/figure/Confusion-matrix-for-the-dev-set-of-ECHR_fig1_372989679)
- Trøndelag School Holidays 2023 and 2024. (2023). Publicholidays.No. <https://publicholidays.no/school-holidays/trondelag/>
- Visual Crossing Documentation. (2023, March 23). Visual Crossing Weather Data Documentation. <https://www.visualcrossing.com/resources/documentation/weather-data/weather-data-documentation/>
- Visualcrossing.com. (2023, February 15). Visual Crossing. <https://www.visualcrossing.com/resources/documentation/weather-data/how-historical-weather-data-records-are-created-from-local-weather-station-observations/>
- Wade, C. (2020). *Hands-On Gradient Boosting with XGBoost and scikit-learn*.
- Wei, M. (2022). How does the weather affect public transit ridership? A model with weather-passenger variations. *Journal of Transport Geography*, 98. <https://doi.org/10.1016/j.jtrangeo.2021.103242>
- Wei, M., Liu, Y., Sigler, T., Liu, X., & Corcoran, J. (2019). The influence of weather conditions on adult transit ridership in the sub-tropics. *Transportation Research Part A: Policy and Practice*, 125, 106–118. <https://doi.org/10.1016/j.tra.2019.05.003>
- XGBoost Documentation. (2022). *Introduction to Boosted Trees*. Introduction to Boosted Trees. <https://xgboost.readthedocs.io/en/stable/tutorials/model.html>
- Zhou, M., Wang, D., Li, Q., Yue, Y., Tu, W., & Cao, R. (2017). Impacts of weather on public transport ridership: Results from mining data from different sources. *Transportation Research Part C: Emerging Technologies*, 75, 17–29. <https://doi.org/10.1016/j.trc.2016.12.001>

## APPENDICES

Daily passengers each month	APPENDIX 1
Correlation between temperature and no of passengers	APPENDIX 2
Correlation between precipitation and no of passengers	APPENDIX 3
Number of passengers categorized by the day of the week	APPENDIX 4
Number of passengers in relation to temperature, day by day	APPENDIX 5

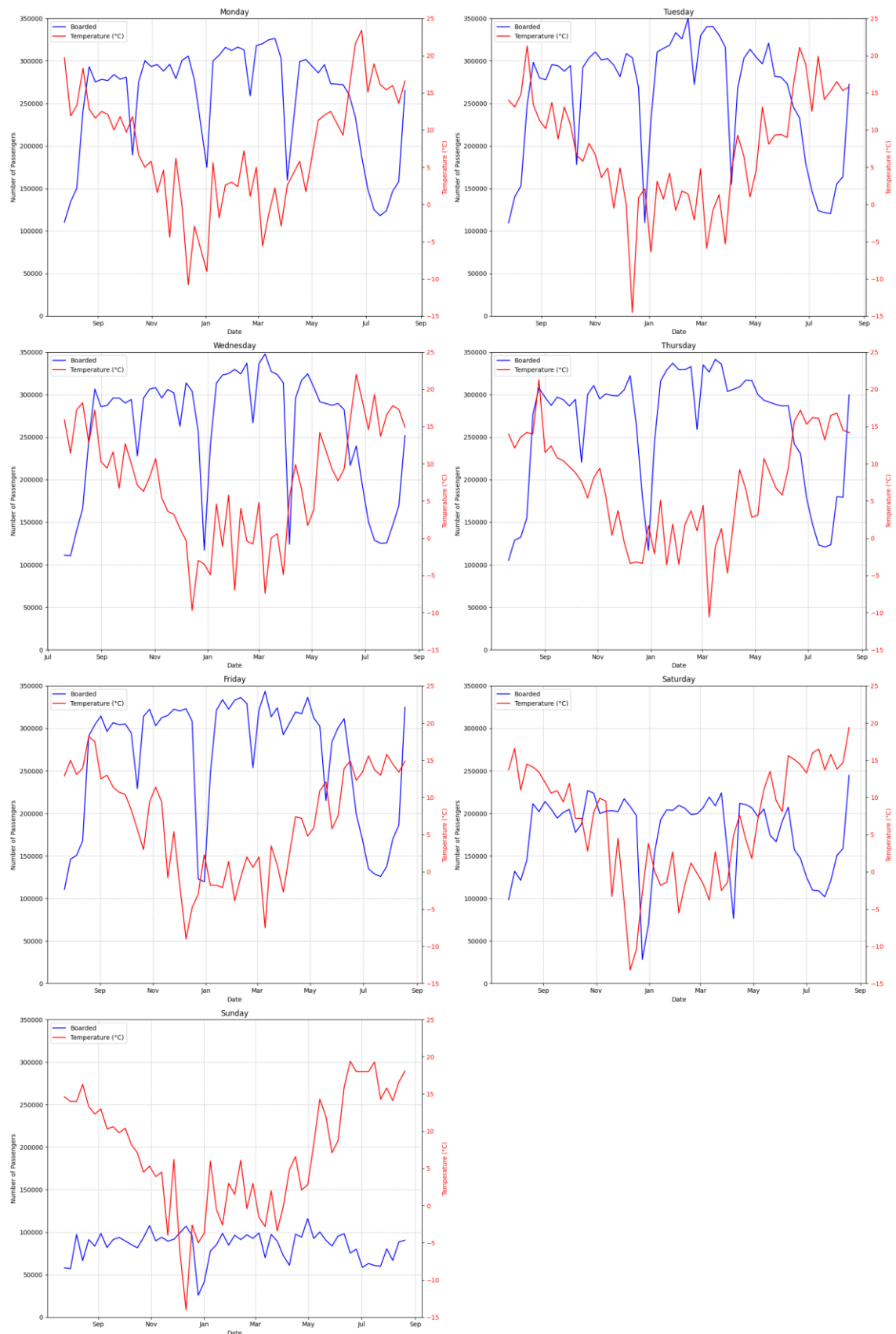
# DAILY PASSENGERS EACH MONTH

# APPENDIX 1



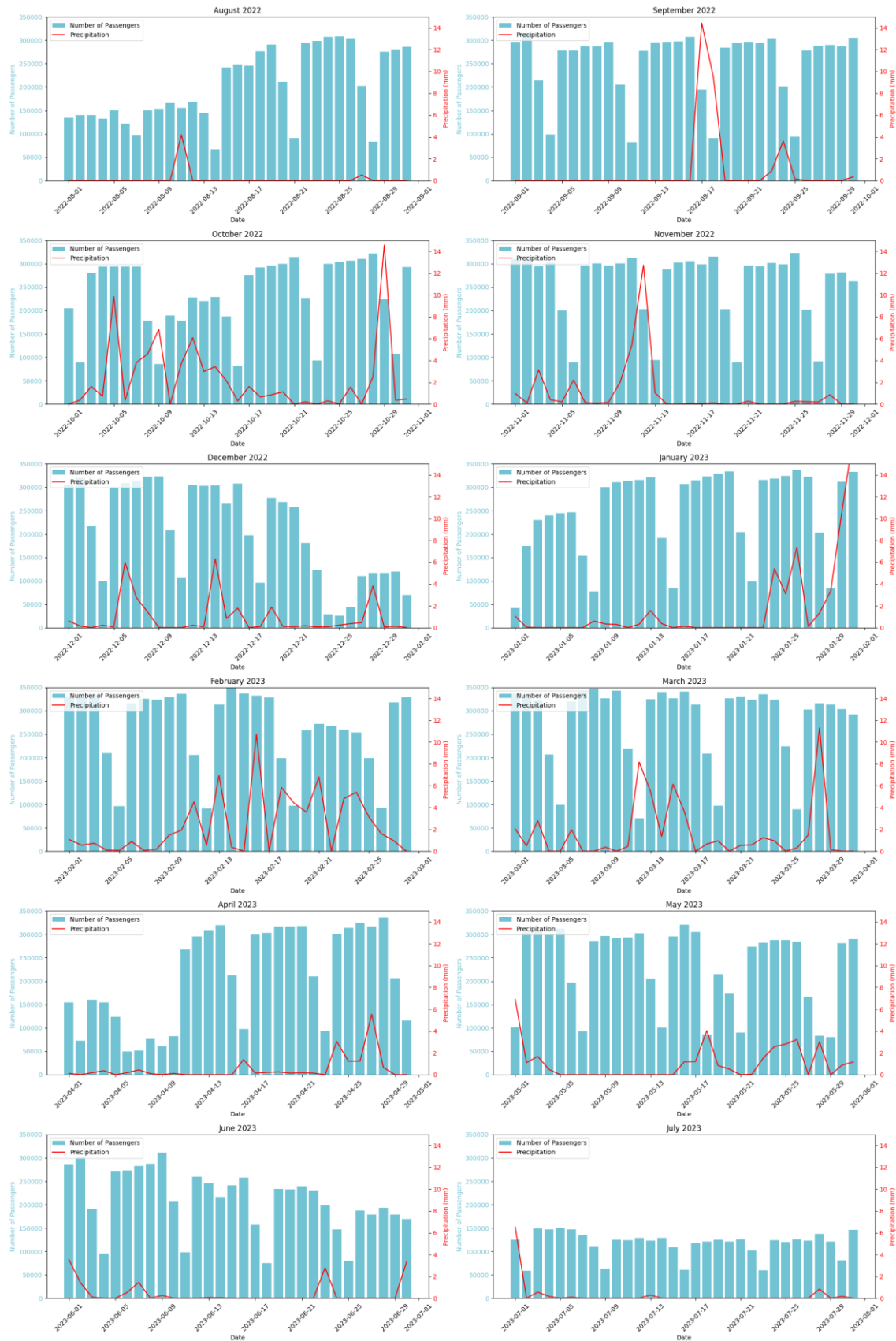
# CORRELATION BETWEEN TEMPERATURE AND NO OF PASSENGERS

# APPENDIX 2



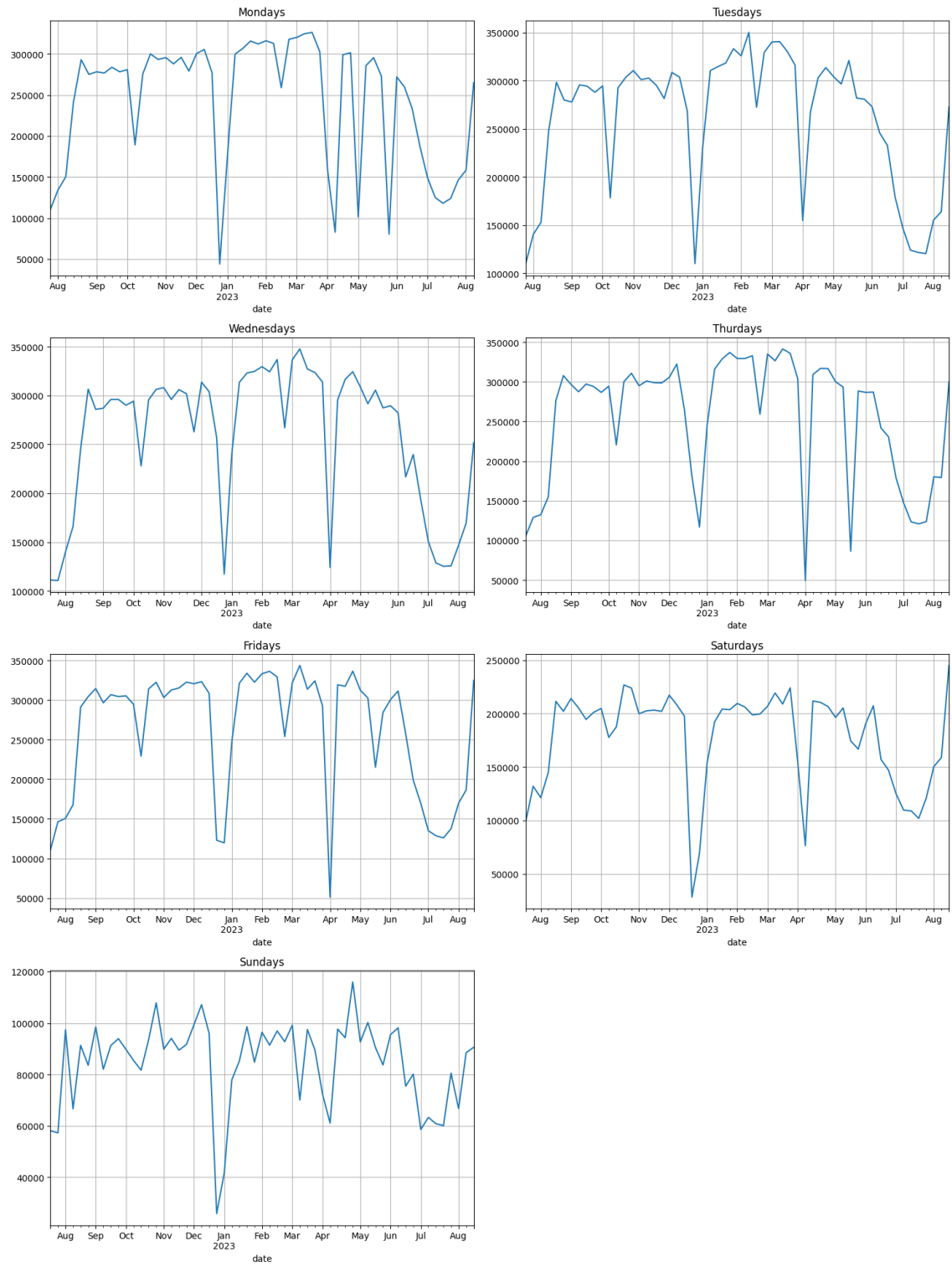
# CORRELATION BETWEEN PRECIPITATION AND NO OF PASSENGERS

# APPENDIX 3



# NUMBER OF PASSENGERS CATEGORIZED BY THE DAY OF THE WEEK

# APPENDIX 4



# NUMBER OF PASSENGERS IN RELATION TO TEMPERATURE, DAY BY DAY APPENDIX 5

