Machine Learning

Ready, Steady Ride

Group7
30/04/2023

Anna Kwiatkowska 20222216

Stefano Sperti 20222246

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

ABSTRACT

This report aims to develop a predictive model to forecast customer demand for the Ready, Steady Ride ride-sharing service. The model was designed to improve efficiency and reliability in the allocation of resources by accounting for multiple factors affecting customer demand. The project utilized a representative subset of 13,000 hours from an approximate total of 18,000 hours of data obtained during a pilot test. The data was explored and prepared before several models, including KNN, linear regression, neural networks and a regression tree model, were developed and compared. The model that minimizes the Mean Absolute Error (MAE) was a regression tree, and its hyperparameters were optimized using grid search to improve its performance in predicting customer demand. The model was able to accurately forecast customer demand, making it a robust and efficient solution for ride-sharing companies. The use of MAE as the evaluation metric allowed for consistency with the Kaggle competition and provided a more robust measure that was less sensitive to outliers. By optimizing the feature model with grid search, its performance was improved, demonstrating the importance of hyperparameter tuning for achieving accurate predictions.

KEYWORDS

Machine learning, Predictive modelling, Regression trees

CONTENTS

INTRODUCTION

The Ready, Steady Ride ride-sharing service recognized the need to develop a predictive model to accurately forecast customer demand for any given hour. With the company's expansion into new locations, it was crucial to ensure that resources were allocated efficiently and customer demand was met with adequate supply. By developing a sophisticated model that accounts for multiple factors, the company aimed to improve its overall efficiency and reliability. This project addresses the business needs of Ready, Steady Ride and has important implications for ride-sharing companies seeking to optimize their services and stay competitive in the market.

The model was constructed utilizing a representative subset of 13,000 hours from an approximate total of 18,000 hours of data obtained during a pilot test. The objective of the model development was to facilitate training the model to forecast the demand for Ready, Steady Ride company's services during any given hour. This report explains the approach used to create the best model, starting with exploring and preparing the data and ending with choosing the best model. Furthermore, the report offers comprehensive conclusions regarding the project.

METHODOLOGY

DATA EXPLORATION AND UNDERSTANDING

Initially, we imported the essential libraries into Jupyter Notebook and proceeded with the data integration stage. Furthermore, to perform the required modifications, we created a duplicate of the train dataset. Later, an assessment of the dataset was conducted to identify inconsistencies and missing data. To achieve this, the pandas library's "info" and "describe" functions were utilized. The initial statistics of the dataset are shown in Table 1 and Table 2.

Specifically, the data was scrutinized for inconsistencies, using both the initial statistics for the numerical variables and the histograms for the categorical variables. However, no such irregularities were detected. Some examples of regularity that we have checked are: positive features do not have negative values, the minimum and maximum value of each feature are “realistic”, the datatype is consistent. The dataset is coherent and can be used for further processing. It was observed that the variable "City_Detroit" had only one value, which was "1". Additionally, all variables were classified into categorical and numerical types, and one variable called "InstanceID", which acted as a unique identifier for each row of the dataset, was identified as a non-feature variable. Subsequently, we generated boxplots, histograms, scatterplots and the pair plot to detect any inconsistencies or outliers. Notably, we observed outliers in three variables, namely Humidity, Windspeed, and RidesBooked (Target). The next step involved identifying missing data by applying the "isna" function, revealing over 1000 instances of missing data in the training dataset, equivalent to 8% of the total data. We observe that there were not missing data in the RidesBooked feature (target), because if there were, the only thing we could have done was delete the row that present this specific missing value. Finally, the training set was examined for duplicate values by utilizing the "value_counts" function for the "InstanceID" variable. Subsequently, a portion of the training dataset, comprising 20% of the total data, was set aside to create a validation dataset. This dataset will be utilized in the project to assess the model's performances.

We also didn’t observe the gaussian property in any of the ordinal features. We can observe that the database was a well balanced dataset with respect to most of the features such as “Season, HourofDay, Year, Month, Day”. An important observation is the omission of two time schedules in the variable HourofDay. We have decided that the missing values are not these two missing hour since for each hours there were at least 600 observations but there were only 60 missing values. Therefore it was impossible that this missing value came from this two missing values..

DATA PREPROCESSING

Duplicate values and outliers

Upon dividing the dataset into two parts, we proceeded with the preprocessing of the training dataset. Specifically, the first step in our preprocessing pipeline involved removing the identified duplicate values. Subsequently, we opted to eliminate the outliers in variable WindSpeed since they represented only a small proportion of the data set. We observe that the only outlier in the variable Humidity was discovered to be a value ‘0’ that could be not an actual outlier. As it is really unlikely for humidity to have value ‘0’, a function was implemented to address this issue. Namely the function

changed the value ‘0’ into a missing value. Furthermore, it was observed that handling outliers in the RidesBooks variable is not useful since it serves as the target variable.

Missing values

Following the removal of outliers, we addressed the issue of missing data by utilizing available variables and logical reasoning to identify potential solutions. These solutions were categorized into two primary paths, which are described below.

The first path involved deterministic methods, whereby we employed our logical thinking to determine how to fill in the missing values. This approach can be further subdivided into two categories, namely the date-alike functions and quantitative functions. The former is based on a specific date and is used in variables such as Season, Month and WeekoftheYear, while the latter involves the use of the group-by function to compute the mean or mode value for a particular group and is used in variables such as WeatherForecast, Temperature and FeltTemperature, Humidity and WindSpeed. All the variables mentioned above were grouped by variable Date that represent the timeline (each day). Indeed those variables were filled with the mean or mode of the day of the revelation. Those a-priori functions suffer in case in a random hold out method it could happen that we are not able to identify the date of the event or there are no events in the same day. Therefore it was also specified that in cases where the functions fail to operate , such as when a missing value is present among the variables utilized in the function, the missing values shall be substituted with the median value of the train dataset specific to the corresponding variable.

The second path utilized a KNN imputer, which is a machine learning algorithm that use the characteristics of similar observations to impute missing values. This method was applied to variables such as Year, DayoftheWeek, HouroftheDay, Holiday and WorkingDay to fill in the missing data.

Feature Engineering

As certain variables were categorical in nature, it was necessary to convert them into numerical values. To accomplish this, the data was transformed into a dummy format using the pd.get_dummies function. This technique allows for the conversion of categorical variables into a numerical format that can be used in the model. The variables that were converted in this way were Season and WeatherForecast. We have also converted the same variable into a new numerical variable in which each element corresponds to one number and we have done this while carefully maintaining the order.

Additionally, two new variables, Time_day and Time_line, were developed. Time_day is intended to transform the continuous variable HouroftheDay into a new numerical variable comprising only four values. This procedure was carried out to aggregate the 24-hour day into four designated time periods, to make it easier to evaluate patterns and correlations with the target variable RidesBooked through histograms. Furthermore, this step improved the effectiveness of the KNN algorithm by converting the continuous variables into discrete variables, which are more adequate to KNN.

The Time_line variable was established by concatenating the variables Month and Year, since the Year was not specified in the dataset, rather only marked as values 3 or 4 we arbitrarily decided that the timeline starts from January of year 3. In this variable each time interval was 1 month. The primary aim of this variable creation was to explore the linear progression of RidesBooked with respect

to time. We also create the variable Date during the missing value procedure with the same idea of the Time_line but with an interval of 1 day. These two variables represent the increasing of the demand during the time. This could be seen as an expansion in the time of the company and this could be useful for better predicting the outcome of a future event.

Scaling and feature selection

Due to the non-normal distribution of variables, it was necessary to perform MinMax Scaling. Min-max scaling is a data preprocessing technique used in machine learning to scale numeric features to a specific range of values. The purpose of scaling the features is to bring all features to a common scale, which is particularly useful for algorithms that use distance-based measures. Hence, this preliminary step was executed to ensure the variables were scaled before commencing model selection, thus preventing the need for scaling the data multiple times.

The following step of the data preparation included feature selection, which required selecting the variables for use in the model. Initially, it was discovered that the FeltTemperature variable could not be utilized in predicting future ride bookings since it reflected the temperature experienced by customers during the ride. These values were measured after the booking had already been made. Thus, it was deemed necessary to exclude this variable from the model since the business goal was to predict something that will happen in the future and therefore in real life we will never have this information. Furthermore, the CityDetroit variable was also eliminated due to its lack of significance in contributing new information to the model. After a little optimization of the regression three model we have also used the feature importance in the regression three for understanding which variable contains important information.

To identify the variables with the most significant impact, ANOVA testing can be performed. ANOVA, which stands for Analysis of Variance, is a statistical method utilized to analyze differences in means among two or more groups. This method is frequently employed in feature selection to determine the relevance of a feature in explaining the variance of the target variable. The outcomes of ANOVA can assist in identifying the most relevant features that are highly correlated with the target variable, which, in turn, can lead to the creation of more accurate predictive models. However, ANOVA is not suitable for variables that do not exhibit a normal distribution. In the case of our model, ANOVA could not be employed as the variables did not conform to normal distribution requirements.

The Spearman correlation was employed to assess the correlation between the variables. We have applied the Spearman correlation only on the train dataset to avoid the problem presented in Jerome H. Friedman, Robert Tibshirani e Trevor Hastie, “Elements of Statistical Learning”. The decision to employ the Spearman correlation, rather than the Kendall and Pearson correlation, was based on several factors. The primary reason is that it utilizes nonparametric measures of the data, so it doesn't assume a particular distribution of the data that we don’t have in our dataset. Additionally, the Spearman correlation is more widely known and more commonly used in research, as well it is easier to compute.

To see patterns more effectively, a heatmap was employed, presented in Figure 8 and Figure 9. Using the heatmap, 8 variables that exhibited the strongest correlation with the target were selected. These variables were HourOfDay, Temperature, Humidity, WindSpeed, Date,

WeatherForecast_ord, Season_0, and Season_2. However, the remaining variables could potentially be utilized in certain models to improve efficiency.

Following the development of all models, a review was conducted to analyze whether using different variables would enhance the models' efficacy. Variables were selected using feature importance in the regression tree and heuristic methods. The second approach utilize feature importance. We established a threshold of 0.002 for the feature importance instead of 0.0 in the regression tree for avoid overfitting. The second approach utilized a trial and error methodology that relied on Spearman's correlation coefficient. The latter was performed through the elimination of the least correlated features with the target variable. As a result of this move, the issue of multicollinearity was resolved as these variables were found to be correlated with one another. The multicollinearity is particularly problematic in linear regression models. That is because if two or more predictor variables in a statistical model are highly correlated with each other, it becomes difficult to identify which variable is actually contributing more towards the outcome. Table 3 shows the features selected to use in models based on different approaches.

MODELS

K-Nearest Neighbours

K-nearest neighbors (KNN) is a non-parametric supervised learning algorithm that can be used for regression and classification tasks. It predicts the value of a new data point by comparing it to the k-nearest neighbors in the training data, based on a similarity measure. KNN can handle numerical and categorical input variables and does not assume any underlying distribution. However, it can be sensitive to the choice of k and the presence of irrelevant features. KNN is an algorithm mostly used for classification problems. In our case we can see the target (number of races in a day) as different integers that classify the state.

Linear Regression

Linear regression is a statistical technique employed for modeling the connection between a dependent variable and one or more independent variables. Linear regression presumes that there exists a linear relationship between the variables, which leads to the value of the dependent variable. Consequently, a linear regression model outputs a linear equation that can be utilized to predict or estimate the value of the dependent variable based on the values of the independent variables.

Additionally, in order to optimize the model performance, linear regression algorithm was used without the intercept. When fit_intercept is set to False in the sklearn LinearRegression object, we are instructing the model to exclude the intercept term from the regression equation. As a result, the regression line will start at the origin (0,0) and all predictions will be solely based on the independent variables and their coefficients, with no additional constant term added to the predictions.

Regression Trees

Regression trees are a type of decision tree that are used for predicting continuous values. They work by splitting the data into smaller and smaller groups based on input variables, and then averaging the target variable within each group to make a prediction. Regression trees can handle both

categorical and continuous input variables and missing data. However, they can overfit the training data, so we pruning in several stages the parameters.

Neural Networks

Neural network is a type of machine learning model inspired by the structure and function of the human brain. It consists of layers of interconnected nodes (neurons) that process information and make predictions based on patterns in the input data. The model learns to map input data to output values through the adjustment of the weights between the neurons.

PERFORMANCE OF THE MODELS

K fold implementation

K-fold cross-validation is a technique used for assessing the performance of a machine learning model. It involves dividing the dataset into k equal-sized partitions or folds, where k is a positive integer. Then, the model is trained on k-1 folds and validated on the remaining fold. This process is repeated k times, with each fold serving as the validation set once. For the project, we have designated the value of k as 10. Furthermore, we have developed a function to perform the entire process with missing values and data preprocessing in order to ensure the robustness and reliability of our findings.

OPTIMIZATION OF THE MODELS

Grid Search

Grid search is a hyperparameter tuning technique used in machine learning to find the optimal combination of hyperparameters for a given model. Hyperparameters are the parameters that are not learned during the training process, but instead are set before training begins. The grid search technique works by specifying a range of values for each hyperparameter, and then exhaustively searching all possible combinations of these values. The grid search optimization was implemented twice time after we have selected the modelt. Firstly, to optimize the model with all the features since the regression tree is not affected of multicollinearity problem . Secondly, after the variables with the smallest importance were deleted we have implement another time the alghoritm.

We have decided to use Grid search since the computational problem of trying all the parameter was “feasible” and therefore since Grid search doesn’t introduce any randomness in the alghoritm is more robust then a random search.

RESULTS

In order to evaluate all of the models present in the section above, the statistical parameters were calculated. We have chosen to compare the models based on Mean Absolute Error (MAE). It measures the average of the absolute differences between the predicted values and the actual values. Lower MAE indicates better performance. We have chosen to use MAE because it is a more robust measure that is less sensitive to outliers. Moreover, as the Kaggle competition utilized the MAE metric for evaluating the performance of the models, it was considered convenient to use the same metric in our analysis for the sake of consistency and comparability with the competition results. The results of the models are presented in Table 4. To choose the best model we were looking at the performance in the validation dataset. As it is shown in the table, the feature model had the significantly lowest MAE compared to the other models.

We can observe that regression trees perform better in pur dataset than Neural networks. Indeed, regression trees can handle non-linear relationships between variables without requiring the use of non-linear activation functions. Neural networks, on the other hand, often require non-linear activation functions to model non-linear relationships, and the choice of activation function can greatly impact performance. Regression trees can perform well on small datasets, whereas neural networks often require large amounts of data to train effectively. In our case the dataset is a medium dataset and therefore could suffer from small database problem. This is because neural networks have a large number of parameters that need to be tuned, and this requires a lot of data to prevent overfitting. Even if we have used a preliminary optimization of the hyperparameters of the neural networks our performances remain really low.

For the basic linear regression the results were quite low, as the MAE was 317.16. As well for the linear regression without the intercept the MAE was quite similar, namely 317.22. It means that on average, the absolute difference between the predicted values and the actual values is 317. The problem of the linear regression is due to the fact that the model assumes a linear relationship between the independent variables and the dependent variable, but a non-linear relationship may exist, resulting in limited explanatory power. Secondly, the dependent variable may have a functional form that is better represented by a different type of regression model, such as a polynomial regression or a logarithmic regression. Thirdly, the independent variables included in the model may be insufficient or irrelevant to the research question, despite being the most correlated with the dependent variable.

The K-Nearest Neighbours has really low performance and this is due to the fact that we used both at the same time with continuous and binary variables. The results are driven by the fact that individual observations nearest neighborhood by distance would be much more heavily informed by the binary variable than by the scaled real-value variable and this causes a problem in the model. Even reducing the number of the neighborhoods we can see (as we expected) an overfitting problem and increasing the neighborhood we increase the bias due to the difference between binary and real-value variables. The only thing that we can try heuristically is to change variables but this is not a good decision since the most correlated with the target are the real-value variables.

The feature model is the regression tree model optimized with grid search. The grid search technique was employed to optimize the model's hyperparameters, which included setting the criterion to Poisson, the maximum depth to 14, and the split strategy to best. By tuning these

hyperparameters, the model's performance was improved, and it was able to better predict the target variable.

CONCLUSION

The aim of this study was to develop a predictive model that accurately estimates ride demand for any given hour based on various features. To achieve this, a dataset was collected through a pilot test conducted by the company. Following this, the dataset was put through a series of processing steps that involved identifying and resolving inconsistencies, preprocessing the data, selecting the most relevant features, and testing various models. The performance of each model was evaluated based on the mean absolute error, and the model with the lowest error was selected as the best fit. In this study, the optimal model was found to be a regression tree that was optimized using grid search.

All of our result are driven by a back test validation of the model. In any case our model is able to predict a possible event in the future since the feature selected are consistent with real world application.

Despite achieving satisfactory results, this study identifies opportunities for further improvement. Firstly, exploring other models for feature selection such as random forests, could potentially reduce the mean absolute error. Secondly, it could usefull to combine different models for improve the performances. However, certain limitations, such as computational resources or the size of the research team, may have restricted the extent of these improvements.

REFERENCES

1. Dennis E. Hinkle, William Wiersma, Stephen G. Jurs, "Applied Statistics for the Behavioral Sciences"
2. Alan Agresti, Barbara Finlay, "Statistical Methods for the Social Sciences"
3. Myles Hollander, Douglas A. Wolfe, "Nonparametric Statistical Methods"
4. Jerome H. Friedman, Robert Tibshirani e Trevor Hastie, Elements of Statistical Learning, 2001, pages 245-247
5. https://stattrek.com/tutorials/regression-tutorial
6. https://www.datacamp.com/

APPENDIX

Table 1 Initial statistics for numerical variables

	count	mean	std	min	max
InstanceID	13955.0	55010.909137	26206.253028	10004.0	99978.0
Year	13888.0	3.503528	0.500006	3.0	4.0
HourofDay	13883.0	11.541742	6.889074	0.0	23.0
Holiday	13870.0	0.0292	0.168372	0.0	1.0
WorkingDay	13886.0	0.682558	0.465498	0.0	1.0
Temperature	13894.0	13.075341	7.722183	-6.001	33.15
FeltTemperature	13892.0	16.189178	5.849787	0.0	32.453
Humidity	13898.0	0.626759	0.192956	0.0	1.0
WindSpeed	13876.0	0.189967	0.122566	0.0	0.8507
DayofMonth	13887.0	15.637287	8.757647	1.0	31.0
AverageRideDurationPreviousDay_Min	13882.0	19.845969	8.700009	5.001	35.0
City_Detroit	13876.0	1.0	0.0	1.0	1.0
WeekofYear	13888.0	26.659778	15.017542	1.0	52.0
RidesBooked	13955.0	569.746972	546.770205	3.0	2931.0

Table 2 Initial statistics for categorical variables

	count	unique	top	freq
Season	13870	4	Summer	3604
Month	13886	12	December	1209
DayofWeek	13887	7	Sunday	2023
WeatherForecast	13890	4	Clear	9079

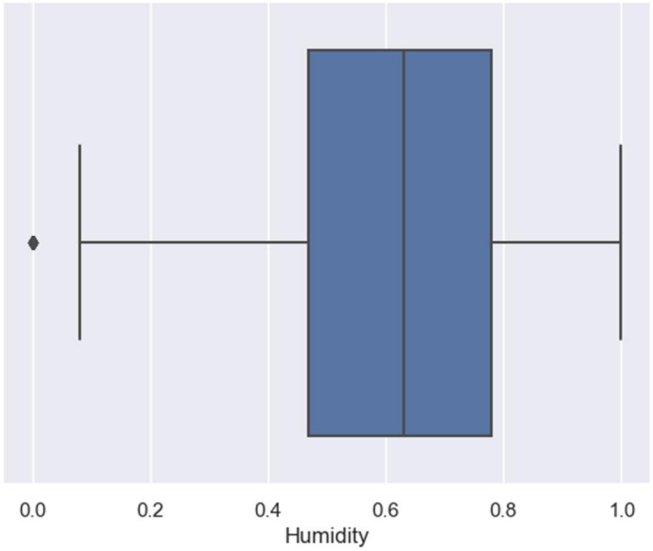
Table 3 Feature selection

	Spearman	Heuristic	Feature importance
Year	-	-	-
Month	-	-	-
HourofDay	YES	YES	YES
Holiday	-	-	-
DayofWeek	-	-	YES
WorkingDay	-	-	YES
Temperature	YES	YES	YES
Humidity	YES	YES	YES
WindSpeed	YES	YES	YES
DayofMonth	-	-	YES
AverageRideDurationPreviousDay_Min	-	-	YES
Date	YES	YES	YES
Season_ord	-	-	-
WeatherForecast_ord	YES	-	YES
Season_0	YES	-	-
Season_1	-	-	-
Season_2	YES	-	-
Season_3	-	-	-
WeatherForecast_0.0	-	-	-
WeatherForecast_1.0	-	-	-
WeatherForecast_2.0	-	-	YES
WeatherForecast_3.0	-	-	-
Time_line	-		-

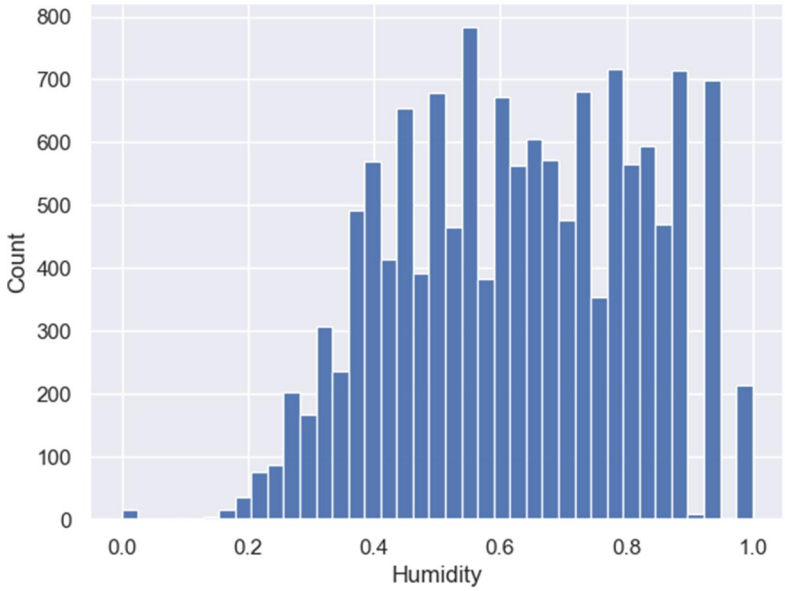
Table 4 Model selection

	Time	Train	Validation
feature_model	0.157+/-0.02	39.285+/-1.03	102.884+/-2.78
linear	0.014+/-0.02	319.264+/-0.67	319.559+/-4.9
KNN	0.077+/-0.01	280.865+/-0.88	311.125+/-6.61
KNN_1	0.09+/-0.04	0.0+/-0.0	279.604+/-9.32
KNN_10	0.101+/-0.04	337.761+/-0.97	352.218+/-9.51

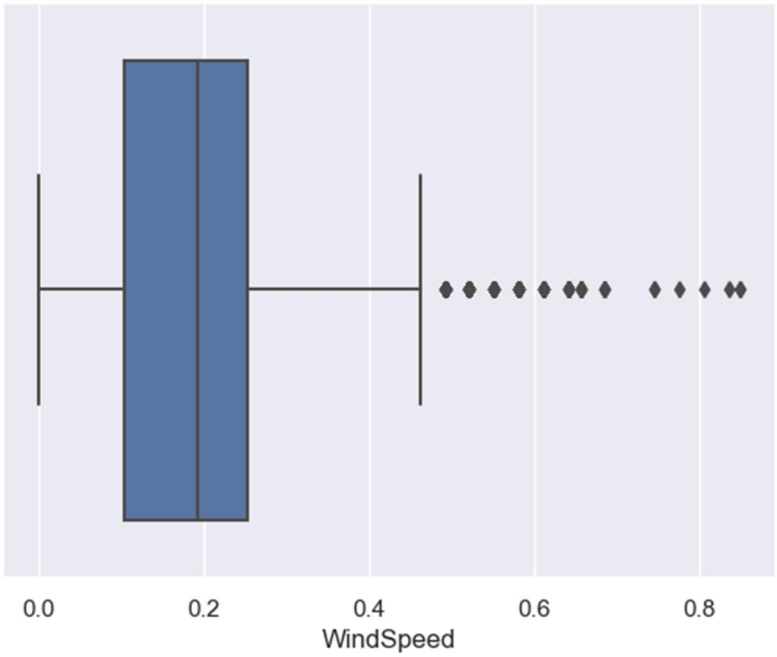
Figure 1 Humidity's boxplot



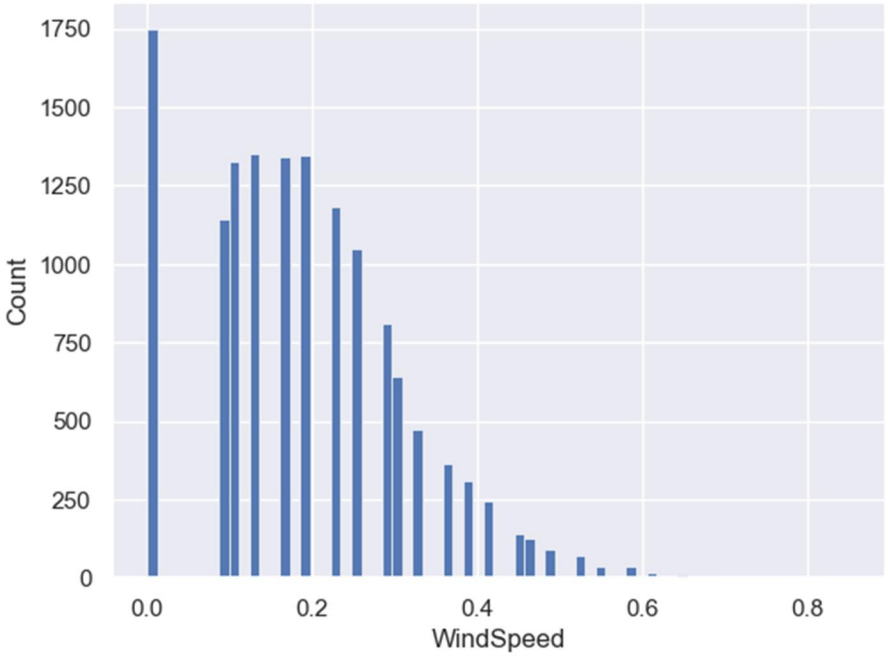
Graph 2 Humidity's Histogram



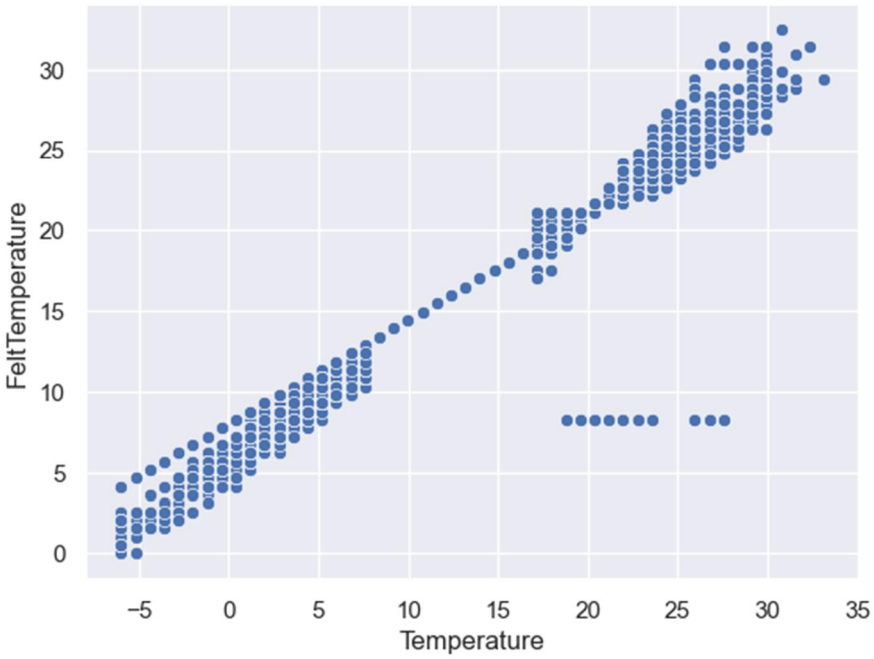
Graph 3 Windspeed's Boxplot



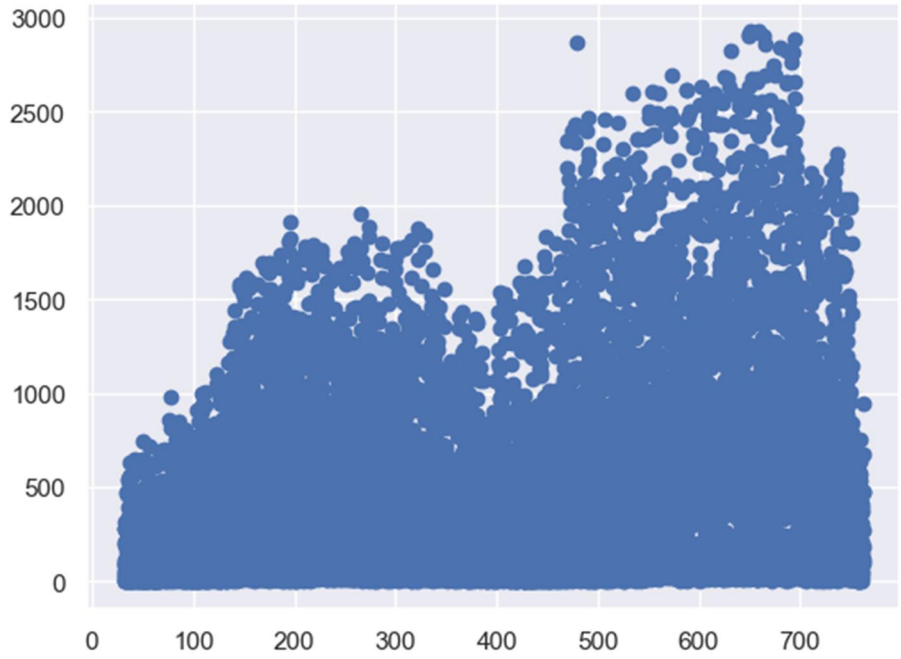
Graph 4 Windspeed's Histogram



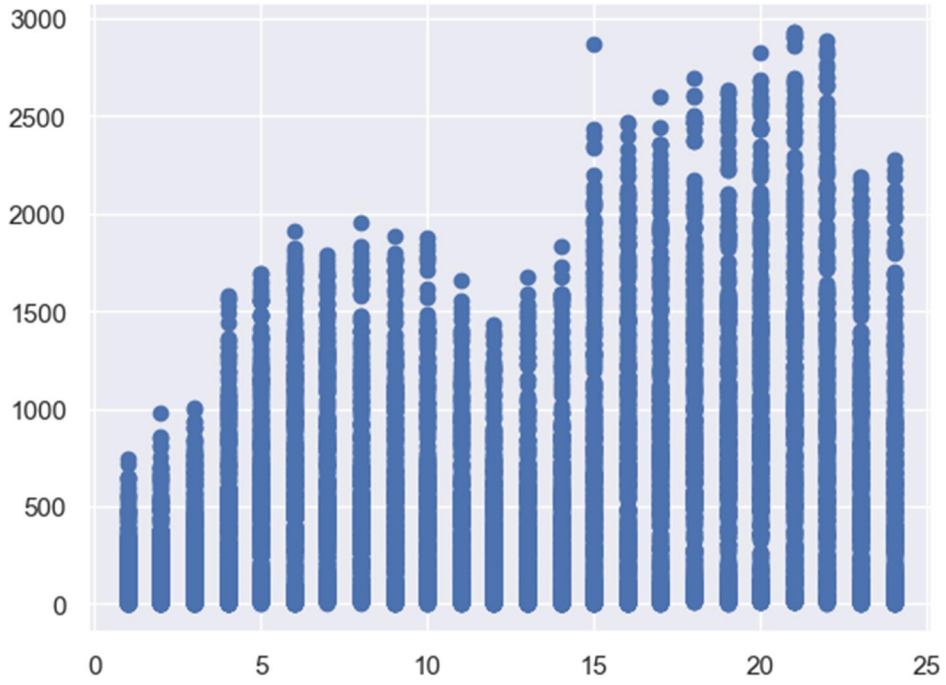
Graph 5 Felt Temperature and Temperature correlation



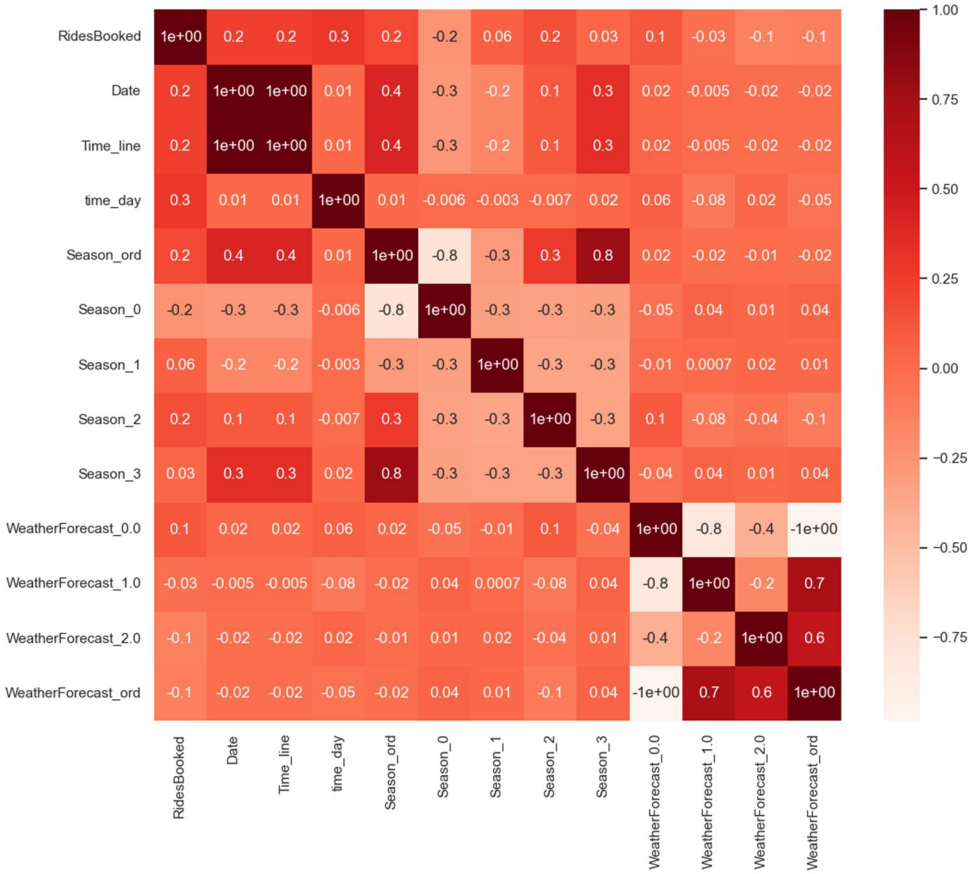
Graph 6 Date and RidesBooked (before preprocessing)



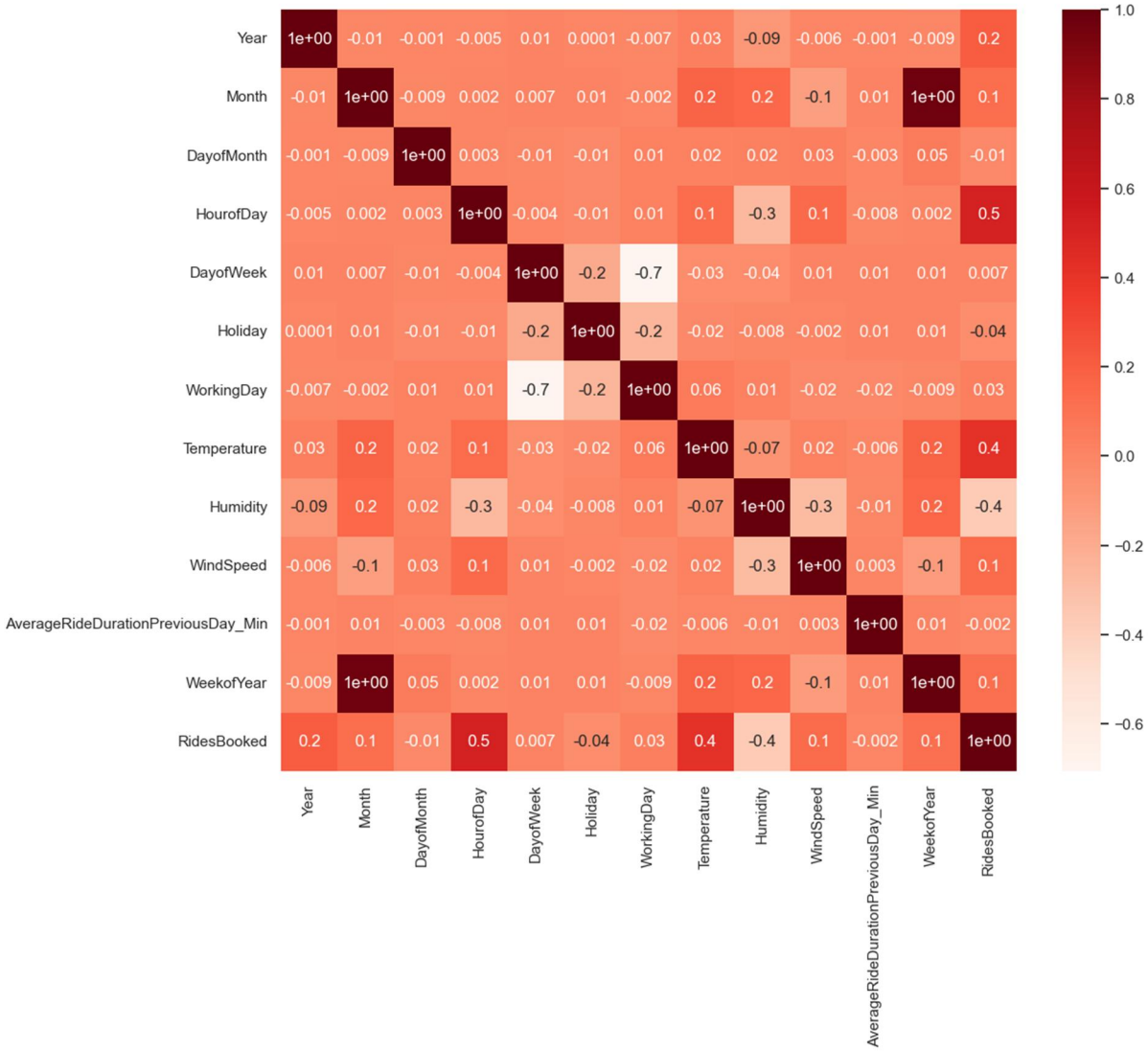
Graph 7 Time_line and RidesBooked (after preprocessing)



Graph 8 Spearman heatmap part1



Graph 9 Spearman heatmap part2



NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa