Predicting hourly demand for shared bicycles with weather data and machine learning models

Dijora Peja

Dissertation written under the supervision of Professor Nicolò Bertani

Dissertation submitted in partial fulfilment of requirements for the MSc in Business Analytics, at the Universidade Católica Portuguesa, January 4th, 2023.

Table of Contents

List of Figures

List of Tables

Abstract

Title: Predicting the hourly demand for shared bicycles with weather data and machine learning models

Author: Dijora Peja

This thesis aims to analyze the bike sharing system in Chicago and apply predictive models that accurately predict the hourly demand for shared bicycles by using time-related and weather-related features. The dependent variable is Count, expressing the sum of the number of bicycles used per hour. Predictive models that are used for this regression problem are Linear Regression, Random Forest, Gradient Boosting, Light Gradient Boosting, Extreme Gradient Boosting, and Multi-Layer Perceptron. Accuracies of these predictive models are measured by R2_score, Root Mean Square Error and Mean Absolute Error. For better predictions, different hyperparameters are used in predictive models.

Without hyperparameters, Random Forest achieves the best accuracy measures. However, after using hyperparameters, Gradient Boosting predicts the most accurate results. The accuracy of Gradient Boosting boosts with hyperparameters, whereas Random Forest is almost unaffected by them for this regression problem. The second-best model when using hyperparameters is Extreme Gradient Boosting. The neural network model, Multi-Layer Perceptron presents less accurate results than the Random Forest and the Boosting models for this type of problem.

Features that are most important for predictive models to forecast accurately were Temperature, Hour, Weekend, Pressure, Uv_Index, and Day.

Keywords: Demand forecasting, Shared bikes, Weather data, Machine Learning, Predictive models

Resumo

Título: Previsão da procura por hora de bicicletas partilhadas com base em dados meteorológicos e modelos aprendizagem automática

Autor: Dijora Peja

Esta tese, ao debruçar-se sobre o sistema de partilha de bicicletas em Chicago, pretende contribuir para a implementação de modelos que permitem analisar, com rigor, a procura por hora de bicicletas partilhadas, utilizando componentes temporais e climatéricas.

A variável dependente é o Count, que representa o somatório do número de bicicletas utilizadas por hora. Os modelos preditivos utilizados neste problema de regressão são: Linear Regression, Random Forest, Gradient Boosting, Light Gradient Boosting, Extreme Gradient Boosting, e Multi-Layer Perceptron. A precisão destes modelos é medida através do R2_score, Root Mean Square Error e Mean Absolute Error. No intuito de minimizar o grau de erro são utilizados vários hiperparâmetros para os diferentes modelos preditivos.

Sem hiperparâmetros, o Random Forest alcança as melhores previsões. Contudo, após a utilização de hiperparâmetros, o Gradient Boosting prevê resultados mais precisos.

A precisão do Gradient Boosting aumenta com a utilização de hiperparâmetros, enquanto que o Random Forest não é afetado por eles, de modo significativo.

O segundo melhor modelo ao utilizar hiperparâmetros é o Extreme Gradient Boosting. O modelo de rede neural Multi-Layer Perceptron, apresenta resultados menos precisos do que o Random Forest e os modelos de Boosting.

As características mais importantes para que os modelos preditivos revelem maior exatidão foram: Temperature, Hour, Weekend, Pressure, Uv_Index, e Day.

Palavras-chave: Procura, Bicicletas partilhadas, Dados meteorológicos, Aprendizado de máquina, Modelos Preditivos

Acknowledgement

I want to express immense gratitude to my parents and family for being the main pillar of making the journey to complete my MSc in Business Analytics degree possible. Your belief in me and my abilities and the constant support gave me the confidence to keep pushing forward.

My Lisbon friends, thank you for making this difficult journey filled with amazing and fun times. Your friendship means so much to me, and I am so grateful to have met you and have you in my life. Special appreciation to my friend Catarina for being my 'thesis partner' and going alongside me every day through the process of finishing our theses. And always, forever grateful for my best friends for being the constant source of support, happiness and love even though we are miles apart.

Finally, I want to thank my mentor, Nicolò Bertani, for his willingness and guidance. His expertise and encouragement have been invaluable to me, and I am so grateful to have had the opportunity to work with him.

Thank you all again for your love and support. This road would have been much more difficult without you.

1. Introduction

The rise of environmental and economic concerns has caused an enormous increase in shared transportation, especially in large and crowded cities. The bike-sharing system is a convenient green transportation for citizens of metropolitan cities that contributes to not only less traffic but also a healthy lifestyle. The system was first introduced in Amsterdam, Netherlands in 1965. It certainly ages back into the past, however, its fluke started with the advancement of technology, hence in the area of Intelligent Transportation, after 2010, making it still considered a new and attractive service (Transportation Institute 2014). Nowadays, it is much easier to use this system because of modern technology and the vast accessibility of smart devices.

Due to the spread of the concept of bike-sharing and the population's interest in this type of transportation, city companies need to know the number of bikes to run proper logistics for their clients and prevent over-demand problems. The accurate prediction for the number of bikes rented per hour around the city is a strategy that solves this problem. Hence, this research tackles the issue of hourly demand for bikes in the city of Chicago by analyzing the data on bike count and using predictive models to predict accurately the proper hourly demand for shared bicycles.

This research provides detailed data analyses on the bike count hourly demand for the city of Chicago. The main dependent variable is the *Count* which signifies the sum of bicycles used per hour. Bike usage is highly influenced by exterior factors. Hence, this research presents the effect that time features and weather features have on the variable *Count*. The relationship between these variables is further explained through graphs by finding patterns. This is done by combining datasets of weather-related variables gathered by a weather website through API and a dataset that consisted of every single bike ride in Chicago city. After analyzing the relationship of independent variables with the dependent variable, machine learning models are used to make accurate predictions for hourly bike usage demand. Two types of predictions are done, one by following the random sampling technique for the testing set and the other by using the last part of the dataset as a test set. Statistical measures that are used for evaluating the accuracy of models are R squared for in-sample data and out of sample data, RMSE, and MAE for out of sample data. At the beginning of the predictive model's phase, three main models are used: Linear Regression, Random Forest Model, and Gradient Boosting Model. These models are tried with and without hyperparameters. In order to expand the analysis and search deeper for potential better accuracies three other models

are added. Two of them are part of the boosting family, Light Gradient Boosting, and Extreme Gradient Boosting, and the last model is a deep neural model, the Multi-layer Perceptron Regressor. After using hyperparameters, Gradient Boosting provides the best accuracies when using the random technique for the train and test set, whereas when testing the last part of the dataset, Random Forest achieves the best accurate score.

Bike sharing prediction is essential to the companies and managers however only historical data of the bike usage will probably be enough to forecast the hourly demand. That is why it is essential for managers to seek for external information such as the weather data as in our case to improve their forecasts.

The thesis is divided into the following sections: Section 2 covers the Bike Sharing Context which presents elaborate context behind the bike-sharing system and a literature review of similar work predicting the demand for bike count in other cities/continents; Section 3 includes Data and Modelling explaining the process, predictive models and techniques that are used in the research; Section 4 provides Data Analysis on the dependent variable - Count and its time and weather covariates; Section 5 presents the journey of results depict from predictive models and the last section 6 provides conclusion and discussion on the findings.

2. Bike Sharing System Context

The bike-sharing system is a worldwide used system. By 2016, there were more than 3000 bike-shared stations that operated in 104 cities in the United States of America. Of all the stations, 77 percent of them had a direct connection to another public transportation mode within one block, and 13 percent had a connection within 1 to 2 blocks. Less than 10 percent had either no connection or a connection within more than 2 blocks. These data represent that the bike-sharing system is a convenient transportation mode to use if you want to arrive somewhere fast even if you plan to use it only halfway and use another transportation such as the bus or metro. Around 75 percent of the connections were with bus stations, transit buses were the most typical connection (Firestine, 2016).

This system allows people to avoid car traffic that contributes to an increase in potential productivity, reduction of environmental pollution, decreases transportation costs, and be more physically active throughout the day. The bike-sharing system is an easy system where finding the nearest station that has free bicycles ready to use. They can easily unlock these bicycles with just a barcode and ride to another station around the city that is close to their destination point where they check out and leave the bike. This type of transportation has proven to entail lots of benefits to the users beginning with the most important one, a healthy lifestyle.

Due to the high expansion of bike-sharing systems across the world, except for the wide focus on social and environmental benefits, researchers also tackled the issue of possible congestion in the city. This research was done in 94 urban areas in the US between 2005 and 2014, by implementing a difference-in-difference model with two-way fixed-effects panel regressions. The results of this paper depict that the bike-sharing system shows a significant impact on congestion in general, by concluding that larger cities get better regarding congestion whereas richer cities get worse off (Wang and Zhou 2017). Therefore, a proper prediction of bike demand will certainly settle the congestion problem around cities.

Even though it is vastly used, it's still unbalanced and exposes various analytical problems which is why research are focused on improving this system. Bike-sharing research is divided into mainly three focused categories: city, cluster, and station. City-focused research helps with the big picture of the overall demand around the city and certainly helps companies and the city with better logistics. The cluster-based research creates clusters for stations and builds the predictive models

based on those clusters and station one implements predictive models for every station separately and tries to predict their demand. The two latest categories focus more on the reallocation of bikes to stations with less demand, hence trying to tackle the overdemand per station. City-focused predictive models have shown to be much more accurate than cluster and station-focused models. Cluster and station-focused models are most of the time context-based and highly dynamic, showing to be highly unpredictable which results in less accurate models (Li et al., n.d.). Further, reviews of literature that have tackled the problem of predicting the demand for shared bicycles around the world are explained.

In previous city-focused research done by E, Park and Cho the overall city demand prediction for bikes was done for the city of South Korea. The data operates on the Seoul bike-sharing system to predict hourly cycling demand, while using a one-year data of bike usage and weather data. This research shows that the bike-sharing demand is highly influenced by exterior factors, depending largely on the weather data such as temperature. It depicts that there exists a positive link between the demand for cycling trips and temperature, therefore a dataset that has combined the bike-usage data with weather data presents a rich dataset that has higher chances for building predictive models with high accuracy. The Seoul research follows by using machine learning models such as Linear Regression, Support Vector Machine, Gradient Boosting Machine, and Boosting Trees for building a model which brings accurate results regarding the hourly demand of bike usage in that city. Gradient Boosting Machine was the model that presented the most accurate results. These models depict that temperature and hour are the most essential features of the mode (E, Park, and Cho 2020).

Similar research was done using the Washington Capital Bike Share program data, focusing on the relationship between historical usage patterns and weather data for those specific dates. Linear Regression and Random Forest were used to predict the accuracy of the data. Although Linear Regression is considered a good starting point for predictive modeling, Random Forest presented better results (Feng and Wang 2017).

Additional analyses were done with both the Seoul bike-sharing and Capital Share program data by using slightly different predictive models. The models that were fitted into the data were CUBIST, Classification and Regression Trees, Regularized Random Forest, K Nearest Neighbors, and Conditional Inference Tree. In both datasets, CUBIST was shown to be the most accurate

model. The CUBIST model is a decision tree model that establishes a set of rules which divides the data into subsets and then applied regression models to those subsets for prediction (v E and Cho 2020).

Other research done by Pan and others also tackled the shared bicycles system but by using neural network models such as Recurrent Neural Networks and Long Short-term Memory Models. These models were fitted into the Citi Bike System Data for the location of New York and Jersey City. LSTM presents more accurate data when predicting the demand for bike usage (Pan et al. 2019). However, according to another analysis that has used the Citi-Bike electronic bike usage in New York City, LSTM can be limited as a model since it lacks mining complex spatial-temporal reasoning- an artificial intelligence area that develops high-level control systems that are used for navigating and understanding time and space/location. Since time features are relevant for bike sharing predictions, an additional model STG2Vec has been proposed which in this case learns the spatial-temporal reasoning and then can be fit into the LSTM modeling (Li et al., n.d.).

Additionally, other research regarding the demand for bike sharing decided to approach this problem from another angle. The research proposes a two-phase framework that was used to predict the over-demand clusters (group of stations) by using two-year bike-sharing data and geographic data of New York and Washington. A weighted correlation network was built to model the relationship between bike stations by focusing on a geographic constraint clustering method that aimed to group stations over the network. This framework first estimated the count of rented bikes and returned to each station and then implemented a Monte Carlo simulation that predicted the cluster over-demand probability (Chen et al. 2016). Historical data on Montreal bike-sharing system was also used to predict bike-sharing demand per station – a specific model was implemented per station. However, these models tend to overfit the training data and underfit the data by being too general. Predicting the traffic based on the station/ neighborhood is not very accurate to make a robust model. (Hulot, Aloise, and Jena 2018).

After reviewing similar research on the problem of bike sharing, the following section presents the data explanation and modeling that this research has followed to achieve accurate predictions for the hourly demand for shared bicycles.

3. Data and Modeling

This research aims to make an analysis of the hourly demand of shared bicycle data dependent on weather and time-related data and find the best predictive model which displays an accurate prediction of the hourly demand for bicycle usage. The main variable is that will be predicted is the Count which depicts the sum number of bikes used per hour. It is known from previous research that the hourly demand for bikes is strongly influenced by external factors such as the weather on that particular day, the season, month, holidays, etc. Therefore, in order for the predictive models to achieve good accuracy, this dataset consists of combined data. One part of the data is bike usage-related data; this part was collected by a dataset in Kaggle, consisting of every possible single trip of all bikes for a period of one year in Chicago city. The dataset comprises historical data from August 1, 2021, until August 31, 2022, a total of 396 days. Since we aim to do hourly demand prediction, we aggregated these single rides grouping by the hour, getting the total sum of bikes per hour for that period. Hence, we got data for every hour (24 hours) for 396 days, achieving a total of 9504 rows of bike counts per hour.

The second part of the data is weather data. The weather data is collected from a weather website through an application programming interface (API) which offered detailed information on hourly weather data for the city of Chicago. The API link allowed access to only one day of data at a time, therefore, a loop was created to record every sequential day for the same period as the bike usage data. Figure 1 depicts how the weather website looks and some of the features that it presented. For additional information on the weather data details, you can access the website here.

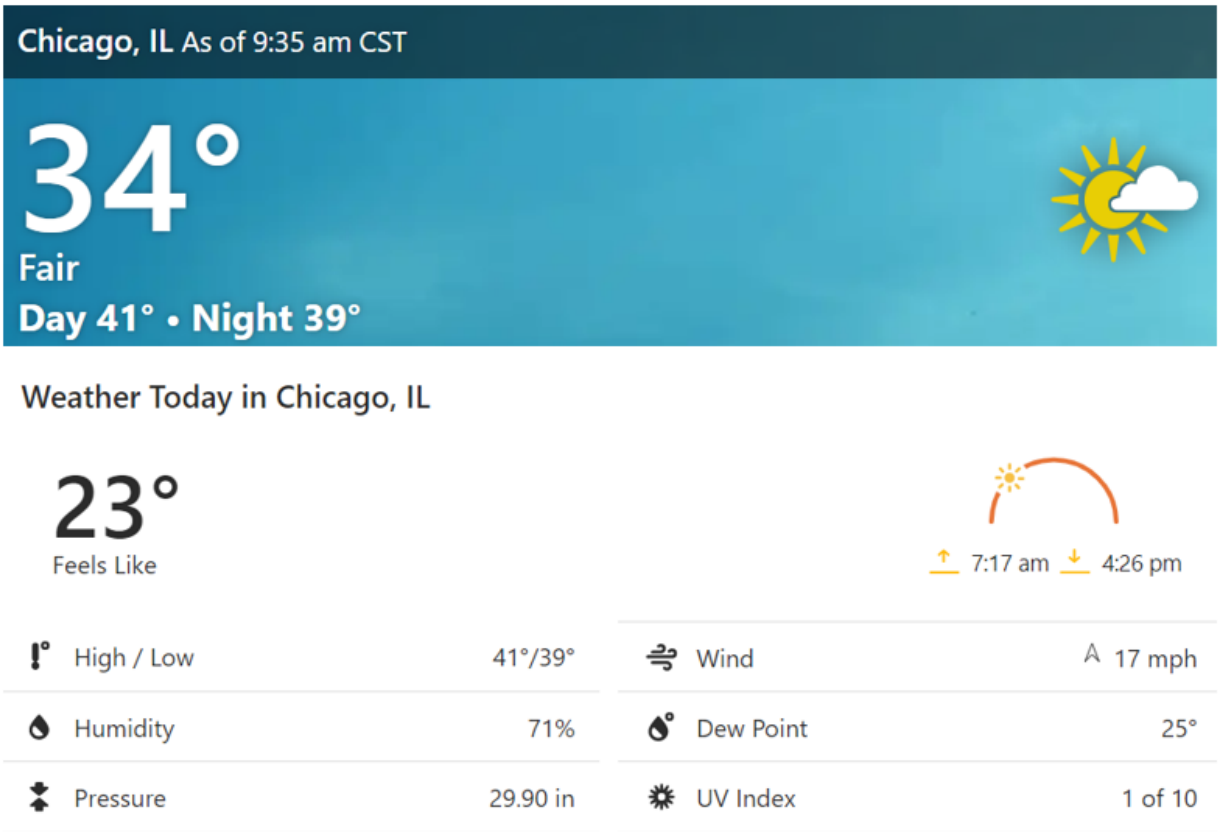


Figure 1 Retrieved from weather website that is used to get the weather data

For further insights on how I implemented the loop for the API or information about the coding part when doing data analysis and predictions, you can find the link of the GitHub repository in Appendix 3 where all the code is downloaded.

3.1. Predictive Models

Six different state-of-the-art predictive models are used to achieve accurate results when predicting the hourly demand of bike count, starting with Linear Regression, followed by more complex models such as Random Forest, Gradient Boosting, Light Gradient Boosting, Extreme Gradient Boosting, and finally, a neural network model such as Multi-layer Perceptron Regressor. The research starts with the Linear Regression model since it is the most simple but concise model, Random Forest which creates a set of decision trees in the training set, and Gradient Boosting, a boosting model that relies on learning from past decision trees. Further, two more boosting models are added, Light Gradient Boosting and Extreme Gradient Boosting and finally a deep neural model for comparisons between the above-mentioned models and neural network models regarding continuous variable problems.

3.1.1. Linear Regression

The first model is Linear Regression. Regression analysis is the main statistical analysis that estimates the relationship of a dependent variable with one or more independent variables. Linear regression requires that the model has linear parameters. The regression analysis of the relationship between one dependent variable and only one independent variable is called univariate regression, whereas the analysis of the relationship between one dependent variable and more than one independent variable is called multivariate regression. Our data includes one dependent variable and 21 independent variables presenting a multivariate regression.

The dependent variable can also be known as the explained variable or predicted variable, while the independent variable can be known also as explanatory variables, or control variables.

The multilinear regression model is written in the following form:

$$y = B_0 + B_1 x_1 + B_2 x_2 + \cdots + B_n x_n + \epsilon$$

Where y is the dependent variable, $B_0, B_1, B_2, \ldots, B_n$ are the regression coefficients while $x^1, x^2, \ldots, x^n$ are the independent variables and ϵ is the error term. In linear regression, it is assumed that ϵ follows normal distribution meaning E (ϵ)=0 and the constant variance partially ∂^2.

The main goals of regression analysis are to estimate a relationship between the explained variable with the explanatory variables, predict the dependent variable based on the set of values of the independent variable, and then find out which variables are more essential when it comes to having more accurate prediction regarding the explanation of Y (Xin, 2009). For this research, the Ordinary Least Squares regression is used to reflect the relationship between the independent variables with the dependent one. Ordinary Least Squares are a standard technique when it comes to estimating the coefficients of Linear Regression. The Gauss-Markov theorem claims that the Ordinary Least Squares Regression must satisfy assumptions in order to produce unbiased estimates. OLS is BLUE: Best Linear Unbiased Estimator. The following first four assumptions that need to be met ensure for linear unbiased estimator whereas the fifth ensures low variance:

- Linear Parameters
- Random Sampling - Sampling distribution is centered on the true population
- Exogeneity- independent variables are not correlated with the error term

- No Perfect Collinearity – independent variables are not perfectly correlated with each other
- Homoscedasticity- Variance of the error term is constant (Hansen 2022).

3.1.2. Random Forest Model

Random Forest is the second method that is used to build a predictive model for the hourly demand for bikes. This method is an ensemble of many decision trees that are randomly created that will produce a final single output. The Random Forest model is flexible since you can fit classification and regression model data into it, is trained and predicted relatively fast, and usually uses one or two tuning parameters for better accuracy.

The main aim of the random forest model is finding a prediction function $f(X)$ which will predict Y where X-s are the predictor variables presented as a p-dimensional random vector $X = (X_1, \dots, X_P)^T$. This function is determined by a loss function and aims to minimize the expected value of that loss.

The Random Forest algorithm works by bagging, which is a term used when you randomly divide the features and rows that you are going to select per every tree. In this way, Random Forest makes decision nodes which are counted as means of splitting the data. There is one main tree node, the root, that entails the whole predictor space. From the main tree, other splits (trees) are created by using a sequence of binary partitions on independent variables. When the splits are formed, new nodes are created which will be treated the same as the original node. This course of action continues recursively until certain criteria are met that stop the procedure. Finally, the nodes which are not split are called the terminal nodes and create a final partition in the predictor space. That is when the predicted value is created by observing all the terminal nodes by using the average of them for regression. Figure 2 presents a simple visualization of how random forest works (Cutler, Cutler, and Stevens 2012).

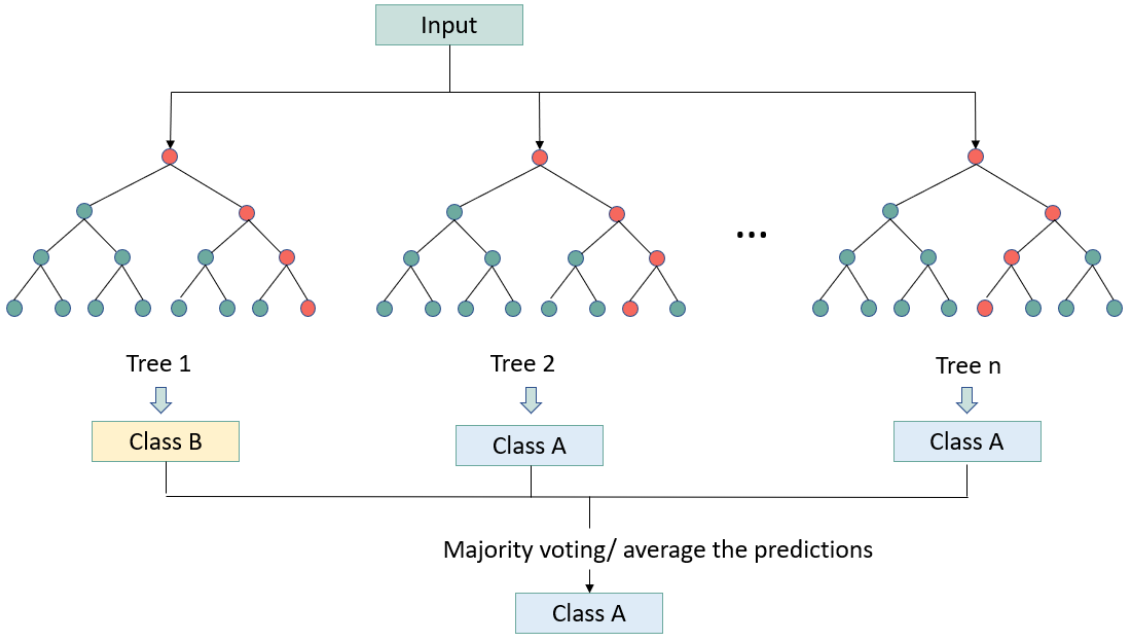


Figure 2 Visual explanation of Random Forest

3.1.3. Boosting Models

To begin with, the boosting models use algorithms that combine several simple models and create a more complete model which indeed results in better prediction. These simple models are the weak models by which you try to learn and improve.

3.1.3a Gradient Boosting

Gradient Boosting is a powerful machine-learning model from the boosting family which is very adaptable since it can be customized for various practices but at the same time simple to use. While Random Forest relies on averaging the model on the ensemble, boosting models work differently by adding new models sequentially to the ensemble. This a more constructive strategy where for every iteration, we have a new base-learner model that is trained by giving attention to the error that is learned so far by the ensemble. This way they infuse the learners that are weak and make them powerful learners. The main goal is to minimize the loss function of the model. So, the model starts at the beginning to predict only the mean of the Y-dependent variable and gradually improves itself by learning from past errors by adding these weak learners using gradient descent. Gradient

descent is an optimization algorithm that finds the local minimum of a function, in this case, the loss function.

Gradient boosting allows different loss functions to be used for the model, therefore giving freedom to the researchers to try different ways until they arrive at the final model which achieves the best results when compared to the conventional single machine learning models. Depending on the type of dependent variable, the loss functions differ. Loss functions that are used for the continuous variable are the Gaussian L2 loss function, Laplace L1 loss function, Huber loss function, and Quantile loss function. Whereas for categorical variables, loss functions that are used are: the binomial loss function and Adaboost loss function. The most commonly used loss function for the continuous variable is the squared-error L2 loss whose derivative is the residual y-f. In this case, the Gradient Boosting model does residual refitting, penalizing large deviations from the targeted output and ignoring small residuals (Natekin and Knoll 2013).

3.1.3b Light Gradient Boosting

Light Gradient Boosting is another decision-tree algorithm, a form of GBM which is implemented first by Microsoft Research. It has proven to be a powerful algorithm that is fast and presents high performance when it comes to predicting regression and classification cases. LGBM uses the gradient-based one-side sample technique which is efficient, especially for managing large data sets and exclusive feature bundling which prevents overfitting of the data. This algorithm follows the tree leaf-wise growth approach where it splits the tree leaf-wise in the best fit compared to other boosting algorithms which split the tree depth-wise or level-wise, by decreasing the loss and resulting in better accuracies. Boosting algorithms that use the level-wise approach split all the leaves that are in the same layer at the same time, even though they contain different information whereas algorithms such as LGBM that follow the leaf-wise approach, on one layer, only split the leaf that has the most information (Abdulalim Alabdullah et al. 2022).

3.1.3c Extreme Gradient Boosting

Extreme Gradient Boosting is a new algorithm part of the boosting team, so a distributed boosted decision tree which gathers information from weak learners and uses additive training approaches to create powerful learners. It is another effective algorithm for classification, regression, and

ranking problems. This algorithm is usually considered a new improved version of the Gradient Boosting Algorithm. Compared to LGBM which uses the leaf-wise approach, this algorithm uses the level-wise approach (Abdulalim Alabdullah et al. 2022).

3.1.4. Multi-layer Perceptron

Artificial Neural Networks can be another proposed system of Machine Learning that can be used for regression problems. The artificial neural network that is used for this bike-sharing problem was the Multi-Layer Perceptron (MLP) which provides a continuous output layer suitable for regression problems. An Artificial Neural Network is a computing system whose main operation is based on the analogy of biological neural networks. So, ANN tries to imitate and work the same way the human brain functions.

The analogy with the human brain comes from the following ANN, usually called a Neural Net which is like a web that contains a large number of units that are interconnected with each other and pass information between them. These units are called nodes or neurons since they are replicas of neurons in the biological term. Biological neurons are nerve cells that process information. Except for neurons, Artificial Neural Networks also contain sets of links/edges which connected these nodes with each other known as 'Connections'. These nodes produce computations where the connections then convey to other nodes as signals. Every connection link is associated with these signals which are numbers called Weights. (Dongare, Kharde, and Kachare 2008). The MLP learning algorithm works by first initializing the network with weights put to random from -1 to 1 and presenting the first training pattern with its output. The first output is compared to the target out and then it propagates the error backward by correcting the output later of weights and input weights. This process is repeated until the error decreases significantly and the final output's performance is close to or the same as the desired output (Noriega 2005).

3.2. Model Development

To find the most accurate model for this dataset, data is divided into two categories, training and testing part. The majority of data, more specifically 80 percent, is used for training the models and the other 20 percent is used for testing. Two methods are used for testing and training. The main method of splitting was done by using the function Train_test_split from the Sklearn library to divide the data into training and testing sets. This function randomly selects 20 percent of the data

as a testing set and the other part as the training set. Whereas the alternative method was to use the last 20 percent of the data, in our case last two months of data as a testing set and the other part as the training set.

The testing set was selected randomly through the data, and the other part was used for testing. Despite the method differences, the dimensions of these two sets were the same and are presented the table 1.

Table 1 Dimensions of Training and Testing Set

Data Split	Data Points
Training set	7603 rows
Testing set	1901 rows

Standard Scaler, Minmax Scaler, and Robust Scaler are three scaling methods that are tried for the normalization of the data in predictive models. Robust Scaler is used on the final models. The accuracy of the models was determined by looking at statistical measures such as R squared for Root means squared error, and Mean absolute error:

- **R Squared** depicts the coefficient of determination which is the proportion of how much the dependent variable is described by the independent variables. It analyzes how the changes in one variable are explained by changes in another variable, hence in this case, how changes in independent variables will explain the changes in the demand for bike count. R Squared is a statistical measure that we used for in-sample measurements and out-of-sample measurements.

  The formula for in-sample measures of R squared is below where n is the sample size, *x* is the independent variable and *y* is the dependent variable. This is used when running an OLS Regression

$$R^2 = \left[\frac{(n\sum xy - \sum x\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}\right]^2$$

  On the other hand, for out-of-sample measurements, for model prediction accuracy, R Squared is calculated by using the Sklearn metrix R2_Score which presents the R squared

between y_pred which are predicted values from the model.predict(X_test) and the actual values of y_test.

$$r2\_score(y_{pred}, y_{test})$$

- **Root-mean-square-error** is the square root of the average between the predicted values from the model with the true values on the data. The formula of RMSE is found below, where P_i is the predicted value of an observation from the dataset, $O_i$is the observed value from the dataset and *n is* the sample size of the dataset.

$$RMSE = \sqrt{\frac{\sum(P_i - O_i)^2}{n}}$$

- **Mean absolute error** is the absolute difference between the actual and predicted values. Similarly, P_i is the predicted value, $O_i$is the observed value and *n is* the sample size.

$$MAE = \frac{\sum|P_i - O_i|}{n}$$

3.2.2. Hyperparameters

Hyperparameters are used in the models to maximize the model performance. These hyperparameters are selected by using the function GridSearchCV which selects the best parameters for the prediction models. Firstly, predefined values as lists of certain hyperparameters are fitted and then this function selects the best value from the list of those hyperparameters by running multiple trials with all combinations. The selection of the best hyperparameter is done by using the Cross-validation method (CV = 10) which is a method that uses different slices of the dataset for training and testing the model on various iterations. This way, the accuracy for each combination of hyperparameters from the list is achieved, and finally, the function chooses the set that has the best performance.

Table 2 presents the list of parameters that are tried for predictive models to achieve higher accuracy and the values that are fitted into the models by GridSearchCV before the selection of the best ones and their explanation.

Table 2 Hyperparameters that are tried for every prediction model

Hyperparameters	Values	Models	Description
n_estimators	250, 300, 500, 1000	GBM, RF, LGBM, XGBM	Number of trees used (splits)
max_depth	2, 4, 6	GBM, RF, LGBM, XGBM	The maximum depth that the tree will be built – helps to avoid overfitting
learning_rate	0.01, 0.1, 1	GBM, LGBM, XGBM, MLP	How fast learns the values of a parameter estimate
Loss	ls, huber, quantile	GBM	The loss function to be optimized
random_state	0, 18, 42	GBM, RF, LGBM, XGBM, MLP	Reproduce the same train test split each time
'max_bin'	100, 255, 300	LGBM	Maximum of bins that variables are bucketed into
num_leaves	10, 25, 31	LGBM	Number of leaf-wise
Activation	Tanh, relu	MLP	A function that is put in a layer of the neural network (input, hidden layers, output)
Solver (optimizer)	Adam	MLP	Optimization algorithm that is used to update the weights in the training data
Hidden layer	128, 64, 32	MLP	Layers inside the input and output
Batch_size	20, 25, 30	MLP	Number of training samples used in one iteration
Max_iter	25, 50	MLP	The number of epochs- complete passes of the training set through the algorithm

4. Data Analysis and Transformation

This section presents the data transformation, data analysis of the features in the dataset and variable selection for the main model that will be used for prediction.

4.1. Data Transformation

The complete data is a combination of time-related variables and weather-related variables. Weather-related variables are: Temperature put as a temp in the dataset, Perceived temperature (feels_like), Dew Point (dewPt), Wind Direction (wdir), Wind Speed (wspid), Hourly Precipitation (precip_hourly), Could Type (clds), Heat Index (heat_index), UV Index (uv_index), Weather condition types (wx_phrase), and Weather Condition (w_condition). Time-related variables originally consisted only of the Date, from which then are extracted separate variables such as Hour, Day, Month, Season, Year, Weekdays, and Weekend (is_weekend). An additional feature Holiday (is_holiday) is added giving information about whether that particular day is a holiday or not based on the National Holiday Program of Chicago City. Table 3 presents a detailed description of the variables in the dataset.

None of the variables had null values except for the Wind Direction, hence this variable was not included in the final model. Appendix 2 presents more information regarding null values, whereas Appendix 1 presents information about the mean, minimum values, maximum values, and standard deviation of every feature.

Table 3 Details of the variables in our data

Variable name	Variable Type	Description
Count	Continuous	The total sum of all bike trips in that particular hour
Date	DateTime	The date on that day (dd/mm/yy)
Hour	Continuous	Specific Hours of the day
Day	Continuous	Sequential ID for every day from the beginning until the end of the dataset (1-396)
Month	Category	Month on that particular date
Season	Category	Season on that particular date

Year	Category	Year on that particular date
Temp	Continuous	The real temperature at that moment- in Fahrenheit (F)
Feels_like	Continuous	Perception of the temperature by people - in Fahrenheit (F)
Wx_phrase	Category	Detailed weather conditions, whether it was a clear sky, light rain, heavy rain, light snow, heavy snow, storm, etc.
W_condition	Category	Good or bad weather (wx_phrase variable broken into two main categories)
dewPt	Continuous	Dew Point at that hour
Wdir	Continuous	Wind direction
Wspid	Continuous	Wind speed
Precip_hourly	Continuous	Hourly Precipitation
Clds	Category	Cloud Conditions: SKC (sky clear), FEW (1-2 octas), SCT (3-4 octas), BKN (5-7 octas), or OVC (8 octas).
Heat_index	Continuous	The heat index (HI) is an index that combines air temperature and relative humidity
Uv_index	Continuous	The ultraviolet index
Week-days	Category	Every day of the week
Is_weekend	Dummy	Dummy variable, a weekday or a weekend day (0: Weekday, 1: Weekend)
Is_holiday	Dummy	Dummy variable, a working day or a national holiday (0: No holiday, 1: Holiday)

4.2. Data Analysis

Before implementing the predictive models, it is needed to first analyze and understand the data, especially the patterns of bike usage. Figure 3 depicts the distribution of the dependent variable – 'Count' the bike usage per hour which is a continuous variable, with only positive values. The Count of bikes ranges from only 1 bike per hour, minimum value, to 3414 bikes per hour, the

maximum value, with an average of 704 bikes per hour, and a median of around 500 bikes. As observed from the Figure 3 and also proven by the z-score formula, the Bike Count variable does not have any outliers. However, it is seen that the shape of the variable does not comfort the assumption of normal distribution.

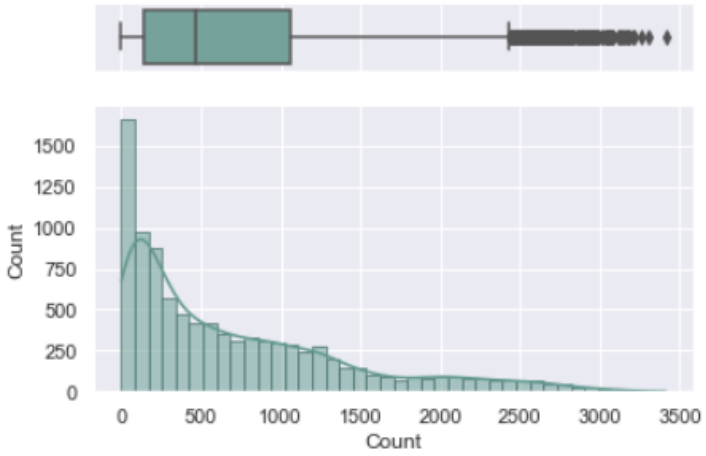


Figure 3 Distribution of the dependent variable Bike Count

Figure 4 presents the mean of bike counts per hour throughout the days for the whole period of the data. As seen, the average bike count per hour is higher on summer days compared to days of other seasons. The highest average bike count is in the summer of 2021, in August, while the lowest averages are shown during the winter of 2022, in February. The maximum bike count per hour point is on July 17th, 2022, with 3414 total bikes at 5 pm on that day, whereas the minimum bike count per hour is found on some of the winter days when only 1 customer uses a bike per hour.

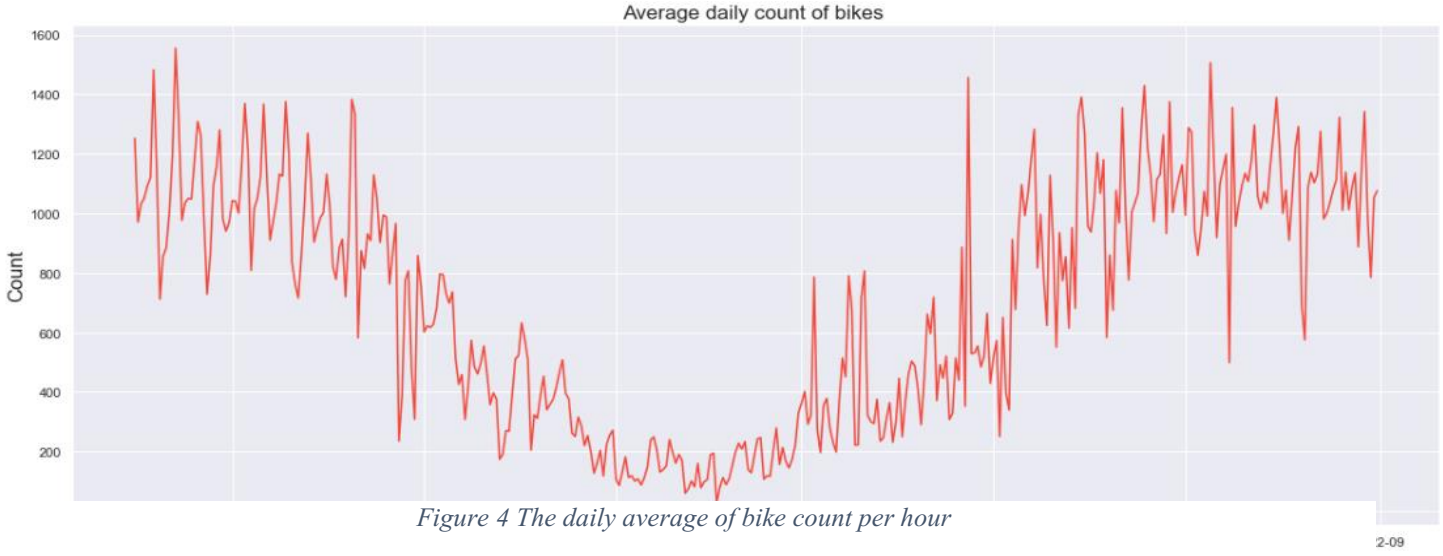


Figure 4 The daily average of bike count per hour

Figure 5 depicts the seasonal and monthly demand for bikes per hour. Certainly, the Figure 4 graph is strongly supported also by Figure 5. There is a gradual increase in the demand from Spring towards the months of Summer. The high demand of bikes per hour is seen from the month of June until September, with July reaching its peak with the highest demand. After summer month is followed by a gradual decrease when entering Autumn and a more evident decrease in Winter- with the lowest demand depicted in January. Winter is certainly low regarding bike-sharing usage.

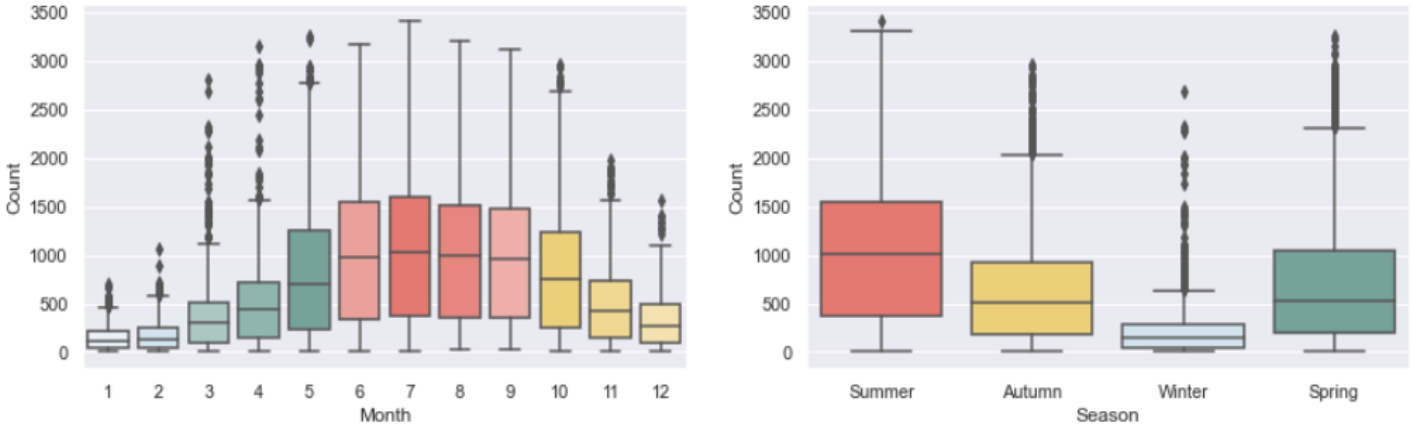
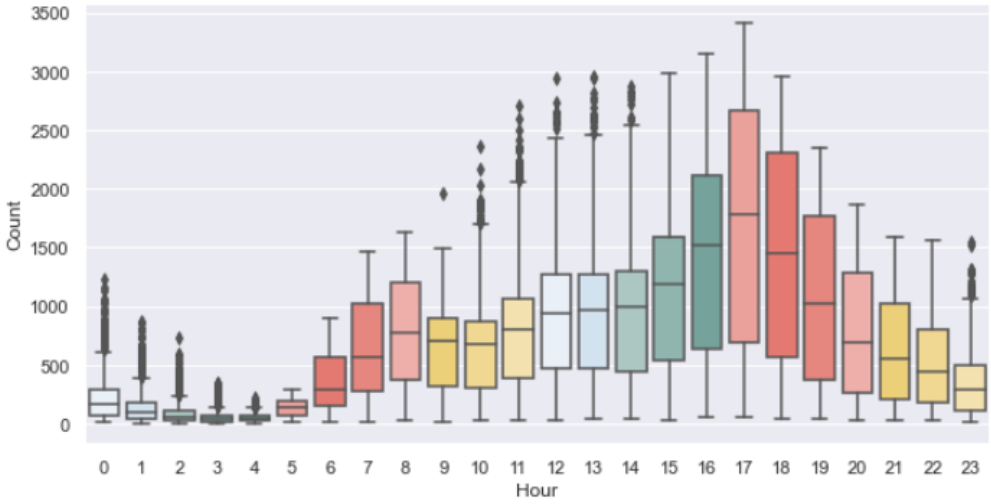


Figure 5 Hourly demand for bike usage depicted in Months and Seasons

Figure 6 presents the bike usage demand throughout the hours. It is evident that customers are more prone to use bikes throughout the daylight since it is presented a sharp decrease in demand from midnight until 6 am. The morning peak is at 8 am, however, it seems that customers prefer to use bikes more during the mid-day - an increase in the demand is presented from 4 until 7 pm. The



highest peak of the day is at 5 pm. After midnight, from 1 am until 5 am, bike usage drastically decreases. This pattern of bike usage per hour was the same every month.

Figure 6 Bike usage demand for every hour

Figure 7 presents the hourly bike count throughout all weekdays and a comparison between workdays and weekends. There is a clear pattern that there is a slight increase in bike demand during the weekend, compared to work days. The highest hourly demand for bikes was shown on Saturdays, whereas the lowest usage was on Mondays. However, bike usage during workdays was similar throughout all days, with only small differences. Even though there is a difference in the overall demand for bikes between weekends and weekdays, the median is the same in both categories.

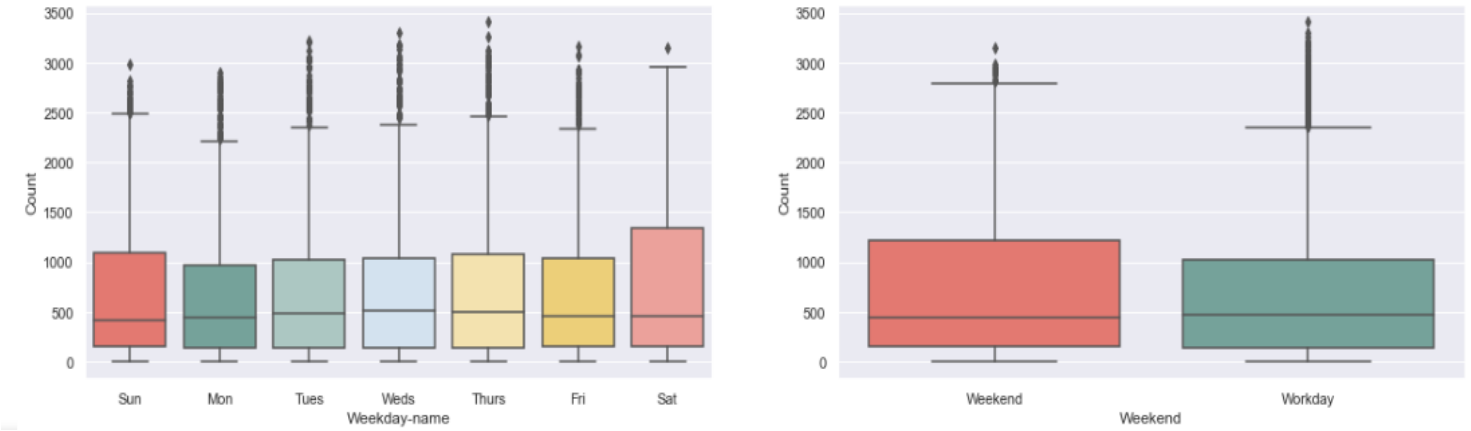


Figure 7 Hourly demand for bike usage throughout the weekdays and comparison between workdays and weekends

Figure 8 presents the bike count demand throughout hours by comparing holidays with normal days. The average bike usage is usually bigger when it is a working day and not a holiday. For a working day, the pattern is the same as depicted in Figure 7, so an increase of demand between 4 pm to 7 pm with a peak at 5 pm. Whereas, on holidays there is a different demand for bikes. The highest demand is during the day from 12 pm until 6 pm. However, there is quite a high variance throughout the hours on holidays which means that customers are more flexible and unpredictable during the holidays.

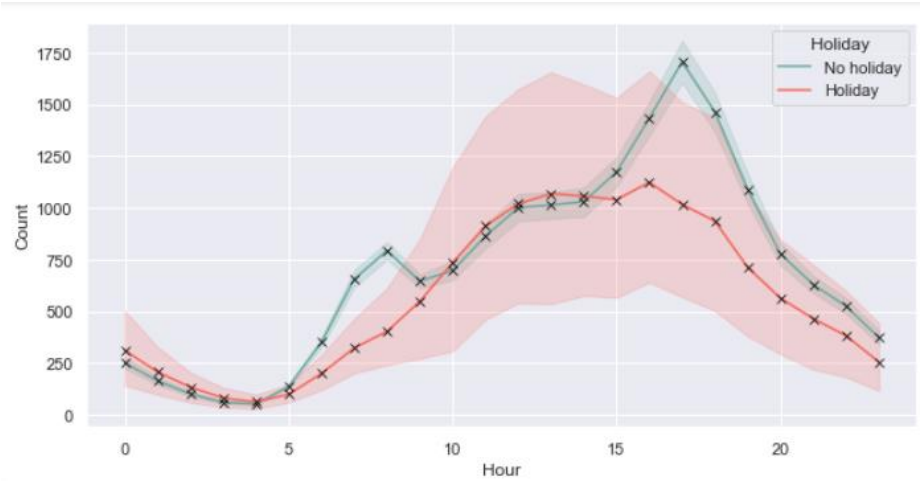


Figure 8 Bike usage demand for every hour by comparing holidays and normal days

Figure 9 depicts important weather-related independent variables that are incorporated into our dataset which might affect the hourly demand for the bike count. Therefore, it is important to analyze them. As observed, Temperature, dew Point, and Pressure are almost normally distributed, whereas the Ultra Violet is not and shows that in most cases the UV Index is 1.

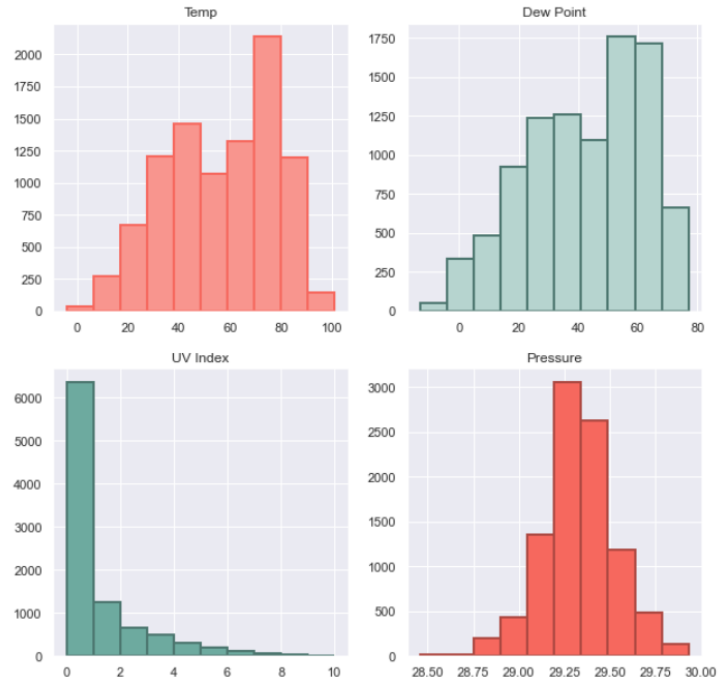


Figure 9 Distribution of the independent weather-related variables: Temperature, Dew Point, UV Index, and Pressure

Figure 10 presents the relationship between the dependent variable, bike count with temperature. It is noticeable that there is a direct positive link between the average bike count and temperature. As the temperature increases, the average bike count also increases. This pattern is not so stable for high temperatures, above 90 Fahrenheit, where the bike counts start to decrease.

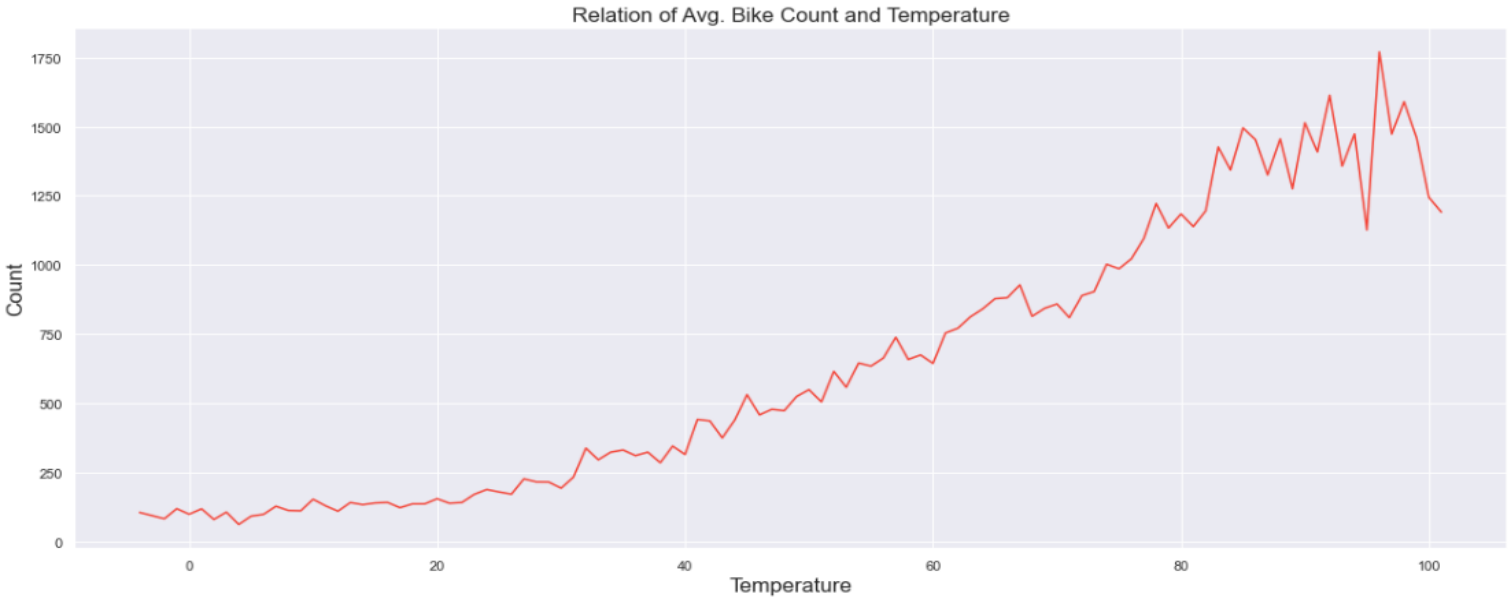


Figure 10 Relationship between daily average of bike count per hour and Temperature

Figure 11 presents the relationship between the daily average bike count per hour with Pressure. There is seen a positive link between them until a certain value of pressure. So, as pressure increases, bike usage also increases until the pressure is 29.3. After that point, as pressure increases further, the bike count starts to drastically decrease. This concludes that customers tend to avoid bicycle use when there are high air pressures (>29.4).

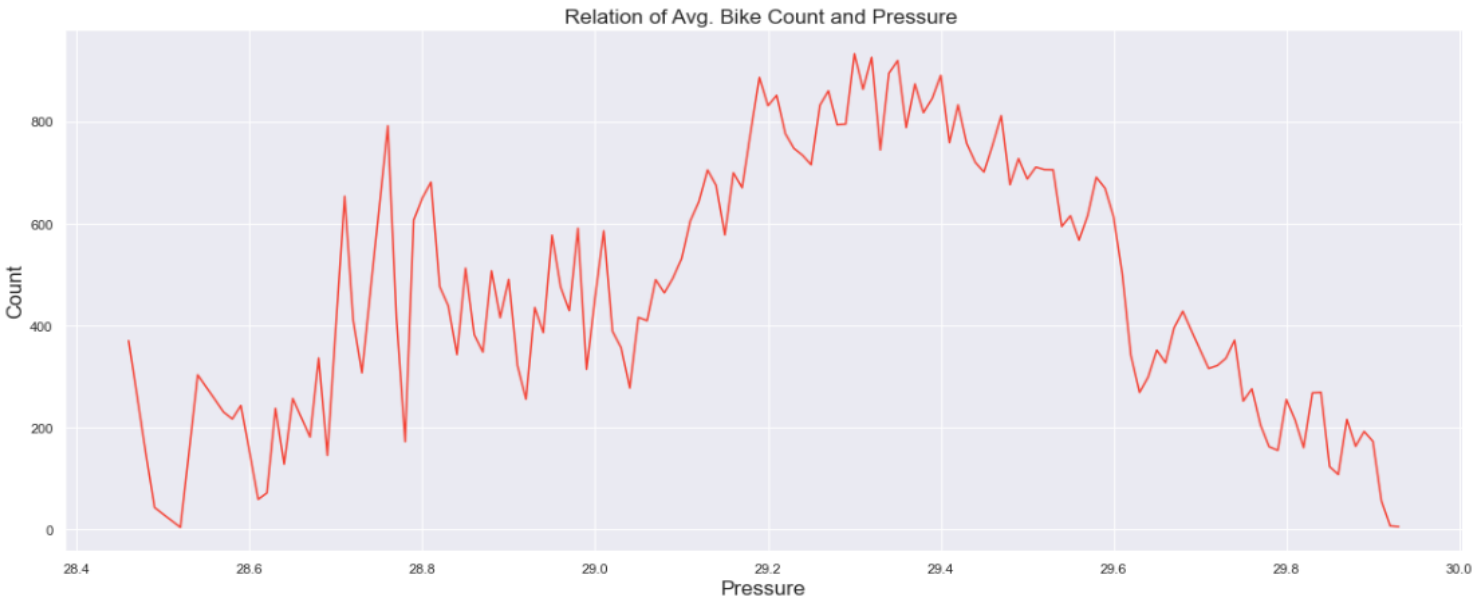


Figure 11 Relationship between the daily average of bike count per hour and Pressure

4.2.1. Transformation of dependent variable *Count*

Even though predictive models can be robust to violations of assumptions such as normal distribution of the data, often analyses still benefit from prior data transformations. Transformation of data are tools that are used when doing quantitative analysis to make sure that assumptions such as normal distribution and homoscedasticity hold.

These transformations are important; however, they certainly change the nature of the variable, hence, the interpretation of the variable will change. That is why detailed analyses are done before choosing the transformation for the main variable Bike Count. Different data transformation types are tried to help achieve a closer version of normal distribution. Since the Bike Count variable contains does not have any negative values and is rightly skewed, Logarithmic, Square Root, and Box-cox are three transformations that are used on the variable to reduce the skewness of the original data. The idea behind the Square Root transformation is compressing high values and spreading out the low ones. When applying this transformation, the square root of every value is done, a method that is proven to be effective for normalizing Poisson distributions. Logarithmic transformations are more common for variables that are highly influenced by independent factors, which seems a right fit for our variable. The logarithm is the exponent for which a base number must be raised for it to achieve the original number, so logarithmic transformations contain power transformations. The log transformation usually uses the base 2, 10, or natural log (base e), and in

this case, we are using the natural log. This transformation is widely used and it is easily interpretable. Box-cox transformation is a transformation established by Box and Cox where:

$$y^{\lambda} = \frac{(y^{\lambda}-1)}{\lambda} \quad \text{where } \lambda \neq 0;$$

$$y^{\lambda} = log_e(y_i) \text{ where } \lambda = 0;$$

As seen, this transformation relies highly on lambda. If the lambda is 0, then a log transformation is a better fit, whereas if the lambda is 1, there is no need for transformation. The calculated best lambda for the Bike Count variable was 0.24, therefore further analyses were done to know which transformation is better suited for this variable (Osborne 2010).

Figure 12 presents the distribution of all the transformation forms of Bike Count: Normal, Squared, Log, and Box-Cox, and their respective Probability Plots. After looking at the histograms and probability plots in figure 12, it can be observed that box-cox and log transformation are the two most appropriate transformations that almost achieved normal distribution. As observed, box-cox provides a closer version when it comes to a normal distribution than the other transformations, however, when looking at the probability plots, it is evident that the difference between logarithmic transformation and box-cox transformation is really small. As mentioned above, these transformations of the data, change the interpretations of the variable. Hence, in order to obtain a more simplified and familiar interpretation of the data, logarithmic transformation is chosen as a transformation type when building the predictive models.

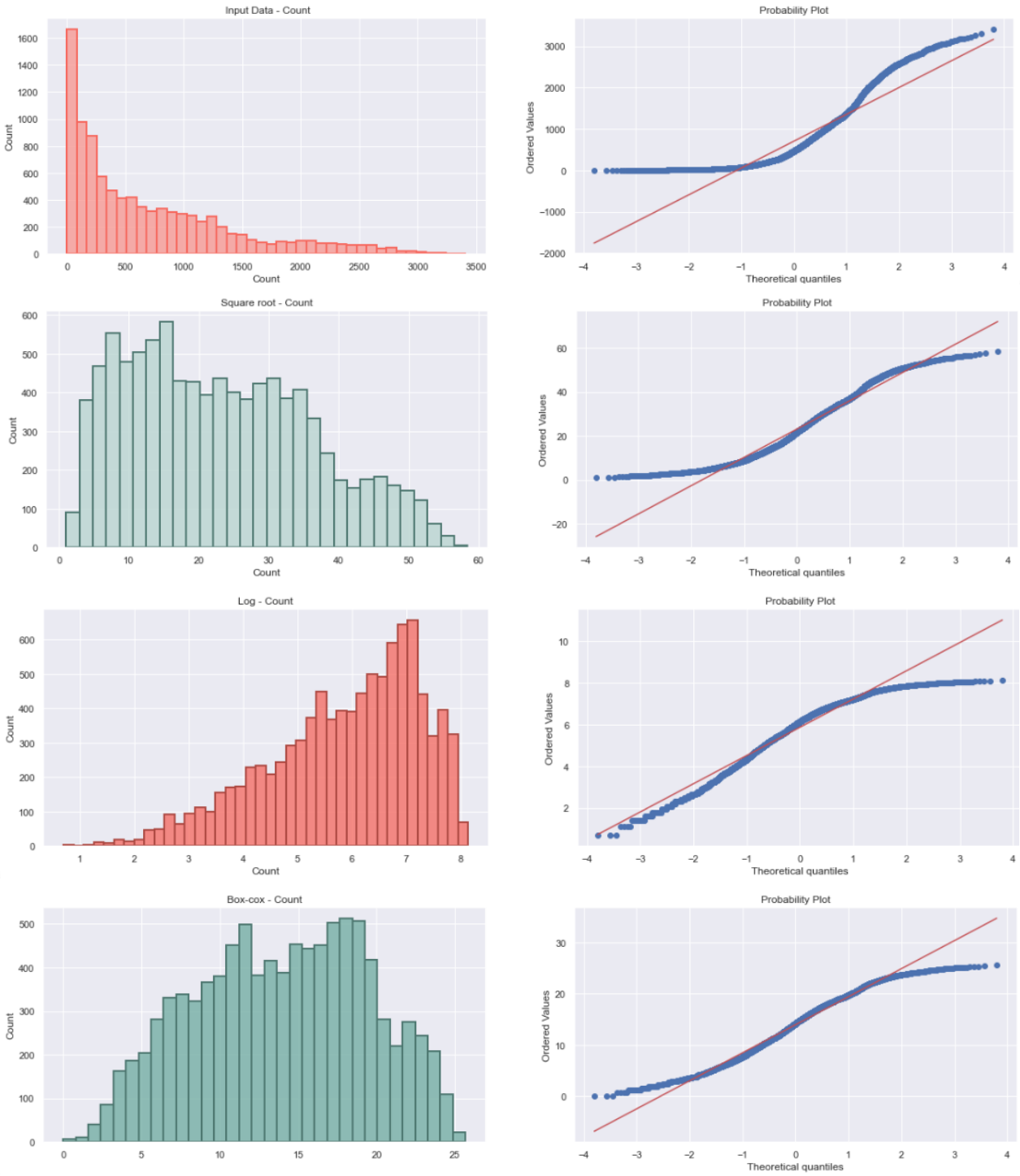


Figure 12 Distribution of Bike Count in all transformation forms: Normal, Squared, Log, and Box-Cox and their respective Probability Plots

4.3. Covariate Correlation and Variable Selection

Before prediction, the variables are analyzed to check whether there exists a high correlation between each other. Correlation is a statistical measure that explains when two variables change at a constant rate. The goal of regression analysis is to know the relationship between an independent variable with the dependent variable, by holding everything else constant. However, when two independent variables are highly correlated, this means that change in one of them is linked to the change in the other, which causes a problem for the 'Ceteris Paribus' goal. Therefore, it is preferred to not include variables which are highly correlated when running predictive models because it will cause biasedness in the accuracy.

Table 13 depicts the correlation matrix between variables. Analyzing the data and observing the correlation table, it can be seen that weather-related variables, *Temperature*, Perceived Temperature, Dew Point, and Heat Index are highly correlated with each other (>0.9), therefore only Temperature is selected to be part of the final model, as it is the most representative feature compared to the others.

Season and *Month* are also highly correlated (>0.8). Since it can be depicted in Figure 5 that the demand of bike usage differs in every month, only the feature Month was left on the final model.

Other highly correlated variables were *Weekdays* and *Weekend,* and Cloud Types and Weather Conditions. Since the demand for bikes is similar throughout the weekdays and there is shown a difference in demand when comparing weekends and workdays, only *Weekend* is left on the regression. Moreover, Weather Condition was chosen instead of Cloud Types since it is seen to be more important for predictive models.

Figure 13 Correlation Matrix presenting the correlation of features with each other

5. Results

The following section presents the results of predictive models regarding the bike-sharing demand. It begins by presenting three predictive model accuracies on the final model, final model variations, and then an expansion of three additional models for comparisons.

After doing this a thorough analysis of the variables, the main model contains the following features: *Hour, Temperature, Pressure, Windspeed, Hourly Precipitation, UV Index, Month, Day, Holiday, Weekend, Weather Condition, and Log Bike Count.*

The predictions journey begins with three main predictive models: Linear Regression, Random Forest and Gradient Boosting. Table 4 presents the accuracy of each of these predictive models on the final model. The accuracy for the out-of-sample data is measured by the R2 score, Root Mean Squared Error, and Mean Absolute Error. It can be seen that the predictive model with the highest accuracy was Random Forest with a R2 score of 95 percent, RMSE of 0.3 and MAE of 0.21. Gradient Boosting is the second best model, whereas Linear Regression presents accurate results of only slightly more than half of the data (55 percent).

Table 4 Evaluating the accuracy of Linear Regression, Random Forest, and Gradient Boosting with R2_Score, Root Mean Squares Error, and Mean Absolute Error for the main model

Predictive Models	R2_Score	RMSE	MAE
LR	0.552	0.90	0.74
RF	0.950	0.30	0.21
GBM	0.925	0.37	0.27

Scatter plots in figure 14 depict the variance between the actual and predicted value for each of the models: Linear Regression, Random Forest, and Gradient Boosting for the final model. From the scatter plots also, it can also be observed that Random Forest works slightly better, since a smaller variance in the linear line between the actual and predicted values when compared to the Gradient Boosting model, hence the predicted values are closer to the actual ones.

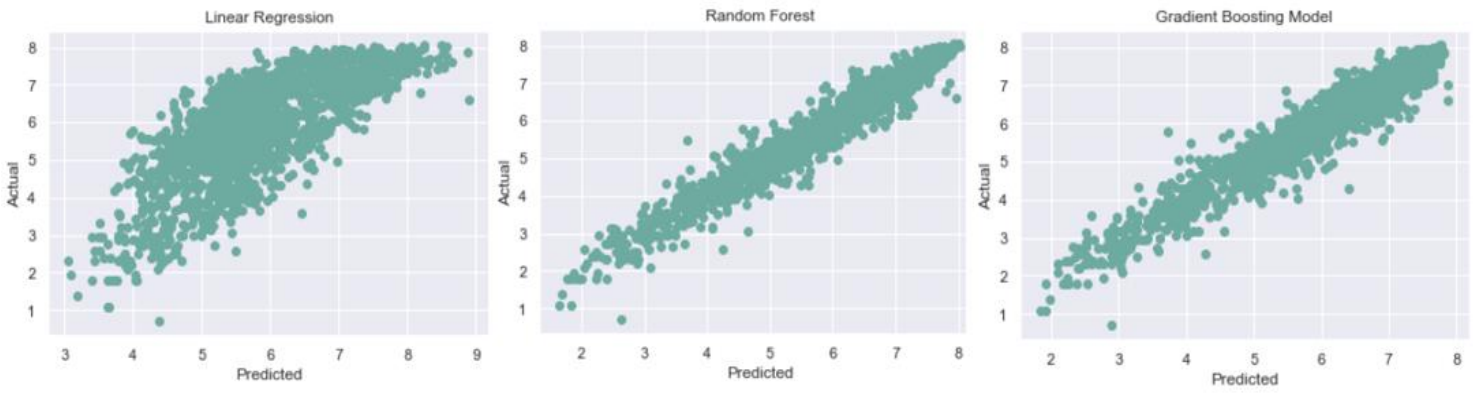


Figure 14 Scatter plots of actual and predicted values for Linear Regression, Random Forest and Gradient Boosting

5.1. Main Model Variations

This subsection presents variations of the main model and checks which of the variation provides more accurate results. Table 5 presents three variations of the model fitted into the Ordinary Least Squared Regression. Model 1 is the simple model, Model 2 depicts the month as a dummy variable that is omitted from the table, and Model 3 has both features, month and hour as dummy variables in order to capture a better effect of these features. Both dummy variables in Model 2 and 3 are omitted from the table for easier comparisons of the coefficients. Creating dummies helps to grasp which of the options is more important, whether a separate effect per every month and every hour or the effect between changing from one month/hour to the other.

It can be observed that a separate effect for every month and hour explains better the dependent variable, count_log according to the OLS regression. To compare the accuracy of these in-sample data, we focused on the R-squared of the model. R Square increases when adding dummy variables, from 55 percent to 58 percent in Model 2. Whereas Model 3 presents a significant increase with 87 percent explainability of the dependent variable. The vast majority of variables in the three model variations are shown to be statistically significant.

Table 5 OLS Regression Results for Model Variations 1, 2, and 3

OLS Regression Results

	Dependent variable:Count_log		
	Model 1	Model 2	Model 3
const	0.968	5.423***	-6.691***
	(1.689)	(1.724)	(0.973)
Hour	0.078***	0.083***	
	(0.002)	(0.002)	
Month	0.060***		
	(0.003)		
temp	0.021***	0.001	0.020***
	(0.001)	(0.001)	(0.001)
Is_holiday	-0.212***	-0.041	-0.116***
	(0.057)	(0.055)	(0.031)
Is_weekend	0.162***	0.130***	0.154***
	(0.021)	(0.021)	(0.012)
W_conditions	-0.298***	-0.349***	-0.262***
	(0.034)	(0.033)	(0.018)
precip_hrly	-0.060	-0.257	-0.507***
	(0.322)	(0.312)	(0.174)
pressure	0.094*	-0.048	0.344***
	(0.057)	(0.058)	(0.033)
uv_index	0.221***	0.227***	0.028***
	(0.008)	(0.008)	(0.006)
wspd	-0.028***	-0.021***	-0.009***
	(0.002)	(0.002)	(0.001)
Day	0.000**	-0.000	-0.000
	(0.000)	(0.000)	(0.000)
Observations	9,504	9,504	9,504
R^2	0.551	0.581	0.869
Adjusted R^2	0.551	0.580	0.869
Residual Std. Error	0.932 (df=9492)	0.900 (df=9482)	0.504 (df=9460)
F Statistic	1059.851*** (df=11; 9492)	626.828*** (df=21; 9482)	1461.969*** (df=43; 9460)
Note:			*p<0.1; **p<0.05; ***p<0.01
			Model 2 - Omit: Month_dummy
			Model 3 - Omit: Month_dummy + Hour_dummy

Having a separate effect for every category of features Month and Hour seems to be important for OLS Regression. Table 6 shows that Linear Regression presents similar accuracies even when using out-of-sample data. However, table 6 shows that this pattern does not follow RF and GBM with out-of-sample data, especially when making variable Hour as a dummy since it worsens the

accuracy of the predictive models. When comparing Model 2 with Model 1, RF and GBM accuracies do not change that much with a slight decrease to 94.8 and 91.8 percent, respectively.

In contrast to Linear Regression, in Model 3, Random Forest and Gradient Boosting predictions worsen. The R2_score accuracy of RF and GBM sharply decreases. The accuracy of Random Forest decreases to 91.8 percent and RMSE increases to 0.39 and the accuracy of Gradient Boosting decreases to 88.3 percent and RMSE increases to 0.47.

Table 6 Evaluating the accuracy of LR, RF, and GBM for Model 2 and Model 3

Predictive Models	R2_Score	RMSE	MAE
LM – M2	0.58	0.87	0.71
LM – M3	0.867	0.71	0.37
RF – M2	0.948	0.31	0.22
RF – M3	0.917	0.39	0.26
GBM – M2	0.918	0.39	0.35
GBM – M3	0.883	0.47	0.38

Since the accuracy of the variations of the model varies a lot depending on the predictive model, hyperparameters tuning is done for Random Forest and Gradient Boosting.

5.2. Results with final model after hyperparameter tuning

Hyperparameters are tuned to help predictive models achieve better predictions. Hence, hyperparameters are added to Random Forest and Gradient Boosting when fitting into the three variants of the final model.

Firstly, it can be observed from table 7 that after adding hyperparameters Gradient Boosting accuracy overpasses the accuracy of Random Forest. Additionally, the best accuracy of GBM is achieved in Model variant 2, therefore Table 2 presents accuracy measurements of the predictive models for variant Model 2.

On the contrary, it seems that hyperparameters are insufficient when it comes to Random Forest for this problem since the accuracy of Random Forest is the same for Model 2 with and without hyperparameters, whereas the RSME increases after tuning hyperparameters.

Moreover, since Gradient Boosting has the best accuracy after hyperparameter tuning, two more models from the boosting family are added to the comparison table 7 Light Gradient Boosting and Extreme Gradient Boosting. For a more thorough analysis, a neural network model is also added to compare with the other model.

After hyperparameter tuning, table 7 depicts that Gradient Boosting outperforms every other model in the table by presenting the highest score accuracy with 97.1 percent accurate results, an increase of 5.3 percentage points when compared to the same model 2 without hyperparameter tuning.

Regarding the newly added models, Extreme Boosting seems to have more accurate results, with 96.8 percent accuracy when compared to LGBM with 96.3 percent. However, both of them present lower results of accuracy than Gradient Boosting for this problem. Furthermore, it seems that MLP Regressor, the neural network model seems to work worse than the boosting family models and Random Forest. The accuracy of this model is 93 percent, however, has a low RMSE of 0.42

Table 7 Accuracy of Random Forest, Gradient Boosting, Light Gradient Boosting, Extreme Gradient Boosting, and Multi-Layer Perceptron Regressor with the final model (model variant 2)

Predictive Models	R2_Score	RMSE	MAE
RF	0.949	0.55	0.22
GBM	0.971	0.48	0.16
LGBM	0.963	0.51	0.18
XGMB	0.968	0.49	0.17
MLP	0.930	0.36	0.27

Table 8 presents the final hyperparameters that are tuned for each predictive model.

Table 8 Final hyperparameters used for every predictive model

Predictive Models	Final Parameters
RF	n_estimators: 500; random_state: 18
GBM	n_estimators: 1000; max_depth: 6; learning_rate: 0.1; random_state: 18; loss: ls
LGBM	n_estimators: 1000; max_depth: 6; max_bin: 255; num_leaves: 31; learning_rate: 0.1, random_state: 18

XGMB	n_estimators: 1000; max_depth: 6; learning_rate: 0.1, random_state: 18
MLP	hidden_layer_sizes=128; activation='tanh'; solver='adam'; max_iter=50; learning_rate_init=0.001; batch_size=30

5.3. Feature Importance

The following subsection presents the feature importance for every predictive model. Feature importance is a technique that ranks the input features based on 'importance' for a particular predictive model. Meaning that the highest-ranked feature was the most useful feature for achieving the results of the model and understanding the dependent variable. Feature importance is crucial for the decision-making process since it explains which of the variables are most essential, hence providing insights for the future on which variables should we focus more in research.

Figure 15 presents the feature importance of every predictive model that is used in this research. Regarding the linear regression, the variables that explain the most bike count, according to the coefficient are the month variables, especially Summer months, topping with September (month_9) as the most important. The second most important feature is August (month_8), followed by months July (month_7), June (month_6), and October (month_10). The least important feature according to the Linear Regression is the variable Day.

As observed, Random Forest and Gradient Boosting have similar graphs regarding feature importance. For both, RF and GBM, the hour is the most important feature and has a strong effect on Log Bike Count when compared to the other variables. Temp is the second most important feature and has half of the effect of the Hour on Log Bike Count. Other variables such as Day, UV Index, and Weekend also have an effect but a weaker one, whereas the month variables have almost no effect at all. For Light Gradient Boosting Model, the most important feature is the Day, followed by Pressure as the second and Temperature as the third. Month variables are also the least important features of this model. The extreme Boosting Model's most important feature is the Hour with a strong influence, followed by Temperature and Weekend with half of the effect of the Hour. UV Index is another important feature with a slightly lower influence than the latter. Concerning MLP Regressor, the most important feature is also the Hour, followed by the Day and Temperature which have half of the effect of the Hour.

Overall, Hour, Day, Temperature, and UV Index seem to be the most important features of most predictive models.

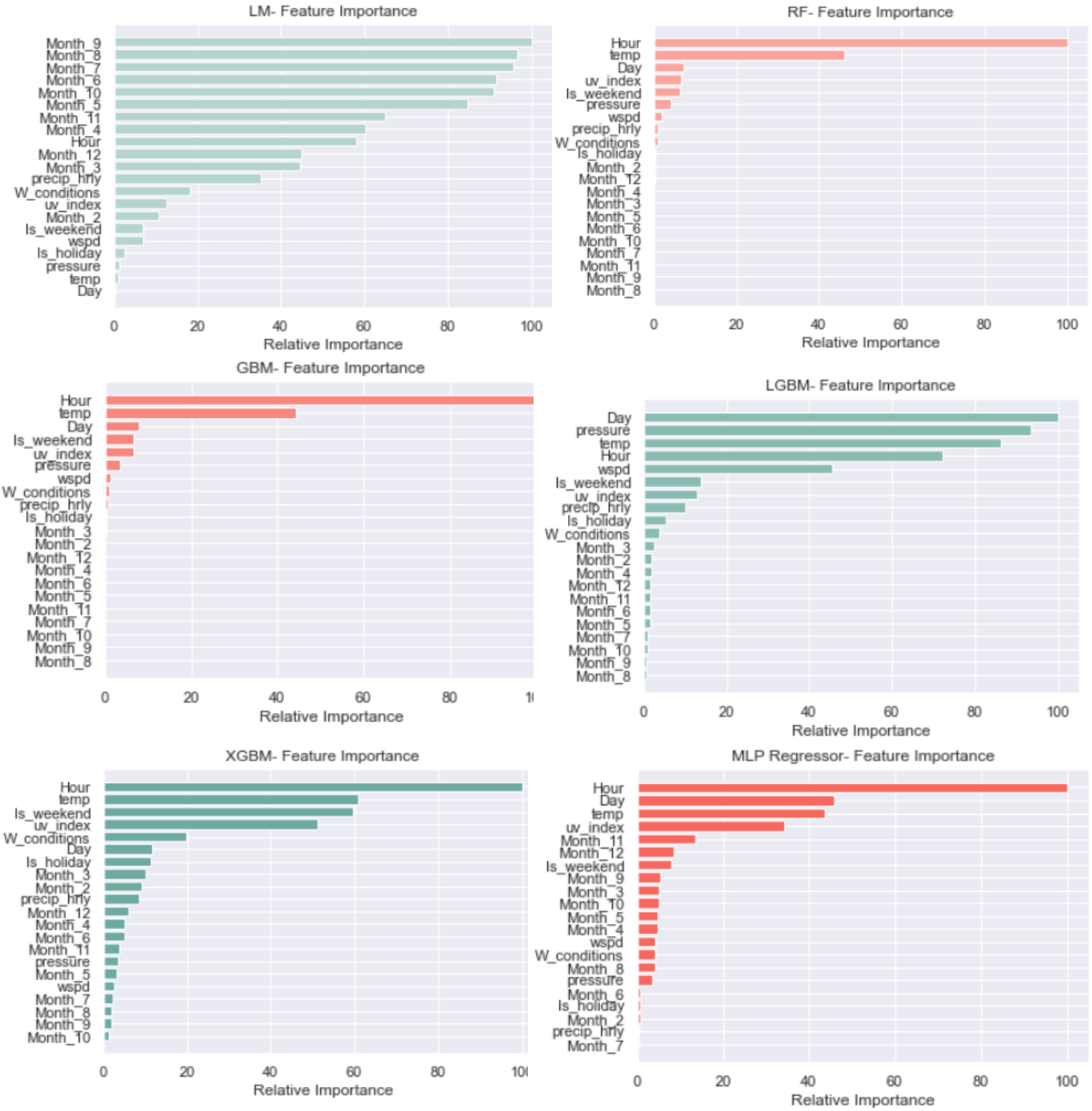


Figure 15 Feature importance for Linear Regression, Random Forest, Gradient Boosting, Light Gradient Boosting, Extreme Gradient Boosting and Multi-Layer Perceptron

5.4. Alternative Train Test Split

This sub-section presents another method of testing the data for prediction. Instead of using the technique split_train_test that divides the train and testing set randomly, this part follows an alternative technique of using the most recent 20 percent of the data as a test set and the rest as a training set. This method is mostly used when you want to predict the future, since in this case, the testing set cannot be influenced by the sideway data points.

When predicting the last part of the data, it seems that Random Forest is the most accurate model with a score accuracy of 92 percent and the RMSE is 0.56. The second-best model seems to be Light Gradient Boosting with 91.8 percent score accuracy and a slightly higher RMSE of 0.57. Gradient Boosting is the third-ranked model with 91.3 percent of accuracy and an RMSE of 0.57, followed by Extreme Gradient Boosting with 89.5 percent and an RMSE of 0.6. Artificial Neural Network is the model with the least accurate model based on the score accuracy with 86 percent accurate results, however, it has the lowest RMSE of 0.54, which is slightly lower than the RSME of the Random Forest.

Table 9 Accuracy of the models when forecasting the most recent part of the dataset

Predictive Models	R2 Score	RMSE	MAE
LM	0.593	0.87	0.71
RF	0.920	0.56	0.22
GBM	0.913	0.57	0.24
LGBM	0.918	0.57	0.22
XGMB	0.895	0.60	0.27
MLP	0.86	0.54	0.44

6. Conclusions

This research has addressed the problem of predicting the hourly demand of shared bicycles by using weather data and predictive models for the city of Chicago. In order to predict the hourly demand of shared bikes, the dataset was enriched with weather and time related variables. It can be stated that the hourly demand for bike count is highly influenced by exterior features such as the Temperature, Hour, Day, Month, UV Index, Weekdays, etc. Using these independent features, models were able to produce accurate results for the demand of bikes per hour.

Six predictive models were considered to achieve accurate predictions. Linear Regression, Random Forest, and Gradient Boosting were the main three predictive models that were used from the beginning of the research to arrive at the final model of the data. For more thorough analysis, two more boosting models were added, Light Gradient Boosting and Extreme Gradient Boosting, and a neural network model, Multi-layer Perceptron. Hyperparameters were also added to the predictive models to increase accuracy.

When sampling the test data uniformly at random, Random Forest was the best predictive model with the highest accuracy when run out-of-the box, i.e. without tuning the hyperparameters. However, when tuning hyperparameters, Gradient Boosting outperforms Random Forest. Tuning hyperparameters highly improved the accuracy of Gradient Boosting since its accuracy increased by 4.7 percentage points. Conversely, Random Forest was not influenced by hyperparameters since its accuracy didn't change after tuning them. The second-best model when tuning hyperparameters was Extreme Gradient Boosting.

On the contrary, when using the alternative split method of using the last 20 percent of data, even after using hyperparameters, Random Forest presented the most accurate results even after tuning hyperparameters. Light Gradient Boosting was the second-best model. The neural network model was the last in both models regarding score accuracy however this model presented a low RMSE. It can be concluded that neural network MLP perform worse than the other models for regression problems such as predicting the hourly demand of bike count.

Regarding feature importance for predictive models, they had different behaviors regarding their importance. The hour was considered to be the most important feature for Random Forest, Gradient Boosting, and Extreme Gradient Boosting. Light Gradient Boosting viewed Day as the most important feature, whereas for Linear Regression, it was Month variables. Overall, Hour, Day,

Temperature, UV Index, Pressure, and Weekend were the most important features. It can be concluded it is essential for forecasting the demand for bike usage to add the weather data to the historical bike usage data since weather data seem to have an key role in understanding the usage. Therefore, managers of bike sharing companies should enlarge the scope of the data that they collect, by recording also weather data in addition to bike usage data.

This research presented city-focused predictions for the demand for bikes. Predicting the overall hourly demand for bikes around the city is indeed important. However, it can be also helpful for companies and the city to predict the hourly demand for bikes per station. This will certainly help companies with better logistics and allocation systems around the city. Therefore, future research can focus on developing predictive models for each station in the city. Additionally, this research was done using only one-year data, which can provide limitations when forecasting. Hence, for future research, a longer time span is preferred.

Appendixes

Appendix 1- Feature Summary

Table 10 describes the Features by providing the mean value, standard deviation, minimum and maximum value and the quartiles of every feature.

Table 10 Feature Summary

	count	mean	std	min	25%	50%	75%	max
Id	9504.0	4751.500000	2743.712813	0.00	2375.75	4751.50	7127.25	9503.00
Hour	9504.0	11.500105	6.922596	0.00	5.75	11.50	17.25	23.00
Count	9504.0	703.646359	701.288020	1.00	146.00	465.00	1061.00	3414.00
temp	9504.0	56.317656	21.184676	-4.00	39.00	59.00	74.00	101.00
dewPt	9504.0	41.872264	19.570337	-13.00	27.00	44.00	58.00	77.00
heat_index	9504.0	56.672664	21.581038	-4.00	39.00	59.00	74.00	108.00
pressure	9504.0	29.331850	0.195537	28.46	29.22	29.33	29.45	29.93
wdir	8972.0	195.328801	94.287234	10.00	120.00	210.00	270.00	360.00
wspd	9504.0	10.082597	5.119555	0.00	7.00	9.00	13.00	38.00
precip_hrly	9504.0	0.003713	0.031260	0.00	0.00	0.00	0.00	1.31
feels_like	9504.0	53.728114	25.296739	-16.00	33.00	59.00	74.00	108.00
uv_index	9504.0	0.817866	1.513756	0.00	0.00	0.00	1.00	10.00
Week-day	9504.0	2.992424	2.004505	0.00	1.00	3.00	5.00	6.00
Year	9504.0	2021.613636	0.486941	2021.00	2021.00	2022.00	2022.00	2022.00
Month	9504.0	6.641414	3.333918	1.00	4.00	7.00	9.00	12.00
Is_weekend	9504.0	0.285354	0.451606	0.00	0.00	0.00	1.00	1.00
Is_holiday	9504.0	0.030303	0.171429	0.00	0.00	0.00	0.00	1.00
Weather_condition	9504.0	1.831019	0.623526	1.00	1.00	2.00	2.00	4.00
Clouds	9504.0	3.551031	1.376401	1.00	2.00	4.00	5.00	5.00
Day	9504.0	198.500000	114.321003	1.00	99.75	198.50	297.25	396.00
W_conditions	9504.0	1.110690	0.313765	1.00	1.00	1.00	1.00	2.00

Appendix 2- Null Values

Table 11 presents Non-Null Values, Null values and Unique values of every feature in the dataset.

Table 11 Null values records

	DataType	Non-null_Values	Unique_Values	NaN_Values
Id	int32	9504	9504	0
Date	datetime64[ns]	9504	9504	0
Hour	int64	9504	24	0
Count	int64	9504	2316	0
temp	int64	9504	106	0
wx_phrase	object	9504	33	0
dewPt	int64	9504	91	0
heat_index	int64	9504	113	0
pressure	float64	9504	137	0
wdir	float64	8972	36	532
wspd	int64	9504	30	0
precip_hrly	float64	9504	42	0
feels_like	int64	9504	125	0
uv_index	int64	9504	11	0
clds	object	9504	5	0
Holiday	object	9504	2	0
Week-day	int64	9504	7	0
Weekend	object	9504	2	0
Year	int64	9504	2	0
Month	int64	9504	12	0
Weekday-name	object	9504	7	0
Season	object	9504	4	0
Is_weekend	int64	9504	2	0
Is_holiday	int64	9504	2	0
Weather_condition	int64	9504	4	0
Clouds	int64	9504	5	0
Day	int64	9504	31	0
W_conditions	int64	9504	2	0

Appendix 3- GitHub Link

In this link you can find the GitHub Repository where is downloaded the coding part done in Python of my dissertations: https://github.com/DijoraPeja/Dissertation_Code

References

Abdulalim Alabdullah, Anas, Mudassir Iqbal, Muhammad Zahid, Kaffayatullah Khan, Muhammad Nasir Amin, and Fazal E. Jalal. 2022. "Prediction of Rapid Chloride Penetration Resistance of Metakaolin Based High Strength Concrete Using Light GBM and XGBoost Models by Incorporating SHAP Analysis." *Construction and Building Materials* 345 (August). https://doi.org/10.1016/j.conbuildmat.2022.128296.

Chen, Longbiao, Daqing Zhang, Leye Wang, Dingqi Yang, Xiaojuan Ma, Shijian Li, Zhaohui Wu, Gang Pan, Thi Mai Trang Nguyen, and Jérémie Jakubowicz. 2016. "Dynamic Cluster-Based over-Demand Prediction in Bike Sharing Systems." In *UbiComp 2016 - Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 841–52. Association for Computing Machinery, Inc. https://doi.org/10.1145/2971648.2971652.

Cutler, Adele, D. Richard Cutler, and John R. Stevens. 2012. "Random Forests." In *Ensemble Machine Learning*, 157–75. Boston, MA: Springer US. https://doi.org/10.1007/978-1-4419-9326-7_5.

Dongare, A D, R R Kharde, and Amit D Kachare. 2008. "Introduction to Artificial Neural Network." *Certified International Journal of Engineering and Innovative Technology (IJEIT)*. Vol. 9001.

E, Sathishkumar v, and Yongyun Cho. 2020. "A Rule-Based Model for Seoul Bike Sharing Demand Prediction Using Weather Data." *European Journal of Remote Sensing* 53 (sup1): 166–83. https://doi.org/10.1080/22797254.2020.1725789.

E, Sathishkumar v., Jangwoo Park, and Yongyun Cho. 2020. "Using Data Mining Techniques for Bike Sharing Demand Prediction in Metropolitan City." *Computer Communications* 153 (March): 353–66. https://doi.org/10.1016/j.comcom.2020.02.007.

Feng, Youli, and Shanshan Wang. 2017. "A Forecast for Bicycle Rental Demand Based on Random Forests and Multiple Linear Regression." In *Proceedings - 16th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2017*, 101–5. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICIS.2017.7959977.

Firestine, Theresa. “BTS Technical Report: Bike-Share Stations in the United States (Updated April 2016) | Bureau of Transportation Statistics.” Bureau of Transportation Statistics, Apr. 2016, https://www.bts.gov/archive/publications/bts_technical_report/april_2016.

Hansen, Bruce E. 2022. “A Modern Gauss–Markov Theorem.” *Econometrica* 90 (3): 1283–94. https://doi.org/10.3982/ecta19255.

Hulot, Pierre, Daniel Aloise, and Sanjay Dominik Jena. 2018. “Towards Station-Level Demand Prediction for Effective Rebalancing in Bike-Sharing Systems.” In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 378–86. Association for Computing Machinery. https://doi.org/10.1145/3219819.3219873.

Li, Youru, Zhenfeng Zhu, Deqiang Kong, Meixiang Xu, and Yao Zhao. n.d. “Learning Heterogeneous Spatial-Temporal Representation for Bike-Sharing Demand Prediction.” www.aaai.org.

Natekin, Alexey, and Alois Knoll. 2013. “Gradient Boosting Machines, a Tutorial.” *Frontiers in Neurorobotics* 7 (DEC). https://doi.org/10.3389/fnbot.2013.00021.

Noriega, Leonardo. 2005. “Multilayer Perceptron Tutorial.”

Osborne, Jason. 2010. “Improving Your Data Transformations: Applying the Box-Cox Improving Your Data Transformations: Applying the Box-Cox Transformation Transformation.” *Practical Assessment, Research, and Evaluation* 15: 12. https://doi.org/10.7275/qbpc-gk17.

Pan, Yan, Ray Chen Zheng, Jiaxi Zhang, and Xin Yao. 2019. “Predicting Bike Sharing Demand Using Recurrent Neural Networks.” In *Procedia Computer Science*, 147:562–66. Elsevier B.V. https://doi.org/10.1016/j.procs.2019.01.217.

Transportation Institute, Mineta. 2014. “Public Bikesharing in North America During a Period of Rapid Expansion: Understanding Business Models, Industry Trends and User Impacts.” http://transweb.sjsu.edu.

Wang, Mingshu, and Xiaolu Zhou. 2017. “Bike-Sharing Systems and Congestion: Evidence from US Cities.” *Journal of Transport Geography* 65 (December): 147–54. https://doi.org/10.1016/j.jtrangeo.2017.10.022.