Seoul Bike Sharing Demand Prediction

Shadab Shaikh,
Data Science Trainee,
AlmaBetter, Bangalore

Abstract

Modern urban mobility involves commuter convenience by providing the people with optimal transport solutions. Whether it is public transport, private cab service or rental bike service, getting to know the insights of the public behavior and usage of the services is essential for the provider organization. Bike sharing service is heavily dependent on logistics analysis, customer behavior and other factors to provide the users with convenient service and to make sure there is no shortage of bikes.

Our job as data scientists is to help such a service to predict the requirement of bikes per hour depending on different parameters involved in the dataset provided to us. From a business point of view, this prediction can prove crucial for organizations to achieve greater user satisfaction.

1. Introduction

Recent studies predict that, compared to the current situation, more than 60% of the world's population would live in cities by the year 2050. Some nations all across the world are putting virtuous scenarios into practise, providing transportation at a reasonable cost and reducing carbon emissions. Other cities, on the other hand, are well behind schedule. Urban transportation typically accounts for 64% of all kilometers traveled worldwide. It should be imitated and replaced by intermodal, networked self-driving cars that also offer environmentally friendly transportation. Mobility on Demand systems play a critical role in boosting the availability of the cars, which also increases their numbers and idle time.

2.Problem Statement

Currently, Rental bikes are introduced in many urban cities for the enhancement of mobility comfort. It is important to make the rental bike available and accessible to the public at the right time as it lessens the waiting time. Eventually, providing the city with a stable supply of rental bikes becomes a major concern. The crucial part is the prediction of bike count required at each hour for the stable supply of rental bikes.

Given the dataset for Seoul Bike Sharing Prediction, we have to predict the hourly count of bikes required given various factors affecting this number.

With this project, we find the factors and reasons that affect bike shortages and wait times for renting bikes. This study seeks to identify the variables that are connected with the prediction of bike demand using the data presented. An anticipated hourly bike rental count will also be provided.

3. Data Description

The dataset contains weather information (Temperature, Humidity, Windspeed, Visibility, Dewpoint, Solar radiation, Snowfall, Rainfall), the number of bikes rented per hour, and date information.

Attribute Information:

- Date: year-month-day
- Rented Bike count - Count of bikes rented at each hour
- Hour - Hour of the day
- Temperature-Temperature in Celsius
- Humidity - %
- Windspeed - m/s
- Visibility - 10m
- Dew point temperature - Celsius
- Solar radiation - MJ/m2
- Rainfall - water in mm
- Snowfall - Thickness cm
- Seasons - Winter, Spring, Summer, Autumn
- Holiday - Holiday/No holiday
- Functional Day - NoFunc (Non Functional Hours), Fun (Functional hours)

Duplicate Observations:

When similar entries are not all in the same set, duplicate entries can sabotage the separation of the train, validation, and test sets. This may result in skewed performance estimations, which could make final models perform poorly. In our dataset, however, we have no duplicate rows present.

Missing Values:

The absence of data reduces statistical power, which refers to the probability that the test will reject the null hypothesis when it is false.The lost data may introduce bias into the parameter estimate process. It might make the samples less representative. Our dataset does not contain any missing values, which saves us some hassle and favors model building.

4. Exploratory Data Analysis and Feature Manipulation

We compared our target variable, the Rented Bike count, with other independent variables after importing the dataset. We were able to determine many facets and connections between the dependent and independent variables thanks to this procedure. We now have a clearer understanding of which property, when compared to the dependent variable, exhibits which behavior.

Feature Transformation:

The distribution of the variables might also be improved by transforming the skewed variables. These transformations could be logarithmic, square root, or square. Given that the dependent variable in our dataset, Rented Bike count, is moderately right skewed. Dependent features must follow the normal distribution in order to use linear regression. Here, we carried out the square root transformation of our dependent variable.

Outlier Treatment:

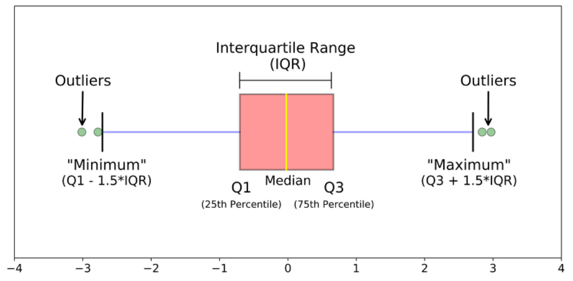
Many times, in a large dataset, some features may contain values that differ significantly from other observations. In statistical terminology, such values are called Outliers.

These values may be included in the dataset due to variability in data, data entry errors, measurement errors or experimental errors.

Outliers may lead to various adverse impacts on the dataset, such as, affecting the normality of features, stretching mean and standard deviation of the data, raising problems during statistical analysis, etc.

Thus, treating these outliers becomes the utmost priority before building the models.

In this project we used a method called Outlier removal using Inter Quartile Range (IQR).



Interquartile Range score is a measure of

statistical dispersion, being equal to the difference between the75th percentile and 25th percentile i.e., third quartile(Q3) and first quartile(Q1). We identify the outliers as values less than Q1 - (1.5 * IQR) or greater than Q3+(1.5 * IQR).

IQR=Q3-Q1

Feature Engineering

The approach of leveraging domain expertise to extract features (characteristics, qualities, and attributes) from unprocessed data is known as feature engineering or feature extraction. The goal is to employ these additional features to enhance the quality of machine learning process results as opposed to only giving the process raw data.

Building new features:

Feature engineering should be carried out with specific domain knowledge, so that these prove to be valuable from a business point of view.

From our dataset, we employed the ‘Date’ column to extract the feature called ‘Is_Weekend’ that specifies whether the given date is a weekend or a week day.We also extracted a feature ‘Time_of_the_day’ from ‘Hour’ Column. This feature has 4 values, i.e., night, morning, afternoon and evening.

Encoding Categorical Features:

Converting categorical data into integer format is the process of encoding categorical data so that the data with transformed categorical values can be delivered to the various models.

We used the following methods to encode categorical features:

Label Encoding:

Label encoding is the process of transforming the labels into a numeric form so that they may be read by machines.

We performed label encoding some features yielding the following results

- Holiday: 1 = Holiday, 0 = No Holiday
- Functioning Day: 1 = Yes, 0 = No
- Time_of_the_day: 0 = night, 1 = morning, 2 = afternoon, 3 = evening

One Hot Encoding:

Data can be converted using one hot encoding as a means of getting a better prediction and preparing the data for an algorithm. With one-hot, we create a new category column for each categorical value and give it a binary value of 1 or 0.

We performed One hot encoding on ‘Seasons’ column, to get new features i.e. ‘Autumn’, ‘Spring’, ‘Summer’, ‘Winter’

Feature Selection:

When two or more independent variables in a regression model have a high correlation with one another, multicollinearity arises. This implies that in a regression model, one independent variable can be predicted from another independent variable.

Multicollinearity is a concern since it reduces the independent variable's statistical significance. Hence it must be carefully handled, by calculating VIF for each variable and dropping features with high VIF.

Variance Inflation Factor

In regression analysis, multicollinearity is found using a variance inflation factor (VIF). The VIF calculates how much multicollinearity in the model has inflated a regression coefficient's variance.

$$\mathrm{VIF} = \frac{1}{1 - R_i^2}$$

The maximum value of VIF to be kept in features in the data changes with context. For our project however, we limited ourself to a VIF of 10.

Model Building

Prerequisites:

Train-Test Split

Train test split is a model validation process that allows you to simulate how your model would perform with new data. The train_test_split() method is used to split our data into train and test sets.

Feature Scaling

Scaling data is the process of increasing or decreasing the magnitude according to a fixed ratio, in simpler words you change the size but not the shape of the data.

Various methods of feature scaling:

1. Standardization

It calculates the z-score of each value and replaces the value with the calculated Z-score. The Z-score can be calculated by the following formula:

$$x' = \frac{x - \bar{x}}{\sigma}$$

Where σ is the variance and x̄ is the mean.The features are then rescaled with x̄ =0 and σ=1.

Library used: StandardScalar

2. Min-Max Scaling:

It is also referred to as Normalization. The features are scaled between 0 and 1. Here, the mean value remains the same as in Standardization, that is, 0. The formula is given as:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

3. Normalizing

It is used to rescale each sample. Each sample (i.e. each row of the data matrix) with at least one non zero component is rescaled independently of other samples so that its norm (l1 or l2) equals one.

Library used: Normalizer

For this project, we utilized Min-Max Scaling

Evaluation Metrics

Evaluation metrics are used to gauge how well a statistical or machine learning model is performing. Every project has to evaluate machine learning models or algorithms. To test a model, a wide variety of evaluation measures are available. We will use the following evaluation metrics to estimate the performance of our models:

1. Mean Squared Error

MSE is the most frequently used statistic for regression problems. Its form is convex. It is the squared difference between the predicted and actual values, on average.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2$$

2. Root Mean Squared Error

Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). It is the square root of the average of the squared difference of the predicted and actual value.

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

3. R-Squared

R-squared is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable. R-square is a comparison of residual sum of squares (SSres) with total sum of squares(SStot).

$$R^2 = 1 - \frac{RSS}{TSS}$$

4. Adjusted R-squared

Adjusted R squared is calculated by dividing the residual mean square error by the total mean square error (which is the sample variance of the target field). The result is then subtracted from 1. Adjusted R2 is always less than or equal to R2.

$$Adjusted\ R^2 = 1 - (1 - R^2) * \frac{n - 1}{n - k - 1}$$

Defining Hyperparameters

Hyperparameters are the parameters that are explicitly defined to control the learning process before applying a machine-learning algorithm to a dataset.

We defined the ranges of hyperparameters to be used in the later stages of model building.

For the scope of this project we will utilize Cross Validation with the ensemble models. Specifically, we used GridSearchCV with these models to come up with ideal hyperparameters.

GridSearchCV is a method for adjusting hyperparameters to find the best values for a particular model. As was already noted, a model's performance is strongly influenced by the value of its hyperparameters. Noting that there is no way to determine the best values for hyperparameters in advance, it is ideal to explore every conceivable value before deciding what the best ones are. We utilize GridSearchCV to automate the tweaking of hyperparameters because doing it manually could take a lot of time and resources.

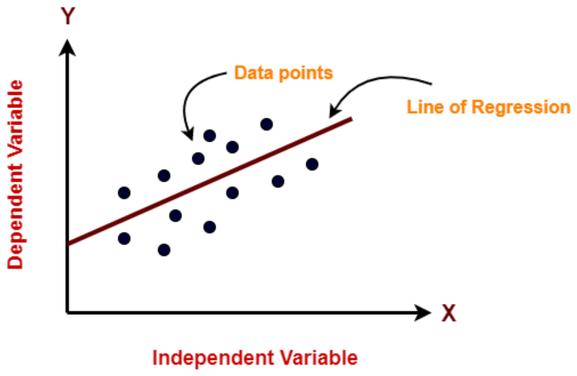
This function aids in fitting your estimator (model) to your training set by looping through predefined hyperparameters. So, from the list of hyperparameters, we may choose the best parameters in the end.

Model Selection:

We used the following models to fit to our data and predict hourly bike counts.

1. Linear Regression

It is a statistical technique for performing predictive analysis. For continuous/real or numerical variables like sales, age, product price, etc., linear regression makes predictions. The term "linear regression" refers to a procedure that displays a linear relationship between a dependent (y) and one or more independent (y) variables.



We can represent it mathematically as:

$$Y = b0+B1x+ \varepsilon$$

Y = Dependent Variable

X = Independent Feature

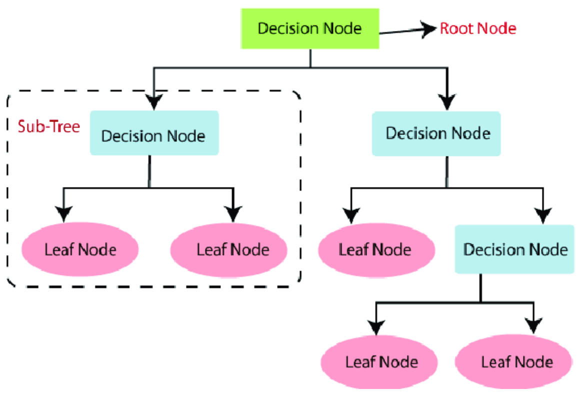
b0 = intercept of the line.

b1 = Linear regression coefficient.

ε = random error

2. Decision Trees

A supervised learning method called the decision tree can be applied to classification and regression problems. It is a tree-structured classifier, where internal nodes stand in for a dataset's features, branches for the decision-making process, and each leaf node for the classification result. The Decision Node and Leaf Node are the two nodes of a decision tree. The CART algorithm, which stands for Classification and Regression Tree algorithm, is used to construct a tree. Both numerical data and categorical data (YES/NO) can be included in a decision tree.



Following terminologies are used for decision trees.

● Root Node: Root node is from where the decision tree starts.

● Leaf Node: Leaf nodes are the final output node

● Splitting: Splitting is the process of dividing the decision node/root node into sub-nodes

● Branch/Sub Tree: A tree formed by splitting the tree.

Ensemble Learning:

Ensemble methods in statistics and machine learning combine several learning algorithms to achieve higher predicted performance than any one of the individual learning algorithms could.

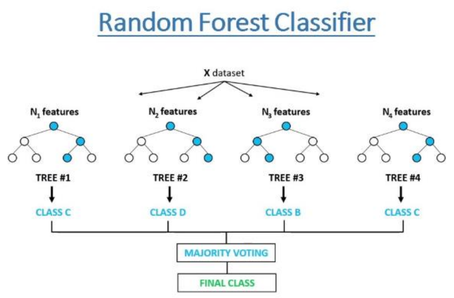
There are two main types of ensemble learning techniques that we will use for the scope of this project:

● Bagging– It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example, Random Forest.

● Boosting– It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. For example, XGBoost, Gradient Boosting Machine, LightGBM, CatBoost.

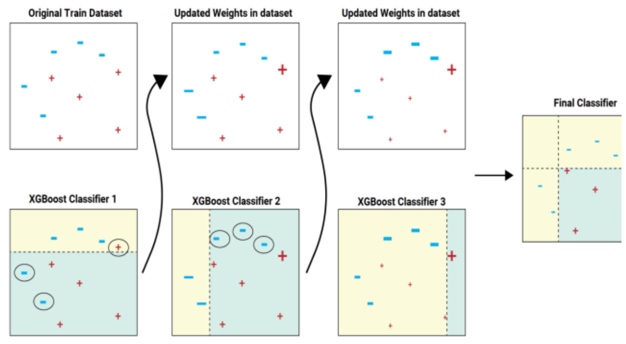
3. Random Forest

Random Forest is a classifier that uses many decision trees on different subsets of the input dataset and averages the results to increase the dataset's predicted accuracy. Instead of relying on a single decision tree, the random forest takes the prediction from each tree and predicts the outcome based on the majority votes of predictions. Higher accuracy and overfitting are prevented by the larger number of trees in the forest.



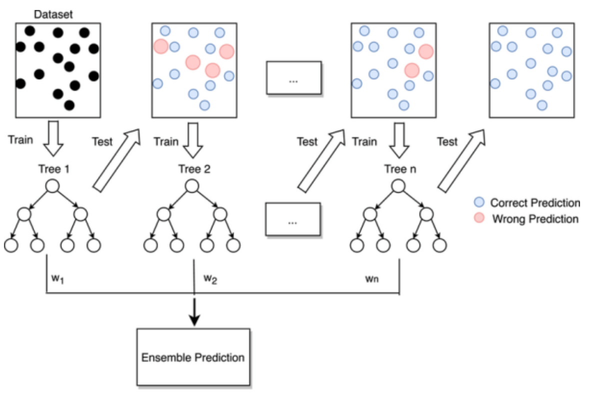
4. XGBoost

Decision trees are generated sequentially in this algorithm. Weights are important in XGBoost. Each independent variable is given a weight before being fed into the decision tree that predicts results. Variables that the tree incorrectly predicted are given more weight before being placed into the second decision tree. These distinct classifiers/predictors are then combined to produce a robust and accurate model. The boosting ensemble strategies include XGBoost, which combines the weaknesses of primary learners with the following strong and compatible learners.



5. Gradient Boosting Machine

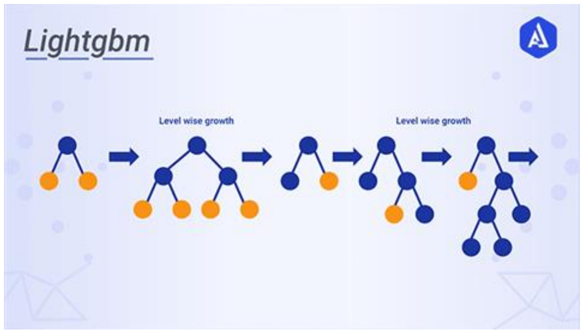
One common approach for boosting is gradient boosting. Each predictor in gradient boosting corrects the error of its predecessor. Gradient Boosted Trees is a method that uses CART (Classification and Regression Trees) as its foundation learner.



6. LightGBM

LightGBM is a gradient boosting framework that uses tree-based learning algorithms. It is designed to be distributed and efficient with the following advantages:

- Faster training speed and higher efficiency.
- Lower memory usage.
- Better accuracy.
- Support of parallel, distributed, and GPU learning.
- Capable of handling large-scale data



7. CatBoost

CatBoost is a recently open-source machine learning algorithm from Yandex. It can easily integrate with deep learning frameworks like Google's TensorFlow and Apple's Core ML. It can work with diverse data types to help solve a wide range of problems that businesses face today. To top it up, it provides best-in-class accuracy.

It is especially powerful in two ways:

- It yields state-of-the-art results without extensive data training typically required by other machine learning methods, and

●Provides powerful out-of-the-box support for the more descriptive data formats that accompany many business problems.

“CatBoost” name comes from two words “Category” and “Boosting”.

Feature Importance

The term "feature importance" describes methods that assign a score to each input feature for a particular model; the scores only indicate the "importance" of each feature. A higher score indicates that the particular characteristic will have more of an impact on the model being used to forecast a particular variable. The tree-based algorithms like random forest, XGboost, and others are linked to the feature approach. We use coefficients as a type of feature relevance in linear regression.

In a model where the prediction is the weighted sum of the input values, linear learning techniques fit the data. Examples include logistic and linear regression, as well as regularization-adding extensions like ridge regression and the elastic net.

To make a prediction, each of these algorithms identifies a set of coefficients to add to the weighted total. One can utilize these coefficients directly as a basic kind of feature importance score.

Conclusion

That's it! We reached the end of our exercise.
Starting with loading the data so far, we have done EDA, null values treatment, encoding of categorical columns, feature selection, and then model building.
In all of these models, our accuracy revolves in the range of 52% to 80%.
Among all the other models used, the XGBoost Regressor gives the best model performance of about 80% on the test data.
We can conclude this performance to be satisfactory, and with more data and careful feature selection from a business point of view and in-depth domain knowledge on the subject matter, there is a chance of getting an even greater prediction accuracy.