Seoul Bike Sharing Demand Prediction

Gaurav Yogeshwar

ABSTRACT

The objective of this work is to predict the trip duration of rental bikes in the Seoul Bike sharing system. The data used include Temperature, Humidity, Wind speed, Visibility, Dew point, Solar radiation, Snowfall, Rainfall, the number of bikes rented per hour and date information.

Our experiment can help understand what could be the reason for the classification of such labels by feature selection, data analysis and prediction with machine learning algorithms taking into account previous trends to determine the correct count.

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.

The main objective is to make a predictive model, which could help them in predicting the bike demands proactively. This will help them in stable supply of bikes wherever needed.

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 - %
- Wind speed - m/s
- Visibility - 10m
- Dew point temperature – Celsius
- Solar radiation - MJ/m2
- Rainfall – mm
- Snowfall – cm
- Seasons - Winter, Spring, Summer, Autumn
- Holiday - Holiday/No holiday
- Functional Day –Yes/No

INTRODUCTION

Today, bike-sharing systems are blooming across more than 1000 cities around the world, particularly in big or large cities like New York City, Paris, Washington DC, London, Beijing and Barcelona. To complete a short trip renting a bike is a faster way when compared to walking. Moreover, it is eco-friendly and comfortable compared to driving.

Due to global warming, continuous pollution and depletion of sources of energy. Many countries have been focused on using renewable energy which doesn't harm the environment and can be reused as well. South Korea is one country which has adapted to it and their most used service is rented bikes in Seoul. But in order to avoid

Any difficulties such as waiting time it is necessary to have an estimate of future demand. Our goal here is to build a model that can predict bike sharing demand considering all the factors which have their effects.

Major Factors Affecting Bike Demand

- **Rainfall:** People tend to use rented bikes quite frequently due to the fact that They can be easily rented from any place and can be dropped off any other place, cheap enough to rent daily, but conditions like Rainfall affects its rental count a lot, People don't rent bikes during the rainy season. So we can say rainfall is negatively correlated with rented bike count.
- **Snowfall:** Similarly, as rainfall, snowfall negatively affects rented bike count as it's hard to drive on snowy roads
- **Visibility:** At times when one can't see properly, it's natural for them to avoid driving, and this is what affects the rented bike count. Although in Seoul the cases of these are quite low.
- **Temperature:** It is seen that people avoid renting bikes at low temperatures. Seoul is a place with an average temperature of 27 to 32 degree Celsius. So, when temperature becomes warm people tend to enjoy it which has an effect in renting bikes as well.
- **Working Day or Not:** Compared to an Off day, people rent bikes more on a working day. The reason behind this is being the same I.e. They can be easily rented from any place and can be dropped off any other place, cheap enough to rent daily.
- **Traffic:** Even though this isn't mentioned in data, traffic also supports renting bike count indirectly. If traffic is high or large people visiting nearby walk or rent a bike for purpose.

STEPS INVOLVED

In order to go ahead for data visualization upon key factors we need to go for certain extra steps before proceeding to the main segment. In this part we are going with the following steps:

1. Importing Analytical necessary library classes for future analysis.
2. Reading the csv data file from Google drive.
3. Setting figure size for future visualization.
4. Removing future warnings in seaborn plots.
5. Visualizing all the columns of the respective Data frame.
6. Viewing all data information
7. Checking the Unique values in the column (if any)
8. Converting the data types to similar objects as the Analysis Demands.
9. Formatting the "size" column into a single column in the dataset.
10. Eradicating special characters from the dataset columns.

- **EXPLORATORY DATA ANALYSIS**

  Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypotheses and to check assumptions with the help of summary statistics and graphical representations. It gives us a better idea of which feature behaves in which manner compared to target variable.After loading the dataset we performed this method by comparing our target variable that is Rented Bike count with other independent variables. This process helped us figuring out various aspects and relationships among the target and the independent variables.

- **EXAMINING NULL VALUES**

  The most critical thing from which we can draw some observations is Dataset, however data comes with unexpected values too i.e. sometimes it may be Null

or missing in other words the space might be blank. Thus, at the time of analysing the first thing which we will do is to examine the null or missing values on the Dataset. It is the first step that will make the results "more" accurate and should be handled before it affects the performance of the models that predict the outcome.

- **ENCODING OF COLUMNS**

  We used One Hot Encoding to produce binary integers of 0 and 1 to encode our categorical features because categorical features that are in string format cannot be understood by the machine and needs to be converted to numerical format.

- **SCALING OF FEATURES**

  Our main motive through this step was to scale our data into a uniform format that would allow us to utilize the data in a better way while performing fitting and applying different algorithms to it.
  The basic goal was to enforce a level of consistency or uniformity to certain practices or operations within the selected environment.

- **Square Root transformation reduce its skewness**

  The square root method is typically used when your data is moderately skewed. Now using the square root (e.g., sqrt(x)) is a transformation that has a moderate effect on distribution shape. It is generally used to reduce right skewed data. Finally, the square root can be applied on zero values and is most commonly used on counted data.

- **Stratified Shuffle Split**
  Stratified ShuffleSplit cross-validator. Provides train/test indices to split data in train/test sets. This cross-validation object is a merge of StratifiedKFold and ShuffleSplit, which returns stratified randomized folds. The folds are made by preserving the percentage of samples for each class.

- **FITTING DIFFERENT MODELS**

  - Linear Regression
  - Lasso Regression
  - Ridge Regression
  - Linear Support Vector Machine
  - Decision Tree
  - Random Forest
  - Extra Trees
  - Gradient Boosting
  - Xgboost
  - LightGBM

ALGORITHMS

I. Linear Regression

Linear regression (LR) is the simplest statistical regression method for identifying the linear link between the independent and the dependent variables. It is done by fitting a linear equation of line to the observed data. For fitting the model, it is utmost important to check, whether there is a connection between the variables or features of interest, which is supposed to use the numerical variable, that is the correlation coefficient.

$$y = k + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n$$

where X is the independent variable whereas Y is a dependent variable. b is the slope of the line and a is the intercept (the value of y when x = 0).

II. Lasso Regression

Lasso regression is a regularization technique. It is used over regression methods for a more accurate

prediction. This model uses shrinkage. Shrinkage is where data values are shrunk towards a central point as the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of multicollinearity. Lasso Regression uses L1 regularization technique.

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j| = \mathrm{RSS}+\lambda\sum_{j=1}^{p}|\beta_j|.$$

This variation differs from ridge regression only in penalizing the high coefficients. It uses |βj| (modulus) instead of squares of β, as its penalty.

III. Ridge Regression

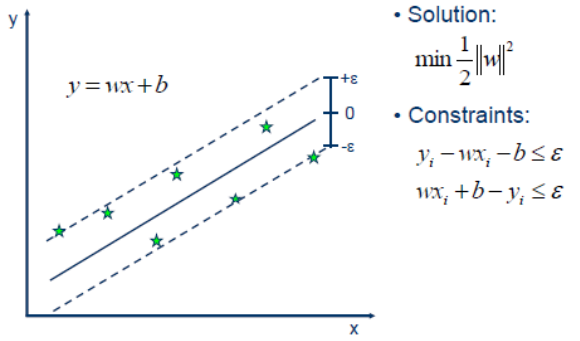
Ridge regression is a model tuning method that is used to analyse any data that suffers from multicollinearity. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values being far away from the actual values.

$$\mathrm{Min}(||Y - X(\mathrm{theta})||^2 + \lambda||\mathrm{theta}||^2)$$

Lambda is the penalty term. λ given here is denoted by an alpha parameter in the ridge function. So, by changing the values of alpha, we are controlling the penalty term. The higher the values of alpha, the bigger is the penalty and therefore the magnitude of coefficients is reduced.

IV. Support Vector Machine

Support Vector Regression is a supervised learning algorithm that is used to predict discrete values. Support Vector Regression uses the same principle as the SVMs. The basic idea behind SVR is to find the best fit line. In SVR, the best fit line is the hyperplane that has the maximum number of points.



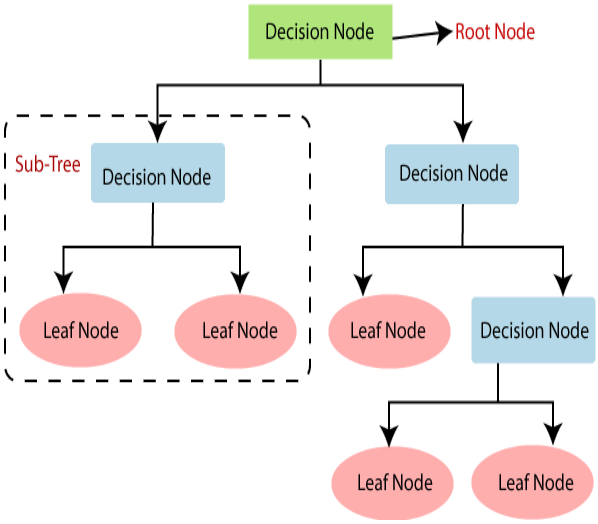
VI. Decision Tree

Decision Tree is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs and utility. Decision-tree algorithms fall under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables. The branches/edges represent the result of the node and the nodes have either:

1. Conditions [Decision Nodes]
2. Result [End Nodes]

The branches/edges represent the truth/falsity of the statement and makes a decision based on that in the example below which shows a decision tree.

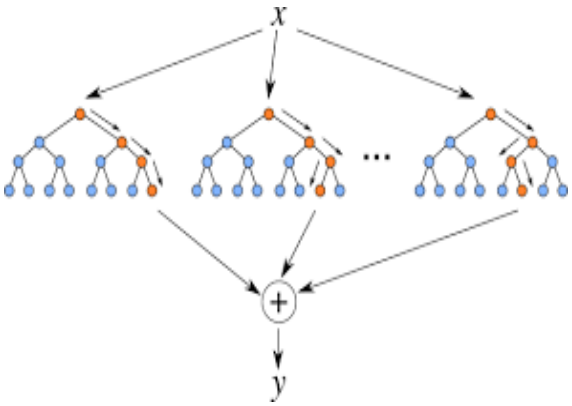
Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values.



VII. Random Forest

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression

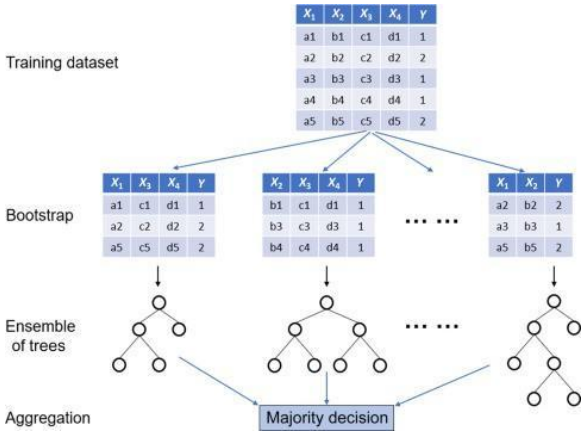
One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification.



VIII. Extra Trees Regressor

Extra trees (short for extremely randomized trees) are an ensemble supervised machine learning method that uses decision trees and is used by the Train Using AutoML tool. See Decision trees classification and regression algorithm for information about how decision trees work. This method is similar to random forests but can be faster.

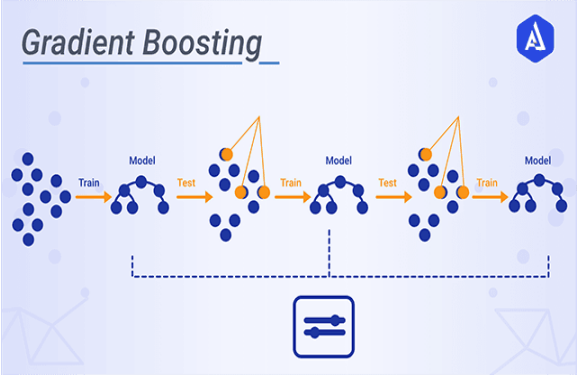
The extra trees algorithm, like the random forests algorithm, creates many decision trees, but the sampling for each tree is random, without replacement. This creates a dataset for each tree with unique samples. A specific number of features, from the total set of features, are also selected randomly for each tree. The most important and unique characteristic of extra trees is the random selection of a splitting value for a feature. Instead of calculating a locally optimal value using Gini or entropy to split the data, the algorithm randomly selects a split value. This makes the trees diversified and uncorrelated.



IX. Gradient Boosting Regressor

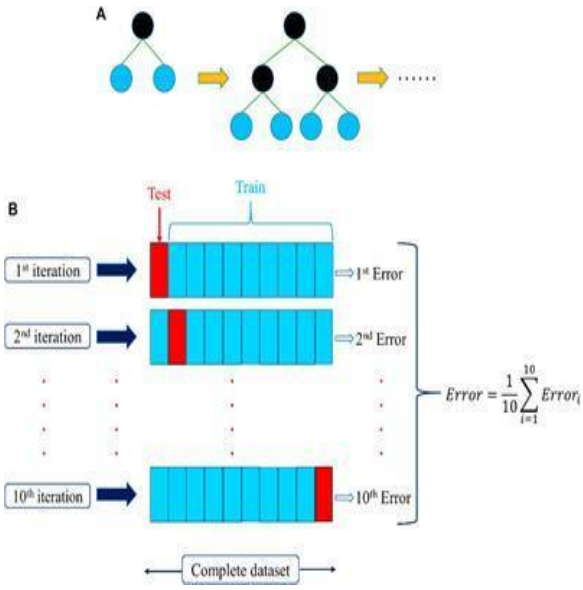
Gradient boosting is one of the most popular machine learning algorithms for tabular datasets. It is powerful enough to find any nonlinear relationship between your model target and features and has great usability that can deal with missing values, outliers, and high cardinality categorical values on your features without any special treatment. While you can build barebone gradient boosting trees using some popular libraries such as XGBoost or LightGBM without knowing any details of the algorithm, you still want to know how it works when you start tuning

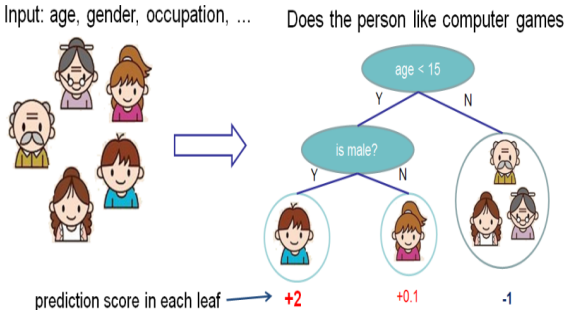
hyper-parameters, customizing the loss functions, etc., to get better quality on your model.



X. XGBoost Regressor

Extreme Gradient Boosting (XGBoost) is an open-source library that provides an efficient and effective implementation of the gradient boosting algorithm. Shortly after its development and initial release, XGBoost became the go-to method and often the key component in winning solutions for a range of problems in machine learning competitions. Regression predictive modeling problems involve predicting a numerical value such as a dollar amount or a height. XGBoost can be used directly for regression predictive modeling



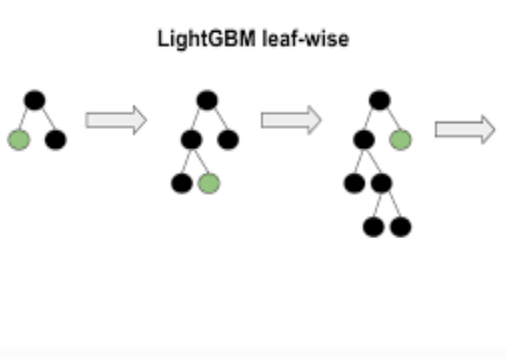


XGBoost is one of the fastest implementations of gradient boosting. trees. It does this by tackling one of the major inefficiencies of gradient boosted trees: considering the potential loss for all possible splits to create a new branch (especially if you consider the case where there are thousands of features, and therefore thousands of possible splits). XGBoost tackles this inefficiency by looking at the distribution of features across all data points in a leaf and using this information to reduce the search space of possible feature splits.

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



MODEL PERFORMANCE

Mean Squared Error (MSE)

MSE or Mean Squared Error is one of the most preferred metrics for regression tasks. It is simply the average of the squared difference between the target value and the value predicted by the regression model.

MSE = mean ((observed - predicted) ^2)

Root Mean Squared Error (RMSE)

which measures the average error performed by the model in predicting the outcome for an observation. Mathematically, the RMSE is the square root of the mean squared error (MSE), which is the average squared difference between the observed actual outcome values and the values predicted by the model.

RMSE = sqrt (MSE). The lower the RMSE, the better the model.

Mean Absolute Error (MAE)

Unlike the RMSE, the MAE measures the prediction error. Mathematically, it is the average absolute difference between observed and predicted outcomes.

MAE = mean (abs (observed - predicted)). MAE is less sensitive to outliers compared to RMSE.

R-squared (R²):

which is the proportion of variation in the outcome that is explained by the predictor variables. In multiple regression models, R2 corresponds to the squared correlation between the observed outcome values and the predicted values by the model.

R² = 1- MSE(model)/MSE(Baseline). The higher the R-squared, the better the model.

Adjusted R²

Adjusted R^2 depicts the same meaning as R^2but it is an improvement. R² suffers from the problem that the scores improve on increasing terms even though the model is not improving which may misguide the researcher.

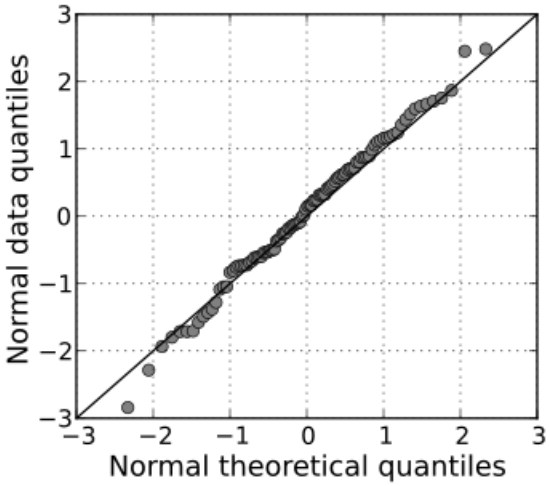
$$R_a^2 = 1 - \left[\left(\frac{n-1}{n-k-1}\right) \times (1 - R^2)\right]$$

where:
n = number of observations
k = number of independent variables
R_a^2 = adjusted R^2

Adjusted R² is always lower than R² as it adjusts for the increasing predictors and only shows improvement if there is a real improvement

Q-Q PLOT

The quantile-quantile (q-q) plot is a graphical technique for determining if two data sets come from populations with a common distribution.



On the X-axis we plot theoretical quantiles of normal distribution and on Y- axis we plot

quantiles of data. If the graph follows a straight line then our given data is normally distributed.

HYPER PARAMETER TUNING

Hyperparameters are sets of information that are used to control the way of learning an algorithm. Their definitions impact parameters of the models, seen as a way of learning, change from the new hyperparameters. This set of values affects performance, stability and interpretation of a model. Each algorithm requires a specific hyperparameters grid that can be adjusted according to the business problem. Hyperparameters alter the way a model learns to trigger this training algorithm after parameters to generate outputs.

We used Grid Search CV, Randomized Search CV and Bayesian Optimization for hyperparameter tuning. This also results in cross validation and in our case we divided the dataset into different folds. The best performance improvement among the three was by Bayesian Optimization.

Grid Search CV

Grid Search combines a selection of hyperparameters established by the scientist and runs through all of them to evaluate the model's performance. Its advantage is that it is a simple technique that will go through all the programmed combinations. The biggest disadvantage is that it traverses a specific region of the parameter space and cannot understand which movement or which region of the space is important to optimize the model.

CONCLUSION

The analysis is done with Seoul Bike data. Four regression techniques Linear Regression, Decision Tree, XG Boosting and Random Forest are used to predict the trip duration. This statistical data analysis shows interesting outcomes in prediction methods and also in an exploratory analysis.

The experimental results show that:

- Most bikes were rented in **summer**, followed by **autumn**, **spring**, and **winter**. **May-July** is the peak Bike renting Season, and **Dec-Feb** is the least preferred month for bike renting.
- **The majority** of the **clients** in the bike rental sector belong to the **Working class**. This is evident from EDA analysis where bike demand is more on weekdays, working days in Seoul.
- **Temperature** of **20-30 Degrees**, **evening time 4 pm- 8 pm, Humidity** between **40%-60%** are the most favourable parameters where the Bike demand is at its peak.
- **Temperature, Hour** of the day, **solar radiation**, and **Humidity** are major driving factors for the Bike rent demand.

Feature and Labels had a weak linear relationship; hence the prediction from the linear model was very low. Best predictions are obtained with a **LightGBM** model with an R^2 score of **0.9523** and **RMSE** is **156.26.**