**Seoul Bike Sharing Demand Prediction**

**Zuber Ahmad**

**Data science trainee,**

**AlmaBetter, Bangalore**

**Abstract:** Bike sharing systems are a means of renting bicycles where the process of obtaining membership, rental, and bike return is automated via a network of kiosk locations throughout a city. Using these systems, people are able rent a bike from a one location and return it to a different place on an as-needed basis. Currently, there are over 500 bike-sharing programs around the world.

The data generated by these systems makes them attractive for researchers because the duration of travel, departure location, arrival location, and time elapsed is explicitly recorded. Bike sharing systems therefore function as a sensor network, which can be used for studying mobility in a city.

This research paper presents a rule-based regression predictive model for bike sharing demand prediction. In recent days, Pubic rental bike sharing is becoming popular because of is increased comfortableness and environmental sustainability.

**1.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.

**Keynotes:** One of the first community bicycle projects in the United States was started in Portland, Oregon in 1994 by civic and environmental activists Tom O'Keefe, Joe Keating and Steve Gunther.

**Data Description**

* Date : year-month-day
* Rented Bike count - Count of bikes rented at each hour
* Hour - Hour of he day
* Temperature-Temperature in Celsius
* Humidity - %
* Windspeed - 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 - NoFunc(Non Functional Hours), Fun(Functional hours)

**2. Introduction**

Foreseeing the rental bike demand is a significant piece of any rebalancing system, such many researches have concentrated on this problem. Predictive models for evaluating the probable range of complete bike request in a given geographic zone dependent on information from drive to-work studies are created.

Our experiment can help understand that how can we examine the rental bike system will be success or not. When the demand is high or low on which factor it depends, factors like weathers, time, day or night, distance etc., by data analysis and prediction with machine learning algorithm.

Our goal is to build the predictive model which predict the consistency of bike sharing system with respect to different variables (like temperature, windspeed, solar radiation etc.) in the city.

## **3. How Bike Sharing Works**

Bike sharing systems are a means of renting bicycles where the process of obtaining membership, rental, and bike return is automated via a network of kiosk locations throughout a city. Using these systems, people are able rent a bike from a one location and return it to a different place on an as-needed basis.

Bike sharing relies on a system of self-service bike stations. Users typically check out a bike using a membership or credit/debit card. They can then ride to their destination and park the bike in a nearby docking station.

## **4. Benefits of Bike Sharing**

## The benefits of bike sharing schemes include transport flexibility, reductions to vehicle emissions, health benefits, reduced congestion and fuel consumption, and financial savings for individuals. But the most special quality of public bicycles is the idea of sharing.

Bike sharing is the right model for supporting commuting systems in cities to decrease CO2 emissions

# **5. Risks in bike sharing**

Helmets – or lack thereof: One major issue that needs to be tackled by the bike-sharing community is bicycle helmets. We all know that bicycle helmets prevent injuries and should be worn by all cyclists.

Docking stations: Bike share programs tend to be congested during heavy traffic hours, this can mean that bicycles might not be available for use. Conversely, during non-typical hours, bike sharing stations can become full, making it difficult to store additional bikes.

**6. Steps involved:**

* **Preprocessing the Datasheet**

The real-world data often has a lot of missing values. The cause of missing values can be data corruption or failure to record data. The handling of missing data is very important during the preprocessing of the dataset as many machine learning algorithms do not support missing values. That’s why we check missing values first.

After checking missing values, we have to check the duplicate values. "Duplication" just means that you have repeated data in your dataset. This could be due to things like data entry errors or data collection methods. by removing duplication in our data set, Time and money are saved by not sending identical communications multiple times to the same person.

**Breakdown of Our Features:**

**Date**: The date of the day, during 365 days from 01/12/2017 to 30/11/2018, formatting in DD/MM/YYYY, type: str, we need to convert into datetime format.

**Changing data type**

As "Hour", "month", "Weekdays weekend" column are show as a integer data type but actually it is a category data type. So, we need to change this data type if we not then, while doing the further analyses and corelated with this then the values are not actually true so we can mislead by this.

**Rented Bike Count**: Number of rented bikes per hour which our dependent variable and we need to predict that, type : int

**Hour**: The hour of the day, starting from 0-23 it's in a digital time format, type: int, we need to convert it into category data type.

**Temperature**(°C): Temperature in Celsius, type: Float

**Humidity** (%): Humidity in the air in %, type: int

**Wind speed** (m/s) : Speed of the wind in m/s, type : Float

**Visibility** (10m): Visibility in m, type : int

**Dew point temperature**(°C): Temperature at the beginning of the day, type: Float

**Solar Radiation (MJ/m2**): Sun contribution, type: Float

**Rainfall**(mm): Amount of raining in mm, type: Float

**Snowfall** (cm): Amount of snowing in cm, type: Float

**Seasons**: \*Season of the year, type: str, there are only 4 seasons in data \*.

**Holiday**: If the day is holiday period or not, type: str

**Functioning Day**: If the day is a Functioning Day or not, type: str

* **Missing and duplicate values Treatment**

Our dataset contains a large number of null values which might tend to disturb our accuracy hence we dropped them at the beginning of our project in order to get a better result.

By doing the preprocessing the datasheet we came to know that there is neither missing and nor duplicate values in our dataset.

* **Exploratory Data Analysis**

After loading the dataset, we performed this method by comparing our target variables that is data variables. This process helped us understanding more about the data and figuring out various aspects and relationships among the dependent variables and bike count. It gave us a better idea of which feature behaves in which manner compared to the target variable. We can do univariate analysis; it is to simply describe the data to find the patterns within the data.

Through different patterns we can analyses both data type categorical variable and numerical variables

Prediction of some EDA analysis mention below:

**Count of rented bikes according to Month:** It is found that from the month 5 to 10 the demand of rented bike is high as compare to other months. These months come inside the summer season.

**Count of rented bikes according to Weekdays-weekend:** It is found that the demand of bike is higher because of the office, peak time are 7 am to pam and 5 pm to 7 pm.

**Count of rented bikes according to Functioning Day:** It is found that peoples don’t use rented bikes in no functioning day.

**Count of rented bikes according to Seasons:** It is found that demand of rented bike is high in summer season and the peak time is 7 am to 9 am and 5 pm to 7 pm. In winter season the use of rented bike is very low.

**Count of rented bikes according to holiday:** It is found that the rented bike in holiday is low as compared to no holiday. The peak time of uses of rented bikes on holiday is 2pm to 8pm.

* **Analyzing Numerical Variables**

The quantitative data or numerical data is always collected in number form. Numerical data differentiates itself from other number form data types with its ability to carry out arithmetic operations with these numbers.

In our dataset the numerical data are

Rented bike count, temperature, humidity, wind speed, visibility, dew point temperature, solar radiation, rainfall and snowfall.

Analyzing result of some Numerical variables are:

**Temperature:** It is found that people like to ride bikes when it is pretty hot around 25°C in average.

**Dew Point Temperature:** The result is similar to temperature analysis.

**Solar Radiation:** It is found that the amount of rented bikes is huge, when there is solar radiation, the counter of rents is around 1000.

**Snowfall:** It is found that the amount of rented bike is very low When we have more than 4 cm of snow, the bike rents is much lower.

**Rainfall:** It is found that v even if it rains a lot the demand of rent bikes is not decreasing.

**Wind Speed:** It is found that the demand of rented bike is uniformly distribute despite of wind speed but when the speed of wind was 7 m/s then the demand of bike also increase that clearly means peoples love to ride bikes when its little windy.

* **Analyzing Catagorical Variables**

A dataset may contain various type of values, sometimes it consists of categorical values. So, in-order to use those categorical value for programming efficiently we create dummy variables.

A **one hot encoding** allows the representation of categorical data to be more expressive. Many machine learning algorithms cannot work with categorical data directly. The categories must be converted into numbers. This is required for both input and output variables that are categorical.

* **Normalizing Column Data:**

The data normalization (also referred to as data pre-processing) is a basic element of data mining. It means transforming the data, namely converting the source data in to another format that allows processing data effectively. The main purpose of data normalization is to minimize or even exclude duplicated data.

* **Outliers Detection and Treatment**

Outliers are nothing but data points that differ significantly from other observations. They are the points that lie outside the overall distribution of the dataset. Outliers, if not treated, can cause serious problems in statistical analyses. Two of the most common graphical ways of detecting outliers are the boxplot and the scatterplot.

There are Five Different ways to treat outliers:

* Set up a filter in your testing tool. Even though this has a little cost, filtering out outliers is worth it.
* Remove or change outliers during post-test analysis.
* Change the value of outliers.
* Consider the underlying distribution.
* Consider the value of mild outliers.

In our model we applying square root on Rented Bike Count check whether we still have outliers.

* **Checking of Correlation between Variables.**

The statistical relationship between two variables is referred to as their correlation. A correlation could be positive, meaning both variables move in the same direction, or negative, meaning that when one variable's value increases, the other variables' values decrease.

we check correlation between variables using Correlation heatmap, it is graphical representation of correlation matrix representing correlation between different variables.



* **Fitting different models**

For modelling we tried various classification algorithms like:

1. **Linear Regression**
2. **Lasso and Ridge Regression**
3. **Elastic Net Regression**
4. **Decision Tree**
5. **Random Forest.**
6. **Gradient Boosting Regressor**
7. **Hyperparameter Tuning**

**7.1. Model Training:**

**Train Test split for regression**

Before, fitting any model it is a rule of thumb to split the dataset into a training and test set. This means some proportions of the data will go into training the model and some portion will be used to evaluate how our model is performing on any unseen data. The proportions may vary from 60:40, 70:30, 75:25 depending on the person but mostly used is 80:20 for training and testing respectively. In this step we will split our data into training and testing set using scikit learn library.

**Techniques & Terms:**

* The mean squared error (MSE) tells you how close a regression line is to a set of points. It does this by taking the distances from the points to the regression line (these distances are the “errors”) and squaring them. It’s called the mean squared error as you’re finding the average of a set of errors. The lower the MSE, the better the forecast.
* MSE formula = (1/n) \* Σ(actual – forecast)2 Where:
* n = number of items,
* Σ = summation notation,
* Actual = original or observed y-value,
* Forecast = y-value from regression.
* Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors).
* Mean Absolute Error (MAE) are metrics used to evaluate a Regression Model. ... Here, errors are the differences between the predicted values (values predicted by our regression model) and the actual values of a variable.
* R-squared (R2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model.
* Formula for R-Squared
* R2=1−Unexplained Variation Total Variation. ​
* R2 = 1− Total Variation Unexplained Variation​
* Adjusted R-squared is a modified version of R-squared that has been adjusted for the number of predictors in the model.

**7.2. Algorithms:**

1. **Linear Regression:**

Regression models describe the relationship between variables by fitting a line to the observed data. Linear regression models use a straight line.

Linear regression uses a linear approach to model the relationship between independent and dependent variables. In simple words its a best fit line drawn over the values of independent variables and dependent variable. In case of single variable, the formula is same as straight line equation having an intercept and slope.

y\_pred=β0+β1x

where β0 and β1 are intercept and slope respectively.

In case of multiple features the formula translates into:

y\_pred=β0+β1x1+β2x2+β3x3+.....

where x\_1, x\_2, x\_3 are the features values and β0,β1,β2..... are weights assigned to each of the features. These become the parameters which the algorithm tries to learn using Gradient descent.

Gradient descent is the process by which the algorithm tries to update the parameters using a loss function. Loss function is nothing but the difference between the actual values and predicted values (aka error or residuals). There are different types of loss function but this is the simplest one. Loss function summed over all observation gives the cost functions. The role of gradient descent is to update the parameters till the cost function is minimized i.e., a global minima is reached. It uses a hyperparameter 'alpha' that gives a weightage to the cost function and decides on how big the steps to take. Alpha is called as the learning rate. It is always necessary to keep an optimal value of alpha as high and low values of alpha might make the gradient descent overshoot or get stuck at a local minima. There are also some basic assumptions that must be fulfilled before implementing this algorithm. They are:

* No multicollinearity in the dataset.
* Independent variables should show linear relationship with dv.
* Residual mean should be 0 or close to 0.
* There should be no heteroscedasticity i.e., variance should be constant along the line of best fit.

We will be using Linear Regression from scikit library.

1. **Lasso and Ridge Regression**

The specific regularization techniques we'll be discussing are Ridge Regression and Lasso Regression.

Regularized linear regression models are very similar to least squares, except that the coefficients are estimated by minimizing a slightly different objective function. we minimize the sum of RSS and a "penalty term" that penalizes coefficient size.

**Ridge regression** (or "L2 regularization") minimizes:

**RSS+λ∑j=1pβ2j**

**Lasso regression** (or "L1 regularization") minimizes:

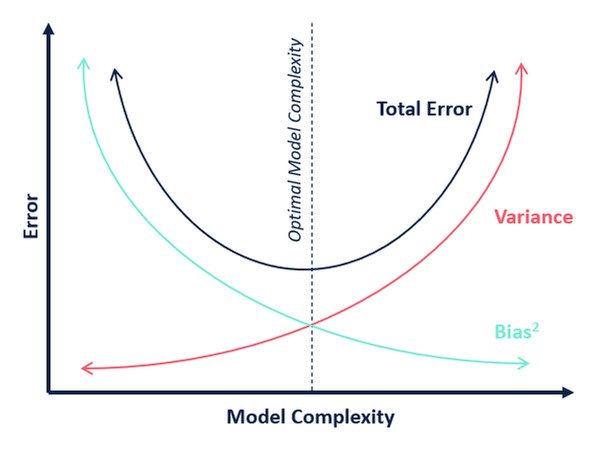
**RSS+λ∑j=1p|βj|**

Where λ is a tuning parameter that seeks to balance between the fit of the model to the data and the magnitude of the model's coefficients:

A tiny λ imposes no penalty on the coefficient size, and is equivalent to a normal linear regression.

Increasing λ penalizes the coefficients and thus shrinks them towards zero.

Lasso stands for least absolute shrinkage and selection operator.

****

1. **Elastic Net Regression**

Elastic net is a penalized linear regression model that includes both the L1 and L2 penalties during training. Using the terminology from “The Elements of Statistical Learning,” a hyperparameter “alpha” is provided to assign how much weight is given to each of the L1 and L2 penalties.

1. **Decision Tree**

Decision tree is a type of supervised learning algorithm that is mostly used in classification problems. It works for both categorical and continuous input and output variables.

It is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes).

The decision of making strategic splits heavily affects a tree’s accuracy. The decision criteria is different for classification and regression trees.

Decision trees use multiple algorithms to decide to split a node in two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that purity of the node increases with respect to the target variable. Decision tree splits the nodes on all available variables and then selects the split which results in most homogeneous sub-nodes.

Methods to determine best split:

* Information Gain
* Gini
* Chi Square
* Reduction in Variance

1. **Random Forest Classifier**

Random Forest is a bagging type of Decision Tree Algorithm that creates a number of decision trees from a randomly selected subset of the training set, collects the labels from these subsets and then averages the final prediction depending on the most number of times a label has been predicted out of all.

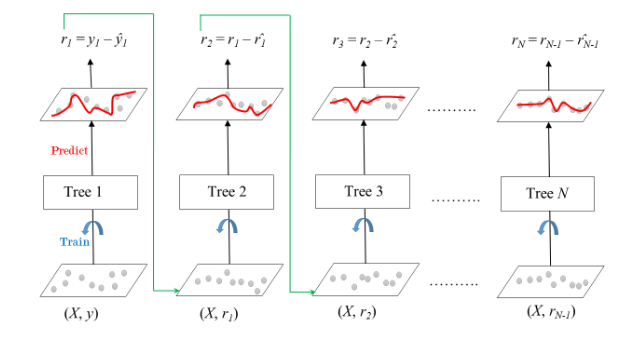


1. **Gradient Boosting**

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.

In gradient boosting, each predictor corrects its predecessor’s error.

Gradient Boosted Trees for Regression shown below image.



1. **Hyperparameter Tuning**

Hyperparameter tuning is the process of choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a model argument whose value is set before the learning process begins. The key to machine learning algorithms is hyperparameter 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.

**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.

GridSearchCV helps to loop through predefined hyperparameters and fit the model on the training set. So, in the end, we can select the best parameters from the listed hyperparameters.

**8. Conclusion:**

That's it! We reached the end of our exercise.

Starting with loading the data so far we have done EDA , outliers detection and treatment, normalizing columns, on hot encoding for categorical data, checking of correlation among variables. Next we implemented 7 machine learning algorithms Linear Regression, lasso, ridge, elastic net regression, decision tree, Random Forest and gradient boosting. We did hyperparameter tuning to improve our model performance.

We conclude that:

No overfitting is seen.

• Random forest Regressor and Gradient Boosting gridsearchcv gives the highest R2 score of 99% and 95% respectively for Train Set and 92% for Test set.

• Feature Importance value for Random Forest and Gradient Boost are different.

• We can deploy this model.

However, this is not the ultimate end. As this data is time dependent, the values for variables like temperature, windspeed, solar radiation etc., will not always be consistent. Therefore, there will be scenarios where the model might not perform well. As Machine learning is an exponentially evolving field, we will have to be prepared for all contingencies and also keep checking our model from time to time. Therefore, having a quality knowledge and keeping pace with the even evolving ML field would surely help one to stay a step ahead in future. So the accuracy of our best model is 73% which can be said to be good for this large dataset. This performance could be due to various reasons like: no proper pattern of data, too much data, not enough relevant features.

**References-**

1. Stacksoverflow.com
2. GeeksforGeeks
3. Kaggle.com
4. analyticsvidhya.com
5. Machine Learning using Python by Manaranjan Pradhan (Book)