# Seoul Bike Sharing Demand Prediction

**KOTA LAKSHMAN RAO,**

**ISHWARYA,**

**Data science trainees,**

**Alma Better, Bangalore**

**Abstract:**

### **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**

**Our experiment can help understand**

**what could be the reason for the prediction of such labels by feature selection, data analysis and prediction with machine learning algorithms taking into account previous trends to determine the correct prediction**

***Keywords :machine learning, Bike sharing demand prediction***

**1.Problem Statement**

**Seoul is a city spread over more than 600 km2, six times the size of Paris. It is necessary to take precautions when it comes to traveling, especially at peak hours, which generate end less traffic jams in the city's streets Their customers can download their app on smartphones and book a bike from anywhere in the cities they operate in.**

**They, in turn, search for cabs from various service providers and provide the best option to their clients across available options. .**

**During this period, they have captured bike availability and peack time of service**

**The main objective is to build a predictive model, which could help them in predicting the bike sharing demand prediction . This would in turn help them in matching the right bikes with the right customers quickly**

. **2.Data Description:**

**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 -mm**
* **Snowfall – cm**
* **Seasons- winter, spring, summer, autumn**
* **Holiday- Holiday/No holiday**
* **Functional Day- No Func (Non Functional Hours),Fun(Functional Hours)**

**3. Introduction**

### **The bike platforms prediction of counting of bikes for each hour using a specific algorithm which is real time and dynamic known as “BIKE SHARING DEMAND PREDICTION”. This algorithm automatically counting of bikes for each hour**

### **The machine learning algorithm generally outputs a multiplier which is adjusted along with the seasons like summer ,winter, spring, autumn, Rainfall , temperature, Humidity ,Hours of the day and This counts of the bikes is done , when customer taking the ride.**

### **Our goal here is to build a predictive model, which could help Counting of bikes for each hour**

## 

## **4. Reasons for less Renting bikes**

**The reasons for less counting of bikes:**

* **More Humidity**
* **More Rainfall**
* **Heavy Snowfall**

# **5. How Renting bikes Increase**

## **Demand for rent increases**

**There are times when so many people are requesting renting for bikes that there don’t have bikes and to go quickly any where on the road to help take them all. Good whether, rush hour, and special events, for instance, may cause unusually large numbers of people to take renting bike****s**

**6. Steps involved:**

* **Exploratory Data Analysis**

**After loading the dataset we performed this method by comparing** **our target** **variable that is RENTED Bikes**  **with Other independent variables. This process helped us figuring out various aspects and relationships among the target and the independent variables. It gave us a better idea of which feature behaves in which manner compared to the target variable**.

* **Null values Treatment**

**Our dataset contains NO null values which may tend to our accuracy will come good and at the beginning of our project in order to get a better result**.

* **Encoding of categorical columns**

**We used Label Encoding to produce binary integers of 0 and 1and 2 and 3 to encode our categorical features (Holiday, Seasons, Functioning day) because categorical features that are in string format cannot be understood by the machine and needs to be converted to numerical format.**

* **Feature Selection**

**In these steps we using information gaining method finding correlation coefficients between variables mostly effects that are removed from our data set i.e temperature and dew-point temperature are most effects for this we taken weighted average and removed this items removed from the dataset**

* **Standardization 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 *Standard scaler after that Normalized*  algorithms to it.**

**The basic goal was to enforce a level of consistency or uniformity to** **certain practices or operations within the selected environment.**

* **Fitting different models**

For modelling we tried various Regression algorithms like:

1. **Linear Regression**
2. **Lasso Regression**
3. **Ridge Regression**
4. **Random Forest Regressor**
5. **Gradient Boost Regressor**
6. **Default XG Booster**
7. **XG boost With Grid search CV**
8. **CAT Boost Regressor**

* **Tuning the hyperparameters for better accuracy**

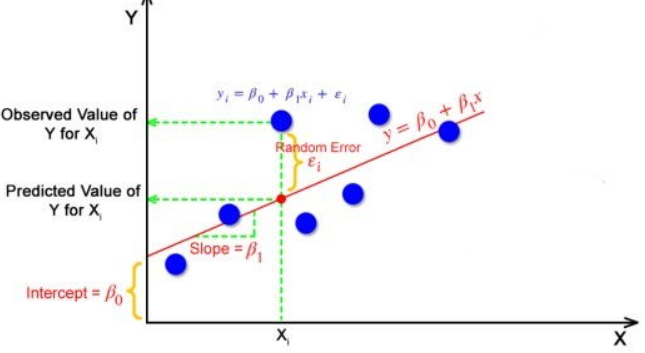
**Tuning the hyperparameters of respective algorithms is necessary for getting better accuracy and to avoid overfitting for this we used Random Forest Regressor and Gradient boost regressor and XG Boost Regressor**

**7.1. Algorithms:**

1. **LINEAR Regression:**

**Linear regression is a quiet and the simplest statistical regression method used for predictive analysis in machine learning. Linear regression shows the linear relationship between the independent(predictor) variable i.e. X-axis and the dependent(output) variable i.e. Y-axis, called linear** **regression*.*If there is a single input** **variable X(dependent** **variable**), such **linear regression is called *simple linear regression***.

**Y=B0 + B1X1 + B2X2 +** …….. BpXp +e



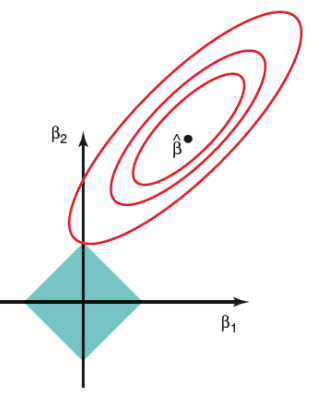
**The Gradient decent is optimization algorithm to finding B0,B1,B2,Bp**

.

1. **Lasso Regression:**

**Lasso is used for Regularization when the coefficients very high or overfitting we use LASSO REGULARIZATION technique and this is used for variable selection and one axis is zero and shrink coefficients all the way to zero**

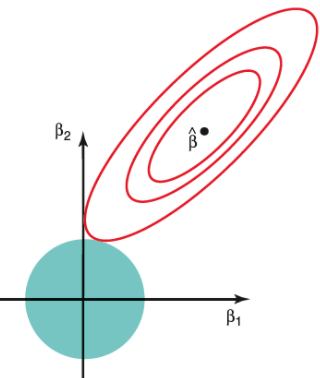
**Using hyper parameter tuning we get good results** :



1. **Ridge Regression:**

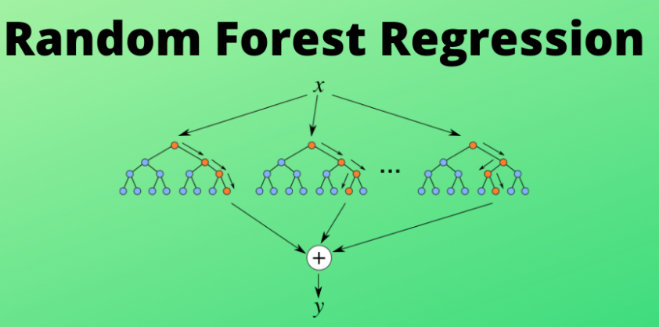
**Ridge is used for Regularization when the coefficients very high or overfitting we use RIDGE REGULARIZATION technique and this is used for better accuracy and there is no variable selection and both axis must have some value shrinking coefficients towards zero but rarely reach zero**

**Using hyper parameter tuning we get good results :**



1. **Random Forest Regressor :**

**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.**



**5.GRADINT BOOST REGRESSOR:**

**Gradient boost sequentially combines many weak learners to form a strong learner. Typically Gradient boost uses decision trees as weak learners.**

**Gradient boost is one of the most powerful techniques for building predictive models for both classification and Regression problems.**

**To understand the Gradient boost below are the steps involved. In Gradient boosting weak learners are decision trees.**

**Step1: Construct a base tree with single root node. It is the initial guess for all the samples.**

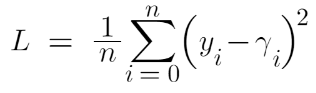
**Step2: Build a tree from errors of the previous tree.**

**Step3: Scale the tree by learning rate (value between 0 and 1). This learning rate determines the contribution of the tree in the prediction**

**Step4: Combine the new tree with all the previous trees to predict the result and repeat step 2 until maximum number of trees is achieved or until the new trees don't improve the fit.**

**The final prediction model is the combination of all the trees.**

**the target column is continuous our loss function will be:**



1. **Default XG Boost-**

**XG Boost 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). XG Boost 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**.

1. **XG\_ boost with Grid \_CV**

**Xg boost with Grid Search cv give very good results because here tuning hyperparameters**

**in this algorithm performed very accuracy , the grid search cv consequently taking every parameter and estimated give good results**

1. **CAT BOOST REGRESSOR**

**Cat Boost builds upon the theory of decision trees and gradient boosting. The main idea of boosting is to sequentially combine many weak models (a model performing slightly better than random chance) and thus through greedy search create a strong competitive predictive model**.

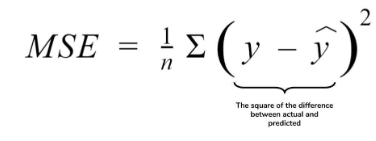
**Cat Boost grows oblivious trees, which means that the trees are grown by imposing the rule that all nodes at the same level, test the same predictor with the same** **condition, and hence an index of a leaf can be calculated with bitwise operations. The oblivious tree procedure allows for a simple fitting scheme and efficiency on CPUs, while the tree structure operates as a regularization to find an optimal solution and avoid overfitting.**

**7.2. Model performance:**

Model can be evaluated by various metrics such as:

1**. 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.**

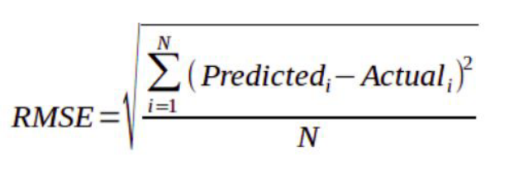
****

**Out of them XG boost with Grid search CV (tuned hyperparameters gave) the best result.**

**MSE is approx 59734,**

**2. Root Mean Squared Error (RMSE):**

**RMSE is the most widely used metric for regression tasks and is the square root of the averaged squared difference between the target value and the value predicted by the model. It is preferred more in some cases because the errors are first squared before averaging** **which poses a high penalty on large errors**.

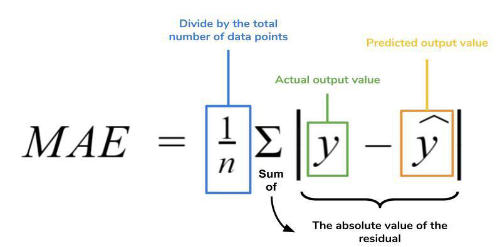


**Out of them XG boost with Grid search CV (tuned hyperparameters gave) the best result.**

RMSE IS approx 245,

**3. MAE (Mean Absolute Error)**:

**MAE is the absolute difference between the target value and the value predicted by the model. The MAE is more robust to outliers and does not penalize the errors as extremely as MSE. MAE is a linear score which means all the individual differences are weighted equally. It is not suitable for applications where you want to pay more attention to the outliers**

**.**

**Out of them XG boost with Grid search CV (tuned hyperparameters gave) the best result**

**MAE is approx 159,**

. **4. R2 (R – Squared)**:

**Coefficient of Determination or R² is another metric used for evaluating the performance of a regression model. The metric helps us to compare our current model with a constant baseline and tells us how much our model is better. The constant baseline is chosen by taking the mean** **of the data and drawing a line at the mean. R² is a scale-free score that implies it doesn't matter whether the values are too large or too small, the R² will always be less than or equal to 1**.

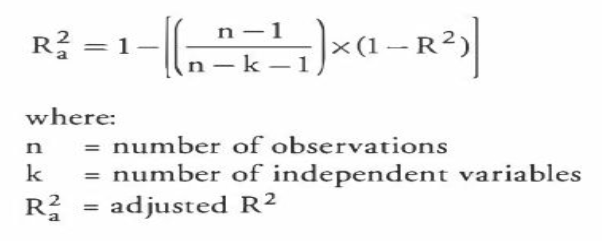
**Out of them XG boost with Grid search CV (tuned hyperparameters gave) the best result**

**R2(R-SQUARED) 0.8593109738470924**

**Approx 86%**

**5.Adjusted R2:**

**Adjusted R² depicts the same meaning as R² but is an improvement of it. R² suffers from the problem that the scores improve on increasing terms even though the model is not improving which may misguide the researcher. Adjusted R² is always lower than R² as it adjusts for the increasing predictors and only shows improvement if there is a real improvement.**

****

**Out of them XG boost with Grid search CV (tuned hyperparameters gave) the best result**

**ADJUSTED R2-SQUARED:** **0.8584215604633672**

**APPROX 86%**

**7.3. 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 for hyperparameter tuning. This also results in** **cross validation and** in **our case we divided the dataset into different folds. The best performance improvement after using this**.

**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 mod**

**8. 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 61 to 86%.**

**And there is a huge improvement in accuracy score after hyperparameter tuning.**

**So the accuracy of our best model R2 SCORE approx is 86% which can be said to be good for this large dataset.**