**Bike Sharing Demand Prediction**

**Saurabh Shinkar**

**Data science trainee,**

**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.

We will analyse each feature and understand about the different factors in the data which affects our target variable. We will apply different machine learning algorithms for regression to our data which will help to understand which model fit the best and then from that model we can estimate target variable.

* **Problem Statement**

The prediction of bike count required at each hour for the stable supply of rental bikes.

**Data Description**:

### 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 (Non-Functional Hours), Yes (Functional hours)

* **Steps involved:**
* **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 hypothesis and to check assumptions with the help of summary statistics and graphical representations. That’s what we have tried to done.

* **Making data in proper format**

We made the Date column in proper format from its string type. Also, we added new columns day, month and year to data from column Date.

* **Analyzing each feature separately**

For numerical feature we look at the distribution of each feature, through Boxplot and Distribution Plot.

* **Fitting different models**

For modelling we tried various classification algorithms like:

➢ Linear Regression

➢ Decision Tree Regression

➢ Decision Tree Regression with Hyperparameter Tuning

➢ Random Forest Regression

➢ Random Forest Regression with Hyperparameter Tuning

➢ Gradient Boosting

➢ Gradient Boosting with Hyperparameter Tuning

➢ XG Boost

➢ XG Boost with Hyperparameter Tuning

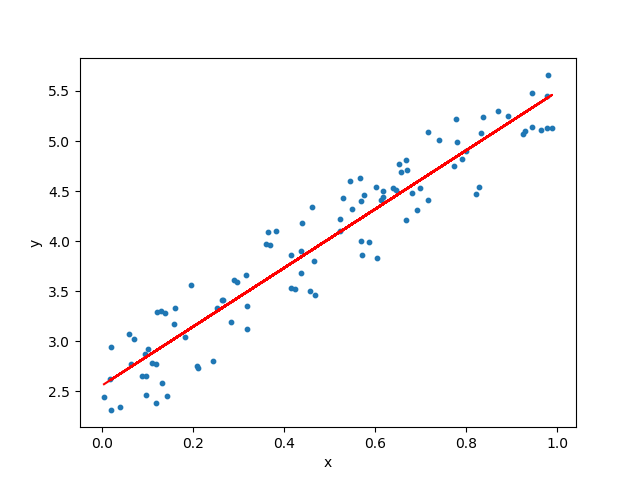
* **Fitting different plots**

For modelling we tried various classification algorithms like:

1. **Histogram**
2. **Bar plot**
3. **Pie Chart**
4. **Density Plot**
5. **Correlation Heatmap**
6. **Regression Line**
7. **Word Cloud**

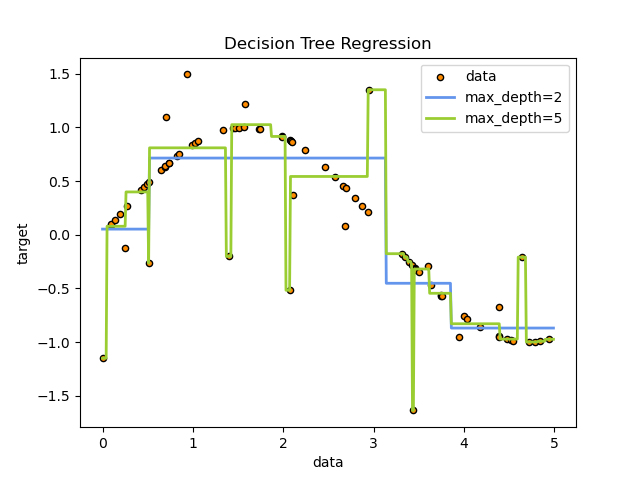
**3. Models:**

1. **Linear Regression:**

 Linear regression analysis is **used to predict the value of a variable based on the value of another variable**. The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable. 

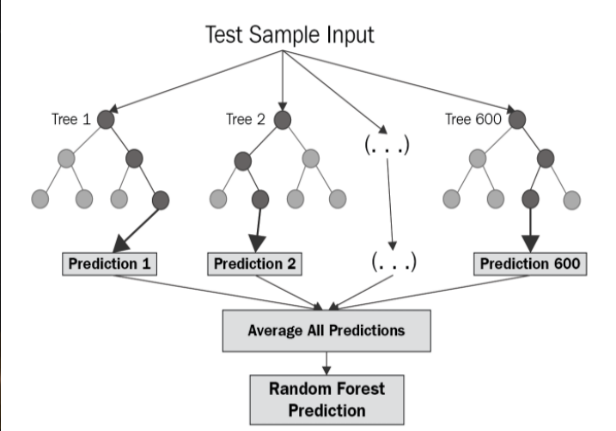
1. **Decision Tree:**

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with **decision nodes** and **leaf nodes**.



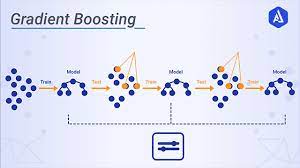
1. **Random Forest:**

A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the max\_samples parameter if bootstrap=True (default), otherwise the whole dataset is used to build each tree.



1. **Gradient Boosting-**

Gradient boosting is a [machine learning](https://en.wikipedia.org/wiki/Machine_learning) technique used in [regression](https://en.wikipedia.org/wiki/Regression_(machine_learning)) and [classification](https://en.wikipedia.org/wiki/Classification_(machine_learning)) tasks, among others. It gives a prediction model in the form of an [ensemble](https://en.wikipedia.org/wiki/Ensemble_learning) of weak prediction models, which are typically [decision trees](https://en.wikipedia.org/wiki/Decision_tree_learning). When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms [random forest](https://en.wikipedia.org/wiki/Random_forest). A gradient-boosted trees model is built in a stage-wise fashion as in other [boosting](https://en.wikipedia.org/wiki/Boosting_(machine_learning)) methods, but it generalizes the other methods by allowing optimization of an arbitrary [differentiable](https://en.wikipedia.org/wiki/Differentiable_function) [loss function](https://en.wikipedia.org/wiki/Loss_function).

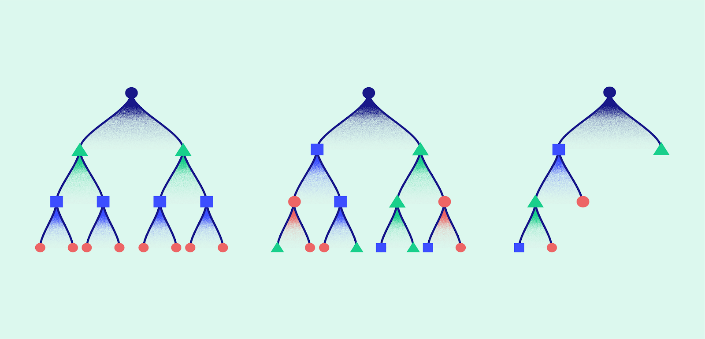


1. **XG Boost-**

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 modelling problems involve predicting a numerical value such as a dollar amount or a height. XG Boost can be used directly for regression predictive modelling.



**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.

**Conclusions-**

* Linear Regression Model Has Accuracy Of 65%.
* Decision Tree Has Accuracy Of 82%.
* Decision Tree with Hyperparameter Tuning Has Accuracy Of 80%.
* Random Forest Has Accuracy Of 91%.
* Random Forest with Hyperparameter Tuning Has Accuracy Of 84%
* Gradient Boosting Has Accuracy Of 87%.
* Gradient Boosting with Hyperparameter Tuning Has Accuracy Of 73%.
* XG Boost Has Accuracy Of 87%.
* XG Boost with Hyperparameter Tuning Has Accuracy Of 92%.
* **From Above We Can Conclude That XG Boost with Hyperparameter Tuning Is the Best Fitted Model to Our Data.**
* **XG Boost with Hyperparameter Tuning Gives About 97% Accuracy in Training Data And 92% Accuracy in Test Data. Also, XG Boost Has Lowest Mean Squared Error in Test Data.**
* Seasons, Temperature, Hour, Functioning Day, Humidity are the most important features which affects our Target variable.

