**Bike Sharing Demand Prediction**

**Viral Bhatu Shewale**

**Data science trainees,**

**AlmaBetter, Bangalore**

**Abstract:**

The project aims to build an ML model that can predict the demand for rental bikes in the city of Seoul. The dataset contains different attributes and the number of bikes rented each hour for 12 months.

***Keywords: machine learning, rental bike count, evaluation,***

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

The main objective is to build a predictive model, which could help them in predicting the demand for rental bikes. This would in turn help them in terms of business development.

**Data summary:**

* 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 - NoFunc(Non Functional Hours), Fun(Functional hours)

**There are no null values present in the data.**

**Steps involved:**

**1. Exploratory Data Analysis:**

After loading the dataset, we performed the EDA method by comparing our target variable (rented\_bike\_count) with other independent variables. This process helped us figure out various aspects and relationships between the target and the independent variables. It gave us a better idea of which feature behaves in which manner compared to the target variable**.**

**2. Multi-collinearity:**

After checking for collinearity, we have seen that **dew point temperature** and **temperature** are highly correlated (**0.91**). Hence we can drop the column from the dataset since it will not increase the prediction accuracy and will only increase the model complexity.

**3. Encoding of categorical columns:**

The categorical features, which are in string format cannot be understood by the machine and needs to be converted to the numerical format. Hence, I used one hot encoding to encode categorical features in binary integers 0 and 1.

**4. Fitting different models:**

For modelling, I tried various regression algorithms like:

**1.** **Decision Tree**

**2. Random forest regression**

**3. XG Boost regression**

**5. Hyperparameter Tuning for better accuracy:**

Tuning the hyperparameters of respective algorithms is necessary to get better accuracy, avoid overfitting and lead to a better-generalized model.

**6. Feature Importance:**

We have visualised feature importance to know the most important features and the features that didn’t put much weight on the performance of our model.

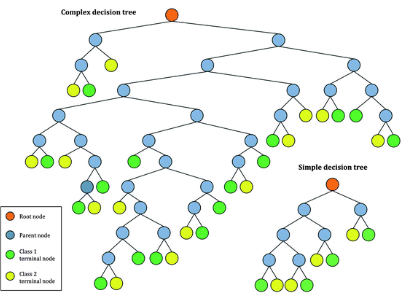
**7. Model evaluation:**

Evaluating the performance of the built model using different metrics.

**Algorithms (ML models):**

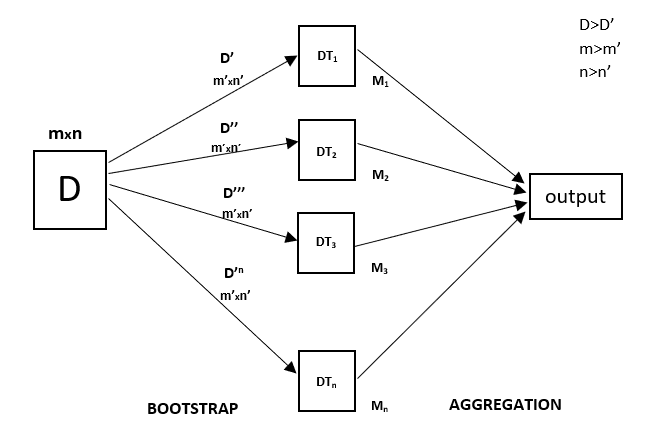
**1. Decision Tree:**

Decision tree uses the tree representation to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree. 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**. A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy), each representing values for the attribute tested. Leaf node (e.g., Hours Played) represents a decision on the numerical target. The topmost decision node in a tree which corresponds to the best predictor is called the **root node**. Decision trees can handle both categorical and numerical data.



**2. Random Forest Regressor:**

Random Forest is a bagging type of Decision Tree Algorithm that creates several decision trees from a randomly selected subset of the training set. Every decision tree has high variance, but when we combine all of them in parallel then the resultant variance is low as each decision tree gets perfectly trained on that particular sample data, and hence the output doesn’t depend on one decision tree but multiple decision trees. In regression problems, the final output is the mean of all the outputs. This part is called **Aggregation**.



**3. XG BOOST:**

**XG Boost** is one of the fastest implementations of gradient boosting trees.**Gradient boosting** refers to a class of ensemble machine learning algorithms that can be used for classification or regression predictive modelling problems.

Ensembles are constructed from decision tree models. Trees are added one at a time to the ensemble and fit to correct the prediction errors made by prior models. This is a type of ensemble machine learning model referred to as boosting.

Models are fit using any arbitrary differentiable loss function and gradient descent optimization algorithm. This gives the technique its name, “gradient boosting,” as the loss gradient is minimized as the model is fit, much like a neural network.

**Hyperparameter Tuning:**

Hyperparameters are sets of information that are used to control the way an algorithm learns. Their definitions impact the parameters of the models, are seen as a way of learning, and change from the new hyperparameters. This set of values affects the performance, stability and interpretation of the 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.

I used Grid Search CV, and along with Grid Search, cross-validation is also performed.

**Grid Search CV:**

**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.

**Cross-validation:**

Cross-Validation is **a statistical method of evaluating and comparing learning algorithms by dividing data into two segments:** one used to learn or train a model and the other used to validate the model.

I used **k-fold cross-validation**. K-fold Cross-Validation is **when the dataset is split into a K number of folds and is used to evaluate the model's ability when given new data**. K refers to the number of groups the data sample is split into. For example, if you see that the k-value is 5, we can call this a 5-fold cross-validation.

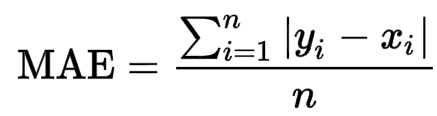
**Model Evaluation:**

The model can be evaluated by various metrics such as:

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

Mean absolute error, also known as L1 loss is one of the simplest loss functions and an easy-to-understand evaluation metric. It is calculated by taking the absolute difference between the predicted values and the actual values and averaging it across the dataset. Mathematically speaking, it is the arithmetic average of absolute errors. MAE measures only the magnitude of the errors and doesn’t concern itself with their direction. The **lower the MAE, the higher the accuracy of a model**.

**MAE is robust to outliers.**

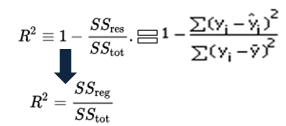
****

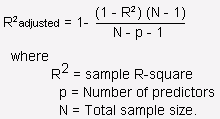
Where, yi = actual value, x\_i = predicted value, n = sample size

**2. R Squared & Adjusted R Squared:**

R Squared & Adjusted R Squared are used for explaining how well the independent variables in the linear regression model explain the variability in the dependent variable. R Squared value always increases with the addition of the independent variables which might lead to the addition of the redundant variables in our model. However, the adjusted R-squared solves this problem.

Adjusted R squared takes into account the number of predictor variables, and it is used to determine the number of independent variables in our model. The value of Adjusted R squared decreases if the increase in the R square by the additional variable isn’t significant enough.

****

****

**Challenges Faced:**

* Comprehending the problem statement and understanding the business implications.
* Feature engineering - deciding on which features to be dropped/kept/transformed.
* Choosing the best visualization to show the trends among different features.
* Deciding on how to handle outliers.
* Choosing the ML models to make predictions.
* Finding the best Hyperparameter which prevents overfitting.
* Evaluating model performance by selecting the best evaluation metric.

**Conclusion:**

* The demand for rental bikes was **highest in the summer** season and **lowest in the winter** season.
* **May-July are peak months** to rent a bike. Dec-Feb is the least preferred month for bike renting.
* The rental bike demand was more on a weekday than on weekends. The majority of **clients belong to the working class.**
* The **temperature of 20-30 Degrees**, evening time 4 pm- 8 pm and the **humidity between 40%-60%** are the most favourable parameters where the Bike demand is at its peak.
* **Temperature, humidity, hour of day, solar radiation and functional day** are major driving factors for the bike rent demand.
* The **XGBoost** model has the **lowest**

 test **MAE**. A low MAE value indicates that the simulated and observed data are close to each other and show better accuracy. Thus **lower MAE is better for model performance**. (XG Boost model with an **accuracy of**88.52**%**).

**References:**

* **pandas documentation**

# **scikit-learn**

* **Analyticsvidhya**
* **GeeksforGeeks**
* **Wikipedia**
* **Medium**