# BIKE SHARING
# DEMAND PREDICTION


## ABSTRACT :

Know a days transportation are bigger problem in society, As a convenient , economical and eco-friendly travel mode, bike sharing greatly improved urban mobility, However it is often very difficult to achieve a balanced utilisation of shared bikes due to the asymmetric user demand distribution and the insufficient numbers of shared bikes, docks, or parking areas. If we can predict the short run bike-sharing demand, it will help operating agencies rebalance bike-sharing systems in a timely and efficient way.


## INTRODUCTION :

According to recent studies, it is expected that more than 60% of the population in the world tends to dwell in cities, which is higher than 50% of the present scenario. Some countries around the world are practising righteous scenarios, rendering mobility at a fair cost and reduced carbon discharge. On the contrary other cities are far behind in the track. Urban mobility usually fills 64% of the entire kilometres travelled in the word. It ought to be modelled and taken over by inter-modality and networked self-driving vehicles which also provides a sustainable means of mobility. Systems called Mobility on Demand have a vital part in raising the vehicles' supply, increasing its idle time and numbers.

## PROBLEM STATEMENT :

**Minimise** - The waiting time to get a bike on rent.

**Maximise** - The availability of bikes to customers.

**Main goal of the project is** - Their central concept is to provide affordable access to bikes for short-distance trips in an urban area as an alternative to private vehicles, thereby reducing congestion, noise, and air pollution.

## FEATURE DESCRIPTION :

- Date (day, month, year),
- 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 FunctionalHours), Fun(Functional hours).

## EXPLORATORY DATA ANALYSIS:

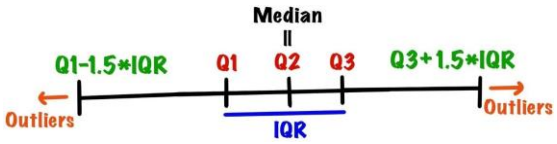
After loading the dataset we perform some tasks like checking null values, missing values, duplicates, outliers, label encoding, feature transformation, comparing our dependent variables with other independent variables and plotting them for better understanding, checking the multicollinearity and treating them using variance inflation factor(VIF), let's deep dive into one EDA part.

**NULL VALUES** :Check the null values using .sum and .isna method, in our bike sharing dataset moreover we don't null values therefore our dataset is pretty good.

**OUTLIERS** :

Checking outliers using box plots and treating them by using interquartile range(IQR) and capping methods.

**Interquartile range(IQR)** -




The interquartile range rule is important for spotting outliers. Interquartile Range score or middle 50% or H-spread is a measure of statistical dispersion,

being equal to the difference between the 75th percentile and 25th percentile i.e., third quartile(Q3) and first quartile(Q1). We identify the outliers as values less than Q1 -(1.5 * IQR) or greater than Q3+(1.5 * IQR). IQR=Q3-Q1.

**Capping and Flooring** -

In this technique, we will do the flooring (e.g., the 10th percentile) for the lower values and capping (e.g., the 90th percentile) for the higher values. The lines of code below print the 10th and 90th percentiles of the variable 'Income', respectively. These values will be used for quantile-based flooring and capping. But in our case, we used the median(50th) percentile to be used both in capping and flooring.

**Why do outliers exist?**

- Variability in data (Natural errors due to few exceptional data readings).
- Data Entry errors (Human errors).
- Experimental errors (Execution errors).
- Measurement errors (instrument errors).

**What impact do outliers have on the dataset?**

- Cause problems during statistical analysis.
- Cause significant impact on the mean and standard deviation of the data.
- Non-random distribution of outliers

**LABEL ENCODING** :

We created new features called **weekend** and **timeshift** using date column, In **weekend** column we give value 1 for saturday and sunday else give 0, In **timeshift** column we give 1 for 0<=x<=6 and 1 for 7<=x<=16 else 2, after label encoding done we drop date column.

**FEATURE TRANSFORMATION** :
The skewed variables may also help correct the distribution of the variables. These could be logarithmic, square root, or square transformations. In our dataset Dependent variable i.e 'Rented Bike Count' having a moderate right skewed, to apply linear regression dependent features have to follow the normal distribution. Therefore, we use square root transformation on top of it.

**DATA PREPROCESSING** :

It is the process of transforming raw data into a useful, understandable format. Real-world or raw data usually has inconsistent formatting, human errors, and can also be incomplete. Data pre-processing resolves such issues and makes datasets more complete and efficient to perform data analysis.

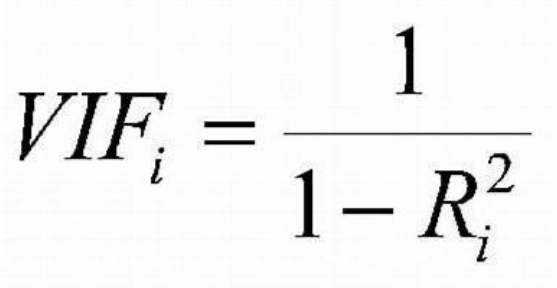
**DATA CLEANING** :

Cleaning is the process of cleaning datasets by accounting for missing values, removing outliers, correcting inconsistent data points, and smoothing noisy data. In essence, the motive behind data cleaning is to offer complete and accurate samples for machine learning models.

**MULTICOLLINEARITY** :

It tells us how one variable depends on other variables that mean if we change one variable value how it's affect our dataset, so we check multicollinearity,

**Detect multicollinearity by variance inflation factor(VIF)**




It measure the amount of multicollinearity in a set of multiple regression variables,

Mathematically, the VIF for a regression model variable is equal to the ratio of the overall model variance to the variance of a model that includes only that single independent variable. This ratio is calculated for each independent variable. Where R^2 is the coefficient of determination in linear regression. A higher R-squared value denotes a stronger collinearity. Generally, a VIF above 5 indicates a high multicollinearity. Here we have taken the VIF for consideration value is 8 for having some important features to accord with the model which we will be using in this dataset.

**FEATURE SCALING** :

Scaling data is the process of increasing or decreasing the magnitude according to a fixed ratio, in simpler words you change the size but not the shape of the data.

There are three different types of feature scaling :

- **Centring** - The intercept represents the estimate of the target when all predictors are at

their mean value, meaning when x=0, the predictor value will be equal to the intercept.

- **Standardisation** - In this method we centralise the data, then we divide by the standard deviation to enforce that the standard deviation of the variable is one.

$$X_{std} = \frac{X - \bar{X}}{s_X}$$

- **Normalisation** - Normalisation most often refers to the process of “normalising” a variable to be between 0 and 1. Think of this as squishing the variable to be constrained to a specific range. This is also called min-max scaling.

$$X_{norm} = \frac{X - min(X)}{max(X) - min(X)}$$

**EVALUATION MATRIX** : Evaluation matrices are measures of how good a model performs and how well it approximates the relationship. Let us look at **MAE**, **MSE**, **R-squared**, **Adjusted R-squared**, and **RMSE**.

**MEAN ABSOLUTE ERROR(MAE) -** This is simply the average of the absolute difference between the target value and the value predicted by the model.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$$

**MEAN SQUARED ERROR(MSE) -** The most common metric for regression tasks is MSE. It has a convex shape. It is the average of the squared difference between the predicted and actual value.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

**ROOT MEAN SQUARED ERROR(RMSE)** - This is the square root of the average of the squared difference of the predicted and actual value.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

**R-SQUARED -** R-square is a comparison of residual sum of squares (SSres) with total sum of squares(SStot).

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}}$$

**ADJUSTED R-SQUARED** - The main difference between **adjusted R-squared** and R-square is that **R-squared** describes the amount of variance of the dependent variable represented by every single independent variable, while **adjusted R-squared** measures variation explained by only the independent variables that actually affect the dependent variable.

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

## HYPERPARAMETER TUNING :

**Hyperparameters** are the variables that the user specifies usually while building the Machine Learning model.

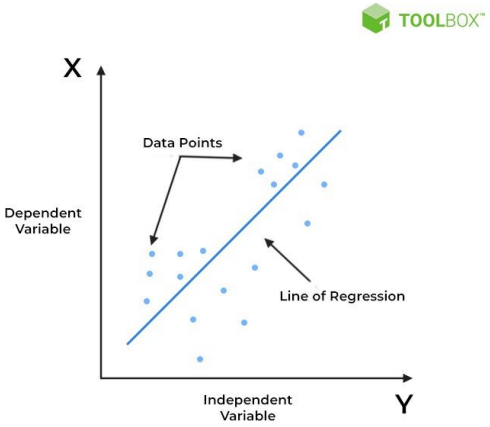
### GRIDSEARCHCV()

GridSearchCV is a library function that is a member of sklearn's model_selection package. It helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, you can select the best parameters from the listed hyperparameters. It generally takes 4 arguments i.e. **estimator**, **param_grid**, **cv**, and **scoring**.

- **n_estimators** - The n_estimator parameter controls the number of trees inside the classifier. We may think that using many trees to fit a model will help us to get a more generalised result. The default number of estimators is 100 in scikit-learn.
- **max_depth** - It governs the maximum height upto which the trees inside the forest can grow. It is one of the most important hyperparameters when it comes to increasing the accuracy of the model. The default is set to None.
- **min_samples_split** - It specifies the minimum amount of samples an internal node must hold in order to split into further nodes. However, the default value is set to 2.
- **min_samples_leaf** - It specifies the minimum amount of samples that a node must hold after getting split. The default value is set to 1.
- **eta/learning_rate** - Learning rate is a hyper-parameter that controls how much we are adjusting the weights of our network with respect to the loss gradient.

## LINEAR REGRESSION :

It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as **sales**, **age**, **product price**, etc. Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called linear regression.




Mathematically, we can represent a linear regression as:

Y = b0+B1x+ ε

Y = Dependent Variable (Target Variable)

X = Independent Variable(predictor

Variable)

b0 = intercept of the line.

b1 = Linear regression coefficient.

ε = random error.

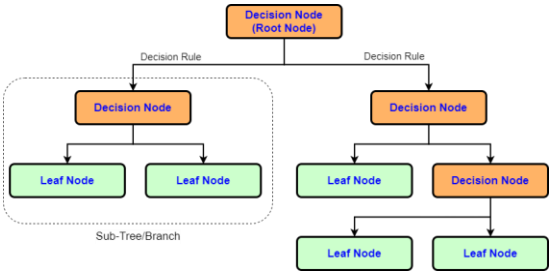
**COST FUNCTION(J)** :

Cost function(J) of Linear Regression is the Root Mean Squared Error (RMSE) between predicted y value (pred) and true y value (y).

$$J = \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

**DECISION TREE :**

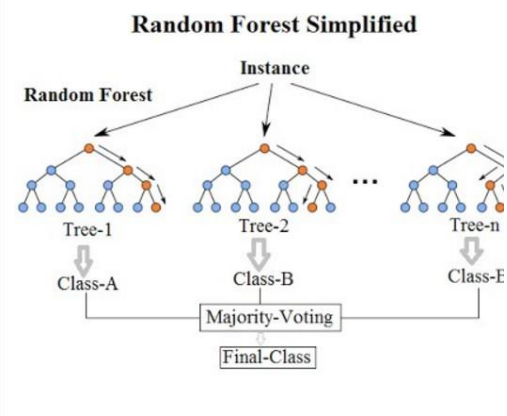
Decision Tree is a **Supervised learning** technique that can be used for both **classification** and **Regression** problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.




In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node**. In order to build a tree, we use the **CART algorithm**, which stands for Classification and Regression Tree algorithm. A decision tree can contain categorical data (YES/NO) as well as numeric data.

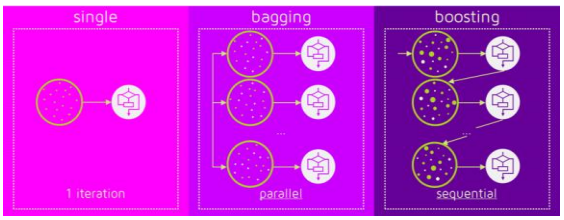
- **Root Node** - Root node is from where the decision tree starts.
- **Splitting** - Splitting is the process of dividing the decision node/root node into sub-nodes.
- **Branch/Sub Tree** - A tree formed by splitting the tree.
- **Leaf Node** - Leaf nodes are the final output node.
- **Pruning** - Pruning is the process of removing the unwanted branches from the tree.
- **Parent/Child node** - The root node of the tree is called the parent node, and other nodes are called the child nodes.

- **RANDOM FOREST** : Random Forest is a classifier that contains a **number of decision trees** on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.



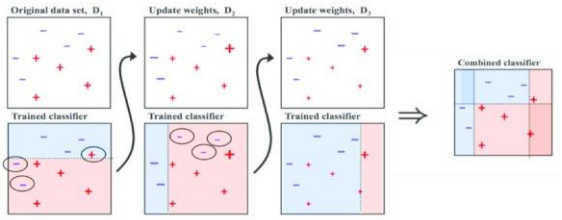

Ensemble uses two types of methods:

- **Bagging**– It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example, Random Forest.
- **Boosting**– It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. For example, ADA BOOST, XGBOOST.



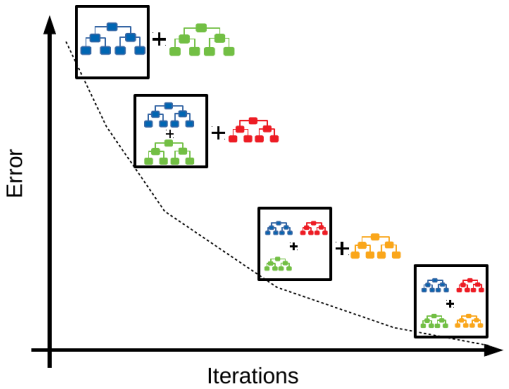

## XGBOOST ALGORITHM :

In this algorithm, **decision trees** are created in **sequential form**. **Weights** play an important role in XGBoost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results. The weight of variables predicted wrong by the tree is increased and these variables are then fed to the second decision tree. These individual classifiers/predictors then ensemble to give a strong and more precise model. XGBoost comes under the boosting ensemble techniques which combines the weakness of primary learners to the next strong and compatible learners.




## GRADIENT BOOSTING :

It is a popular boosting algorithm. In gradient boosting, each predictor corrects its predecessor's error. There is a technique called the Gradient Boosted Trees whose base learner is CART (Classification and Regression Trees).




The ensemble consists of N trees. Tree1 is trained using the feature matrix X and the labels y. The predictions labelled y1(hat) are used to determine the training set residual errors r1. Tree2 is then trained using the feature matrix X and the residual

errors of Tree1 as labels. The predicted results r1(hat) are then used to determine the residual r2. The process is repeated until all the N trees forming the ensemble are trained. Each tree predicts a label and final prediction is given by the formula, y(pred) = y1 + (eta * r1) + (eta * r2) + ..... + (eta * rN)

**FEATURE IMPORTANCE :**

Feature Importance refers to techniques that calculate a score for all the input features for a given model; the scores simply represent the “importance” of each feature. A higher score means that the specific feature will have a larger effect on the model that is being used to predict a certain variable. Feature technique is associated with the tree-based algorithms like random forest, XGboost and so on. In linear regression we use coefficient as a type of feature importance.

Linear learning algorithms fit a model where the prediction is the weighted sum of the input values. Examples include linear regression, logistic regression, and extensions that add regularisation, such as ridge regression and the elastic net. All of these algorithms find a set of coefficients to use in the weighted sum in order to make a prediction. These coefficients can be used directly as a crude type of feature importance score.

**CONCLUSION** :

Starting with loading the data so far, we have done EDA, outlier treatment, encoding of categorical columns, feature selection and then model building.

After trying different models, finally the XGBOOST regressor gives us the highest accuracy ranging between 82-96%. Functioning day is the most important feature and Winter is the second most important feature for XGBoostRegressor.