BIKE SHARING DEMAND PREDICTION

Dhiraj Chaudhari, Data Science Trainee, AlmaBetter, Bangalore

ABSTRACT:

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 utilization 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:

Ride sharing companies like Uber, Ola, and Lyft are great business models that provide conventional, affordable & efficient transportation for the customer who wants to travel without the hassle of owning or operating a vehicle. However, the increase in demand for automobile ride sharing cars is not good, especially in a Crowded and busy area. Therefore, bike sharing is a brilliant idea which provides people another short-range transportation option that allows them to travel without worrying about being stuck in traffic.

PROBLEM STATEMENT:

- Maximize: The availability of bikes to the customer.
- Minimize: Minimise the time of waiting to get a bike on rent.

The main goal of the project is to: Find factors and causes which influence shortages of bikes and time delay of availing bikes on rent. Using the data provided, this paper aims to analyse the data to determine what variables are correlated with bike demand prediction. Hourly count of bikes for rent will also be predicted.

FEATURE DESCRIPTION:

Attribute Information

- Date - year-month-day
- Rented Bike count - Count of bikes re nted 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, Au tumn
- Holiday - Holiday/No holiday
- Functional Day – No Func(Non Functional Hours), Fun(Fun ctional hours)

FEATURE BREAKDOWN:

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

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

Hour: The hour of the day, starting from 0-23 it's in a digital time format

Temperature (°C): *Temperature of the weather in Celsius and it varies from -17°*C to 39.4°C*.*

Humidity (%): Availability of Humidity *in the air during the booking and ranges from 0 to 98%.*

Wind speed (m/s): Speed of the wind while booking and ranges from 0 to 7.4m/s.

Visibility (10m): Visibility to the eyes during driving in "m" and ranges from 27m to 2000m.

Dew point temperature (°C): *Temperature At the beginning of the day, it* ranges from -30.6°C to 27.2°C.

Solar Radiation (MJ/m2): Sun contribution or solar radiation during ride booking which varies from 0 to 3.5 MJ/m2.

Rainfall (mm): The amount of rainfall during bike booking which ranges from 0 to 35mm.

Snowfall (cm): Amount of snowing in cm during the booking in cm and ranges from 0 to 8.8 cm.

Seasons: Seasons of the year and total there are 4 distinct seasons I.e., summer, autumn, spring and winter.

Holiday: If the day is holiday period or not and there are 2 types of data that is holiday and no holiday

Functioning Day: If the day is a Functioning Day or not and it contains object data type yes and no.

MISSING VALUES:

One of the ways to handle missing values is to simply remove them from our dataset. We have known that we can use the null() and not null() functions from the pandas library to determine null values. Since there are no missing values in this data set.

DATA DUPLICATION:

It is very likely that your dataset contains duplicate rows. Removing them is essential to enhance the quality of the dataset. Since there is no duplicate value in these datasets.

EXPLORATORY DATA ANALYSIS:

After loading the dataset, we performed this method by comparing our target variable that is **bike_count** 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.

OUTLIER TREATMENT:

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)

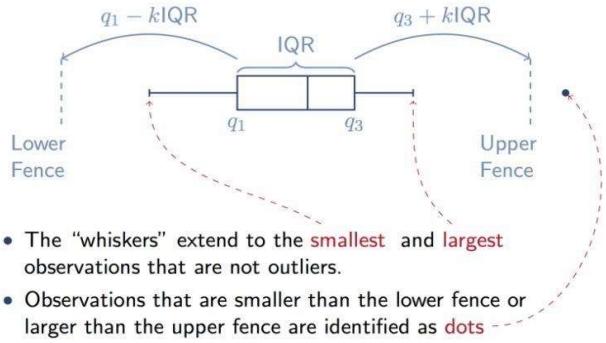
What impact do outliers have on the dataset?

- Cause problems during statistical analysis.

- Cause significant impact on the mean and standard deviation of the data.
- Non-random distribution of outliers can cause decrease in Normality.

There are different types of techniques to handle outliers, we used IQR and capping – flooring method to hands on it .

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**

QUANTILE 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(50^{th}) percentile to be used both in capping and flooring.

FEATURE TRANSFORMATION:

Transformation of 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 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.

CLEANING AND MANIPULATING THE DATASET:

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: Cleansing 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.

NOISY DATA: A large amount of meaningless data is called noise. More precisely, it's the random variance in a measured variable or data having incorrect attribute values. Noise includes duplicate or

semi-duplicates of data points, data segments of no value for a specific research process, or unwanted information fields.

DETECTING MULTICOLLINEARITY BY VARIANCE INFLATION FACTOR(VIF)

$$VIF_i = \frac{1}{1 - R_i^2}$$

Variance inflation factor (VIF) is a measure of 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.

MODEL BUILDING: PREREQUISITES

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 different three types of feature scaling:

- **Centring**: The intercept represents the estimate of the target when all predictors are at their mean value, means when x=0, the predictor value will be equal to the intercept.

- **Standardization**: In this method we centralize the data, then we divide by the standard deviation to enforce that the standard deviation of the

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

variable is one.

- **Normalization:** Normalization most often refers to the process of "normalizing" 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 metrics are a measure 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.

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 *(SS_{res})* with total sum of squares*(SS_{tot})*.

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

ADJUSTED R-SQUARED:

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

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.

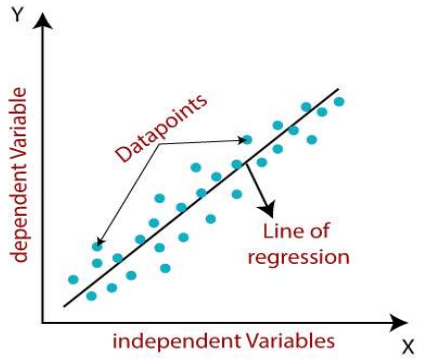
HYPERPARAMETER TUNING:

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

GRIDSEARCH CV()

uses a different combination of all the specified hyperparameters and their values and calculates the performance for each combination and selects the best value for the hyperparameters. This makes the processing time-consuming and expensive based on the number of hyperparameters involved. GridSearchCV() method is available in the scikit-learn class **model_selection.** It can be initiated by creating an object of GridSearchCV().Primarily, it 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 generalized 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 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:

$\mathbf{Y = b_0 + B_1 x + \varepsilon}$

Y = Dependent Variable (Target Variable)

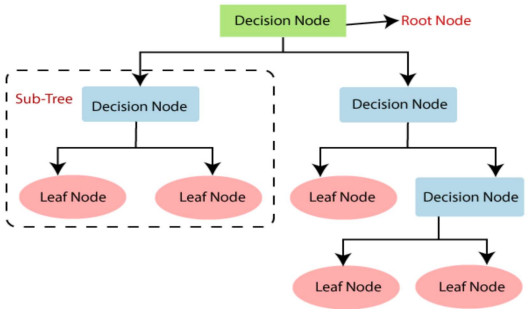
X = Independent Variable(predictor Variable)

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).

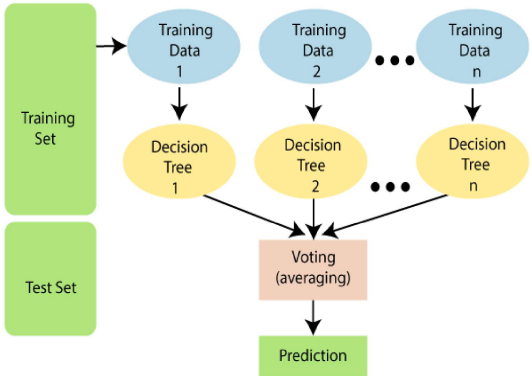


DECISION TREE: Decision Tree is a **Supervised learning** technique that can be used for both classification and Regression problems**.** It is a tree-structured classifier, where **internal nodes** represent the features of a dataset**, branches** represent the decision rules and each **leaf node** 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.
- **Leaf Node:** Leaf nodes are the final output node
- **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.
- **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.

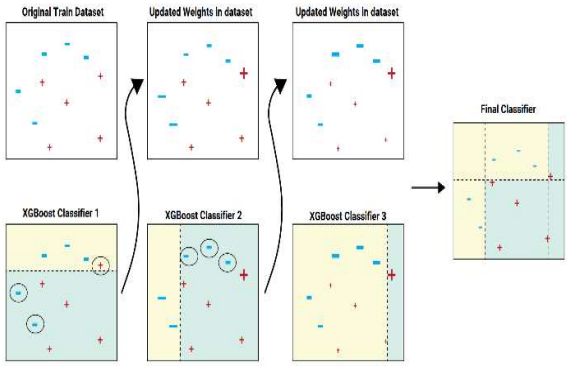
RANDOM FOREST: Random Forest is a classifier that contains a **number of decision trees** on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and 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:

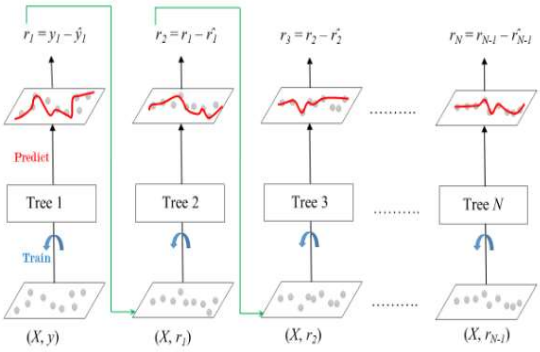
- **Bagging**– It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example, Random Forest.
- **Boosting**– It combines weak learners into strong learners by creating sequential models such that the final model has the highest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

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: Gradient Boosting 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 *r1* 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 regularization, 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:

I fit the data first to linear regression and then I get the baseline accuracy. Then I fit Decision Tree, Random Forest regressor, XGBOOST regressor and gradient Boosting regressor. Then on tunning hyperparameter by using GridSearchCV to find the best hyperparameters & fit to the model and found that the Gradient Boosting and XGBOOST gives highest accuracy with lowest RMSE of 186.30 and 182.46 respectively.

Also, I found the temperature & hour feature are the most important feature in predicting Bike sharing rental count for many algorithms. But for linear regression and XGBOOST winter is most important. Though it is also a function of temperature, it seems it gives the best result.