**Abstract:**

Rental Bike Sharing is the process by which bicycles are procured on several basis (hourly, weekly, monthly, membership-wise, etc. Nowadays from the market survey, we can see that the traffic situation and cost of petrol is too high so the bike sharing system can be the best solution to tackle this problem.

Many people don’t have personal vehicles and public transportation is affordable but due to heavy traffic and time consumption, people are preferring to switch to rental bikes.

The main aim of this project is to make the rental bike available and accessible to the public at the right time as it reduces the waiting time by using the Multi Linear Regression Model.

**Introduction:**

The first bike sharing tasks had been initiated through local community organisations, or as charitable initiatives intended for the disadvantaged, or to promote bicycles as a nonpolluting shape of transport, or they had been business enterprises to hire out bicycles. The first documented bikeshare task started in Europe in 1965, the crew Provo painted fifty bicycles white and placed them unlocked in Amsterdam for all people to use freely. Bike-sharing systems have developed and advanced with society adjustments and technological improvements. Many bicycle programmes paint their bicycles in a sturdy stable colour, such as yellow or white. Painting the bicycles helps to promote the programme, as properly as theft (a painted-over bicycle body is normally much less appropriate to a buyer). However, theft quotes in many bike-sharing programmes stay high, as most shared-use bicycles have price solely as simple transport, and may be resold to unsuspecting consumers after being cleaned and repainted. In response, some large-scale bike sharing programmes have designed their very own bike the use of specialised frame designs and different components to forestall disassembly and resale of stolen parts.

**Problem definition:**

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 various regression models and analyze their performance by comparing various factors like MSE, RMSE, R square value to get the best fit model which helps in predicting the number of bikes required at each hour for the stable supply of rental bikes.

**Data Description:**

The dataset contains weather information (Temperature, Humidity, Windspeed, Visibility, Dewpoint, Solar radiation, Snowfall, Rainfall), the number of bikes rented per hour and date information.

**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 - %
* 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).

Making the availability of bikes at the right time will be a good approach. To maintain the stable supply of rental bikes, the hourly prediction of the bike count is required.

People are able to rent a bike from one location and return it to a different location as needed using these systems. The aim of this project is to build a model using machine algorithms. The historical data of the bike usage pattern and the weather data is used to do this forecast.

**Factors Affecting:**

The factors affecting the rentals of bike are as follows :-

1. **Seasons:** Bike Rental is high during summer and less during spring.

2. **Temperature:** Gradually the bike count increases as the temperature increases.

3. **Hours:** The bike rentals count is in peak form from 8 am morning to 5 pm evening**.**

4. **Weather:** The bike rental count is high when the weather is clear and sunny whereas when there is heavy rainfall the bike rentals count comes down**.**

5. **Working Days:** Bike rental counts are higher during working days.

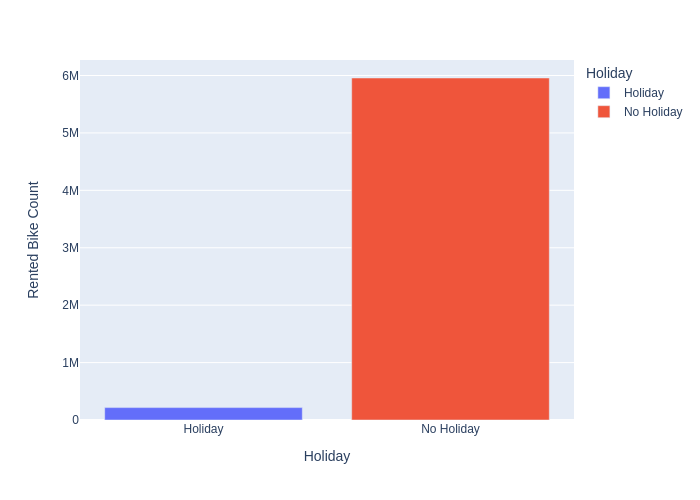
6. **Holidays:** Bike rental is pretty much less as compared to working days.

**Feature Engineering:**

In addition to existing independent variables, we will create new variables to improve the prediction power of the model. Here, “Date” column can be splitted into multiple independent variables such as date, month and year. This will help to classify the rented bike rate day wise, month wise and year wise.

**EDA (Exploratory Data Analysis)**

* **Bar Plot**

****

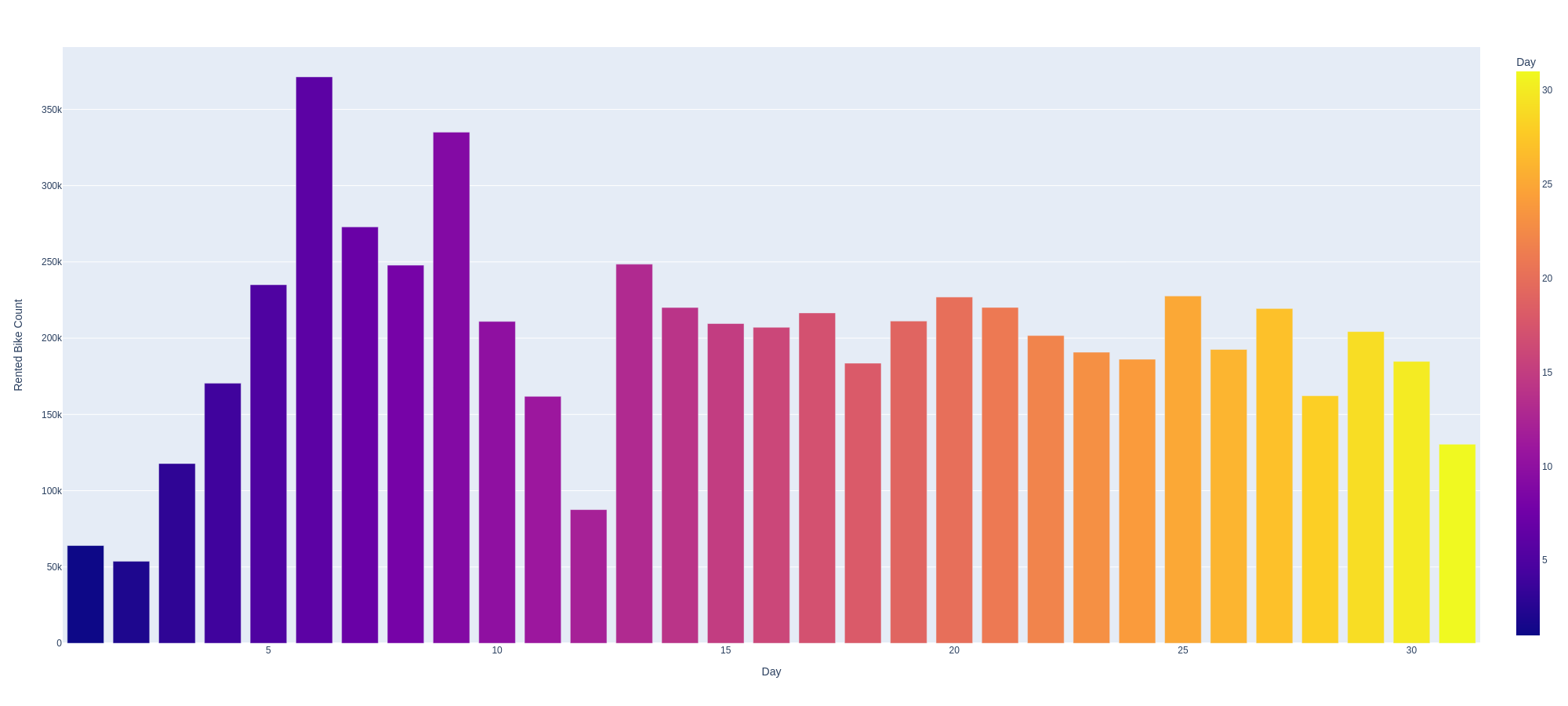
A bar graph is a data visualization technique that can be used to represent numerical values in a dataset to show how different data points vary from each other. It utilizes a bar as a measure of magnitudes. The bigger the bar, the higher the number. For instance, if one is comparing wealth, a bigger bar in a bar graph signifies a lot of money compared to a smaller bar.

Bar graphs are commonly used in financial analysis for displaying data. A stock [volume](https://www.investopedia.com/terms/v/volume.asp) chart is a commonly used type of vertical bar graph. A [histogram](https://www.investopedia.com/terms/h/histogram.asp) is an example of a bar graph used in statistical analysis that depicts a probability distribution in some data or sample.

* **PIE Chart:**

A pie chart is a type of graph that represents the data in the circular graph. The slices of pie show the relative size of the data, and it is a type of pictorial representation of data. A pie chart requires a list of categorical variables and numerical variables. Here, the term “pie” represents the whole, and the “slices” represent the parts of the whole.

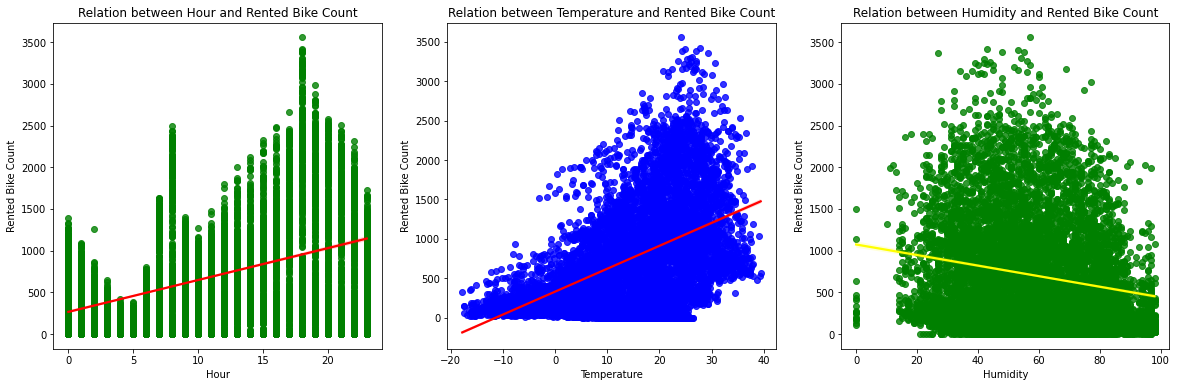
* **Distplot**

****

Seaborn distplot lets you show a histogram with a line on it. This can be shown in all kinds of variations. We use seaborn in combination with matplotlib, the Python plotting module.

A distplot plots a univariate distribution of observations. The distplot() function combines the matplotlib hist function with the seaborn kdeplot() and rug plot() functions.

* **Reg plot()**

****

To evaluate the regression model, there are many other interrelated contradictory approaches. Whenever the predicted output is a continuous as well as a cumulative value, it is referred to as a prediction model. Numerous other approaches can be employed. The most basic of which is the linear model. It integrates the values to the optimal higher dimensional space that passes through all of the vertices. The regplot() function is used to create the regression plots.

**Modelling approach**

Since our data contains no outliers, and many categorical attributes. It would be wise to model in both linear models as well as tree models

In our project we will be using Linear regression, Decision Tree Regressor, XG Boost, Gradient Boost Regressor, Random Forest,Lasso Regression,Ridge Regression and Elastic Net

Final choice of model will depend on whether interpretability or accuracy is important to the stakeholders.

Choice of split is taken as test train split, because of the computational power available and to reduce overfitting

Evaluation metrics are MSE, RMSE, R2 and Adjusted R2 to punish outliers, and choose a model that is able to generalize the results for all points including outliers.

Hyperparameter tuning is done to prevent overfitting, and the best parameters are chosen using GridsearchCV

**Data Preparation for Regression**

1**.**Linear Assumption: Linear regression assumes that the relationship between your independent and dependent is linear. It does not support anything else. This may be obvious, but it is good to remember when you have a lot of attributes. You may need to transform data to make the relationship linear (e.g. log transform for an exponential relationship).

2.Remove Outlier: Linear regression assumes that your independent and dependent variables are not noisy. Consider using data cleaning operations that let you better expose and clarify the signal in your data. This is most important for the output variable and you want to remove outliers in the output variable (y) if possible.

3.Remove Collinearity: Linear regression will over-fit your data when you have highly correlated input variables. Consider calculating pairwise correlations for your input data and removing the most correlated.

4.Gaussian Distributions: Linear regression will make more reliable predictions if your independent and dependent variables have a Gaussian distribution. You may get some benefit using transforms (e.g. log or BoxCox) on you variables to make their distribution more Gaussian looking.

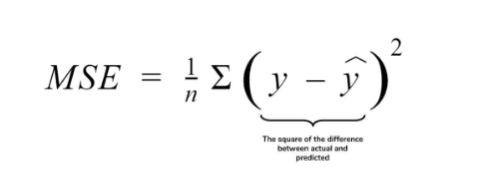
5.Rescale Inputs: Linear regression will often make more reliable predictions if you rescale input variables using standardization or normalization.

**Regression Model Evaluation Metrics**

After the model is built, if we see that the difference in the values of the predicted and actual data is not much, it is considered to be a good model and can be used to make future predictions. Few metric tools we can use to calculate error in the model

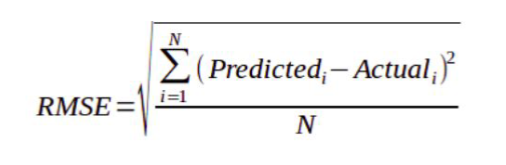
1.MSE (Mean Squared Error)

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.



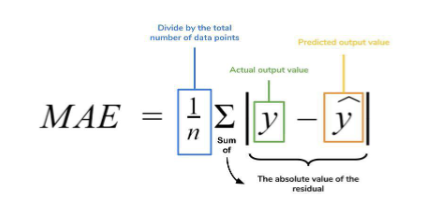
2.RMSE (Root Mean Squared Error)

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.



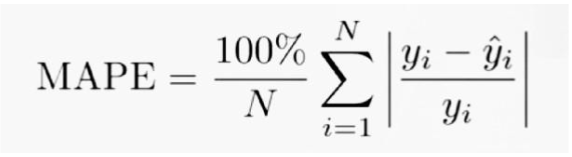
3.MAE (Mean Absolute Error)

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.



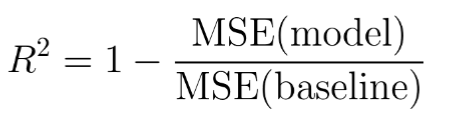
4.MAPE (Mean Absolute Percentage Error)

The mean absolute percentage error (MAPE), also known as Mean Absolute Percentage Deviation (MAPD), is a measure of prediction accuracy of a forecasting method in statistics, for example in trend estimation, also used as a loss function for regression problems in machine learning.



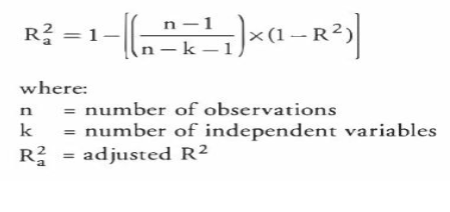
5.R2 (R – Squared)

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.



6.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.

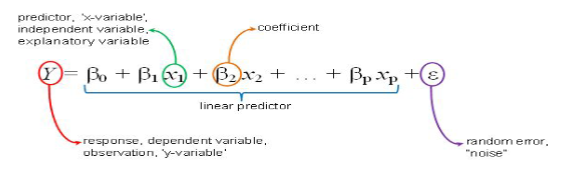


**Linear Regression**

Linear regression is simple to implement and easier to interpret the output coefficients.When you know the relationship between the independent and dependent variable is linear, this algorithm is the best to use because it’s less complex as compared to other algorithms.It works well irrespective of data size.

Outliers can have huge effect on the regression line..Linear regression assumes a linear relationship between dependent and independent variables, which is not the case in most of the real world problems..Prone to underfitting - Linear regression sometimes fails to capture the underneath pattern in data properly due to simplicity of the algorithm

The linear regression model can be represented by the following equation:

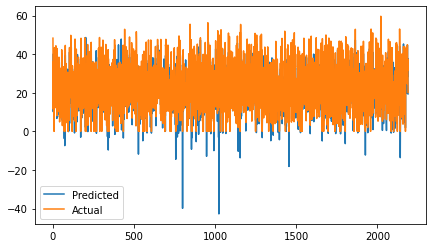


Y is the actual value

𝛃₀ is the bias term.

𝛃₁,...,𝛃p are the model parameters

x₁, x₂,...,xp are the feature values.



Evaluation metrics Test:

MSE : 37.13314681383164

RMSE : 6.093697302445506

R2 : 0.7629334199581261

Adjusted R2 : 0.7576185223205689

Evaluation metrics Actual vs Predicted:

MSE : 37.273629840755824

RMSE : 6.1052133329438885

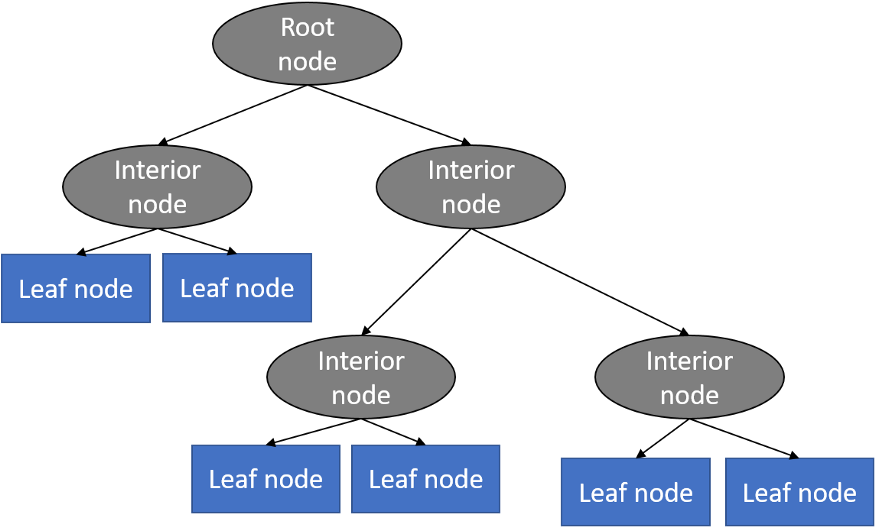
R2 : 0.750827145595724

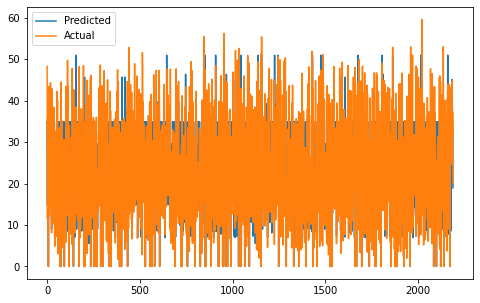
Adjusted R2 : 0.745240832185446

**Decision Tree Regression**

Decision Tree is one of the most commonly used, practical approaches for supervised learning. It can be used to solve both Regression and Classification tasks with the latter being put more into practical application.

It is a tree-structured classifier with three types of nodes. The Root Node is the initial node which represents the entire sample and may get split further into further nodes. The Interior Nodes represent the features of a data set and the branches represent the decision rules. Finally, the Leaf Nodes represent the outcome. This algorithm is very useful for solving decision-related problems.





Evaluation metrics Test:

Model Score: 0.7954152088220128

MSE : 32.04533125397695

RMSE : 5.660859586138571

R2 : 0.7954152088220128

Adjusted R2 : 0.7908285343817777

Evaluation metrics Actual vs Predicted:

MSE : 34.876743604873255

RMSE : 5.905653529023969

R2 : 0.7668502425580743

Adjusted R2 : 0.761623157851296.

**XG Boost**

Extreme Gradient Boosting (XGBoost) is an open-source library that provides an efficient and effective implementation of the gradient boosting algorithm.

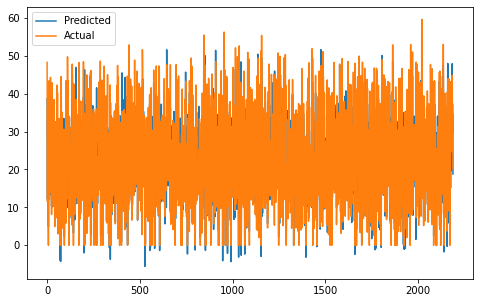
Shortly after its development and initial release, XGBoost became the go-to method and often the key component in winning solutions for a range of problems in machine learning competitions.

Regression predictive modeling problems involve predicting a numerical value such as a dollar amount or a height. XGBoost can be used directly for regression predictive modeling.

Gradient boosting refers to a class of ensemble machine learning algorithms that can be used for classification or regression predictive modeling 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.



Evaluation metrics Test:

Model Score: 0.892549996227313

MSE : 16.830532437483136

RMSE : 4.102503191648135

R2 : 0.892549996227313

Adjusted R2 : 0.8901410283706623

Evaluation metrics Actual vs Predicted:

Model Score: 0.892549996227313

MSE : 19.446866450146597

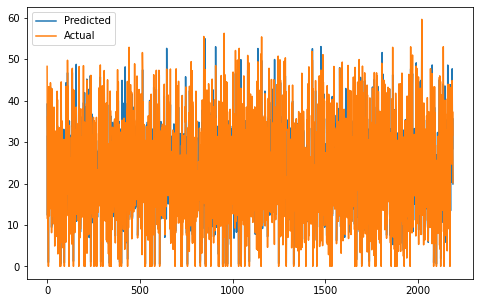
RMSE : 4.4098601395221815

R2 : 0.8699984079011416

Adjusted R2 : 0.8670838462847262

**Gradient Boosting**

Gradient boosting is one of the most popular machine learning algorithms for tabular datasets. It is powerful enough to find any nonlinear relationship between your model target and features and has great usability that can deal with missing values, outliers, and high cardinality categorical values on your features without any special treatment. While you can build barebone gradient boosting trees using some popular libraries such as XGBoost or LightGBM without knowing any details of the algorithm, you still want to know how it works when you start tuning hyper-parameters, customizing the loss functions, etc., to get better quality on your model.



Evaluation metrics Test:

Model Score: 0.9686919285509537

MSE : 4.903969228265

RMSE : 2.214490737904541

R2 : 0.9686919285509537

Adjusted R2 : 0.9679900194292562

Evaluation metrics Actual vs Predicted:

Model Score: 0.8923750050716048

MSE : 16.09956362287185

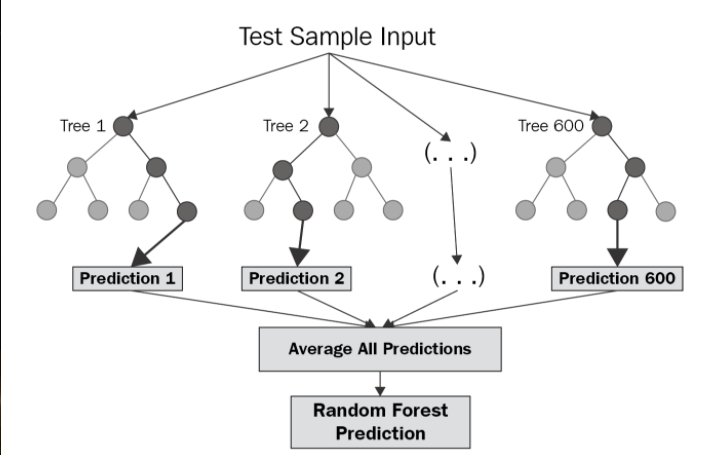
RMSE : 4.012426151703212

R2 : 0.8923750050716048

Adjusted R2 : 0.889962114012958

**Random Forest Regression**

Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.



The diagram above shows the structure of a Random Forest. You can notice that the trees run in parallel with no interaction amongst them. A Random Forest operates by constructing several decision trees during training time and outputting the mean of the classes as the prediction of all the trees. To get a better understanding of the Random Forest algorithm, let’s walk through the steps:

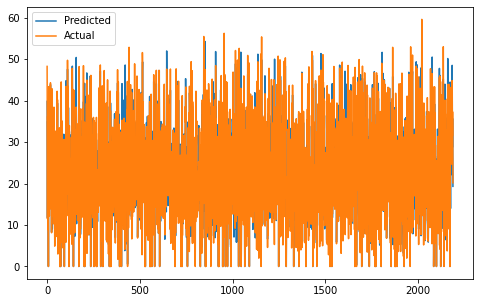
Pick at random k data points from the training set.

Build a decision tree associated to these k data points.

Choose the number N of trees you want to build and repeat steps 1 and 2.

For a new data point, make each one of your N-tree trees predict the value of y for the data point in question and assign the new data point to the average across all of the predicted y values.

A Random Forest Regression model is powerful and accurate. It usually performs great on many problems, including features with non-linear relationships. Disadvantages, however, include the following: there is no interpretability, overfitting may easily occur, we must choose the number of trees to include in the model.



Evaluation metrics Test:

Model Score: 0.985012541920559

MSE : 2.3475745975317563

RMSE : 1.532179688395508

R2 : 0.985012541920559

Adjusted R2 : 0.9846765316506789

Evaluation metrics Actual vs Predicted:

Model Score: 0.8934260679169712

MSE : 15.94233571162436

RMSE : 3.9927854577505615

R2 : 0.8934260679169712

Adjusted R2 : 0.8910367410883933

**Lasso Regression**

Lasso regression is a type of linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, like the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of muticollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination.

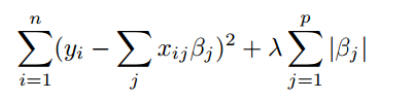
The acronym “LASSO” stands for Least Absolute Shrinkage and Selection Operator.

L1 Regularization

Lasso regression performs L1 regularization, which adds a penalty equal to the absolute value of the magnitude of coefficients. This type of regularization can result in sparse models with few coefficients; Some coefficients can become zero and eliminated from the model. Larger penalties result in coefficient values closer to zero, which is the ideal for producing simpler models. On the other hand, L2 regularization (e.g. Ridge regression) doesn’t result in elimination of coefficients or sparse models. This makes the Lasso far easier to interpret than the Ridge.

Performing the Regression

Lasso solutions are quadratic programming problems, which are best solved with software (like Matlab). The goal of the algorithm is to minimize:



Which is the same as minimizing the sum of squares with constraint Σ |Bj≤ s (Σ = summation notation). Some of the βs are shrunk to exactly zero, resulting in a regression model that’s easier to interpret.

A tuning parameter, λ, controls the strength of the L1 penalty. λ is basically the amount of shrinkage:

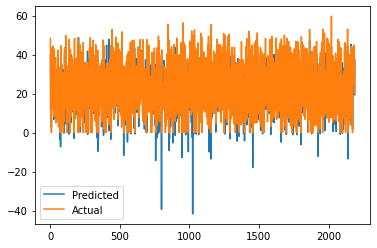
When λ = 0, no parameters are eliminated. The estimate is equal to the one found with linear regression.

As λ increases, more and more coefficients are set to zero and eliminated (theoretically, when λ = ∞, all coefficients are eliminated).

As λ increases, bias increases.

As λ decreases, variance increases.

If an intercept is included in the model, it is usually left unchanged.



Evaluation metrics Test:

MSE : 37.13314681383164

RMSE : 6.093697302445506

R2 : 0.7629334199581261

Adjusted R2 : 0.7576185223205689

Evaluation metrics Actual vs Predicted:

MSE : 37.18201422405232

RMSE : 6.097705652460794

R2 : 0.7514395926479578

Adjusted R2 : 0.7458670099516018

**Ridge Regression**

Ridge regression is a method of estimating the coefficients of multiple-regression models in scenarios where linearly independent variables are highly correlated. It has been used in many fields including econometrics, chemistry, and engineering.

Ridge regression was developed as a possible solution to the imprecision of least square estimators when linear regression models have some multicollinear (highly correlated) independent variables—by creating a ridge regression estimator (RR). This provides a more precise ridge parameters estimate, as its variance and mean square estimator are often smaller than the least square estimators previously derived.

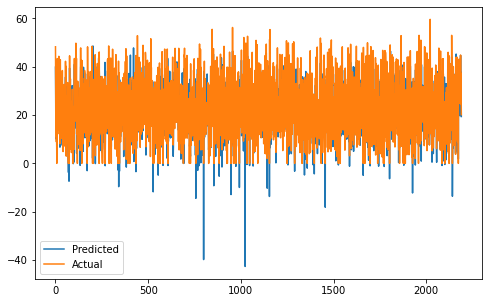
The cost function for ridge regression:

Min(||Y – X(theta)||^2 + λ||theta||^2)

Lambda is the penalty term. λ given here is denoted by an alpha parameter in the ridge function. So, by changing the values of alpha, we are controlling the penalty term. The higher the values of alpha, the bigger is the penalty and therefore the magnitude of coefficients is reduced.

It shrinks the parameters. Therefore, it is used to prevent multicollinearity

It reduces the model complexity by coefficient shrinkage



Evaluation metrics Test:

MSE : 37.13315291299508

RMSE : 6.093697802893993

R2 : 0.762933381019659

Adjusted R2 : 0.7576184825091236

Evaluation metrics Actual vs Predicted:

MSE : 37.26764694572257

RMSE : 6.104723330808905

R2 : 0.7508671410305504

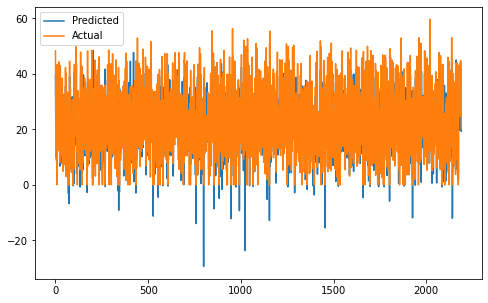
Adjusted R2 : 0.7452817242951307

**Elastic Net Regression**

Regression is a modeling task that involves predicting a numeric value given an input.

Linear regression is the standard algorithm for regression that assumes a linear relationship between inputs and the target variable. An extension to linear regression involves adding penalties to the loss function during training that encourage simpler models that have smaller coefficient values. These extensions are referred to as regularized linear regression or penalized linear regression.

Elastic net is a popular type of regularized linear regression that combines two popular penalties, specifically the L1 and L2 penalty functions.



Evaluation metrics Test:

MSE : 37.47983954768055

RMSE : 6.122078041619573

R2 : 0.7607200535243308

Adjusted R2 : 0.7553555334725643

Evaluation metrics Actual vs Predicted:

MSE : 36.469868314473736

RMSE : 6.039028755890614

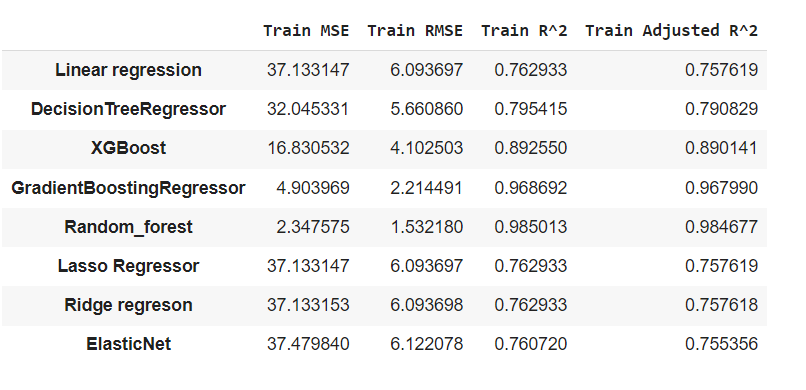
R2 : 0.756200262048822

Adjusted R2 : 0.7507344108476746

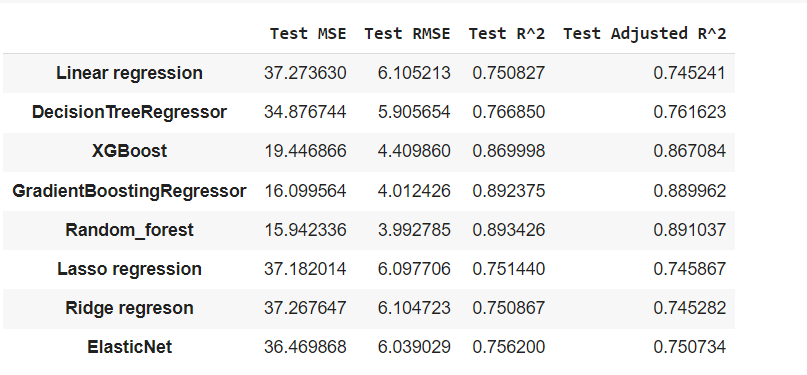
**Modelling Conclusion:**

* Since the alpha coefficient for Lasso, Ridge and Elastic Net came out to be almost 0, Lasso, Ridge, Elastic Net and Linear regressions had almost the same R-squared value.

**R-squared, RMSE values for Trained Dataset**



**R-squared, RMSE values for Test Dataset**



After looking at the both the images we get that RMSE value of random forest is very low thus R-squared values are higher, making it a best fit model for prediction.

Basis of higher accuracy random forest model is the best suited one while basis of interpretability decision tree models is a best fit.

**Limitations and Scope for Model improvement:**

Below are few limitations in this analysis and ideas to improve model prediction accuracy

* Solar Radiation, Rainfall and Snowfall weren’t used due to very low correlation. This might have been due to several instances where there magnitude was equal to 0. One possible method could be to first estimate those and then use it as a feature to estimate count.
* Seasons, Holiday, and Functioning day columns were classification problems.

**Conclusion:**

According to the analysis we conclude that Random Forest is the model which has the highest accuracy compare to any other model if we compare the R^2 value.

As per the EDA perspective we have the analysis that Rented Bike Count is highest when there is No Holiday compare to Holiday

June is the month where we have the highest bike count we see.

According to our seasonal analysis we get that winter has the lowest Rented Bike Count and summer has the highest Rented Bike Count among all the seasons.

As per the multicollinearity perspective we see the “Temperature” and “Dew Point Temperature” are highly correlated to each other.

We have also seen that most of our independent variable are positively correlated to the dependent variable except 3 independent variable which is “Humidity, Rainfall and Snowfall”.

**References**

<https://www.academia.edu/46510368/Regression_Model_to_Predict_Bike_Sharing_Demand>

<https://www.computerscijournal.org/vol10no1/prediction-of-bike-sharing-demand/>

<https://towardsdatascience.com/end-to-end-case-study-bike-sharing-demand-dataset-53201926c8db>

<https://github.com/syedsharin/Seoul-Bike-Sharing-Demand-Prediction>

<https://corporatefinanceinstitute.com/resources/knowledge/other/elastic-net/>

<https://siyuanligit.github.io/Bike-Sharing-Demand-Kaggle/>