Capstone Project

Bike Sharing Demand Prediction

Team Members:

Anamika

Ayush Goyal

Sumaya Bai A R

CONTENT

1. Summary
2. Problem Statement
3. Dataset variable Description
4. Data Preprocessing
5. Exploratory Data Analysis
6. Machine Learning :
    - Defining Independent and Dependent variable
    - Dealing with positively skewed dependent variable
    - Machine Learning models :
        i. Linear Regression
        ii. Decision Tree
        iii. Random Forest
        iv. XgBoost
7. Model evaluation

Summary

Bike sharing is an innovative approach to urban mobility, combining the convenience and flexibility of a bicycle with the accessibility of public transportation. Bike sharing systems are a means of renting bicycles where the process of obtaining membership, rental, and bike return is automated via a network of kiosk locations throughout a city. Using these systems, people are able rent a bike from a one location and return it to a different place on an as-needed basis.

Here, we are using Bike Sharing Demand Prediction dataset and performing various ML techniques in order to predict the bike rental Demand.

Data Set Name: Bike Sharing Demand Prediction

Shape : Rows : 8760, Columns : 14

This dataset gives us the information for two years : 2017 and 2018

Problem Statement

The main motive of this project is to combine different historical usage patterns and predict the future bike rental demands.

The crucial part in a renting bike sharing system, is to make sure to have a stable supply of rental bikes.

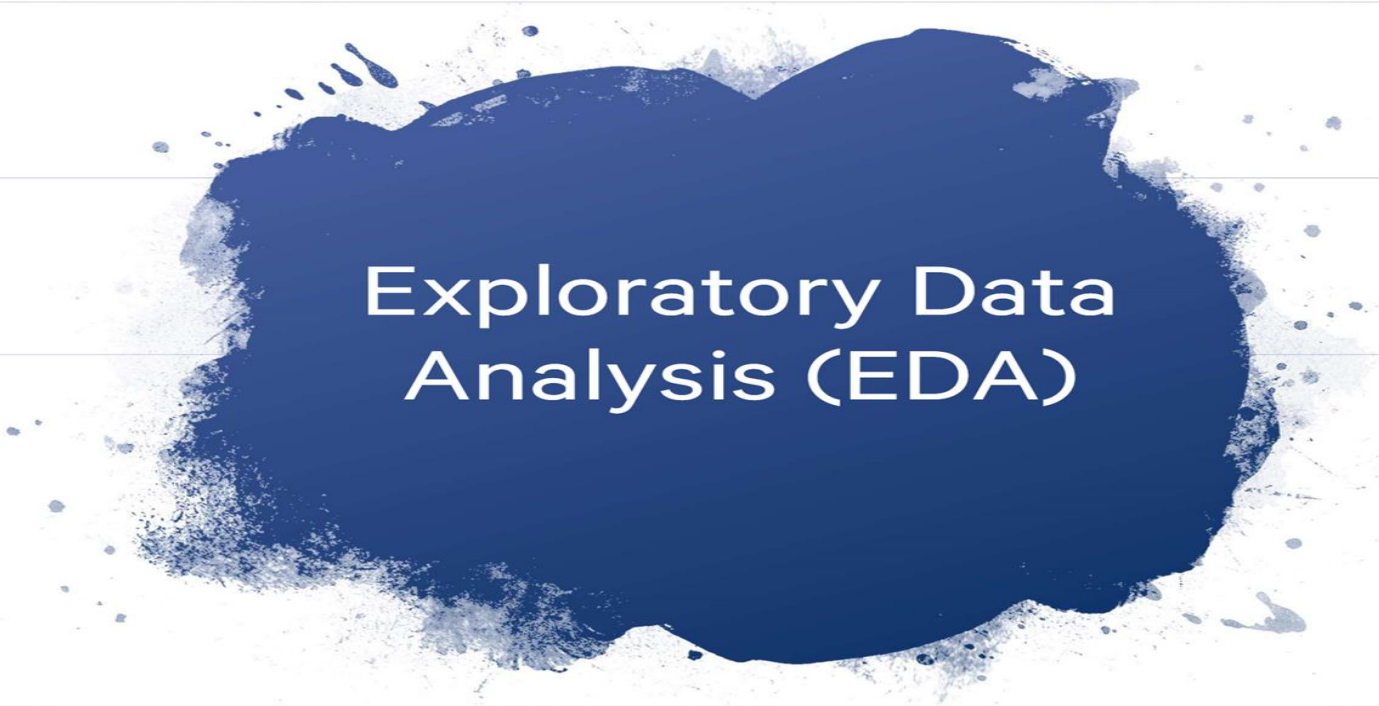
While having excess bikes results in wastage of resource (both with respect to bike maintenance and the land/bike stand required for parking and security), having fewer bikes leads to revenue loss (ranging from a short term loss due to missing out on immediate customers to potential longer term loss due to loss in future customer base), Thus, having a estimate on the demands would enable efficient functioning of these companies.

Thanks to Machine Learning, we can achieve this by predicting rental bike demand by using different machine learning algorithms.

In this project, we have used regression algorithms to achieve this.

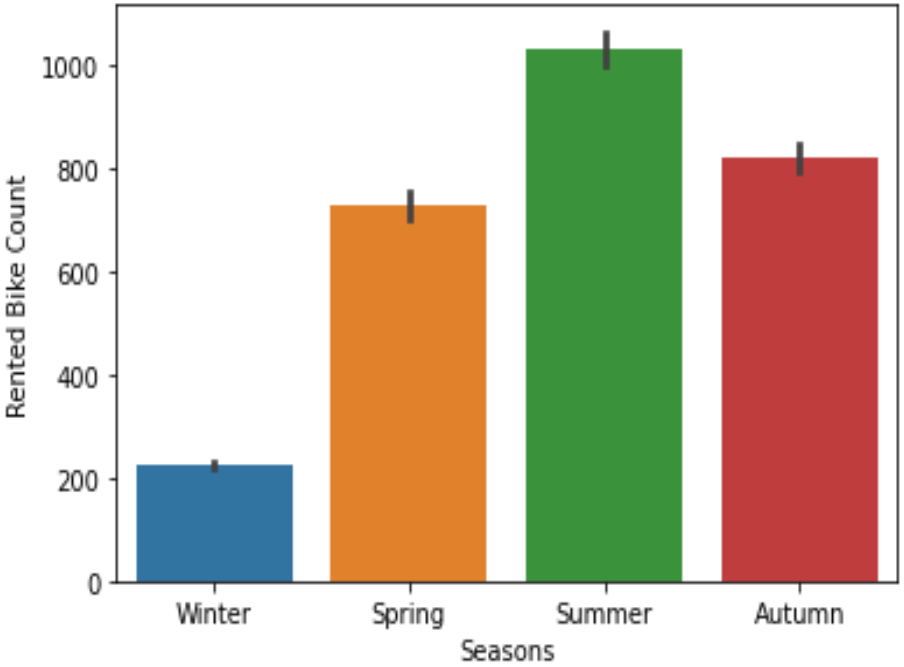
Dataset variable description

1. Date - This describes the actual date at which bike ride was taken
2. Rented Bike Count - This is the target variable and it tells us the number of bike rides taken by individuals.
3. Hour - It describes us the time or bike rides, we can interpret the peak times.
4. Temperature - It describes us the local temperature of the location during the bike rides.
5. Humidity - This describes the level of humidity in the weather during the ride.
6. Wind speed - This describes the average speed of wind while bike ride was taken
7. Visibility - This describes the outside environment visibility which might be affected due to adverse weather conditions sometimes like fog.
8. Dew Point temperature (in celsius) - This indicates the amount of moisture in the air.
9. Solar radiation - MJ/m2 - This describes us the amount of ultraviolet radiation.
10. Rainfall (mm) - This describes us the measurement of rainfall helps us to check if the rainfall is heavy or light.
11. Snowfall (cm) - This describes us the measurement of snowfall helps us to check if the rainfall is heavy or light.
12. Seasons - It indicates the the type of season like autumn, summer,spring,winter.
13. Holiday - It indicates whether it was official holiday or not
14. Functional Day - It indicates whether the bike ride was during functioning hours or non functioning hours .


Exploratory Data
Analysis (EDA)

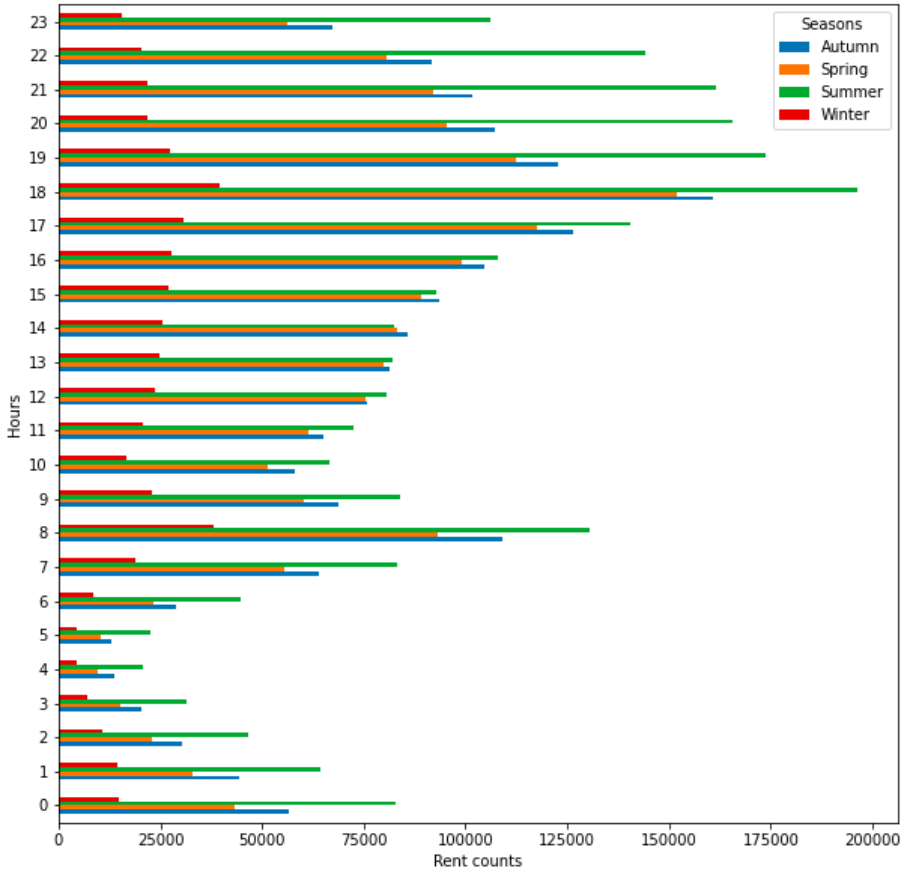
Season Vs Rented Bike Count

Count of bikes according to the Season



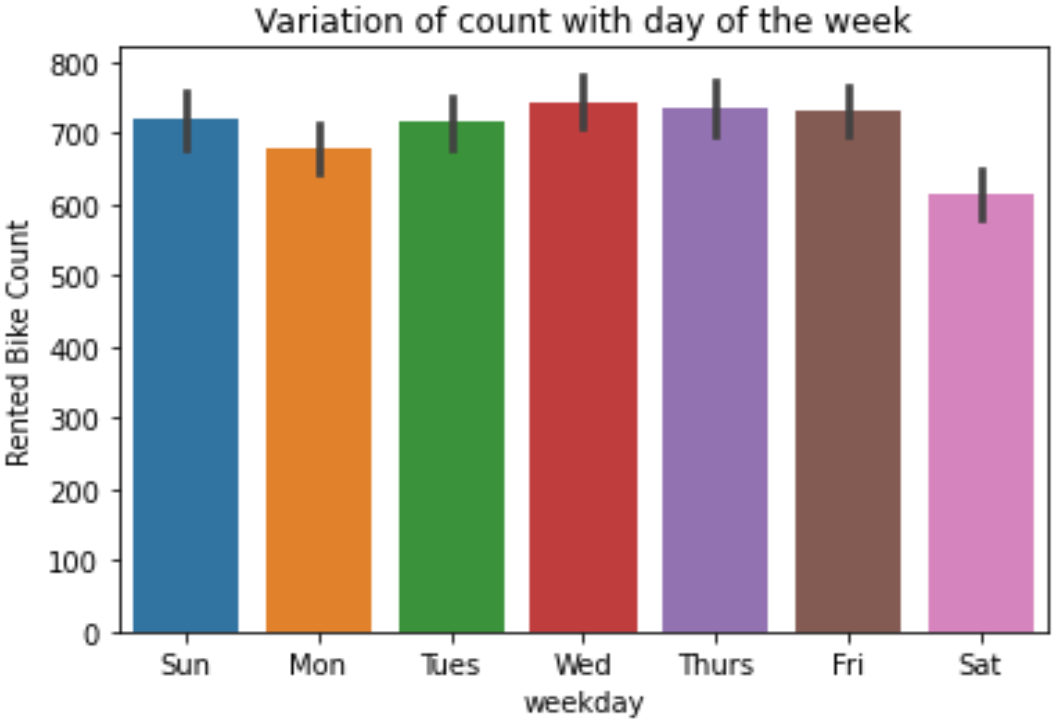
- The graph shows the peak season in which rented bikes were in more demand.
- It is shown that Summer season is the peak season.

Hours vs seasons



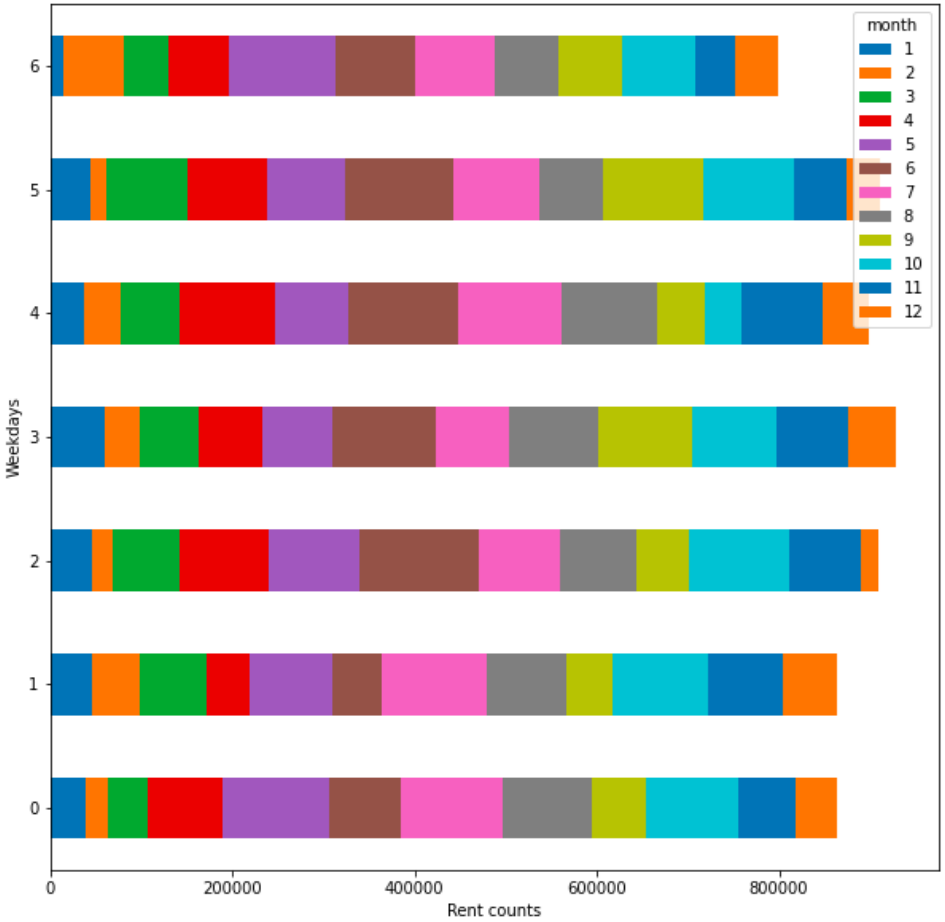
1) As we already concluded summer is the season with the highest bike counts.
2) Our peak times are 7am to 8am and 5pm to 10pm.
3) 6pm records highest number of bike counts.
4) Based upon the timings being officially, looks like bikes are highly in demand by the corporates.
5) Winters have lowest demand
6) Spring and autumn are having almost equal demands with a slight difference in counts.

Weekday Vs Rented Bike Count



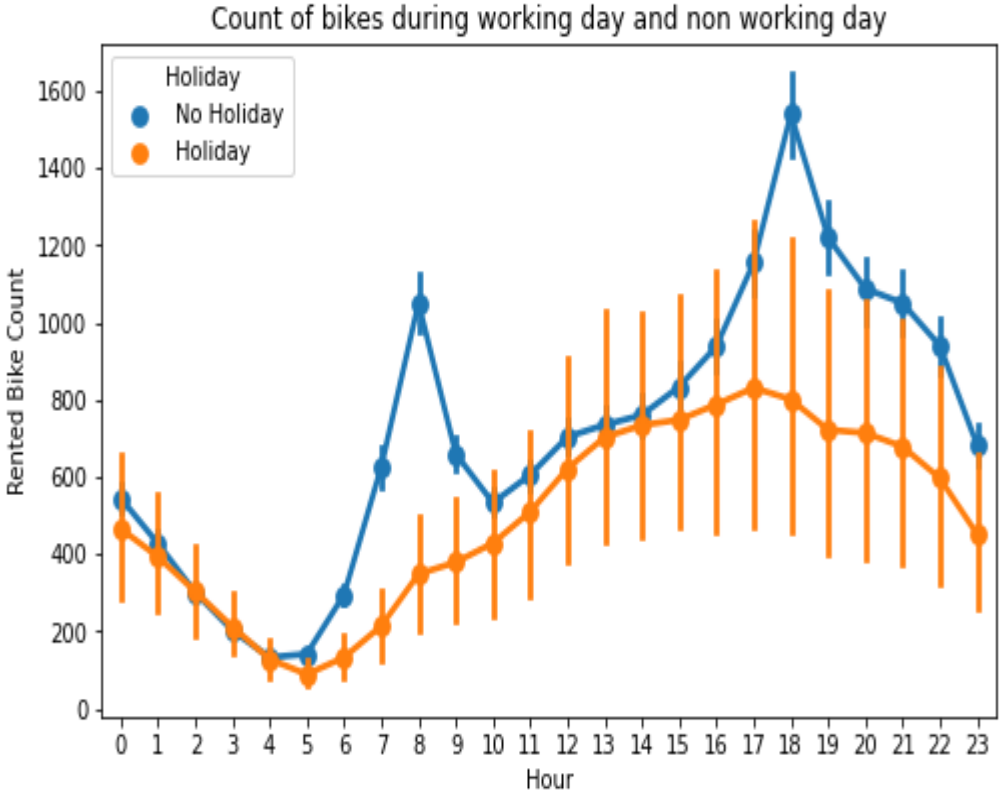
- This graph tells us which was the busiest day.
- We can see that almost everyday we had more or less same demand.
- But Wednesday and Thursday being more busy.

Weekdays vs Months



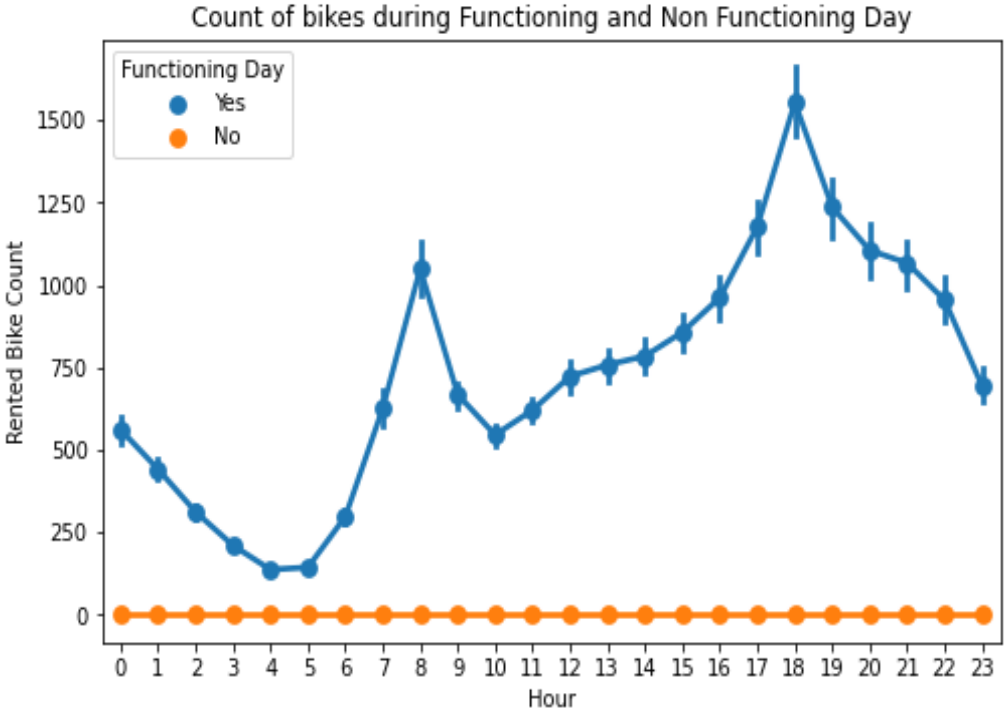
1) Here, we analysed the total bike rides according to given days and months regardless of the years. And to the surprise we can see that may, june, july, august, october has higher number of counts and we have a higher count wednesdays and saturdays recorded the lowest counts.

Holiday Vs Hour



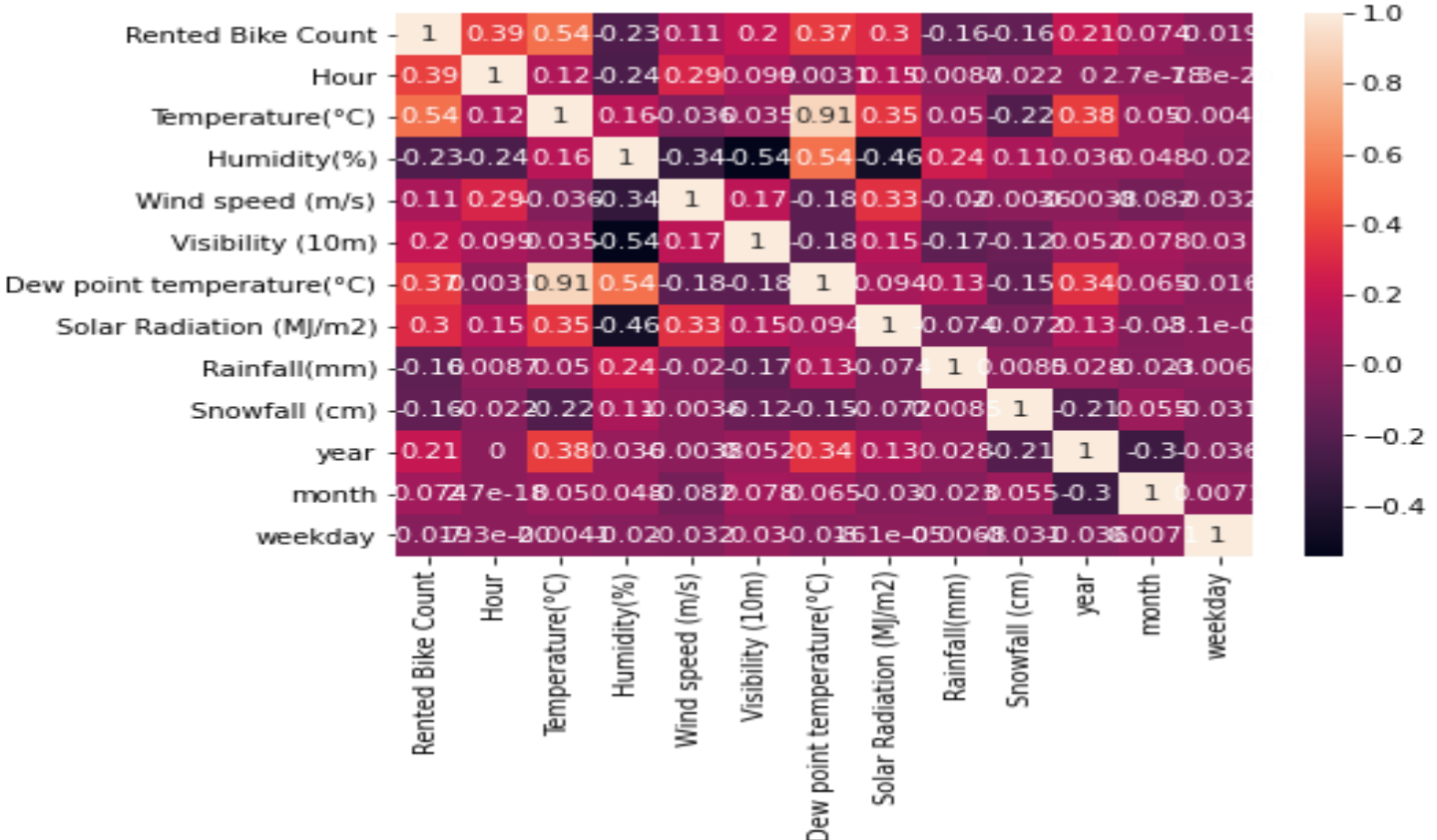
- * During working days there is a high demand around the 7th hour and 18th hour. There is a lower demand during 0 to 5th hour and 10 to 14th hour.
- * During non working days there is a high demand during 15 to 19th hour. There is a lower demand around the 5th hour.

Functioning Day Vs Hour



- This graph shows the rented bike count during functioning and non functioning day..
- Non functioning day is the day on which there is strike of two wheeler vehicles that's why count of rented bike is zero.

Correlation



VIF

	variables	VIF
0	Hour	1.211351
1	Temperature(°C)	5.188875
2	Humidity(%)	2.680056
3	Wind speed (m/s)	1.305459
4	Visibility (10m)	1.730240
5	Solar Radiation (MJ/m2)	1.961804
6	Rainfall(mm)	1.071850
7	Snowfall (cm)	1.137642
8	month	1.222871
9	weekday	1.004545
10	day	1.072503
11	season_Autumn	inf
12	season_Spring	inf
13	season_Summer	inf
14	season_Winter	inf
15	hol_Holiday	inf
16	hol_No Holiday	inf
17	fnc_No	inf
18	fnc_Yes	inf

VIF after Rented Bike Count,Dew point temperature, Date are dropped.

VIF before dropping any features.

```
print(vif_data)
```

```
/usr/local/lib/python3.7/dist-packages/statsmodel
  import pandas.util.testing as tm
                     feature         VIF
0                       Hour    1.211693
1            Temperature(°C)   90.063901
2                Humidity(%)   20.643757
3          Wind speed (m/s)     1.308342
4          Visibility (10m)     1.690745
5   Dew point temperature(°C)  117.747892
6     Solar Radiation (MJ/m2)    2.036299
7              Rainfall(mm)     1.086257
8              Snowfall (cm)    1.131599
9                      year     1.890010
10                    month     1.601193
11                  weekday     1.010118
12            season_Autumn          inf
13            season_Spring          inf
14            season_Summer          inf
15            season_Winter          inf
16              hol_Holiday          inf
17           hol_No Holiday          inf
18                   fnc_No          inf
19                   fnc_Yes         inf
```

Machine Learning:

Target variable and Feature Variable:

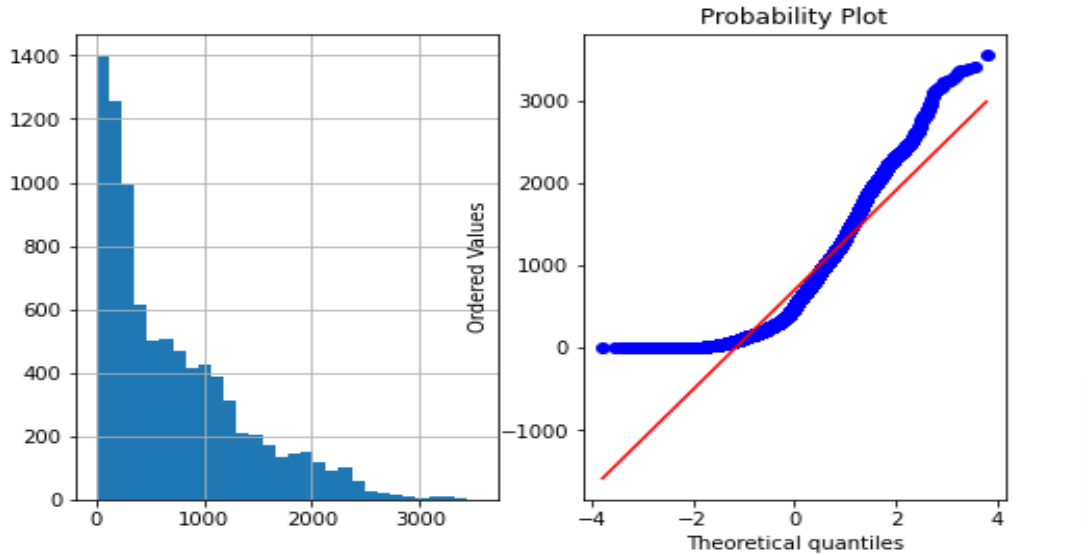
a) Target variable : Our aim is to predict the number of rented bikes that will be needed in a particular point of time, so the column “Rented Bike Count” will be our target Variable.

a) Feature Variable : Except ‘Rented Bike Count, Date and Dew point Temperature’ we have selected every other columns as our features.

Rightly skewed target variable :

Our target variable ‘ Rented Bike Count’ seems to be rightly skewed.

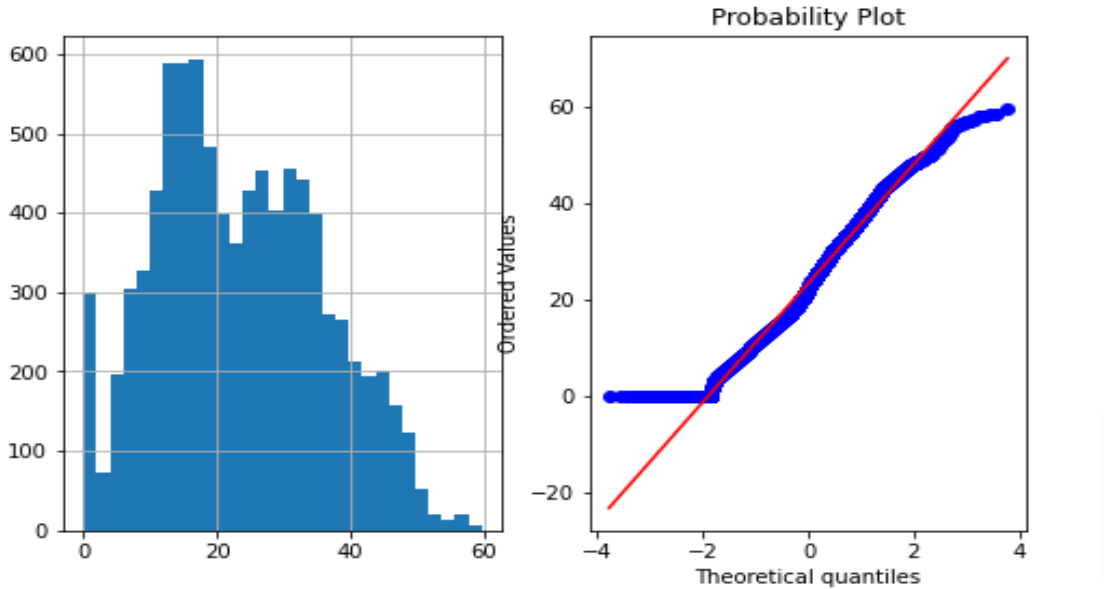
```
plotvariale(df, 'Rented Bike Count')
```



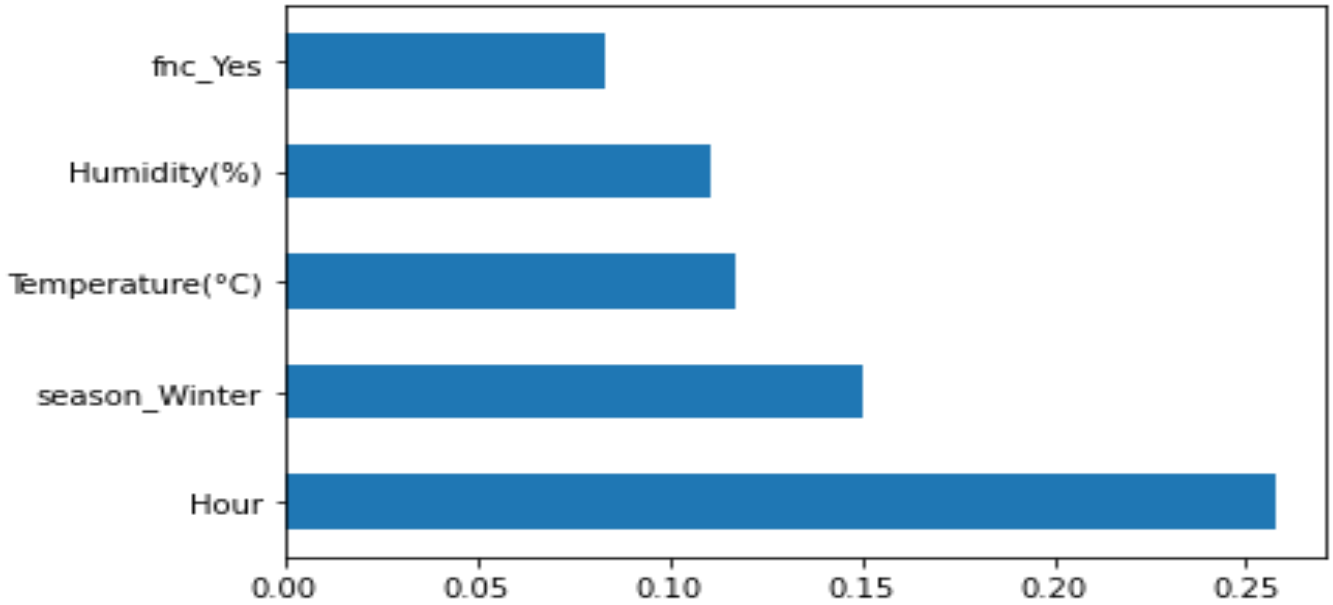
Square Root Transformation:

To change ‘Rented Bike Count’ to a normally distributed variable we are using Square root Transformations.

```
plotvariale(df, "Rented Bike Count")
```



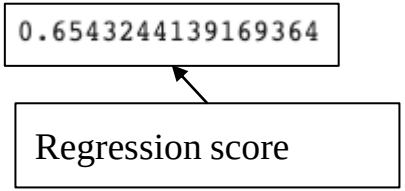
Feature Importance

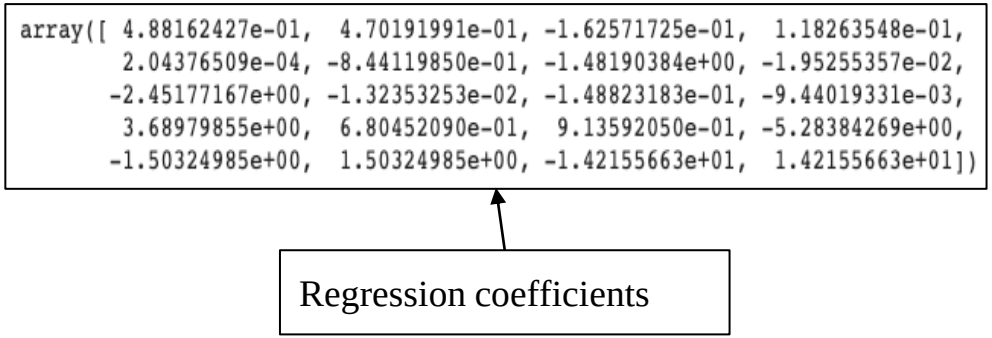


We have used ExtraTreeRegressor to know which are the important features

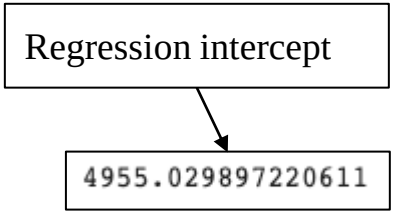

Modeling

Linear regression working





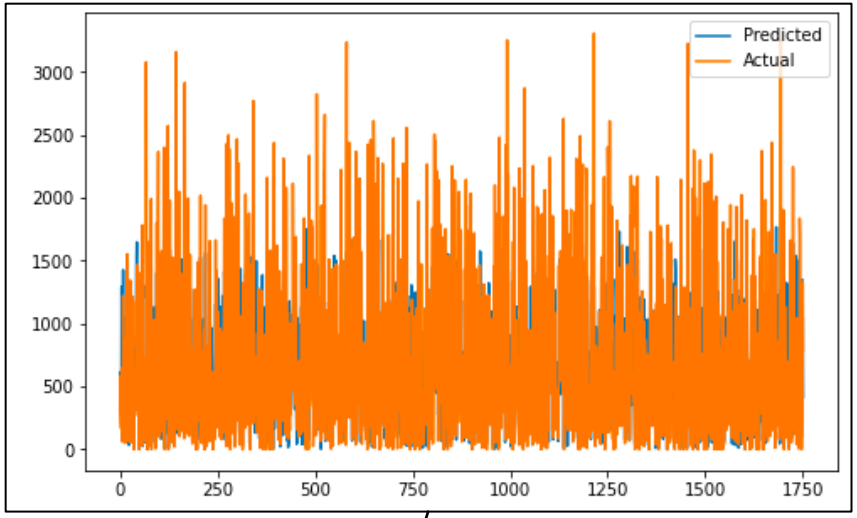
After applying the square root transformation we got a regression score of 0.65 which is a decent score and along with it we used total of 20 coefficients with intercept to make prediction.



Linear regression results

```
MSE : 174539.8488427653
RMSE : 417.779665425168
```

RMSE considered better and shows how concentrated the data is around best fit. This is not a good fit as RMSE and MSE are terrible.




```
R2 : 0.5829619388349918
Adjusted R2 :  0.578143474812288
```

R2 shows how well terms(data points) fit a curve or line. Adjusted R2 also indicates how well terms fit a curve or line, but adjusts for the number of terms in a model. If you add more and more useless variables to a model, adjusted r-squared will decrease. If you add more useful variables, adjusted r-squared will increase. Here, we are considering and comparing on the basis of R2 and we are not satisfied with R2. Hence we will try another model.

Decision Tree working :

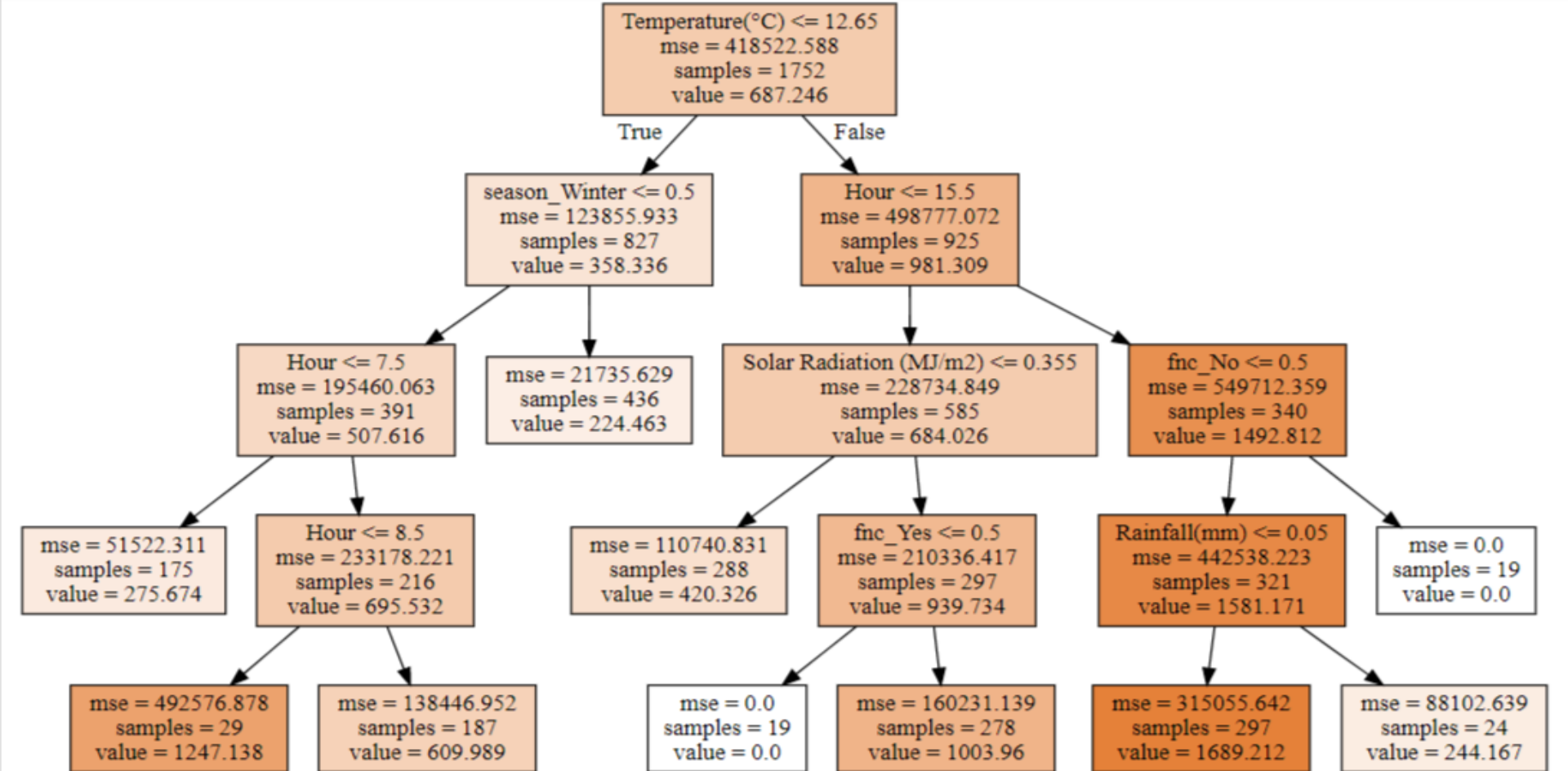
Decision tree when used without any parameter tuning we get a r2 score

```
MSE : 2773.9470863395704
RMSE : 52.66827400190337
```

R2 Score → 0.49009985162534647

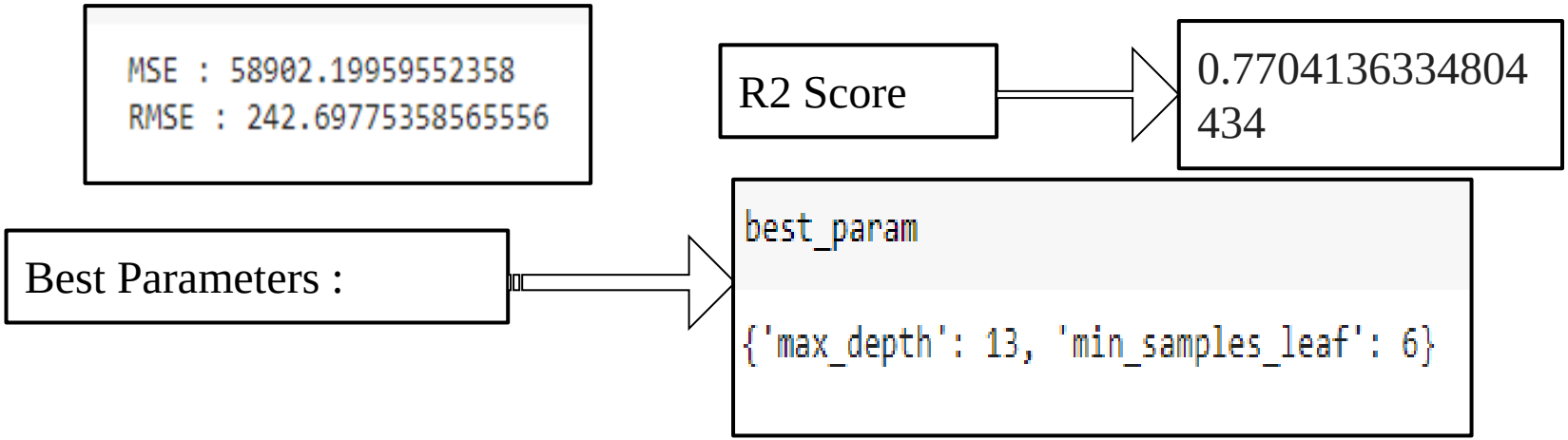
The decision tree gives better result when hyperparameter tuning are used.

Decision Tree Visualization :



Optimized Decision Tree :

Decision Tree can give better results when optimization techniques are used. Here, I've used GridSearchCV as a hyperparameter tuning method to yield better result.



XgBoost Working

We had chosen XgBoost Regressor with Grid Search for our prediction and the best hyperparameter obtained are as below:

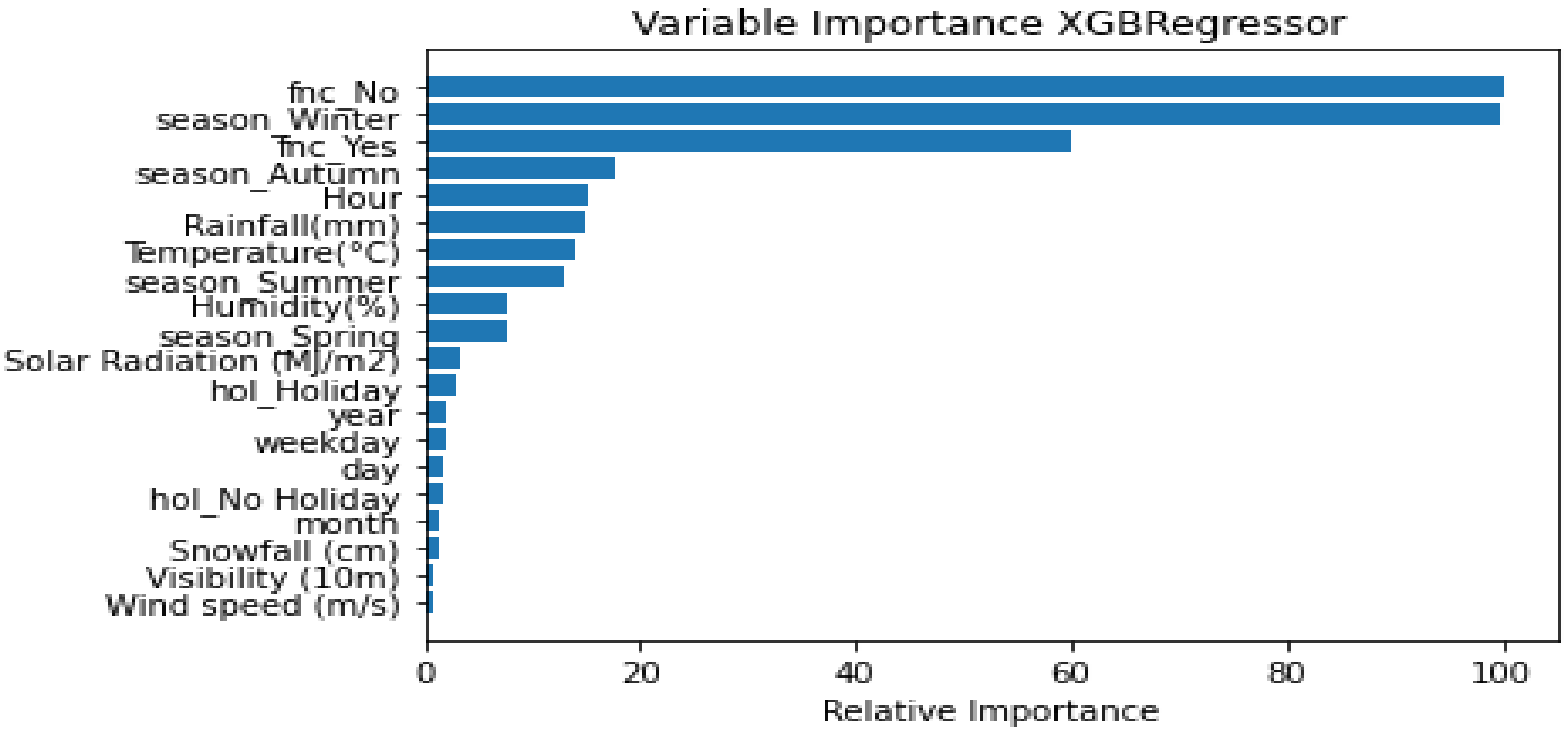
Best Hyperparameters:

Colsample_bytree : 1
Gamma : 0.0
Learning_rate : 0.1
Max_depth : 7
N_estimators : 200

XgBoost Result

R2_Score: 0.921780515480
MSE : 32736.6211288393
RMSE: 180.9326425188096

Variable importance XGB regressor



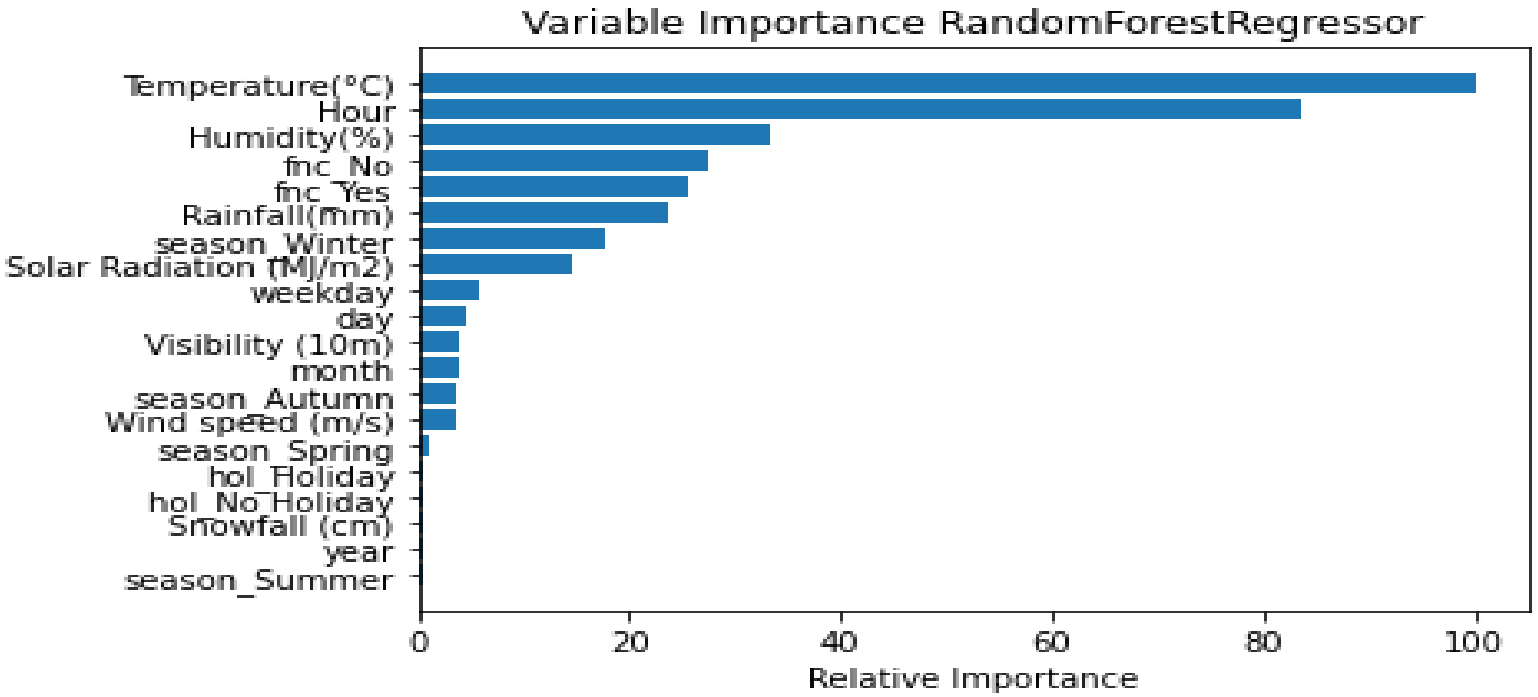
Random Forest Working

We had chosen Random Forest Regressor with Randomized Search CV for our prediction and best hyperparameter obtained as below:

Best Hyperparameter:

```
(bootstrap=True, ccp_alpha=0.0,
criterion='mse',
                       max_depth=17,
max_features='auto',
max_leaf_nodes=None,

max_samples=None,
min_impurity_decrease=0.0,

min_impurity_split=None,
min_samples_leaf=1,

min_samples_split=15,
min_weight_fraction_leaf=0.0,

n_estimators=600, n_jobs=None,
oob_score=False,

random_state=None, verbose=0,
warm_start=False)
```

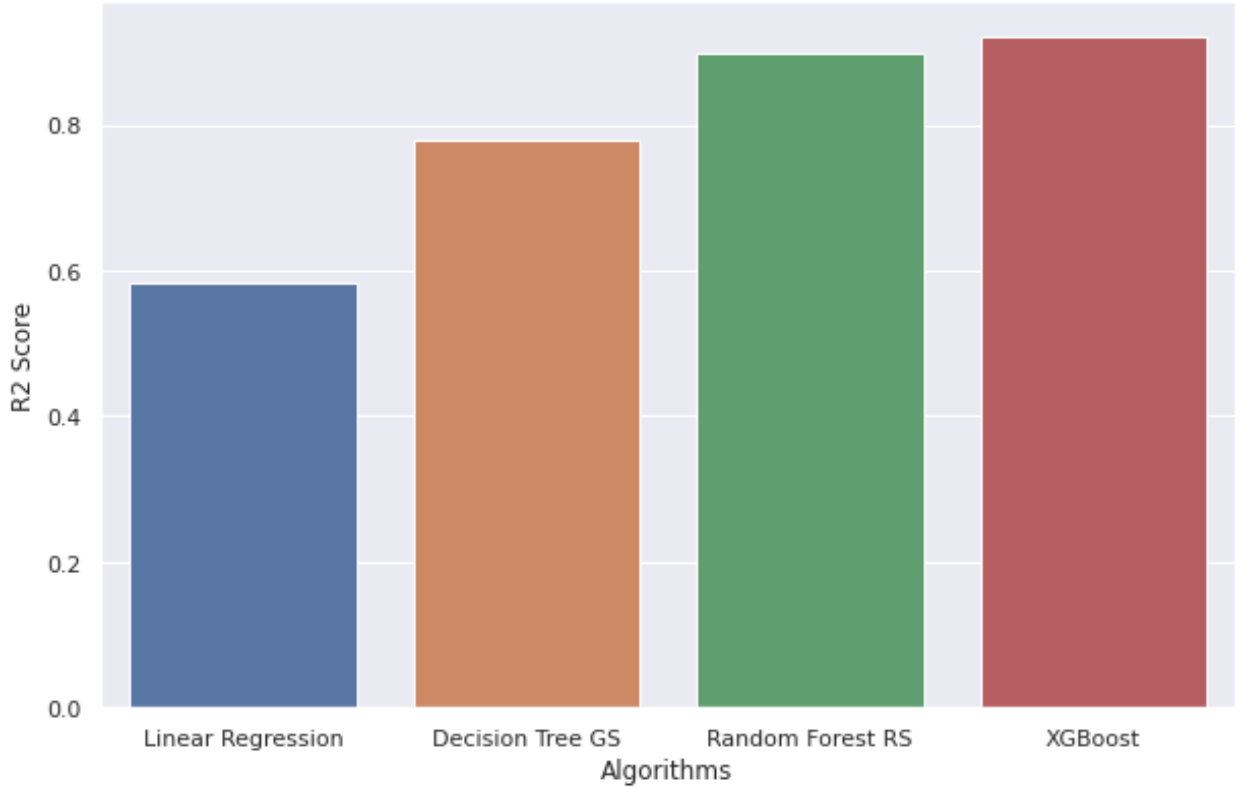
Variable importance Random forest regressor



Model Comparison

Model Name	R2_Score	MSE	RMSE
Linear Regression	0.5829619388349918	174539.8488427653	417.779665425168
Decision Tree	0.49009985162534647	157655.52461655077	3911405367.058909
Decision Tree with hyperparameter tuning	0.7780086150185419	56086.24227401238	236.8253412834285
Random Forest	0.8977625869300374	42788.66675491794	206.8542161884015
XgBoost	0.9217805154800872	32736.621128839934	180.9326425188096

Model Evaluation on the basis of R2 Score



Observation of different Models

Observation 1:

As seen in the table above linear regression is not giving great results, Decision tree with hyperparameter tuning had better R2_Score

Observation 2:

XgBoost Regressor and Random Forest have performed better R2_Score .

Conclusion

1. Out of the 5 models tried, XGBoost gives the best R2 score of 0.92 in test dataset.
2. Optimized random forest is the next best model out of the 5.
3. ‘Hour’ of the day holds the most important feature.