Capstone Project
Bike Sharing Demand Prediction

By

Dinesh Wagh.

CONTENT

Introduction

- Bike sharing systems are a type of bicycle rental service in which the procedure of obtaining a membership, renting a bike, and returning the bike is all done through a network of kiosks located around a city.
- People can rent a bike from one location and return it to a different location on an as-needed basis using these systems.
- The purpose of this study is to estimate bike rental demand by combining past bike usage trends with meteorological data. The data set consists of two year's worth of hourly rental data.

Abstract

- The main goal is to create a prediction model that can be used to anticipate the number of bike rentals each hour based on weather conditions. As a result, it would be easier to anticipate fast and accurately.
- Exploratory Data Analysis is performed on a dataset to determine the graph distribution by comparing the target variable to the other variables.
- We search for outliers and null values that were not discovered. We also use correlation analysis to pick the most critical and relevant features from the dataset, and then train the model using train test split.
- The goal of this project is to combine the historical bike usage patterns with the weather data to forecast bike rental demand. The data set consists of hourly rental data spanning two years

Problem Statement

Problem Description :

- 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 bike count required at each hour for the stable supply of rental bikes.
- The major goal is to create a prediction model that can be used to anticipate the number of bike rentals throughout the year based on weather conditions
- As a result, it will be easier to predict fast and accurately

Steps Involved

- Exploratory Data Analysis.
- Null values Treatment and Outliers
- Numerical and categorical Features
- Label encoding
- Correlation Analysis
- Train test Split
- Fitting different models

  1.Linear Regression.

  2.Lasso Regression.

  3.Polynomial Regression.

  4.Random Forest Regressor.

  5. Hyperparameter Tuning On Random Forest Regressor.

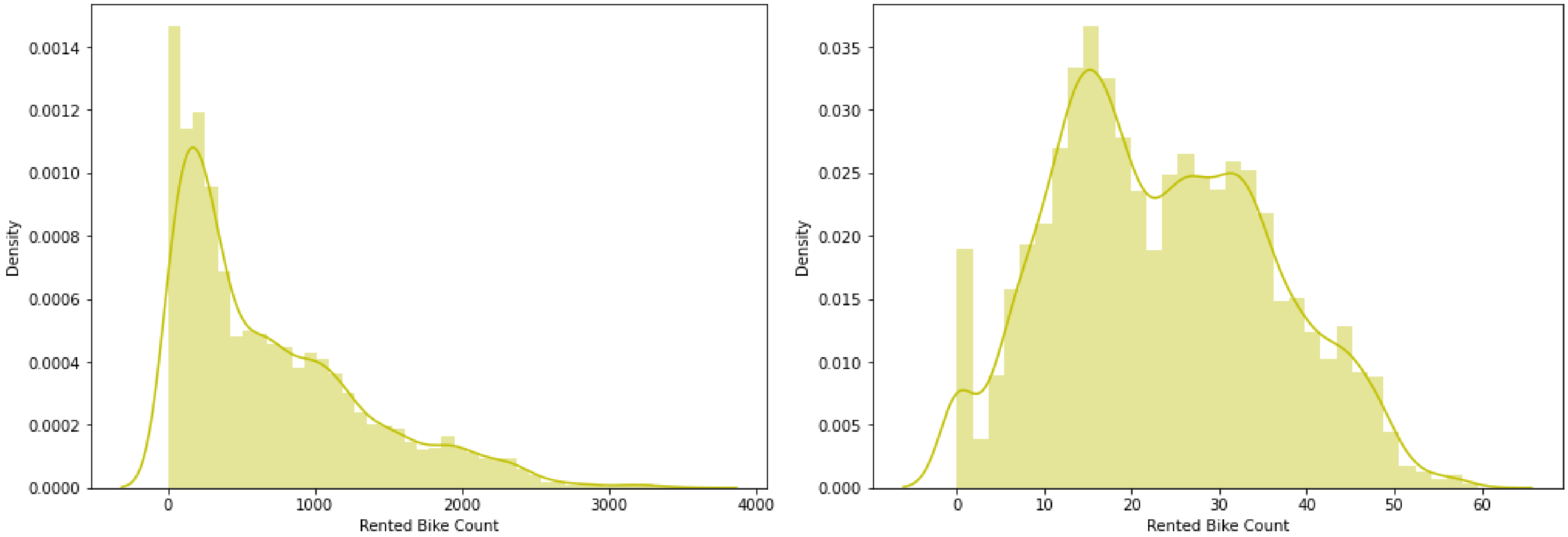
Data Description

- 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:

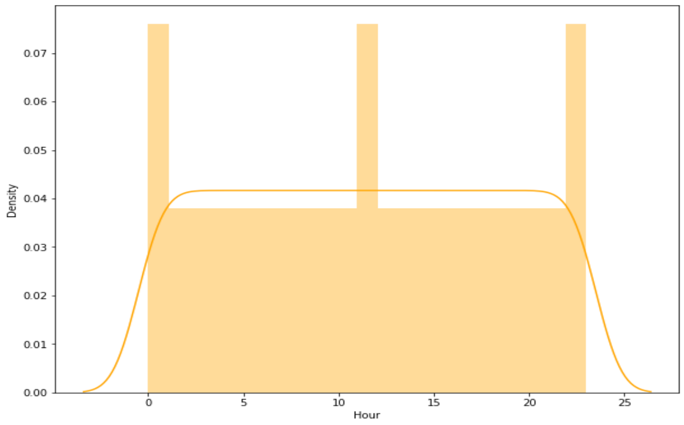
- Date : year-month-day
- Rented Bike count - Count of bikes rented at each hour
- Hour - Hour of he day
- Temperature-Temperature in Celsius
- Humidity - %
- Windspeed - m/s
- Visibility - 10m
- Dew point temperature - Celsius
- Solar radiation - MJ/m2
- Rainfall - mm
- Snowfall - cm
- Seasons - Winter, Spring, Summer, Autumn
- Holiday - Holiday/No holiday
- Functional Day - NoFunc(Non Functional Hours), Fun(Functional hours)

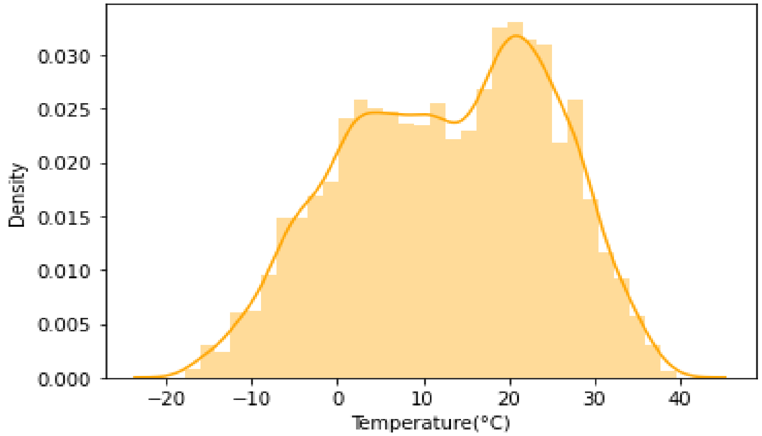
Distribution of our Dependent Variable

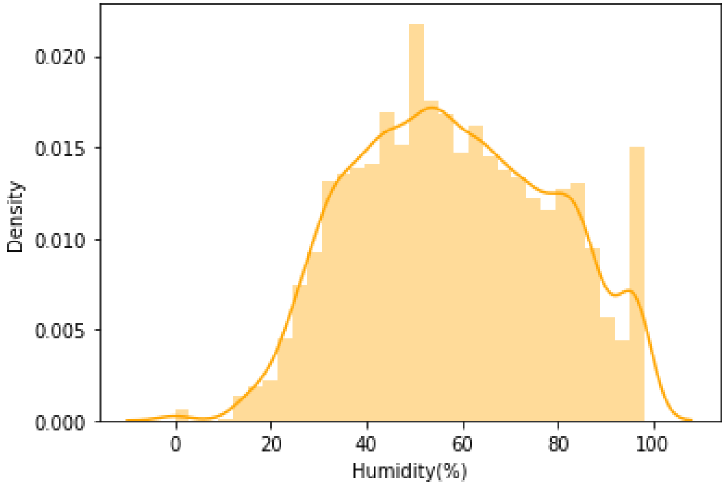


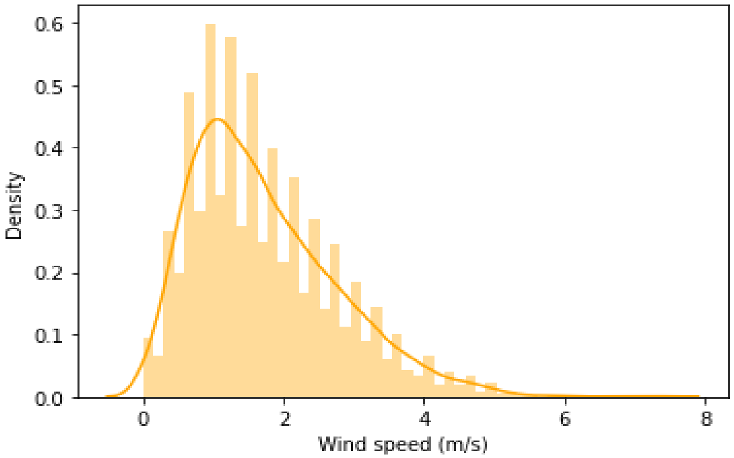
- As it was right skewed, so we have taken square root of dependent variable to visualize it in a better way.

Distribution of independent variables

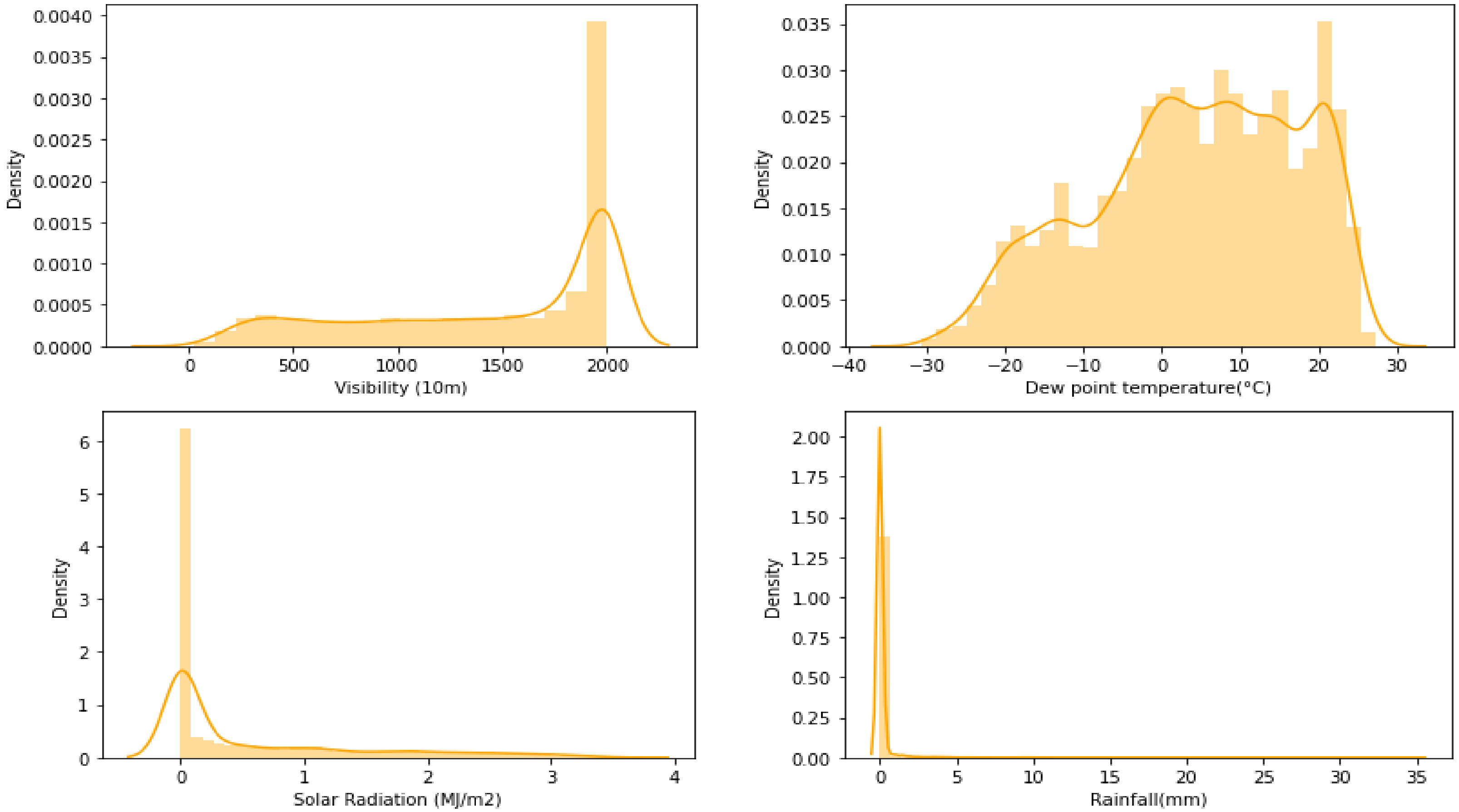




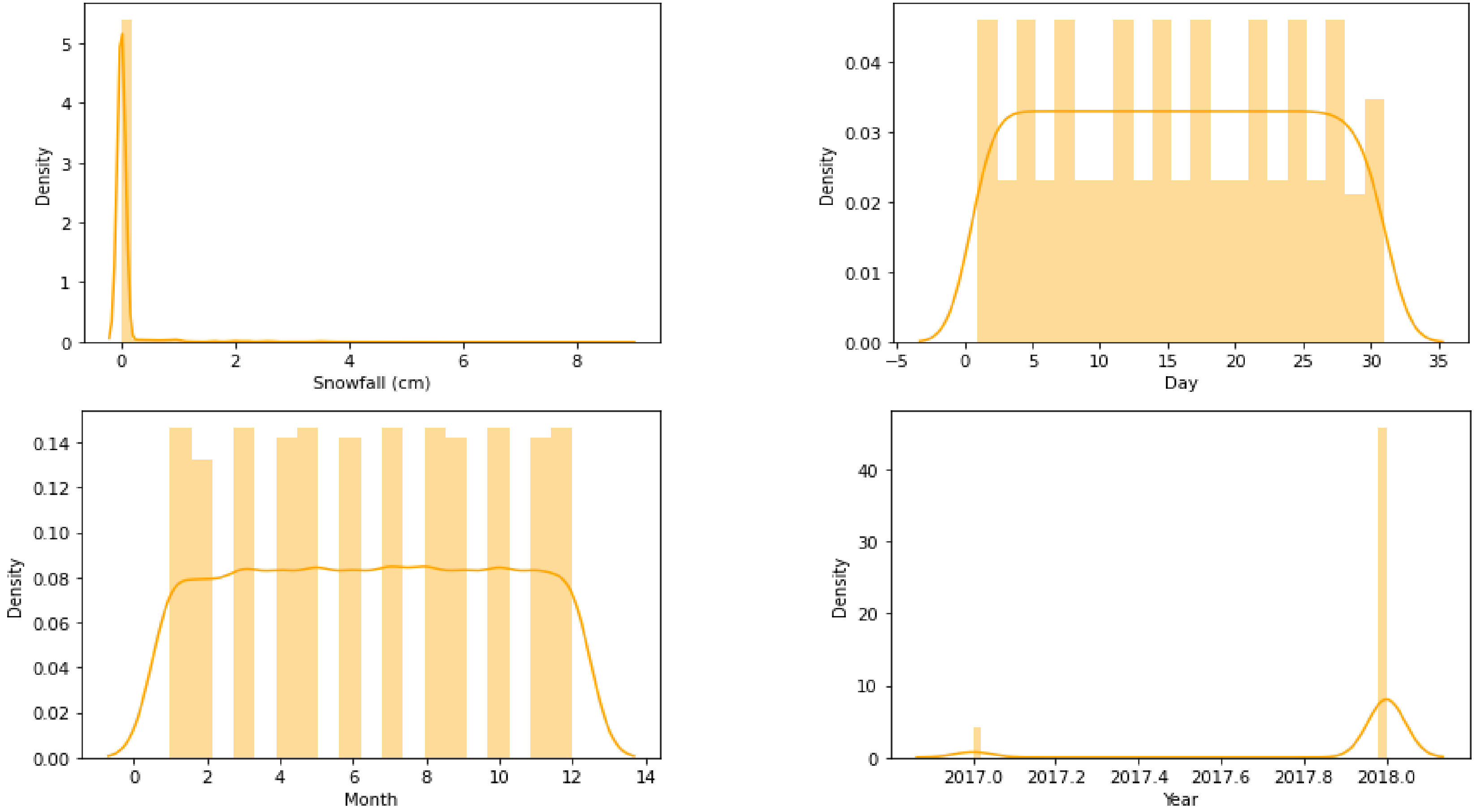




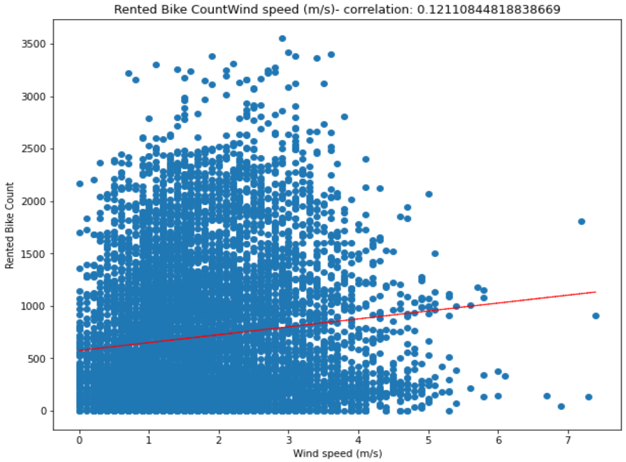
Distribution of independent variables

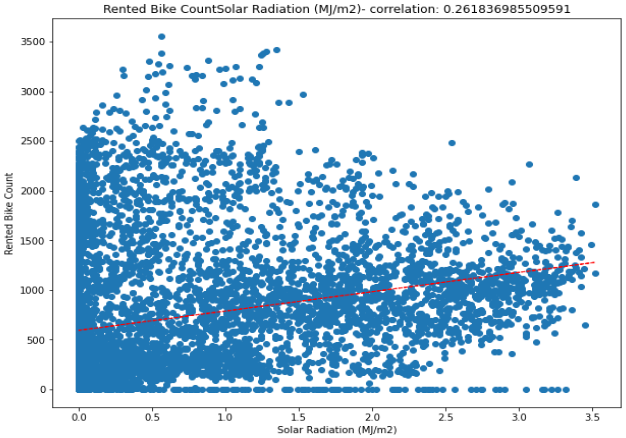


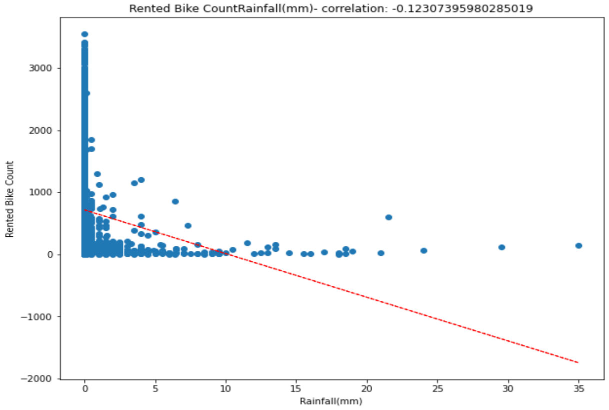
Distribution of independent variables

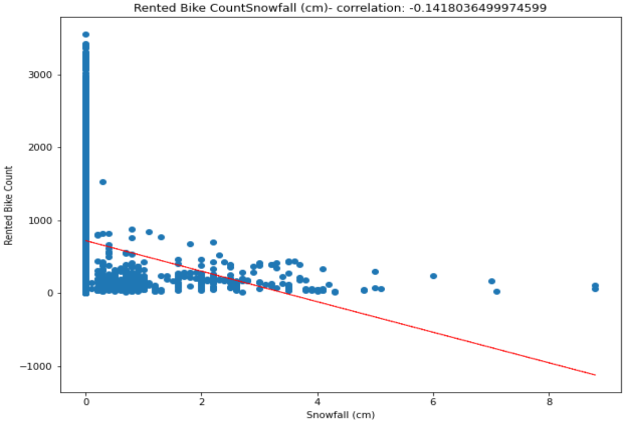


Visualizing the relationship b/w dependent & independent variable after transformation





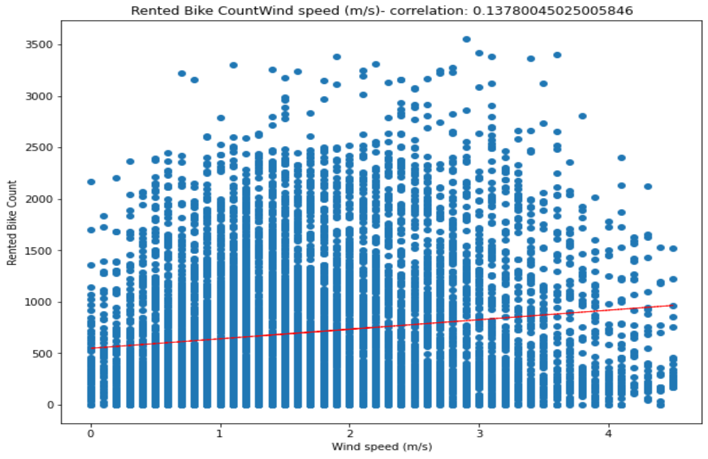


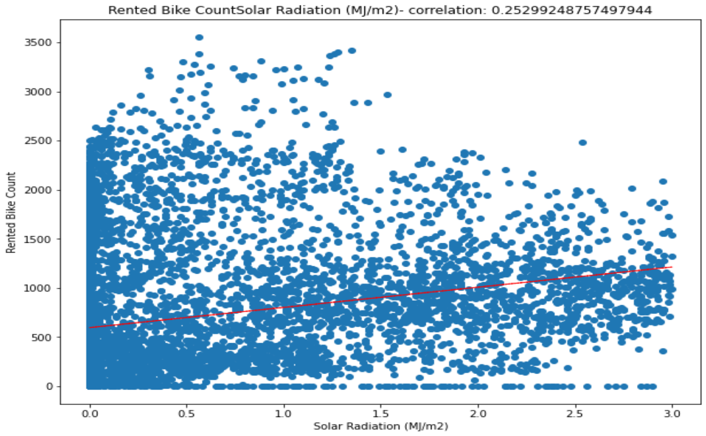


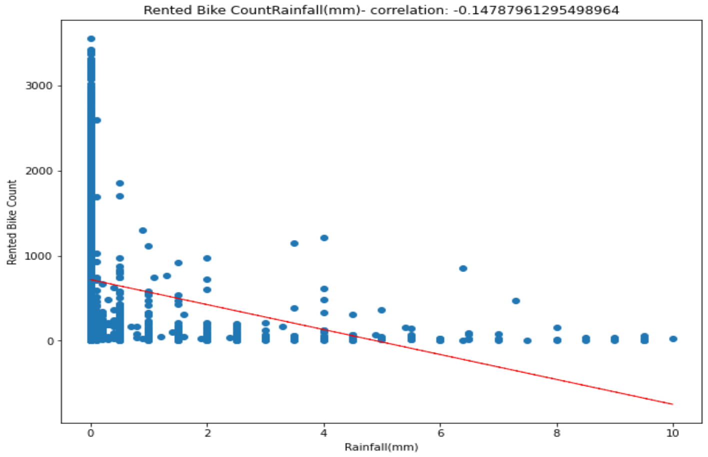
- So after visualizing these scatter plots we removed the unwanted or extra data which were making our dataset quite unwell.

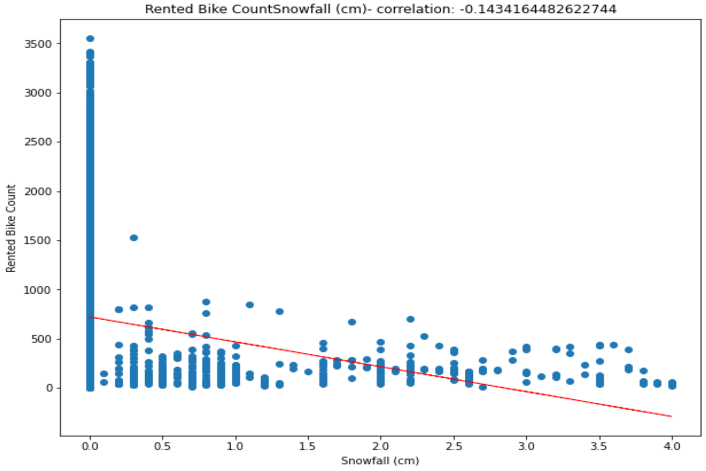
- So, for windspeed – value higher than 4.5m/s.
- Solar Radiation(MJ/m2) value higher than 3MJ/m2.
- Rainfall value higher than 10mm.
- snowfall value higher than 4cm. were not taken

Visualizing the plot after Removing Outliers from the dataset

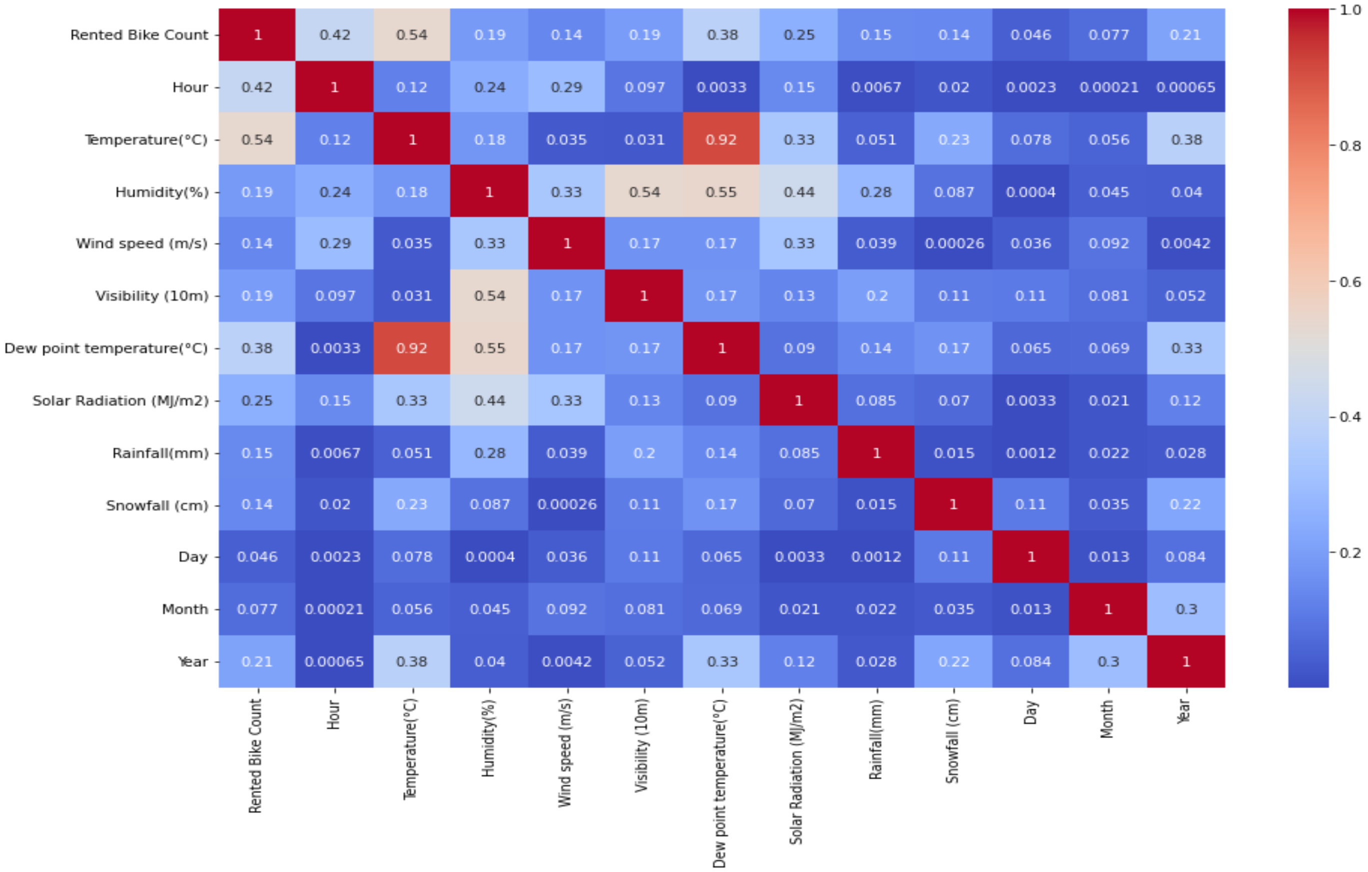






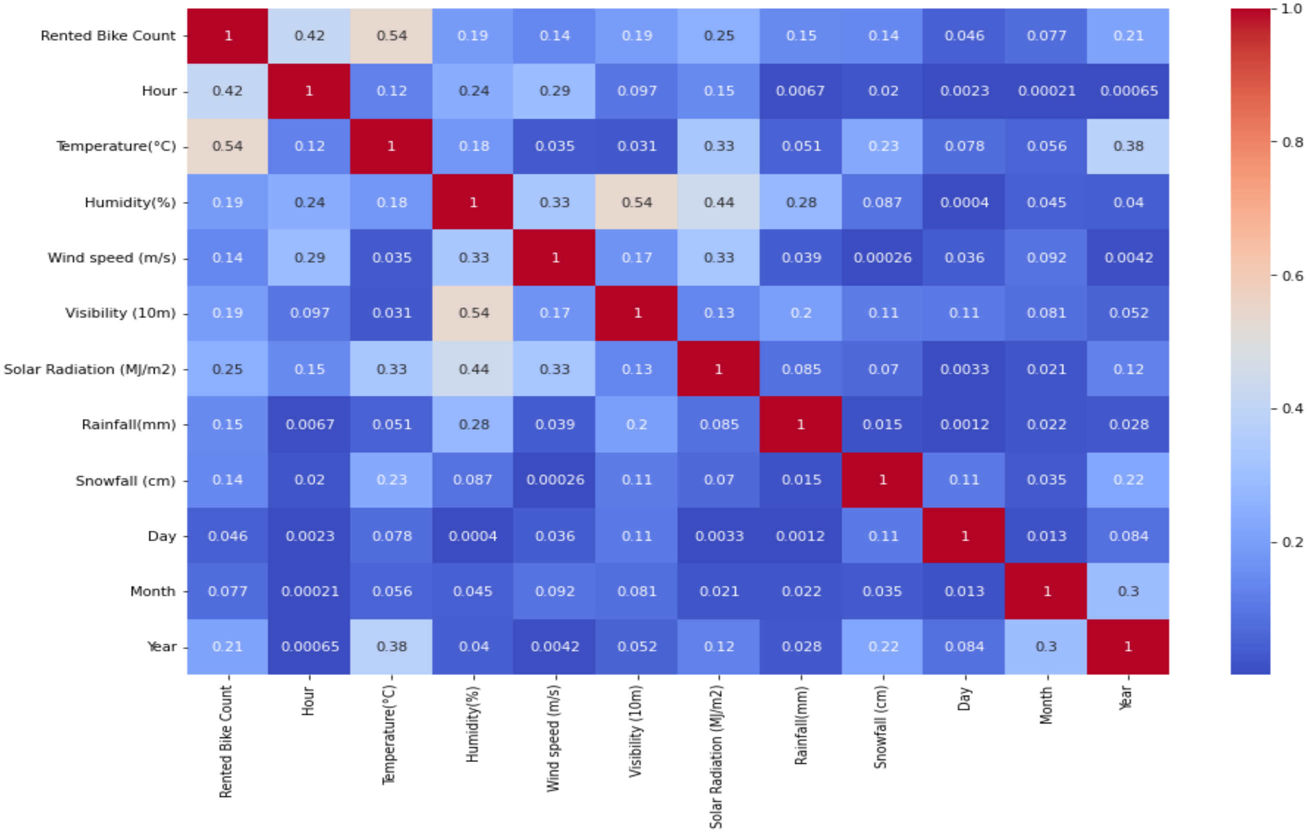


Multicollinearity:



Here Temperature's and Dew Point Temperature's VIF are highly correlated so we will focus on these two to remove collinearity. So we will be focusing on Dew Point Temperature.

After Removing Multicollinearity:



Label Encoding

- Label Encoding is a popular encoding technique for handling categorical variables. In this technique, each label is assigned a unique integer based on alphabetical ordering.

- Here we map the variables like Functioning Day and Holiday in the form of 0 and 1. also convert the seasons column into dummy variables like Spring, Summer and Winter.

- Mapping the Variables

- df['Functioning Day'] = df['Functioning Day'].apply(lambda x : 1 if x == 'Yes' else 0)

- df['Holiday'] = df['Holiday'].map({'Holiday':1,'No Holiday':0})

Feature Engineering & Feature Selection

- It is the process of designing artificial features into an algorithm. These artificial features are then used by that algorithm in order to improve its performance, or in other words for better results.
- In Feature Engineering, we apply lambda function to convert respective columns in the form of 0 and 1.
- Ex. We convert Visibility column in the form of 1 when it is greater than 2000, also for rainfall if the value is greater than 0.148 then it is converted into 1 otherwise 0. Same procedure follows for snowfall and solar radiation

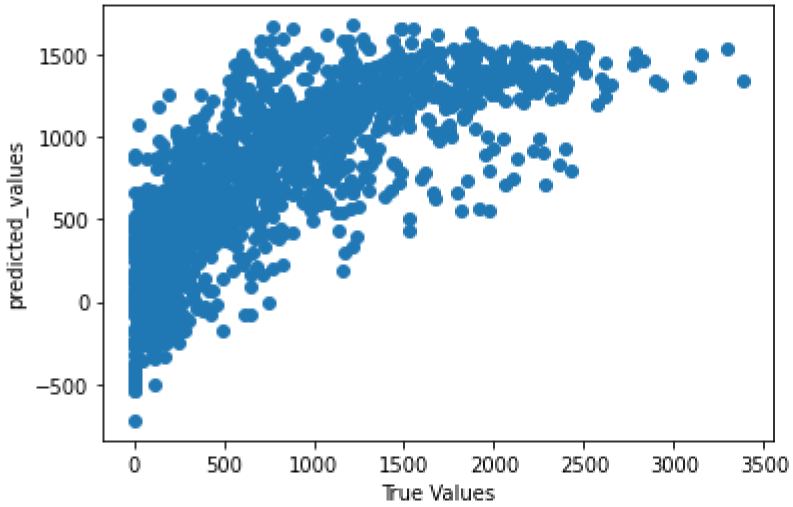
- **Feature Selection :**

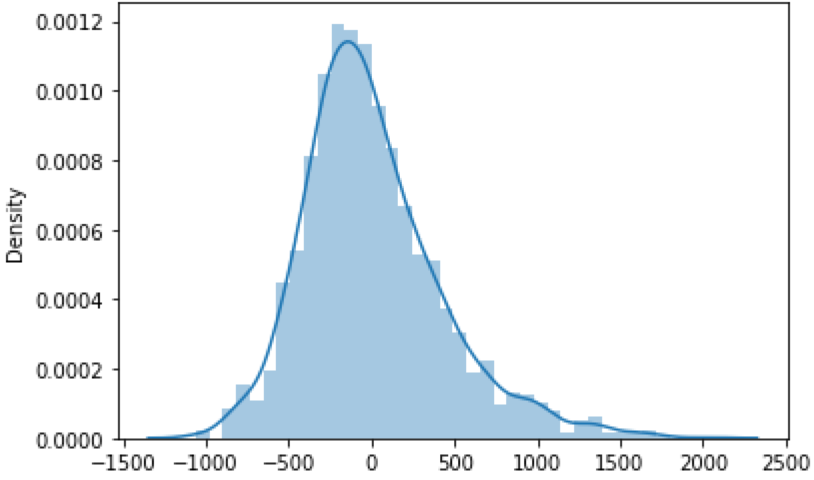
- In Feature selection we remove non-informative or redundant predictors from the model.
- At beginning we have 8760 rows and 14 columns. After label encoding and feature engineering we get 8459 rows and 15 columns

Algorithms

1. Linear Regression :

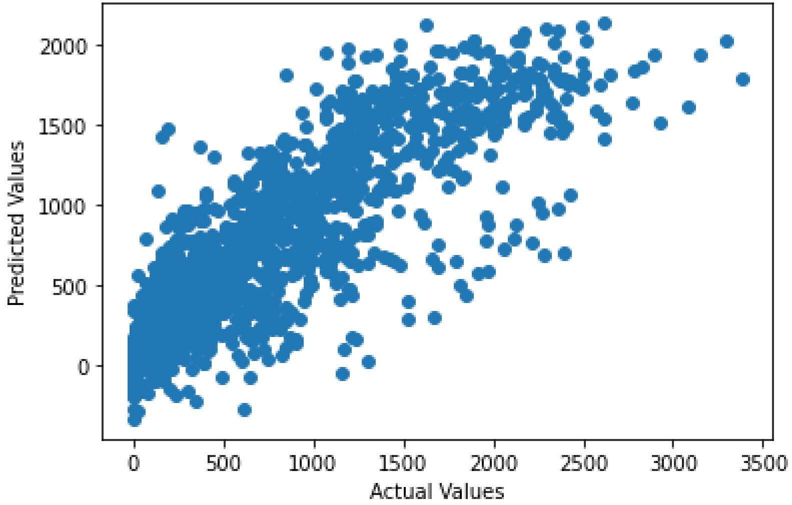
- We have use linear regression model over our data and then we got our prediction model accuracy as in train it was 56.32% and in test it was 57.35%.
- From this we can clearly say that the model is underfitted.
- As we can see from following scatter plot our model is not giving us satisfied result:

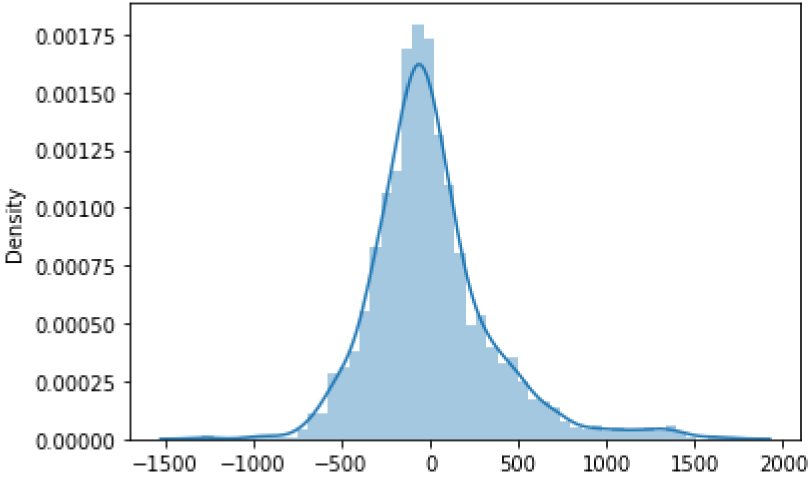




2. Polynomial Linear Regression :

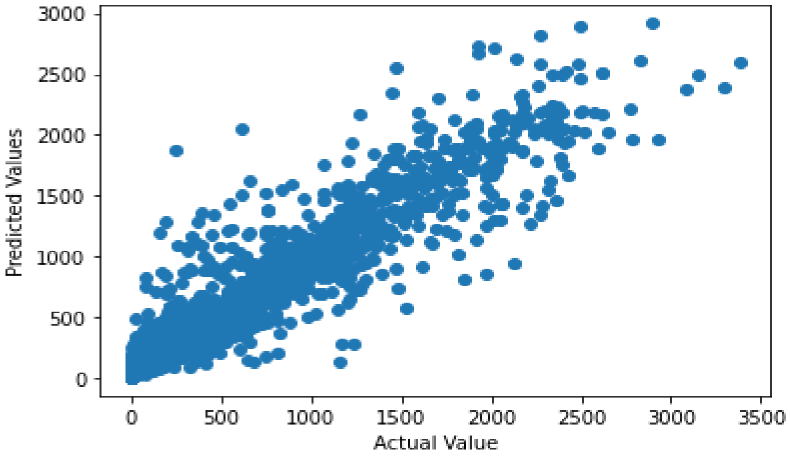
- We received an underfitted model after implementing the linear regression algorithm therefore we're now using the Polynomial Regression algorithm to get an optimal model and satisfied prediction.
- After using the Polynomial Regression algorithm on our data, we were able to achieve better results than the previous algorithm, with 70.89 % on our train data and 71.38 % accuracy on our test data.
- This can be seen in the scatter plot below, our model produces better results as compared to the linear regression algorithm.

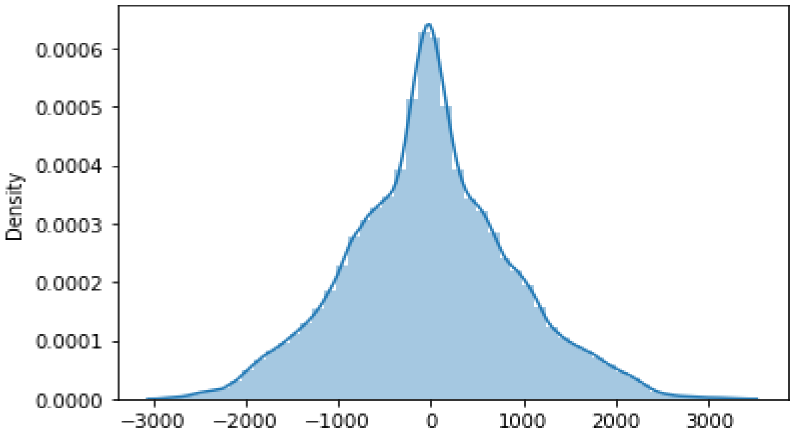




3. Random Forest Regressor:

- We obtained better results after implementing the Polynomial Regression algorithm than we did with previous algorithm, but we still weren't able to get a satisfactory result over our training and testing data, so we used the random forest algorithm to improve our performance.
- After using the random forest algorithm, we were able to improve our model prediction accuracy to 98.16 % on training data and 86.08 % on testing data.
- As it can be observed in the scatter plot below, implementing the random forest algorithm, our model delivers better results as compared to linear regression and Polynomial Regression algorithm.



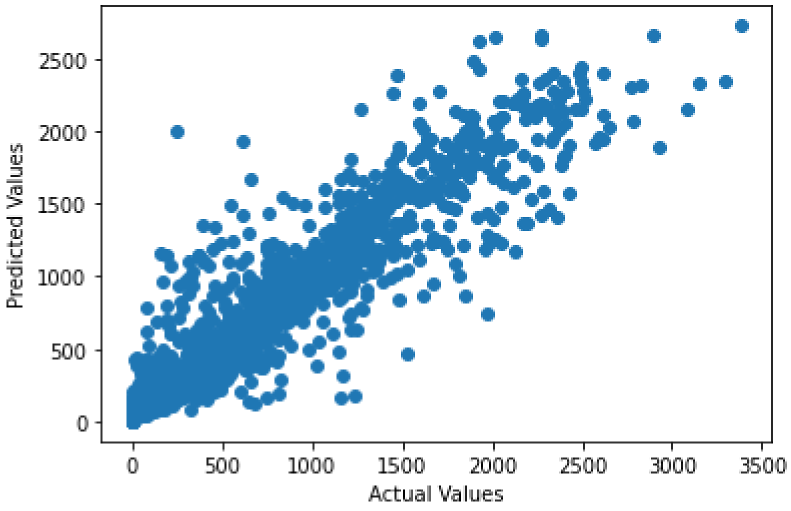


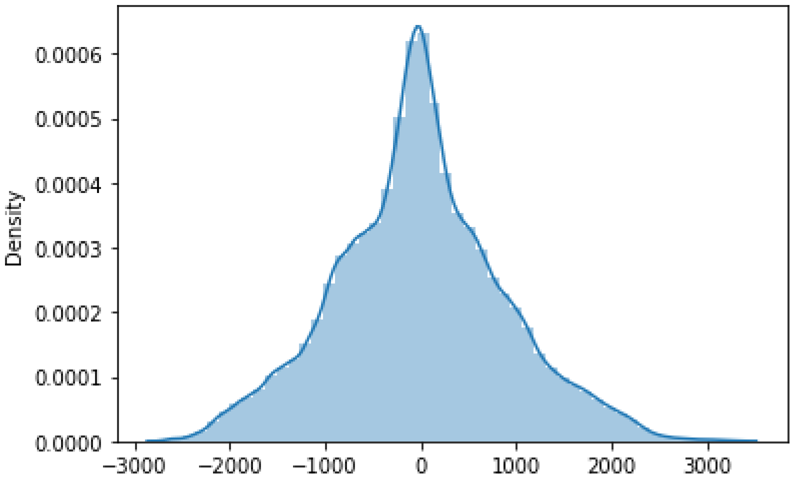
Hyperparameter tuning using Grid Search CV on Random Forest Regressor:

- Using random forest, we were able to achieve 98.16 % accuracy on our training data and 86.08 % accuracy on our testing data, resulting in a gap of 12.08 % between our training and testing accuracy.
- As a result of these numbers, we can say that our model is overfitting.
- The algorithm may overfit if we use this model on unknown data to predict the number of rental bikes required at each hour to maintain a stable supply.
- As a result, we must overcome this challenge, and we are doing so by implementing Hyperparameter tuning using Grid Search CV.
- We were able to get 94.95% accuracy on our training data and 86.25% accuracy on our testing data after implementing Hyperparameter tuning using Grid Search CV on Random Forest Regressor.
- By analyzing this outcome, we can conclude that we now have an optimal model with satisfying results.

Hyperparameter tuning using Grid Search CV on Random Forest Regressor:

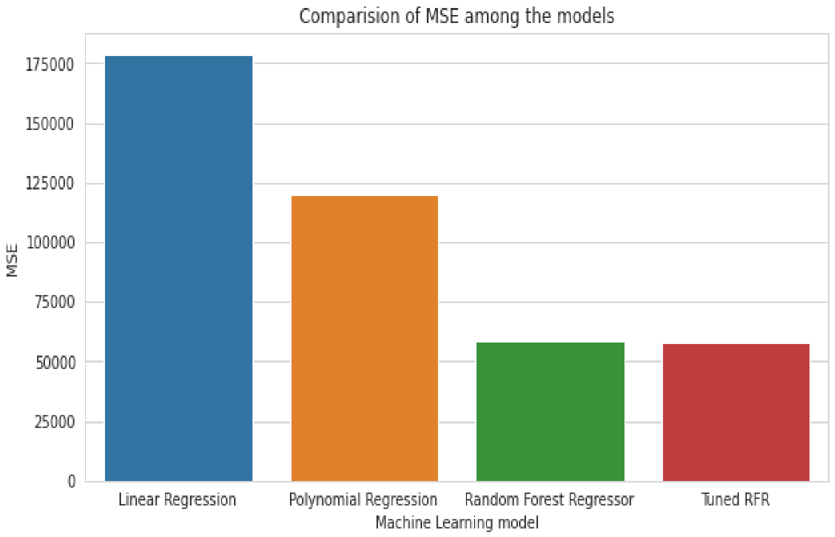
- Implementing Hyperparameter tuning on a Random Forest Regressor using Grid Search CV gives satisfactory results, as shown in the scatter plot below:

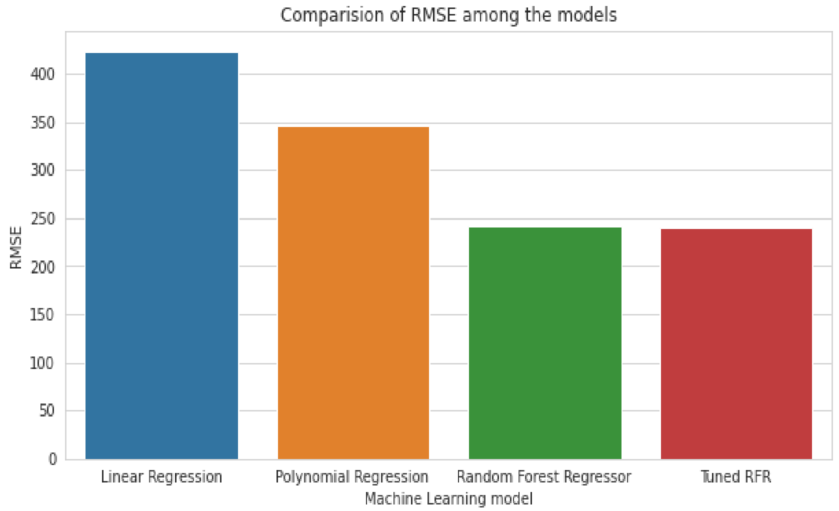




- **Comparing Evaluation Metrics among all the models being used:**

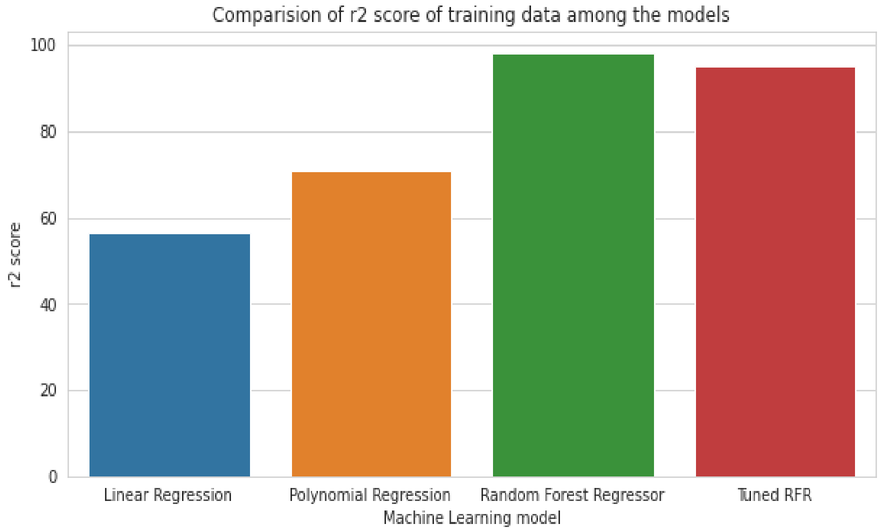
- Comparison of Mean Squared Error and Root Mean Squared Error among all models being used :

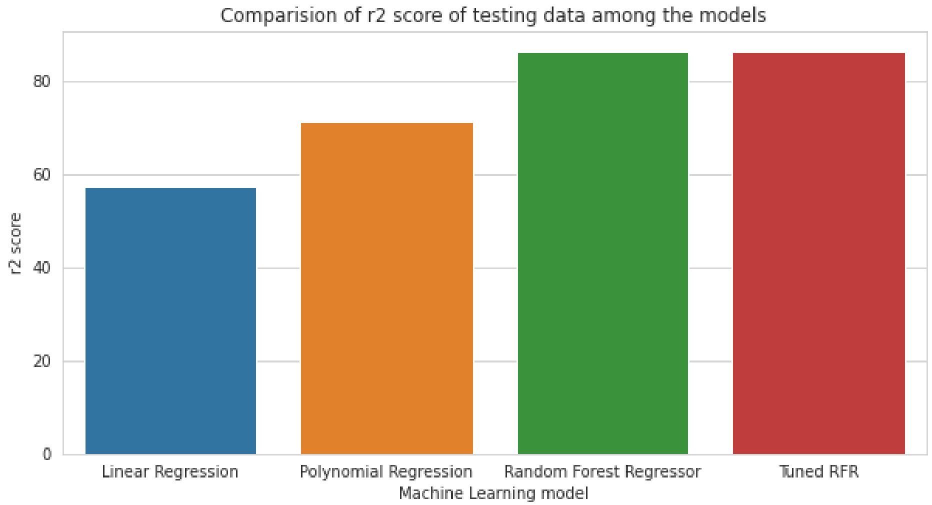




• Comparing Evaluation Metrics among all the models being used:

- Comparing r2 score on training & testing data among all models being used :





Conclusion:

- We have used three different models. Linear Regression, Polynomial Regression, Random Forest Regressor, as per analysis Linear Regression model is underfit.
- After then, polynomial regression produces a model that is slightly overfit. As a result, we tested Bagging Models like the Random Forest Regressor, and our r2 score on both the training and testing datasets increased.
- But RFR was overfitted so we used hyperparameter tuning and we improved the model as a result, Random Forest Regressor has finalized as our final model.
- Rented bike count is dependent on what hour the bike is rented.
- Rented bike count is dependent on what temperature the bike is rented.
- Rented bike count is dependent on Humidity present in atmosphere.
- Rented bike count is also dependent on functioning day or not.
- Rented bike count is also dependent on is it raining outside or not.

Thank You