Capstone Project

Bike Sharing Demand Prediction

By - Shourya Chandra Sai

Project Agenda

- Segment 1: Data Description
- Segment 2: Defining the Problem Statement
- Segment 3: Data Preprocessing
- Segment 4: Exploratory Data Analysis
- Segment 5: Linear Regression
- Segment 6: Regularised Linear Regression
- Segment 7: Random Forest Regressor
- Segment 8: Hyperparameter Tuning
- Segment 9: Conclusion

Segment 1: Data Description

Below is the description of all the columns in the dataset.

- **Date(Year-Month_day)**: This column contains the date on which the observation has occurred.
- **Rented Bike Count**: This column contains the number of bikes that are being rented out each hour.
- **Hour**: This column contains the hour of the day.
- **Temperature**: This column contains the values of temperature.
- **Humidity**: This column contains the values of humidity.
- **Windspeed**: This column contains the values of wind speed.
- **Visibility**: This column contains the values of visibility.
- **Dew point temperature**: This column contains the values of dew point temperature.
- **Solar Radiation**: This column contains the values of solar radiation.
- **Rainfall**: The amount of rainfall.

- **Snowfall**: The amount of snowfall.
- **Seasons**: The various seasons.
- **Holiday**: This column informs us whether that particular day was a holiday or not.
- **Functional day**: This column informs us whether that particular day was a functioning day or a non-functioning day.

Segment 2: Defining the Problem Statement.

Dependent variable(Y): Rented Bike Count

Independent variable(X):

Date = Breaking the column into 3 Separate columns - Day, Month, Year. Treating each column as a categorical variable.

Hour = Numerical Discrete variable.

Temperature(°C) = Continuous Variable.

Humidity(%) = Continuous Variable.

Wind speed (m/s) = Continuous Variable.

Visibility(10m) = Continuous Variable.

Dew point temperature(°C) = Continuous Variable.

Solar Radiation (MJ/m2) = Continuous Variable.

Rainfall(mm) = Continuous Variable.

Snowfall(cm) = Continuous Variable.

Seasons = Categorical Variable of 4 classes(Winter,Spring,Summer,Autumn).

Holiday = Categorical Variable of 2 classes(No Holiday, Holiday).

Functioning Day = Categorical Variable of 2 classes(Yes, No).

Segment 3: Data Preprocessing

- **Dealing with Null and Duplicate values -** There weren't any null or duplicate values that I had to deal with. By removing null and duplicate values the model generalises well with the unseen data.

- **Breaking the date column -** I had broken down the data column in 3 separate columns(year, month, date).

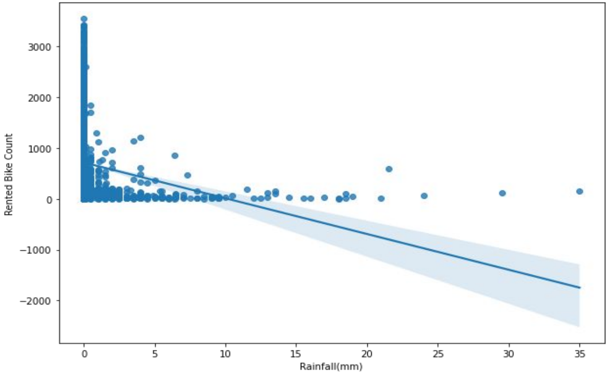
  Year - I had performed binary encoding for year column as there were only 2 classes of values(2017,2018).

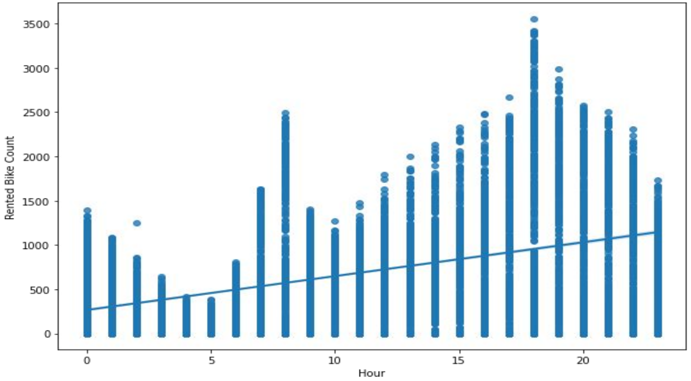
  Month - I had performed one hot encoding on the month column.

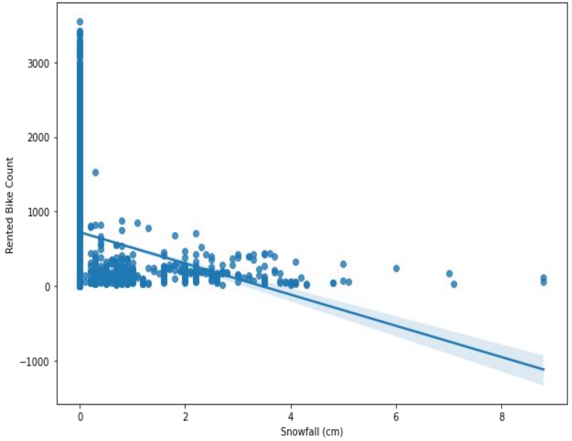
  Date - I had further transformed date column into day column and create a column called Is_weekend. If the day was Saturday or Sunday then I encoded them with 1 and else I encoded with 0.

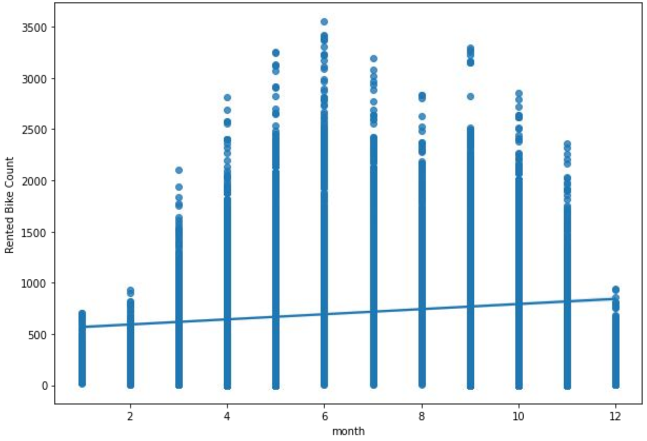
- **Approach used to deal with discrete numeric variable:** Classifying them was a challenge because I had to decide whether to treat these variables the same way as one would treat continuous numeric variables or categorical variables. Using sns.regplot, I got an understanding of how strong the linear relationship is with the dependent variable. If the linear relationship was not strong them I had converted them to categorical variables.
  Using this approach I had converted Hour, Month, Rainfall, Snowfall into categorical.

From the sns reg plots, we can clearly see that the variables rainfall, hour, snowfall, month do not have a strong linear relationship with the dependent variable so I will be converting these into categorical variables.









Segment 4: Exploratory Data Analysis.

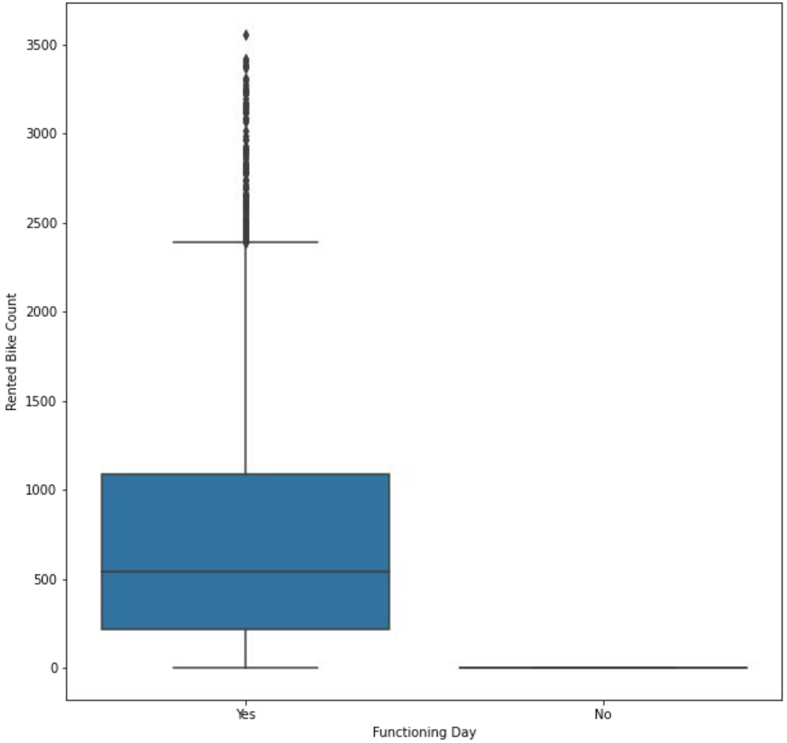
Objective: Exploratory data analysis is performed to understand how are the independent variable related to the dependent variable.

This is a box plot showing us the relationship between Functioning day(X variable) & Rented Bike Count(Y variable).

From the box plot, we can conclude that the count of rented bikes is really high when the day was a functioning day.

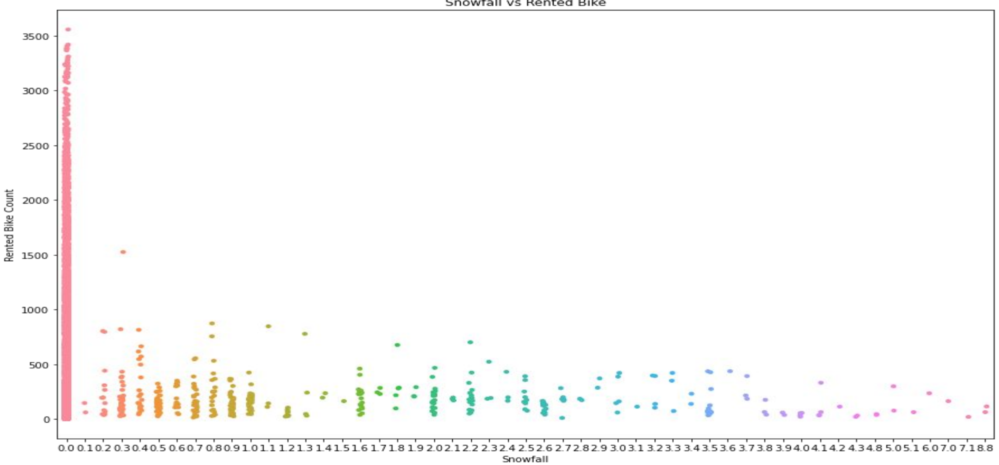
When the day was not a functioning day there are hardly any bikes.

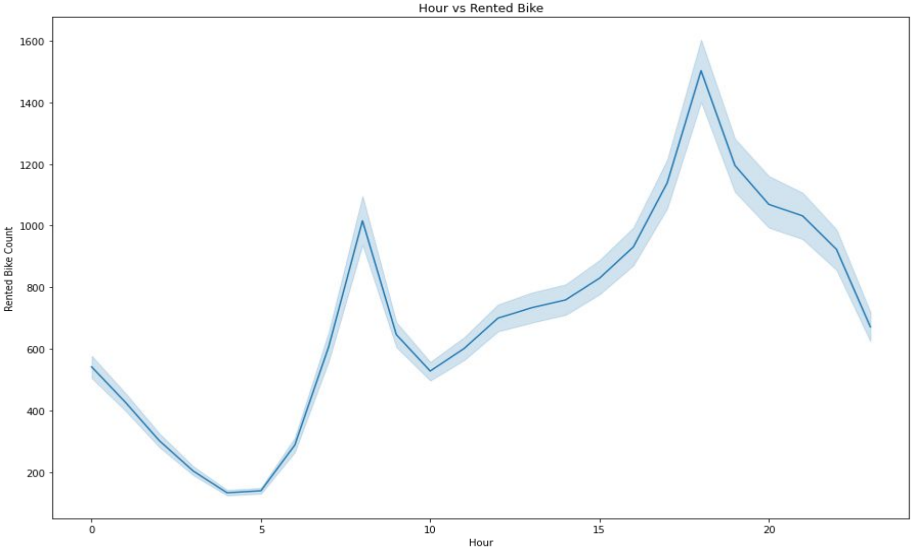
This makes sense right, When it is a functioning day a lot of people go to work and because of this the Rented Bike Count is high.



This is a strip plot that is showing the relationship between Snowfall and Rented Bike Count.

When it is not snowing then, a lot of people rent out bikes.



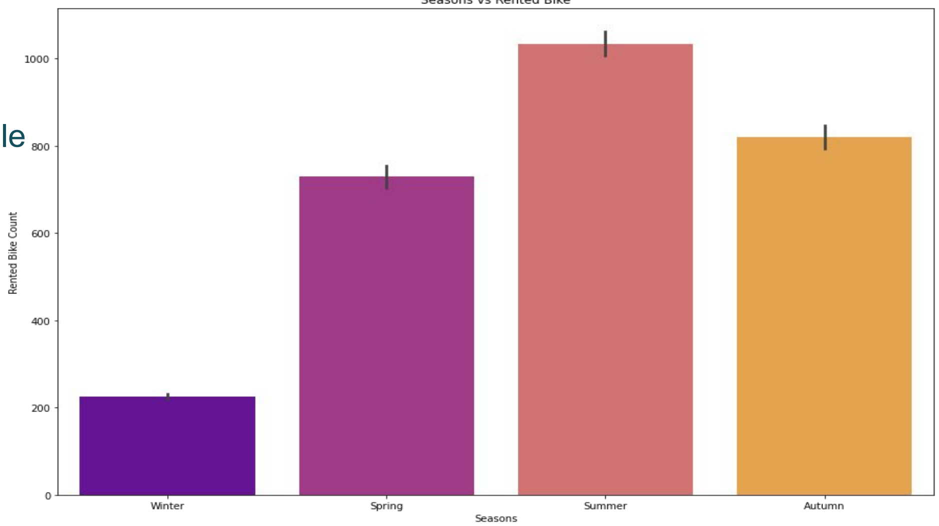


This is a line plot that show the relationship between the Hour of the day(X variable) and Rented Bike Count(Y variable).

As you can see during evening times the number of bikes being rented is the maximum.

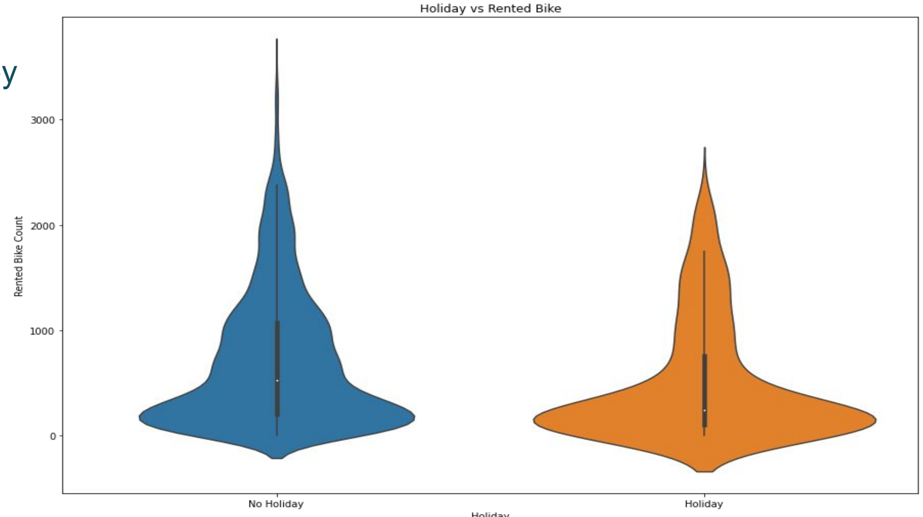
This is a bar plot showing the relationship between Seasons variable and Rented bike count variable.

A lot of people rent bikes during the summer season. This makes total sense as summer season is conducive to riding bikes.



The violin plots provide information as the box plots but the way they represent data is a bit different.

A lot of people rent bikes when it is a working day. They might be using these bike to commute to work.



Segment 5: Linear Regression

Linear regression tries to fit the best fit line along the data points. This best fit line is determined by minimizing the cost function of line using gradient descent.

Objective: In this segment I had made sure that all the assumptions of linear regression have been met, In order to get the best out of Linear Regression. And also I had implemented the Model.

Assumptions of Linear Regression:

- The distributions of the variables should be normal.
- The independent variables should be linearly related to dependent variable.
- No multicollinearity between the independent variables.
- The distribution of the residuals should be normal and the mean of the residuals must be approximated to 0.
- There should be homoscedasticity i.e Variance should be equal along the best fit line.

Making the distributions normal.

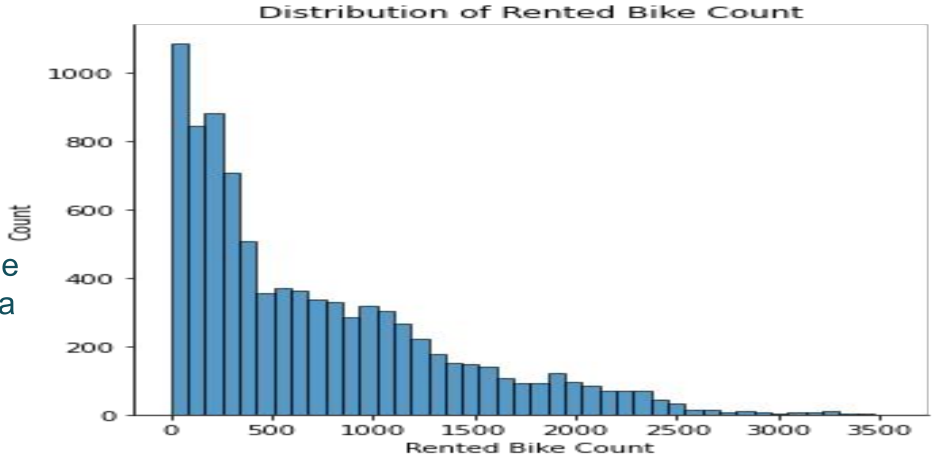
Making the dependent variable distribution normal:

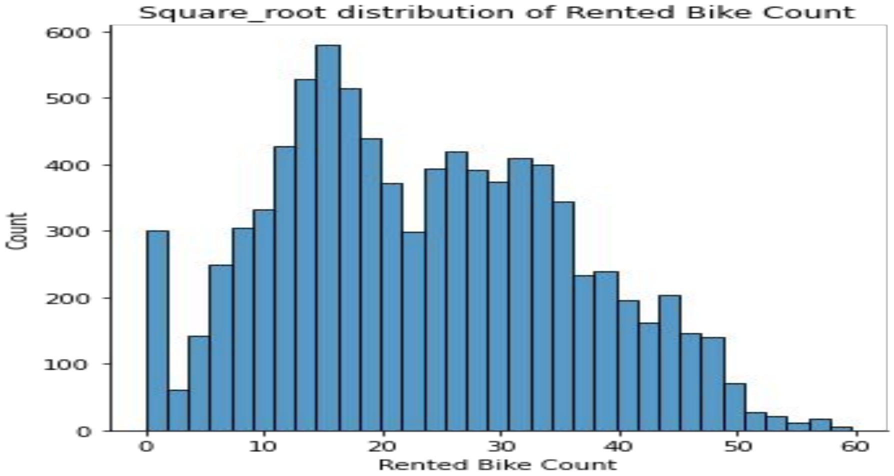
I have used the square root transformation to make the distributions of the dependent variable normal.

From the top right histograms you can see that the Rented Bike Count histogram is right skewed. After applying the square root transformation the distribution is looking more like a normal one(bottom right histogram).

Making the independent variable distribution normal:

I have converted the independent variables distribution into a normal one by removing outliers.

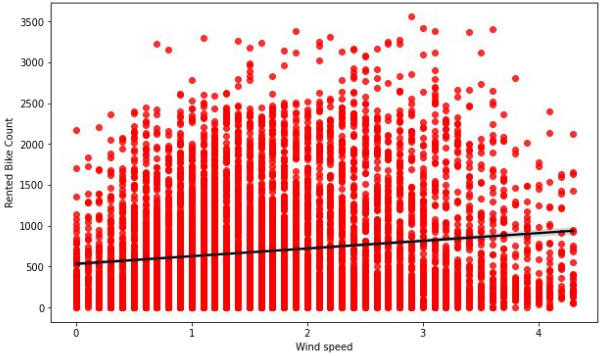


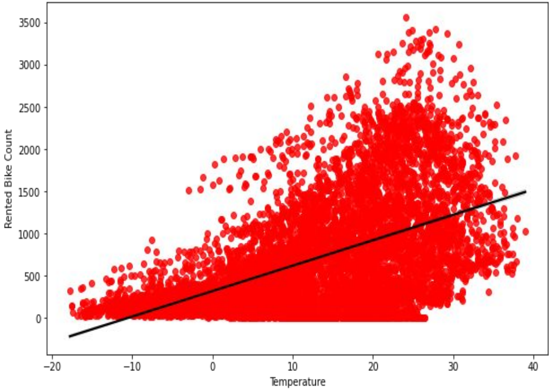


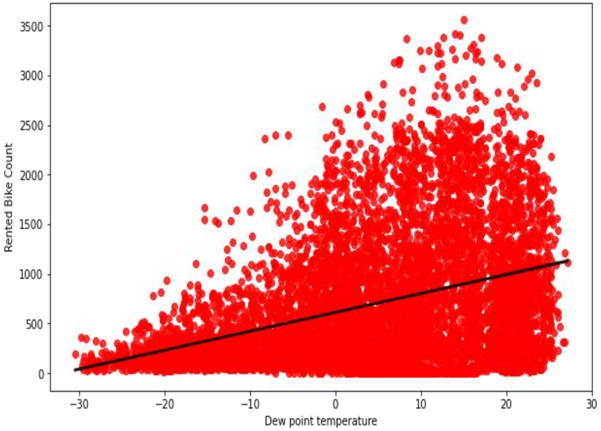
Relation between independent variables and dependent variable.

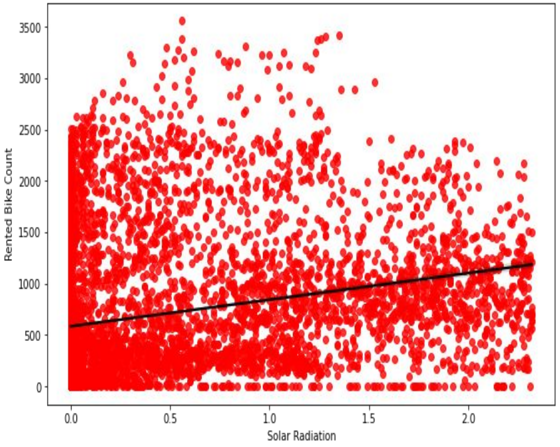
From the regression plots made by the seaborn library we can see that the relationship between dependent variable and independent variables (Wind Speed, Temperature, Dew Point, Solar Radiation) have a linear relationship.

I have not included Visibility and Humidity regression plots because of the lack of space left.









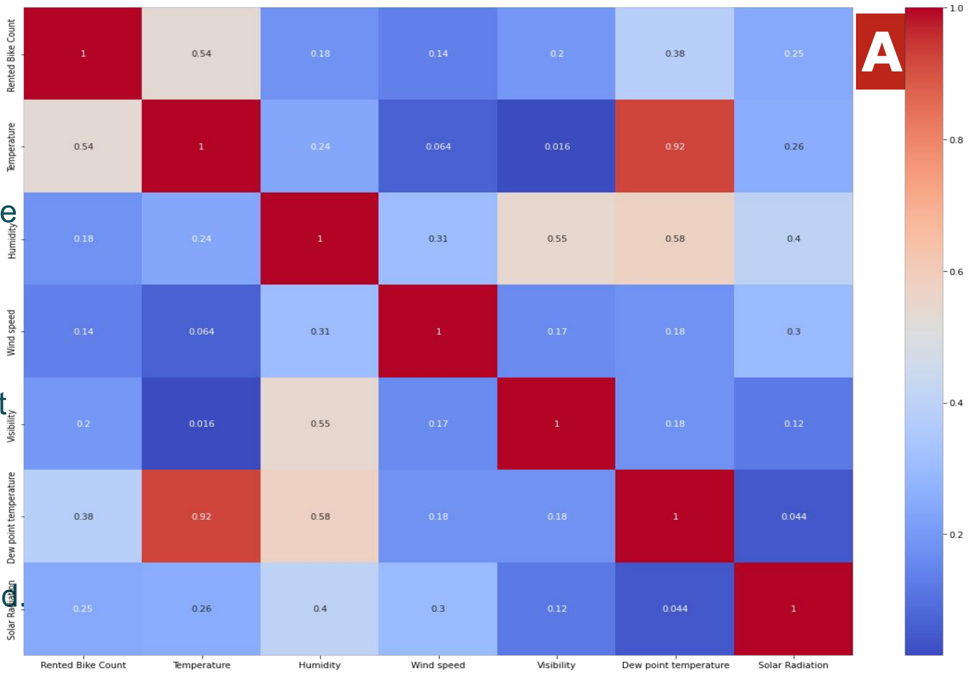
Correlation and Multicollinearity.

Correlation: It tells us how strong the relation is between the variables. The main aim here is to drop those variables that have a low correlation with the dependent variable. Also drop those variables whose correlation with the variables (excluding the dependent variable) is high.

Multicollinearity: Linear regression assumes that there is no correlation between the independent variables. One way to deduct multicollinearity is through VIF(variance inflation factor).

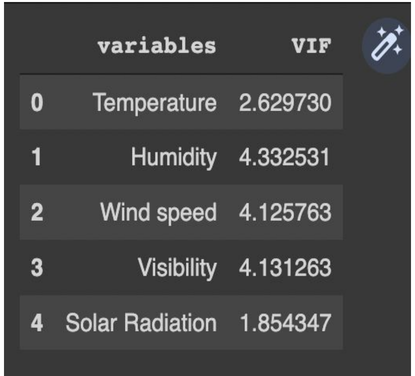
As a rule of thumb if VIF of a particular variable is below 2 we can say that the variable in subject is not related to other independent variables. If the VIF is between 2 and 5 they are moderately related. and if the VIF is above 5 we say that they are highly correlated.

As there are not variables whose VIF is above 5, we don't need to drop any of the variables.



	Rented Bike Count	Temperature	Humidity	Wind speed	Visibility	Dew point temperature	Solar Radiation
Rented Bike Count	1	0.54	0.18	0.14	0.2	0.38	0.25
Temperature	0.54	1	0.24	0.064	0.016	0.92	0.26
Humidity	0.18	0.24	1	0.31	0.55	0.58	0.4
Wind speed	0.14	0.064	0.31	1	0.17	0.18	0.3
Visibility	0.2	0.016	0.55	0.17	1	0.18	0.12
Dew point temperature	0.38	0.92	0.58	0.18	0.18	1	0.044
Solar Radiation	0.25	0.26	0.4	0.3	0.12	0.044	1

1.0
0.8
0.6
0.4
0.2



	variables	VIF
0	Temperature	2.629730
1	Humidity	4.332531
2	Wind speed	4.125763
3	Visibility	4.131263
4	Solar Radiation	1.854347

Working with categorical variables.

Hour variable = I have converted all those values that are equal to 18(6Pm) as peak and the rest as non-peak. From the EDA(line chart) we got to know that at 18 the demand for bikes is really high.

Rainfall variable = If the value is not equal to 0.0, then I have converted into Yes(it is raining) else into No.

Snowfall variable = If the value is not equal to 0.0, then I have converted into Yes(it is snowing) else into No.

Objective: Since the models can't take strings as input we have to encode categorical variables.

Categorical Variables are classified into 3 types:

Dichotomous variables = These are those categorical variables which have only **2 classes** of categories. We can perform binary encoding for these kind of variables.

Ordinal Variables = These are those categorical variables which have **more than 2 classes** of categories and they signify **some order**. We can perform label encoding.

Nominal Variables = These are those categorical variables which have **more than 2 classes** of categories and they do not signify **any order**. We can perform One hot encoding.

There are 5 dichotomous variables[**Holiday, Functioning day, year, Is_raining, Is_snowing**]. I will be performing binary encoding for these.

There is 2 nominal variables[**Seasons, Month**]. I will be performing one hot encoding.

Fitting a Model, Checking Homoscedasticity.

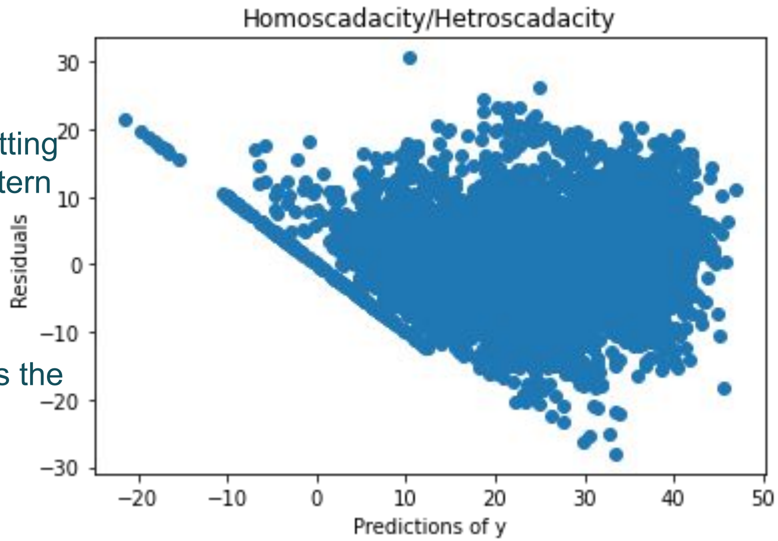
After fitting a linear model to the polished data, from the coef's of the fitted model I got to know that Month_12 variable has a lot of negative influence on the model prediction while season_spring has a lot of positive influence on the model prediction.

Model Performance: There a lot of model evaluation metrics. All the metrics do the same thing i.e, measure the residuals but the way the residuals are measured is different. The most commonly used performance metrics are MSE(The lower the value the better the model), MAE(The lower the values the better the model), R2(The higher the value better the model), Adjusted R2(The higher the value the better the model). I will be only talking about the R2 metric because all the metric speak the information.

Training R2 score: 0.666
Testing R2 score: 0.671
From this we can come to the conclusion that our model is not overfitting.

After cross validation our models average performance was 0.70

Homoscedasticity/Heteroscedasticity: We can check for scedasticity by plotting a scatter graph between the y predictions and the residuals. If there is any pattern in the scatter plot then we say it is heteroscedasticity. As there is no pattern here we can conclude it is homoscedasticity.



Distribution and mean of residuals: The distribution of the residuals is also normal and mean is also close to zero. This tells us that the fitted line indeed is the best fit line.

Segment 6: Regularised Linear Regression.

Regularized Linear Regression is implemented when the model is overfitting. It reduces the complexity of the model by penalising the coefficients. There are 3 regularised linear regressions:

1. Lasso
2. Ridge
3. Elastic Net

I have implemented all of them to see if my model performance improves.Personally, I don't think my model performance improves because my model is not overfitting.

Model Performance Conclusion: At arbitrarily picked hyper-parameters Ridge > Lasso > Elastic Net.

In the later segment we will tune the hyper-parameters alpha and L1 ratio to get the best model performance.

Segment 7: Random Forest Regressor.

Random Forest Regressor is used to predict the dependent variable which is continuous in nature. These are non-parametric in nature. This means they don't have any assumptions unlike linear regression.

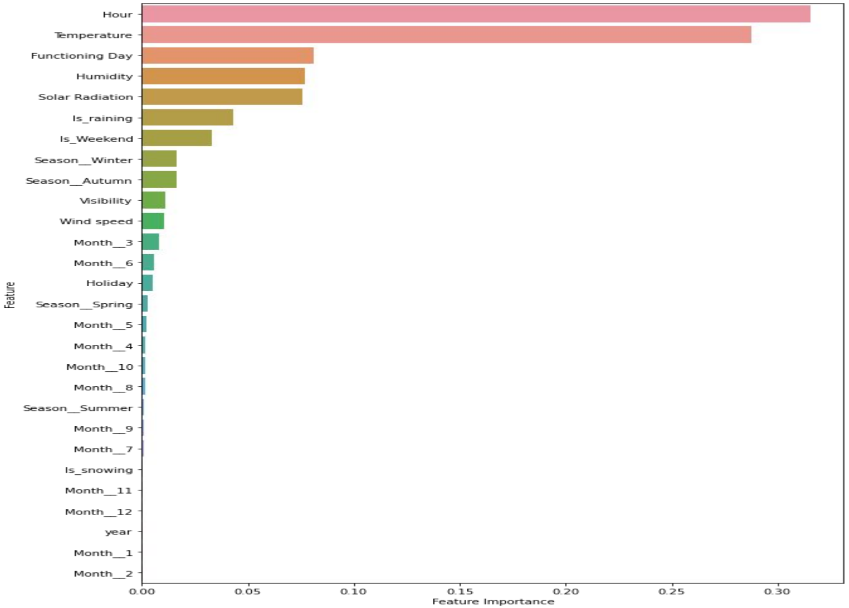
They use a technique called **bootstrap aggregation.**

They build multiple decision trees from randomly sampled rows and columns of the dataset. This random sampling of the data is called **bootstrap**. After building multiple decision trees from the dataset they calculate the average of the values(in case of a regression problem) that have been predicted by all the decision trees. This process of calculating the average prediction by combining all the individual prediction values is called **aggregation**.

Model Performance = Training R2 Score: 0.99, Testing R2 Score:0.94

The model is performing way better than any of the linear and regularised Regression model. But it has been overfit.

Feature Importances: From the horizontal bar chart we can see that hour is the most important feature followed by temperature.



Segment 8: Hyper-Parameter Tuning.

Hyper-parameter Tuning:

Hyperparameter tuning is the process of figuring out the best hyperparameters for the model.Hyperparameter tuning can be done in one of 3 following ways.

* Grid Search - It calculates the scoring for all the possible combinations of hyperparameters and returns that combination of hyperparameters that has the highest scoring.
* Random Search - It calculates the scoring for random combinations of hyperparameters and return that combination that has the highest scoring.
* Bayesian Optimization - Bayesian Optimization is an approach that uses Bayes Theorem to direct the search for hyperparameters in order to find the hyperparameters that return the maximum scoring.

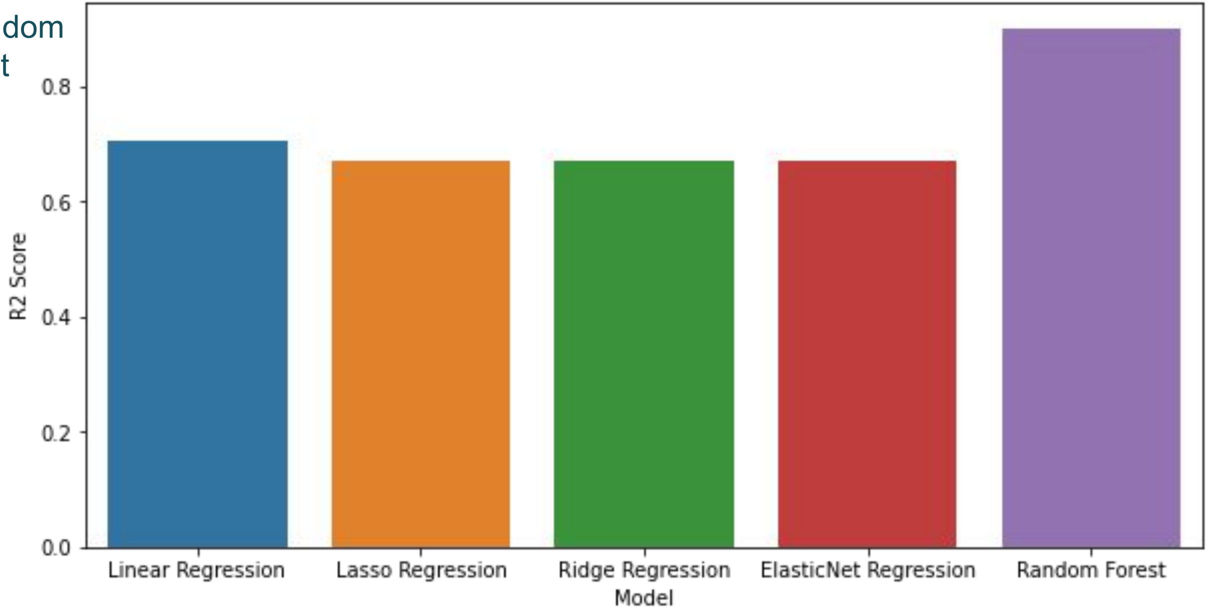
I will be using Grid search to tune hyperparameters when implementing regularised linear regression because implementing regularised linear regression at all possible combinations of hyperparameters is not computational intensive because regularised linear regression is a simple model.

I will be using Random Search CV when tuning hyperparameters for random forest because implementing random forest is computational complex than regularised linear regression.

Optimal Hyper-Parameters model Performance.

From the bar chart we can explicitly see that random forest outperformed all the other models by quiet some margin.

We can improve the performance of the random forest even more by giving more set of hyper-parameters to GridSearchcv or RandomizedSearchcv.



Segment 9: Conclusion.

- After implementing 5 different models on the dataset I came to the conclusion that the data is complex for the linear and regularised regression to understand the patterns. One must use more complex models like random forest, XGBoost to get better results. Hence I have used random forest regressor.

- I had a hunch from the beginning that the linear regression and regularised regression might not relatively perform well. So I have engineered a lot of features like 'Is_raining','Is_snowing' etc... to not loose on any data that has been provided in the dataset.

- Among the linear regression models, the best performing model is linear regression. This makes sense because the linear regression model that we have built was not overfitting in the first place.