A

PROJECT REPORT

ON

"Bike Rental Count"

DESIGN AND DEVELOPED

BY

HARSH.J.JOSHI

UNDER THE GUIDANCE OF

PROF. Muquayyar Ahmed

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Now a day's Bike rental services are expanding with the multiplier rate. The ease of using the services and flexibility gives their customer a great experience with competitive prices. By doing this we can help the customer with the bike rental count.

1.1 Problem Evaluation

The objective of this Case is to Predication of bike rental count on daily based on the environmental and seasonal settings. A bike rental system is a service in which users can rent/use bikes available for shared use on a short term basis. Our goal is to develop and optimize Machine Learning models that effectively predict the bike rental count on a daily basis.

So let's first understand the given problem, we have the data from day.csv from which we can perform our analysis and predict the rental count.

1.2 Understanding Data

Understanding the data is one of the most important and crucial task to perform. It is the first step for every Data Science and Data Analytic projects because by doing so we can easily analyse and give answers to the questions like what is data, Is data proper or not, Is data enough for performing any analysis methodologies.

Here in our case our company has provided a data set with following features, we need to go through each and every variable of it to understand data and for

better functioning. We have been provided with one datasets day.csv. As we see in the data set we have 16 variables and 731 observations. In that 13 variables are independent and 3 dependent variables.

Let's have better understanding of the data so that it will help us to properly understand the data more adequately and come to more accurate decision.

The 16 variables are:-

1. **instant**: Record index
2. **dteday**: Date:
3. **Season** (1:springer, 2:summer, 3:fall, 4:winter)
4. **yr**: Year (0: 2011, 1:2012)
5. **mnth**: Month (1 to 12)
6. **hr**: Hour (0 to 23)
7. **holiday**: weather day is holiday or not (extracted fromHoliday Schedule)
8. **weekday**: Day of the week workingday: If day is neither weekend nor holiday is 1, otherwise is 0.
9. **weathersit**: (extracted from Freemeteo)

   1: Clear, Few clouds, Partly cloudy, Partly cloudy

   2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist

   3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds

   4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
10. **temp**: Normalized temperature in Celsius.
11. **atemp**: Normalized feeling temperature in Celsius.
12. **hum**: Normalized humidity. The values are divided to 100 (max)

13. **windspeed**: Normalized wind speed. The values are divided to 67 (max)
14. **casual**: count of casual users
15. **registered**: count of registered users
16. **cnt**: count of total rental bikes including both casual and registered

Here casual, registered and count are our dependent variables

COUNT = CASUAL+REGISTERED

Remaining all are independent variables.

Below mentioned is a list of all the variable names with their meanings:

Variables	Meaning
instant	Record Index
dteday	Date Time
Season	4 season
yr	year
mnth	month
hr	hour
holiday	Yes/No
weekday	working day or not
weathersit	weather situation
temp	Normal Temperature
atemp	Feeling Temperature
hum	humidity
windspeed	wind speed
casual	casual users
registered	Daily/Registered users
cnt	Casual + registered

Fig 1.1 Variable table

1.3 Previous Assumption

As we are talking about how independent variables will be effect on the target variable. So there will be the multiple assumptions. It also helps to better understand the business process very efficiently so that we can properly come to a conclusion or to get to a final outcome very precisely.

- Rent amount is highly depend on trip distance.
- Rent amount is depending on how much time it will take to travel from one place to another place. Because, in the traffic it may be take more time. So, indirectly it will effect on fare amount.
- Pickup time is also impact on rental charges like suppose journey may be start in night time so night charges will be impact on fare amount.
- Rental amount and count is also depended on the season and weather condition is it rainy or sunny weather.
- Rental count is also depended on the weekdays and also weather there is a holiday on weekday or on weekends.

CHAPTER 1

Data Pre-Processing

Once we have gathered the data and understood the data then we can directly start analysing the data. This step involves cleaning the data, dropping unwanted attributes, conversion of datatypes to machine-understandable format and when our unstructured data come up into structure format then finally we will split the training data into train and validation sets. Here we already removed few unstructured data from our training data set in above chapter but still few impurities are there we will figure it out using below methods.

2.1 Exploratory Data Analysis

When we required to build a predictive model, we require to look and manipulate the data before we start modelling which includes multiple preprocessing steps such as exploring the data, cleaning the data as well as visualizing the data through graph and plots, all these steps is combined under one shed which is Exploratory Data Analysis which includes following steps:

- Data exploration and Cleaning
- Missing values treatment
- Outlier Analysis
- Feature Selection & Features Scaling
- Visualization

In the given project, we have data set of bike rental count, since from 2011. The data which we have is unstructured in nature so, here we need to spend more time for data understanding, data cleaning, and data visualization to figure out new features that are better predictors of bike rental count.

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
0	1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446	331	654	985
1	2	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539	131	670	801
2	3	2011-01-03	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309	120	1229	1349
3	4	2011-01-04	1	0	1	0	2	1	1	0.200000	0.212122	0.590435	0.160296	108	1454	1562
4	5	2011-01-05	1	0	1	0	3	1	1	0.226957	0.229270	0.436957	0.186900	82	1518	1600

Fig 2.1 day.csv in Python

It is very necessary to first know the data types of the given variables so that we can convert them to the type we want the variable to be for the further analysis. So after the data is perfectly loaded next step is to know the data types of the given variables.

```
instant         int64
dteday         object
season          int64
yr              int64
mnth            int64
holiday         int64
weekday         int64
workingday      int64
weathersit      int64
temp          float64
atemp         float64
hum           float64
windspeed     float64
casual          int64
registered      int64
cnt             int64
dtype: object
```

Fig 2.2 Data Types of data

2.2 Missing Value Analysis

Missing value analysis plays a vital role in data preparing. There are many reasons to occur missing values. In statistics while calculating missing values, if it is more than 30% we just drop the particular attribute because it does not carry much information to predict our target variables.

```
instant       0
dteday        0
season        0
yr            0
mnth          0
holiday       0
weekday       0
workingday    0
weathersit    0
temp          0
atemp         0
hum           0
windspeed     0
casual        0
registered    0
cnt           0
dtype: int64
```

Fig 2.3 Missing Values

As we can see that from fig 2.3 that there is no missing values in the data so that there is no need to do any of the missing value analysis on the given data as it is appropriate and valid for further data exploration and analysis.

2.3 Outlier Analysis

Outlier is the observation which is inconsistent related with all data set.

The values of Outliers are the accurate but it is far away from the set of actual values and it heavily impact on the mean so that we consider it as an outlier.

Here we have used box plot method to detect outliers as shown below Fig 2.4

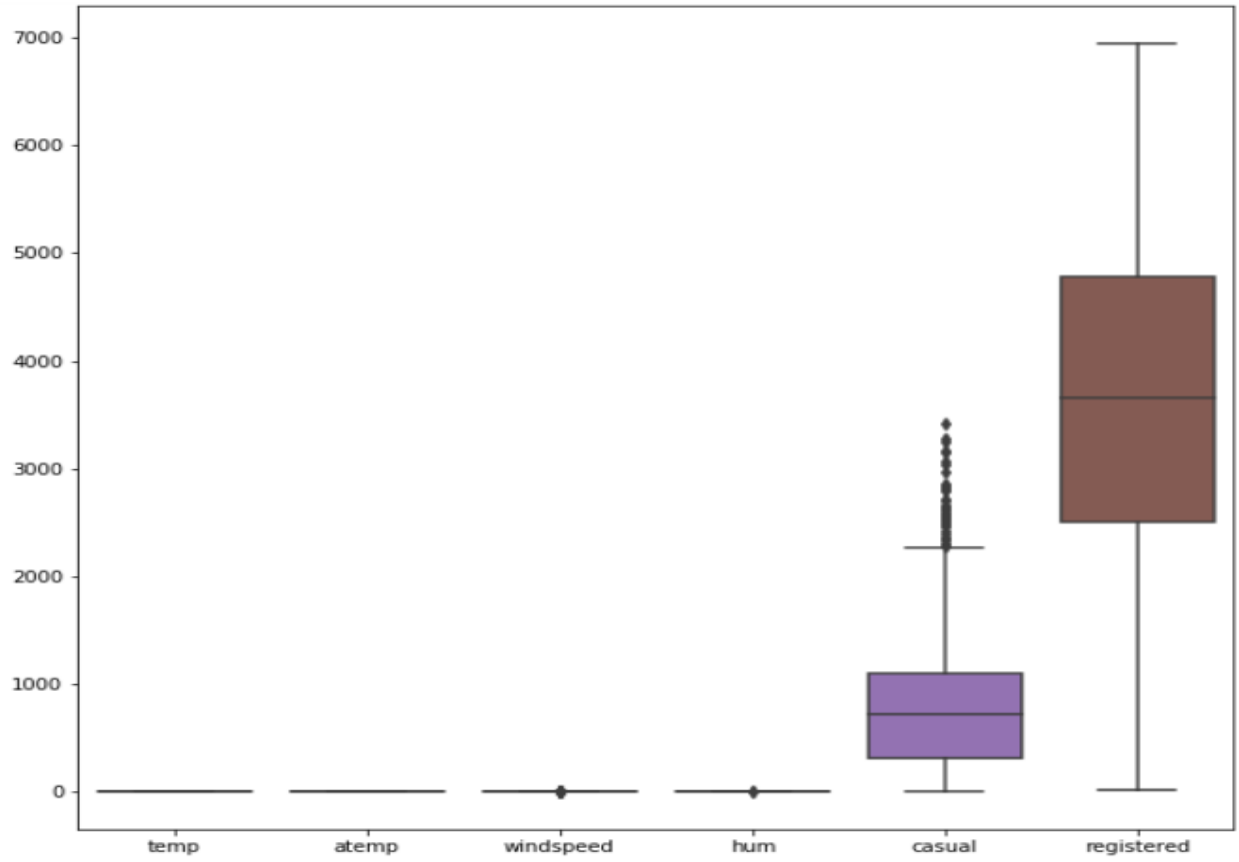

Fig 2.4 Outlier analysis

As we can see from the given fig 2.4 there are outliers present in the 3 data columns wind speed, hum and casual.

So we first need to know how many outliers are present in each of the columns so to evaluate properly. In this case we would ignore the casual column because it is dependent variable and it is not our predicted variable so we may consider the independent variable wind speed and humidity.

So we get to know that humidity has two outliers and wind speed has 13 outliers rows from the below fig 2.5

```
instant        0
dteday         0
season         0
yr             0
mnth           0
holiday        0
weekday        0
workingday     0
weathersit     0
temp           0
atemp          0
hum            2
windspeed     13
casual         0
registered     0
cnt            0
dtype: int64
```

Fig 2.5 Outliers in the data

We can delete those data so that they will not cause any bias to our data set and predict models.

2.4 Feature Selection

As we know, while developing the model if we consider the independent variables which carries the same information to explain the target variables it will create the problem of multi- collinearity. So to avoid our model from the multi-collinearity problem we need to apply Feature Selection or dimensional reduction on the top of our data set. It helps us to sort out the variables which are highly correlated with each other. In our case we applied correlation analysis for numeric variables and ANOVA for the categorical variables.

Below fig 2.6 shows the correlation of all the attributes with each other.

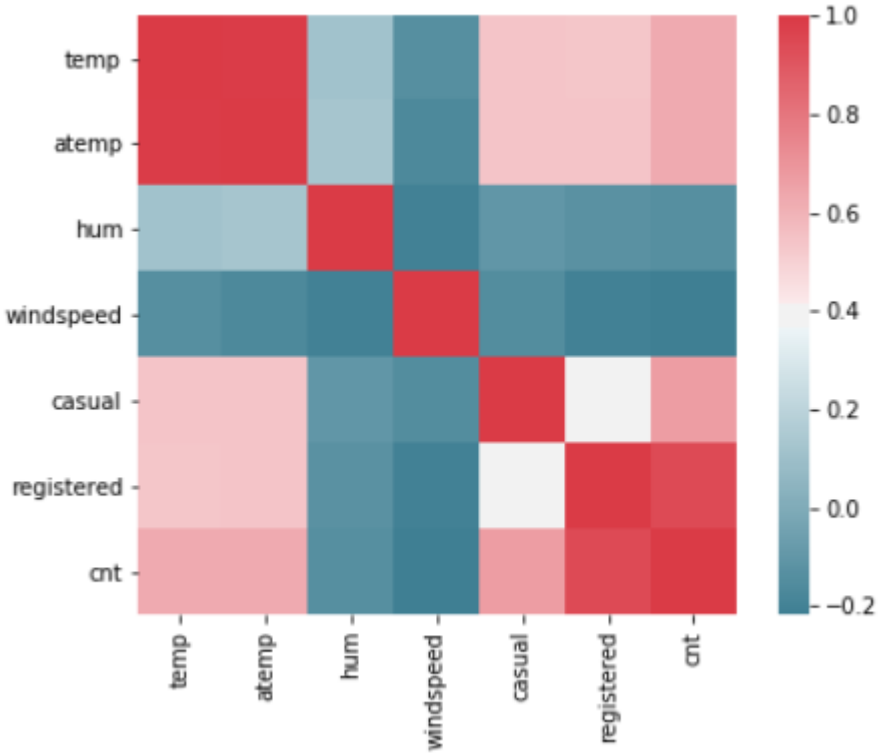

Fig 2.6 Feature Selection

From the above figure we can clearly see that the attributes temp, atemp and attributes registered and count are also closely related to each other.

2.5 Feature Scaling

As we know before passing the data to machine learning algorithm our data should be structure in format. To arrange our data into the desire structure feature scaling technique comes into the picture. We have used VIF to check multicollinearity and chi-square test. To avoid the further biasness in data and it will help to predict our model more effectively and efficiently.

Below given fig 2.7 shows the result of the result of VIF (Variance Influence Factor).

```
const          54.847289
temp           63.442490
atemp          64.309759
hum             1.179328
windspeed       1.154450
casual          1.502061
registered      1.561168
dtype: float64
```

Fig 2.7 Result of VIF

We have also used chi-square test for the categorical variables present in the data. So that we can know which data is contributing or related to the data.

	season	yr	mnth	holiday	weekday	workingday	weathersit
season	-	0.999	0.0	0.641	1.0	0.946	0.013
yr	0.999	-	1.0	0.995	1.0	0.956	0.183
mnth	0.0	1.0	-	0.571	1.0	0.993	0.01
holiday	0.641	0.995	0.571	-	0.0	0.0	0.599
weekday	1.0	1.0	1.0	0.0	-	0.0	0.249
workingday	0.946	0.956	0.993	0.0	0.0	-	0.294
weathersit	0.013	0.183	0.01	0.599	0.249	0.294	-

Fig 2.8 Result of Chi-square test

Observations:

- From heatmap and VIF, we remove variables atemp because it is highly correlated with temp.
- From chi-square test, we remove weekday, holiday variables because they don't contribute much to the independent variables,
- We remove causal and registered variables because that's what we need to predict.
- We remove instant and dteday variables because they are not useful in generating model.

After removing the unwanted attributes we get the clean data-set that we can further use for analysis. The fig 2.9 shows the cleaned dataset

	season	yr	mnth	workingday	weathersit	temp	hum	windspeed	cnt
0	1	0	1	0	2	0.344167	0.805833	0.160446	985
1	1	0	1	0	2	0.363478	0.696087	0.248539	801
2	1	0	1	1	1	0.196364	0.437273	0.248309	1349
3	1	0	1	1	1	0.200000	0.590435	0.160296	1562
4	1	0	1	1	1	0.226957	0.436957	0.186900	1600

Fig 2.9 Cleaned Data

2.6 Data Visualization

On the basis of the cleaned data we can perform different visualization techniques so that we can better understand what actually data is trying to say us. It also helps us to see the hidden aspects of the data that how one attribute of the data is linked to one other.

- Scatter plot between temp and cnt:

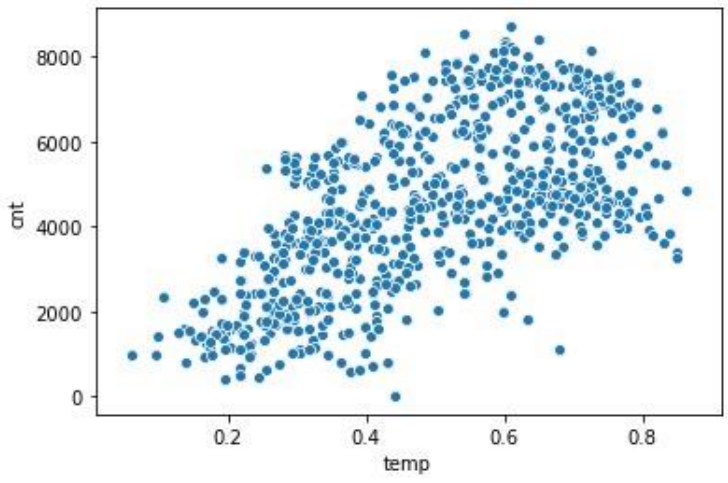

Fig 2.10 temp vs cnt

From the above fig we can see that how temp effects the count on bike rental

- Scatter plot between atemp and cnt:

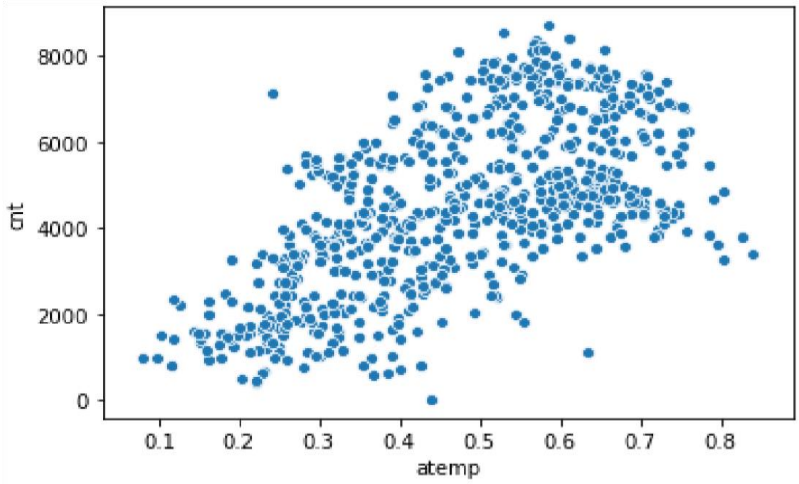

Fig 2.11 atemp vs cnt

From the above fig we can see that how atemp is effecting the count. So from the figure 2.10 and 2.11 we can come to conclusion we can see that attributes temp and atemp both are effecting almost equally on the count.

- Scatter plot between windspeed and cnt:

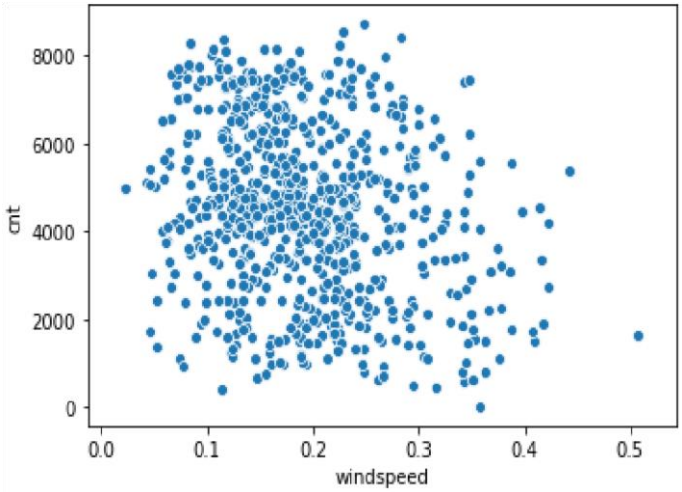

Fig 2.11 windspeed vs cnt

From the above fig we can see how the windspeed attribute is directly affecting the cnt attribute.

- Scatter plot between humidity and cnt

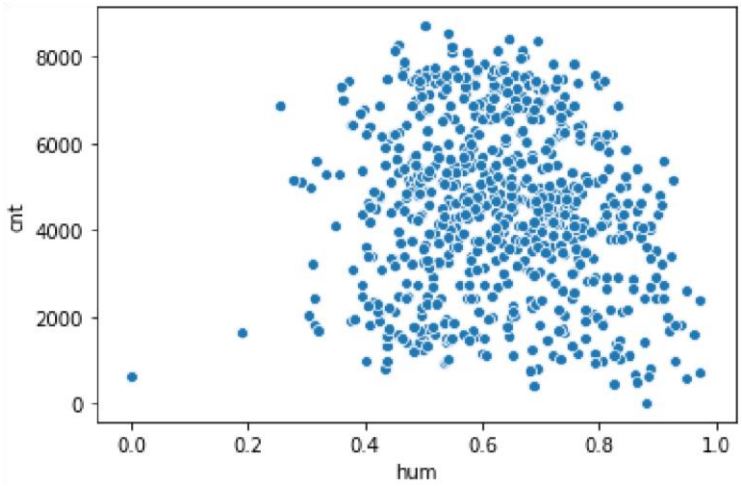

Fig 2.12 humidity vs cnt

From the above fig we can see that how humidity is effecting the bike rental count.

From the figs 2.11 and 2.12 we can come to a conclusion that whenever the wind speed is less count is more and when the humidity is more than count is more.

- Scatter plot between casual vs. count and registered vs. count.

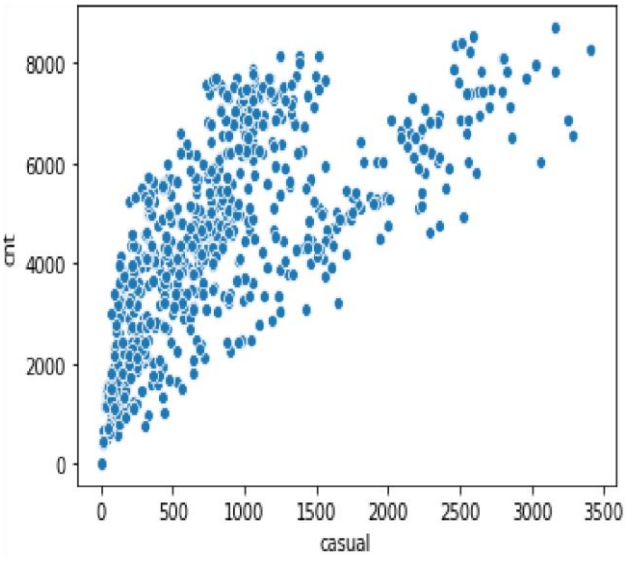

Fig 2.13 casual vs. cnt

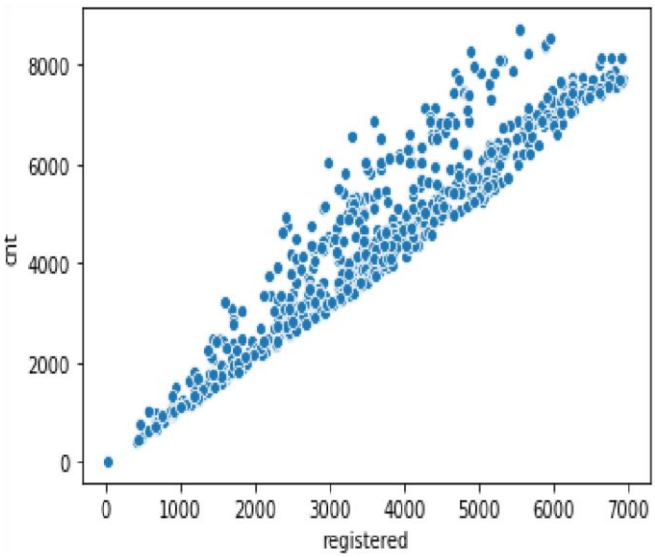

Fig 2.14 registered vs. cnt

There is also some histograms of attribute temperature, humidity and wind speed in fig 2.15, 2.16, 2.17.

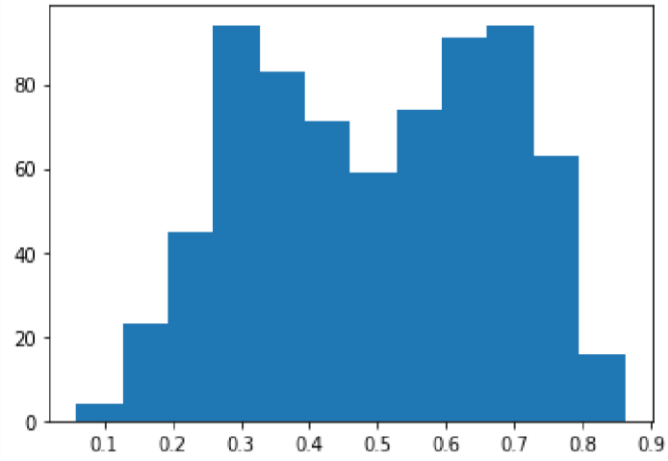

Fig 2.15 temperature

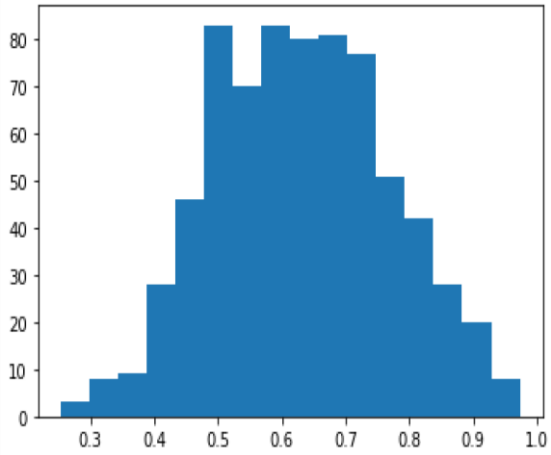

Fig 2.16 humidity

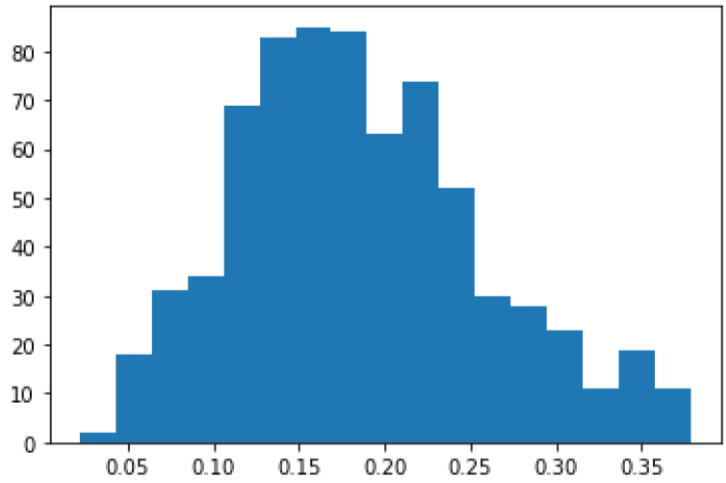

Fig 2.17 windspeed

Seasons widely effect the bike rental count in fig 2.18 we can see that count is more in fall and less in spring

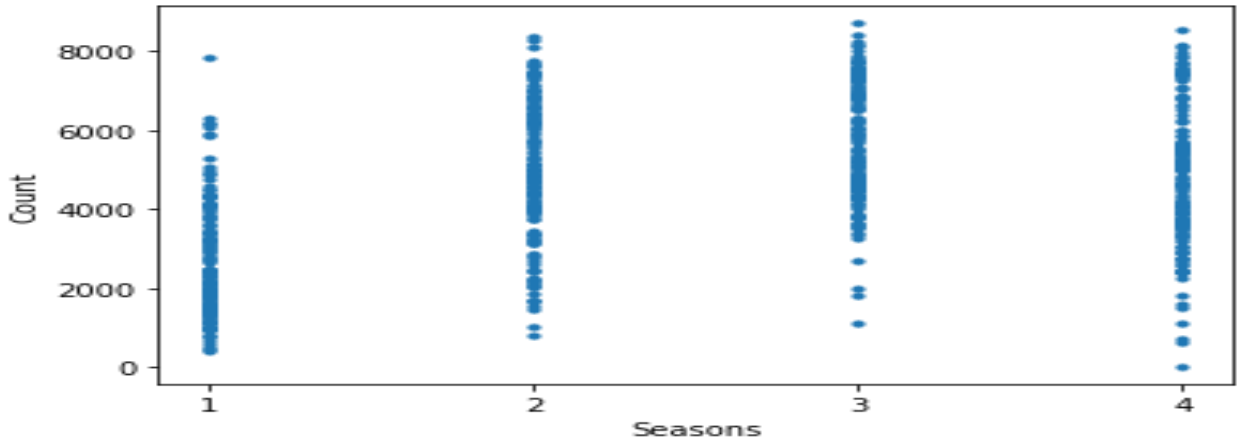

Fig 2.18 season vs count

Based on the season we can see that count of rental of bikes largely effects the count on months as we can see in fig 2.19

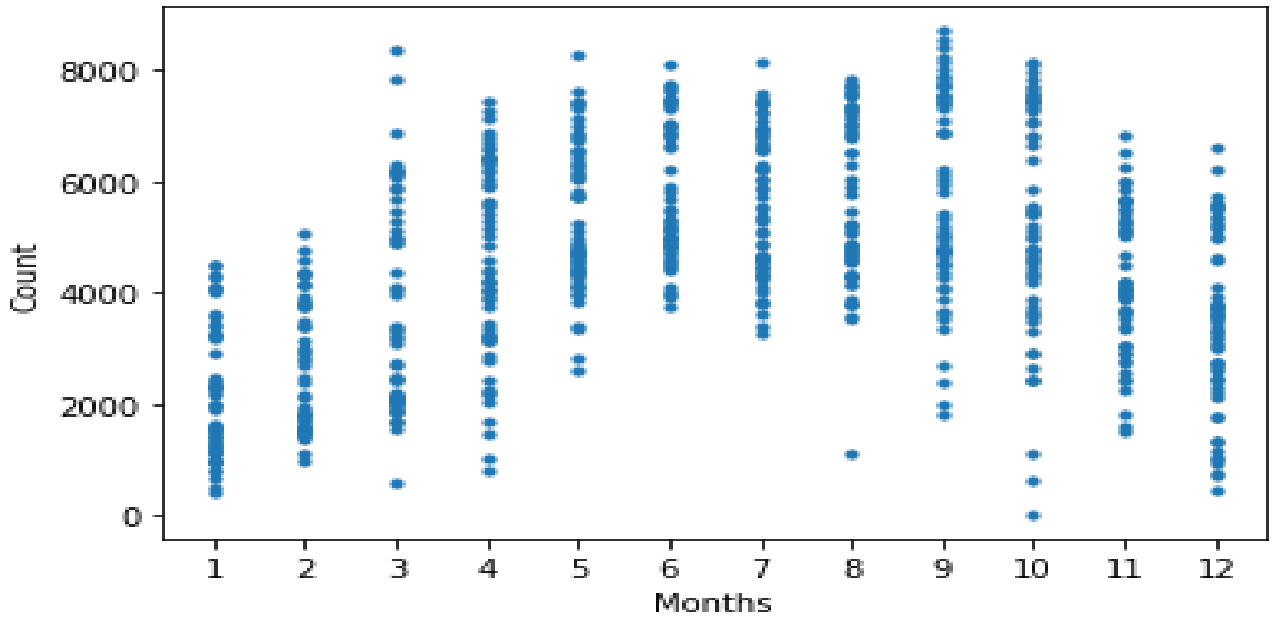

Fig 2.19 months vs. count

From fig 2.19 we can see that the count is less from months Jan to March and it increases from April eventually till October with its peek value in September and again decreasing in months November and December.

We can also see in the fig 2.20 that rental count is increased in the year 2012

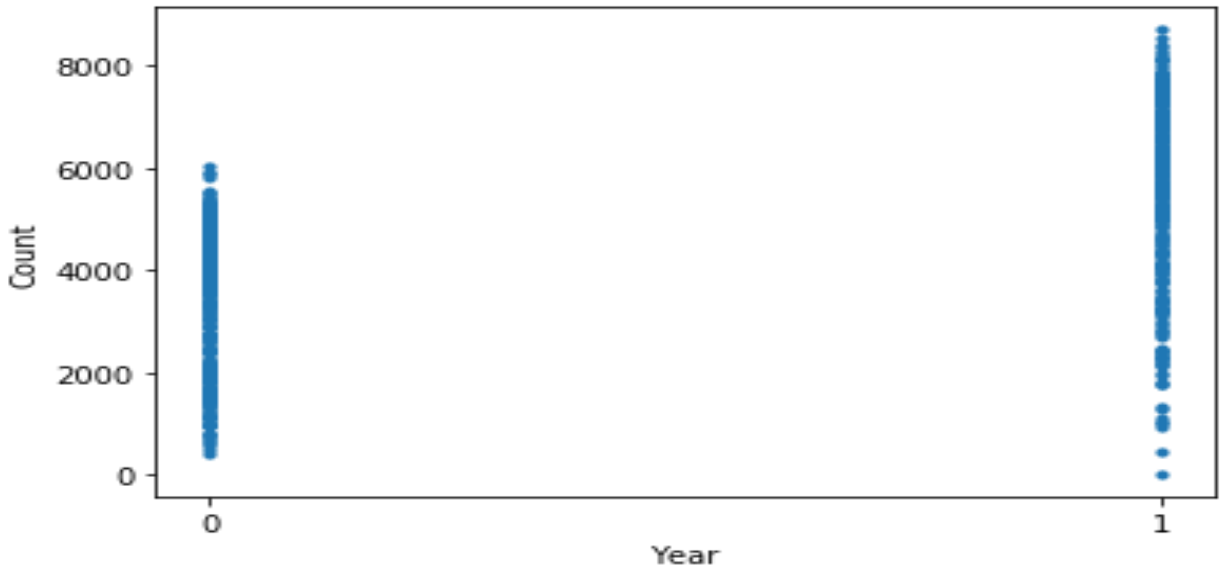

Fig 2.20 Year vs. Count

CHAPTER 3

Modelling

After data cleaning and exploratory data analysis phase, we finally arrived at the model building phase. In this chapter we will applied multiple machine learning algorithm to predict the test case. In cab fare prediction project our target variable i.e. fare amount is numeric (predicting and forecasting type of problem) so that here we are using regression models on structure data to predict test case.

The next step is to differentiate the train data into 2 parts i.e. train and test. The splitting of train data into 2 parts is very important factor to verify the model performance and to understand the problem of over-fitting and under-fitting. Over-fitting is the term where training error is low and testing error is high and under-fitting is the term where both training and testing error is high. Those are the common problem of complex model.

In this analysis, since we are predicting fare amount which is the numeric variable. So, we come to know that, our problem statement is predicting (forecasting) type. So, what we can do is we will apply supervise machine learning algorithms to predict our target variable. As we know our target variable is continuous in nature so, here we will build regression matrix model.

Root Mean Square Error (RMSE) to measures how much error there is between two data sets. In other words, it compares a predicted value and an observed or known value. The RMSE is directly interpretable in terms of measurement units, and so is a better measure of goodness of fit.

So, in our case any model we build should have lower value of an RMSE and higher value of variance i.e. R square.

Following are the models which we will built: –

- Linear Regression
- Decision Tree
- Random Forest
- Gradient Boosting

3.1 Linear Regression

Multiple linear regression is the most common form of linear regression analysis. Multiple regression is an extension of simple linear regression. It is used as a predictive analysis, when we want to predict the value of a variable based on the value of two or more other variables. The variable we want to predict is called the dependent variable (or sometimes, the outcome, target or criterion variable). The main idea is to identify a line that best fits the data. To build any model we have some assumptions to put on data and model. This algorithm is not very flexible, and has a very high bias.

3.2 Decision Tree

A tree has many analogies in real life, and turns out that it has influenced a wide area of machine learning, covering both classification and regression. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions.

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

3.3 Random Forest

Random forest is the collection of multiple decision trees. In Random Forest, output is the average prediction by each of these trees. For this method to work, the baseline models must have a lower bias. The idea behind Random Forest is to build n number of trees to have more accuracy in dataset. Random Forest uses bagging method for predictions. It can handle large no of independent variables without variable deletion and it will give the estimates that what variables are important. To say it in simple words Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.

3.4 Gradient Boosting

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods.

So we have seen the different models that can be applied on the train data and decide which model to use on the data. Below is the table which shows the RMSE and R^2 values, considering this value we choose the model for our data.

Model Name	RMSE		R^2	
	Train	Test	Train	Test
Linear Regression	0.32	0.31	0.56	0.6
Decision tree	0.35	0.35	0.48	0.49
Random Forest	0.1	0.28	0.95	0.68
Gradient Boosting	0.15	0.27	0.89	0.68

Fig 3.1 Modelling Table

Here we can clearly identify that Random Forest and Gradient Boosting values are most suited for the evaluation but to select any one we have to perform tuning on the parameters to evaluate the models more efficiently.

CHAPTER 4

Hyper- Parameters Tuning

Model hyperparameters are set by the data scientist ahead of training and control implementation aspects of the model. The weights learned during training of a linear regression model are parameters while the number of trees in a random forest is a model hyperparameter because this is set by the data scientist. Hyperparameters can be thought of as model settings. These settings need to be tuned for each problem because the best model hyperparameters for one particular dataset will not be the best across all datasets. The process of hyperparameter tuning (also called hyperparameter optimization) means finding the combination of hyperparameter values for a machine learning model that performs the best - as measured on a validation dataset - for a problem.

Here we have used two hyper parameters tuning techniques

➢ Random Search CV

➢ Grid Search CV

1. Random Search CV: This algorithm set up a grid of hyperparameter values and select random combinations to train the model and score. The number of search iterations is set based on time/resources.

2. Grid Search CV: This algorithm set up a grid of hyperparameter values and for each combination, train a model and score on the validation data. In this approach, every single combination of hyperparameters values is tried which can be very inefficient.

We have performed the hyperparameters tuning on both the best fit models Random Forest and Gradient Boosting so that we can choose the best model for our data. For tuning the parameters, we have used Random Search CV and Grid Search CV under which we have given the range of n_estimators, depth and CV folds.

Fig no. 4.1 shows the result that have been achieved after hyperparameters tuning.

Model Name	Parameters	RMSE(Test)	R^2(Test)
Random Search CV	Random Forest	0.28	0.68
	Gradient Boosting	0.32	0.58
Grid Search CV	Random Forest	0.28	0.68
	Gradient Boosting	0.27	0.7

Fig 4.1 Hyperparameter tuning table

Above table shows the results after tuning the parameters of our two best suited models i.e. Random Forest and Gradient Boosting.

CHAPTER 5
Model Selection

In above chapters we applied multiple preprocessing to frame our data into the structural format and different machine learning algorithm to check the performance of model. In this chapter we finalize one of them. Above model help us to calculate the Root Mean Square Error (RMSE) and R-Squared Values. RMSE is the standard deviation of the prediction errors. Residuals are a measure of how far from the regression line data points are, RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. RMSE is an absolute measure of fit. R-squared is a relative measure of fit. R-squared is basically explains the degree to which input variable explain the variation of the output. In simple words R-squared tells how much variance of dependent variable explained by the independent variable. It is a measure if goodness of fit in regression line. Value of R-squared is between 0-1, where 0 means independent variable unable to explain the target variable and 1 means target variable is completely explained by the independent variable. So, Lower values of RMSE and higher value of R-Squared Value indicate better fit of model.

5.1 Model Evaluation

The main concept of looking at what is called residuals or difference between our predictions f(x[I,]) and actual outcomes y[i].

In general, most data scientists use two methods to evaluate the performance of the model:

I. RMSE (Root Mean Square Error): is a frequently used measure of the difference between values predicted by a model and the values actually observed from the environment that is being modelled.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

II. R Squared(R^2): is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression. In other words, we can say it explains as to how much of the variance of the target variable is explained.

III. We have shown both train and test data results, the main reason behind showing both the results is to check whether our data is overfitted or not.

5.2 Model Selection

On the basis RMSE and R Squared results a good model should have least RMSE and max R Squared value. So, from above tables we can see:

• From the observation of all RMSE Value and R-Squared Value we have concluded that,

• Both the models- Gradient Boosting Default and Random Forest perform comparatively well while comparing their RMSE and R-Squared value.

• After this, I chose Random Search CV and Grid Search CV to apply cross validation technique and see changes brought about by that.

• After applying tunings Gradient Boosting model shows best results compared to Random Forest.

• So finally, we can say that Gradient Boosting model is the best method to make prediction for this project with highest explained variance of the target variables and lowest error chances with parameter tuning technique Grid Search CV.

Finally, I used this method to predict the target variable for the test data file shared in the problem statement. Results that I found are attached with my submissions.

Basically this is how we can choose the model for the evaluation of the data that we are using for prediction of the Bike Rental Count based on environmental and seasonal setting.

By using the Gradient Boosting we can predict the Rental Count by using the test data set.

Tools Used

- Jupyter(Python Development)
- R Studio(R code)
- Tableau(Data visualization)
- MS Word(Report Making)
- Excel(Plotting tables and reading Data)

References

1. For Data Cleaning and Model Development - https://edwisor.com/career-data-scientist
2. For other code related queries - https://www.analyticsvidhya.com/blog/2016/03/practical-guide-principal-component-analysis-python/
3. https://stackoverflow.com/

- **Python Code Attached- CAB_Fair_train.ipynb (should be run on jupyter only)**

- **R Studio Code Attached (code must be run on RStudio Only)**

END OF REPORT