BIKE RENTAL PREDICTION

- Rahul kumar

Contents

1. Introduction

1.1 Problem Statement ……..…… 1
1.2 Problem Description…… 1
1.3 Data ……. 2
1.4 Performance Metric ……..... 2

2. Methodology

2.1 Exploratory Data Analysis …….. 3

2.1.1 Data Preparation And Cleaning
2.1.1.1 Missing Value Analysis…….. 3
2.1.1.2 Outlier Analysis ……. 4
2.1.1.3 Feature Scaling …….. 5

2.1.2 Data Visualisation
2.1.2.1 Univariate Analysis ……... 7
2.1.2.2 Bivariate Analysis ……. 13
2.1.2.3 Feature Seletion..... 15

2.2 Modeling

2.2.1 KNN…….……. 17
2.2.2 Ordinary Least Squares……..……... 18
2.2.3 Ridge Regression……... 18
2.2.4 Lasso Regression……... 19
2.2.5 Support Vector Regression ……... 20
2.2.6 Decision Tree ……... 20
2.2.7 Gradient Boosted Decision Tree……..…….. 21
2.2.8 Random Forest ……..……... 21

3. Conclusion

3.1 Model Evaluation ……. 25
3.1.1 Root Mean Square Value……. 25
3.2 Model Selection …… 26
3.2.1 Cross Validation 26
Appendix A - Extra Figures **……** **27**

4. References ……. **30**

Chapter 1

Introduction

1.1 Problem Statement

The objective of this Case is to Predication of bike rental count on daily based on the environmental and seasonal settings.

1.2 Problem Description

Number of attributes:-

instant - Record index.

Dteday - Date

season - Season (1:springer, 2:summer, 3:fall, 4:winter)

yr - Year (0: 2011, 1:2012)

mnth - Month (1 to 12)

hr - Hour (0 to 23)

holiday - weather day is holiday or not (extracted fromHoliday Schedule)

weekday - Day of the week workingday: If day is neither weekend nor holiday is 1, otherwise is 0.

weathersit - (extracted fromFreemeteo) 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 temp: Normalized temperature in Celsius.

The values are derived via (t-t_min)/(t_max-t_min), t_min=-8, t_max=+39 (only in hourly scale) atemp: Normalized feeling temperature in Celsius. The values are derived via (t-t_min)/(t_maxt_min), t_min=-16, t_max=+50 (only in hourly scale)

hum- Normalized humidity. The values are divided to 100 (max)

windspeed- Normalized wind speed. The values are divided to 67 (max)

casual- count of casual users

registered- count of registered users

cnt: count of total rental bikes including both casual and registered

1.3 Data

The data is a Time-Series data but instead we will approach it as Regression Problem.Our task is to build a regression model which will predict the fare of the bike facility based on the customer attributes and all of the information during the journey and general information available to the company about them.

Sample Dataset-

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit
0	1	2011-01-01	1	0	1	0	6	0	2
1	2	2011-01-02	1	0	1	0	0	0	2
2	3	2011-01-03	1	0	1	0	1	1	1
3	4	2011-01-04	1	0	1	0	2	1	1
4	5	2011-01-05	1	0	1	0	3	1	1

temp	atemp	hum	windspeed	casual	registered	cnt
0.344167	0.363625	0.805833	0.160446	331	654	985
0.363478	0.353739	0.696087	0.248539	131	670	801
0.196364	0.189405	0.437273	0.248309	120	1229	1349
0.200000	0.212122	0.590435	0.160296	108	1454	1562
0.226957	0.229270	0.436957	0.186900	82	1518	1600

1.4 Performance Metric

RMSE : Root Mean Square Error (RMSE) is the standard deviation of the residuals (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. Also, Since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors.

So, RMSE becomes more useful when large errors are particularly undesirable. So, Root Mean Square value seems like a perfect choice for our problem at hand.

Chapter 2

Methodology

2.1.1 Data Preparation and Cleaning

2.1.1.1 Missing Value Analysis

One of the most common problems I have faced in Data Cleaning/Exploratory Data Analysis is handling the missing values. Firstly, there is no good way to deal with missing data. But still missing value analysis helps address several concerns caused by incomplete data. If cases with missing values are systematically different from cases without missing values, the results can be misleading. Also, missing data may reduce the precision of calculated statistics because there is less information than originally planned.

```
season        0
year          0
month         0
holiday       0
weekday       0
workingday    0
weather       0
temprature    0
atemp         0
humidity      0
windspeed     0
count         0
dtype: int64
```

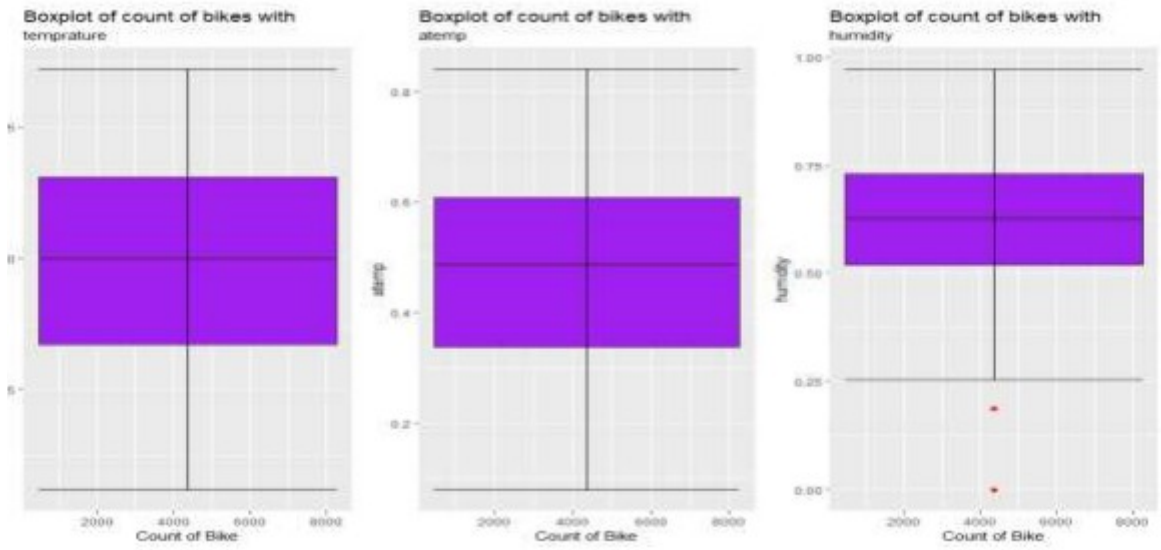
data does not have any missing values.

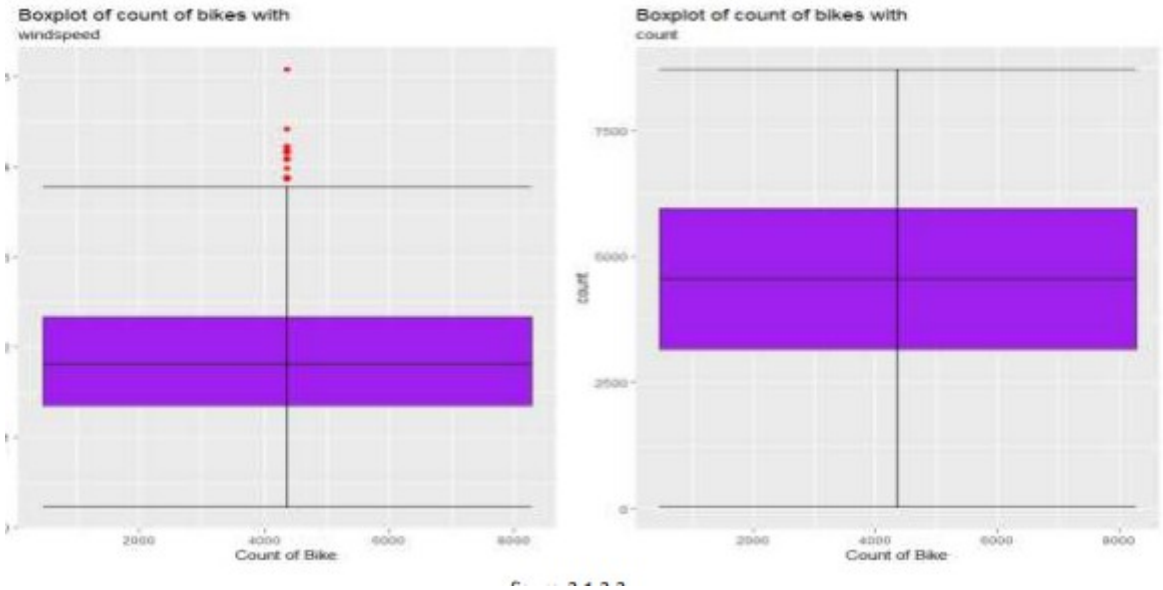
Another concern is that the assumptions behind many statistical procedures are based on complete cases, and missing values can complicate the theory required. So, In our data, there are plenty of missing values available in different variables. So, after computing the percentage of missing data that is available to us in the dataset, it accounts the 0% of the data.

2.1.1.2 Outliers Analysis

In statistics, an outlier is an observation point that is distant from other observations.In layman terms, we can say that an outlier is something which is separated/different from the crowd. Also, Outlier analysis is very important because they affect the mean and median which in turn affects the error (absolute and mean) in any data set. When we plot the error we might get big deviations if outliers are in the data set. In Box plots analysis of individual features, we can clearly observe from these boxplots that, not every feature contains outliers and many of them even have very few outliers.

Also, given the constraint that, we have only 16k data-points and after removing the outliers, the data gets decreased by almost 15%. So, dropping the outliers is probably not the best idea. Instead we will try to visualise and find out the outliers using box plots and will fill them with NA, that means we have created ‘missing values’ in place of outliers within the data.





From the boxplot almost all the variables except "windspeed" and "humidity" does not have outliers.

Now, we can treat these outliers like missing values and impute them using standard imputation techniques. In our case, we use Mean imputation to impute these missing values.

2.1.1.3 Feature Scaling

Normalization rescales the values into a range of [0,1]. This might be useful in some cases where all parameters need to have the same positive scale. However, the outliers from the data set are lost.

$$X_{changed} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

In Linear Algebra, Normalization seems to refer to the dividing of a vector by its length.

	temperature	humidity	windspeed
count	731.000000	731.000000	731.000000
mean	0.495385	0.629354	0.186257
std	0.183051	0.139566	0.071156
min	0.059130	0.254167	0.022392
25%	0.337083	0.522291	0.134950
50%	0.498333	0.627500	0.178802
75%	0.655417	0.730209	0.229786
max	0.861667	0.972500	0.378108

Therefore, the data is already scaled and the data are normalized.

2.2 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is the first step in our data analysis process. We do this by taking a broad look at patterns, trends, outliers, unexpected results and so on in our existing data, using visual and quantitative methods to get a sense of the story this tells. To start with this process, we will first have a look at univariate analysis like plotting Box plot and whiskers for individual features, Histogram plots, Bar plots and Kernel Density Estimation for the same for the same.

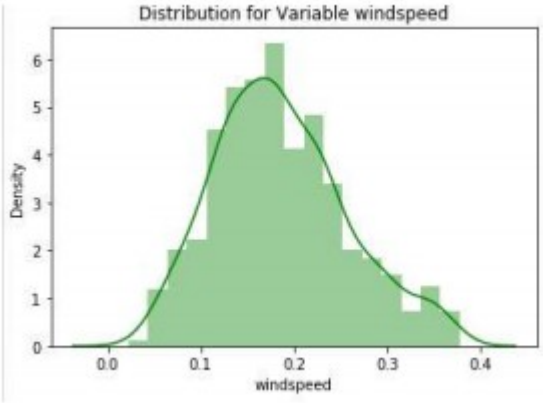
2.2.1 Data Visualisation

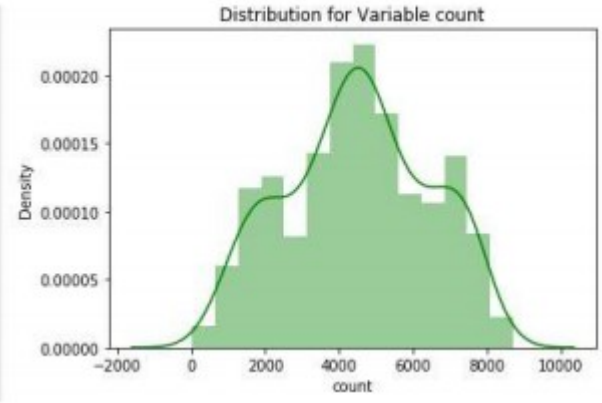
Data visualisation helps us to get better insights of the data. By visualising data, we can identify areas that need attention or improvement and also clarifies which factors influence fare of the cab and how the resources

are used to determine it.

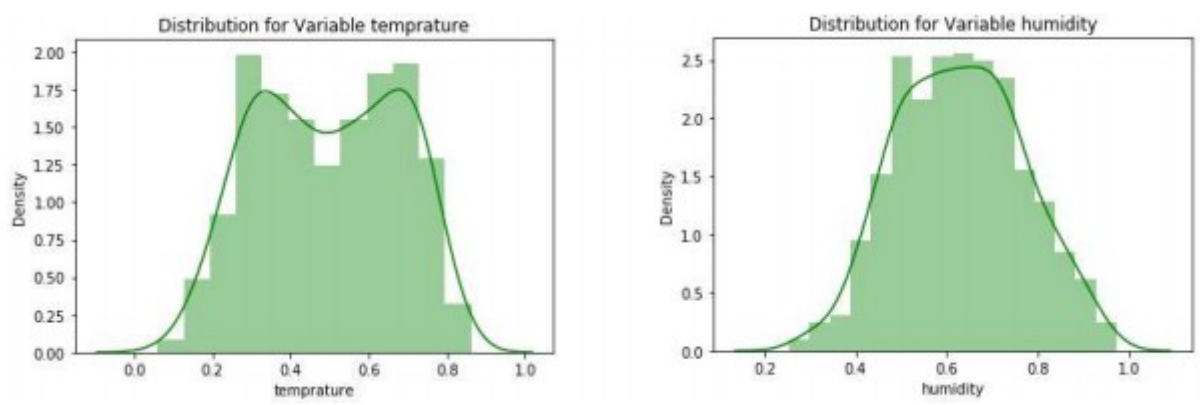
2.2.1.1 Univariate Analysis

Univariate analysis is the simplest form of data analysis where the data being analysed contains only one variable. Since it's a single variable it doesn't deal with causes or relationships. The main purpose of univariate analysis is to describe the data and find patterns that exist within it. So, Lets have a look at histogram plot, to identify the characteristic of the features and the data.





Histograms are constructed by binning the data and counting the number of observations in each bin. The objective of plotting Histogram plot is usually to visualise the shape of the distribution.The number of bins needs to be large enough to reveal interesting features and small enough not to be too noisy.



A Density Plot visualises the distribution of data over a continuous interval or time period. Density plots can be thought of as plots of smoothed histograms. An advantage Density Plots have over Histograms is that they're better at determining the distribution shape because they're not affected by the number of bins used.

2.1.2.2 Bivariate Analysis

1. From season plot in figure-1.4.1 we can see that season 2,3 and 4 have more bike count as compare to season 1. the daily bike count for these season was between 4000 to 8000.

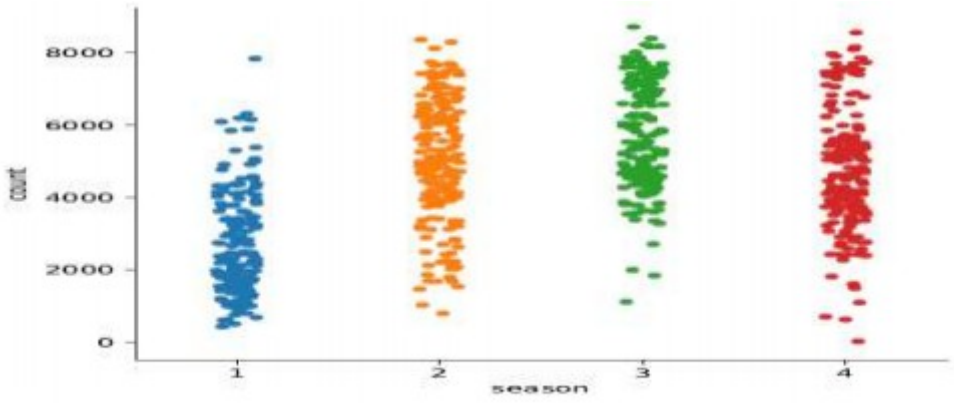


figure-1.4.1

2. Below plot figure-1.4.2 is for month wise count of bikes, so this tells us that the bike counts are higher between month 4 to month 10 as comapre to other months.

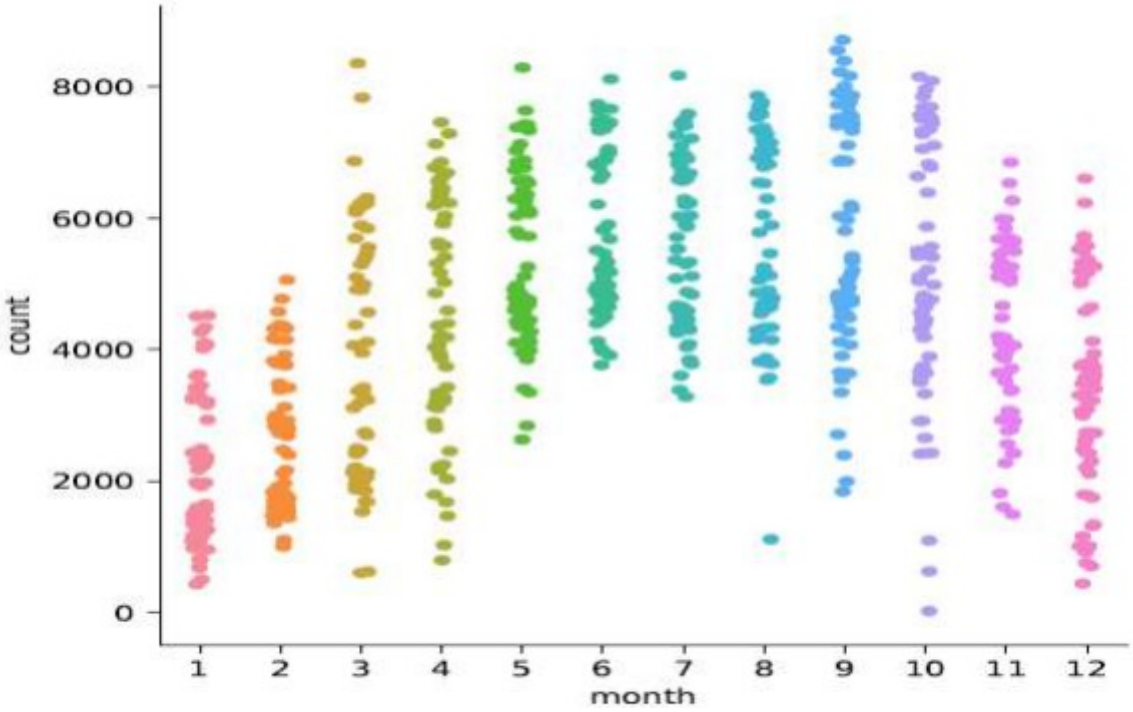


figure-1.4.2

3. Below Plot Figure-1.4.3 is between holiday and count, from this plot we can clearly say count of rented bikes are higher on holiday.

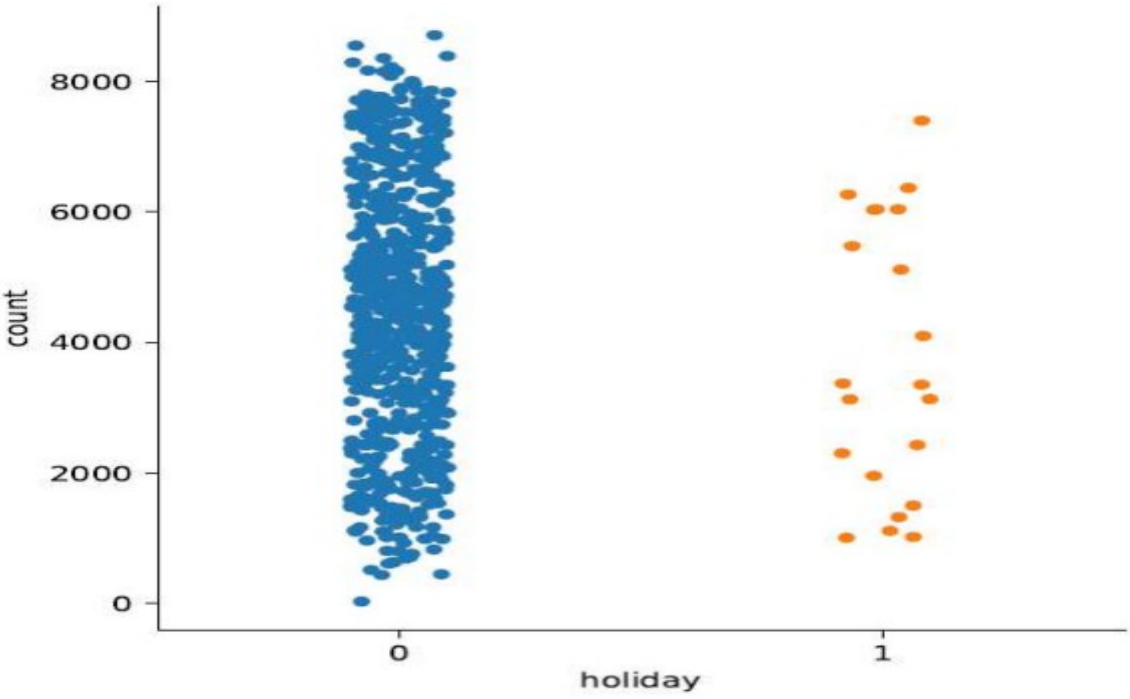


figure-1.4.3

4. In weather-1 in figure-1.4.4 the count of bikes is good as compare to other weather.

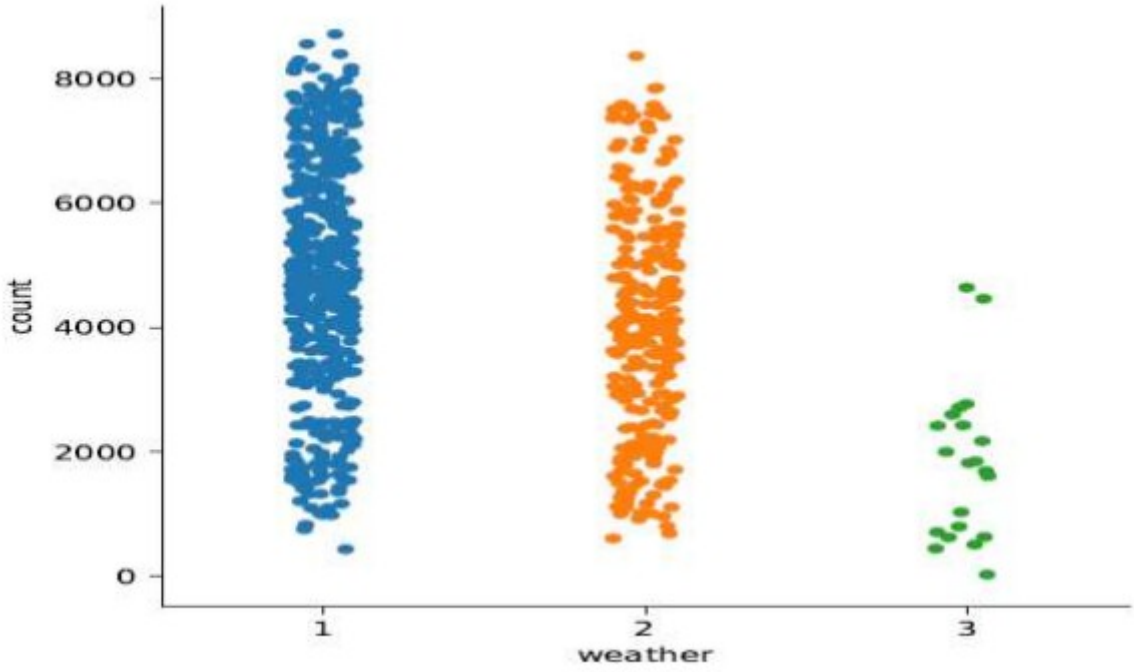


figure-1.4.4

5. Below plot figure-1.4.5 is for count bike with respect to normalized temperature and normalized humidity, from this we can see that count is maximum when temperature 0.4 to 0.7 and humidity below 0.75.

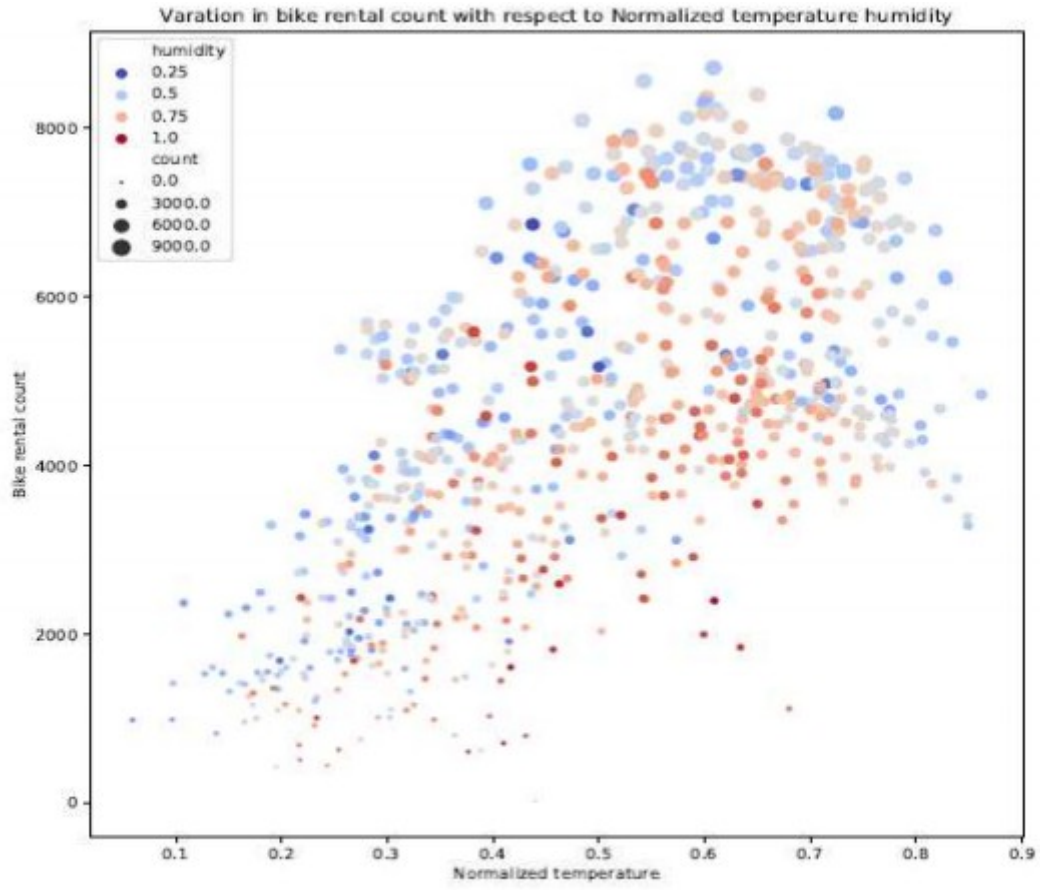


figure-1.4.5

6. The below plot figure-1.4.6 is for bike count with respect to Normalized Temperature and Normalized Humidity, from this plot it is clear that count is higher when the temp is 0.5 to 0.7 and windspeed below 0.15 and humidity less than 0.75.

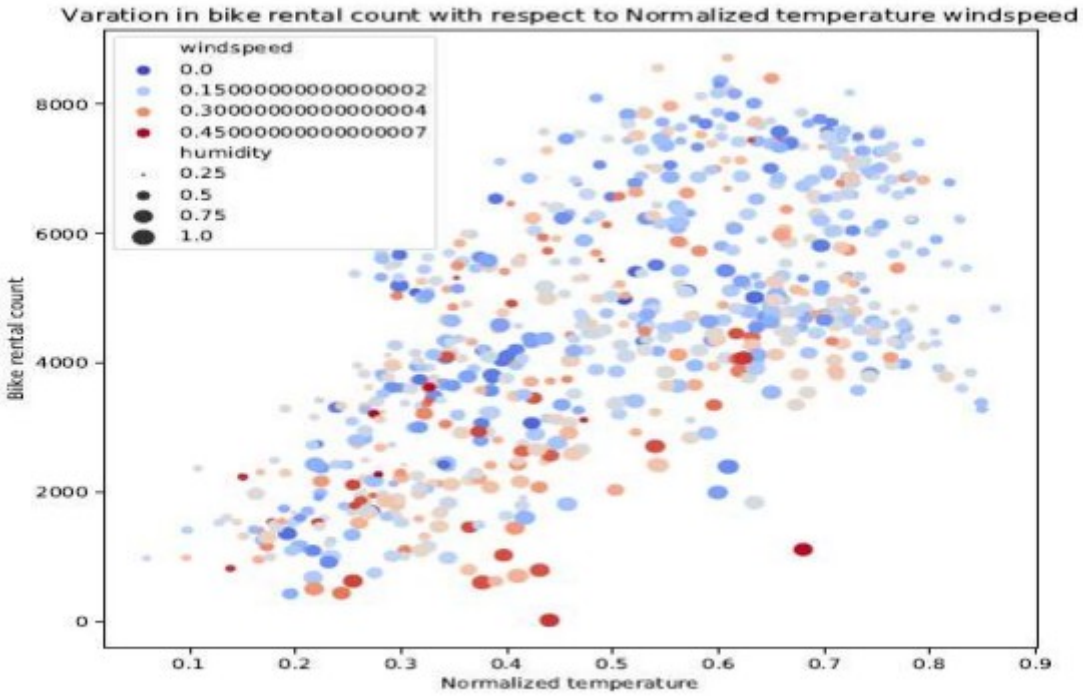


figure-1.4.6

7. Below Plot figure-1.4.7 is plotted for count of bikes with respect to temperature, weather and humidity, and we have found that the count is maximum when temperature is between 0.5 to 0.7, and in season 2 and 3.

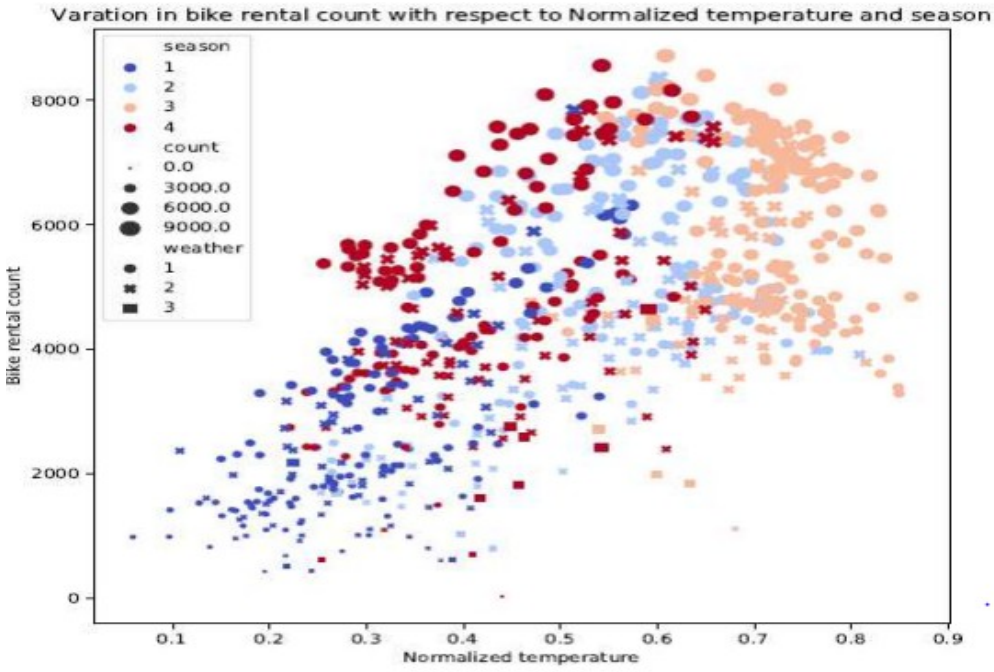


figure-1.4.7

2.1.2.3 Feature Selection

Correlation Analysis:

If the values of one column to other column are similar then it is said to be collinear. Therefore between predictor variables there should be less collinearity as compared to the collinearity among the predictors and variable.

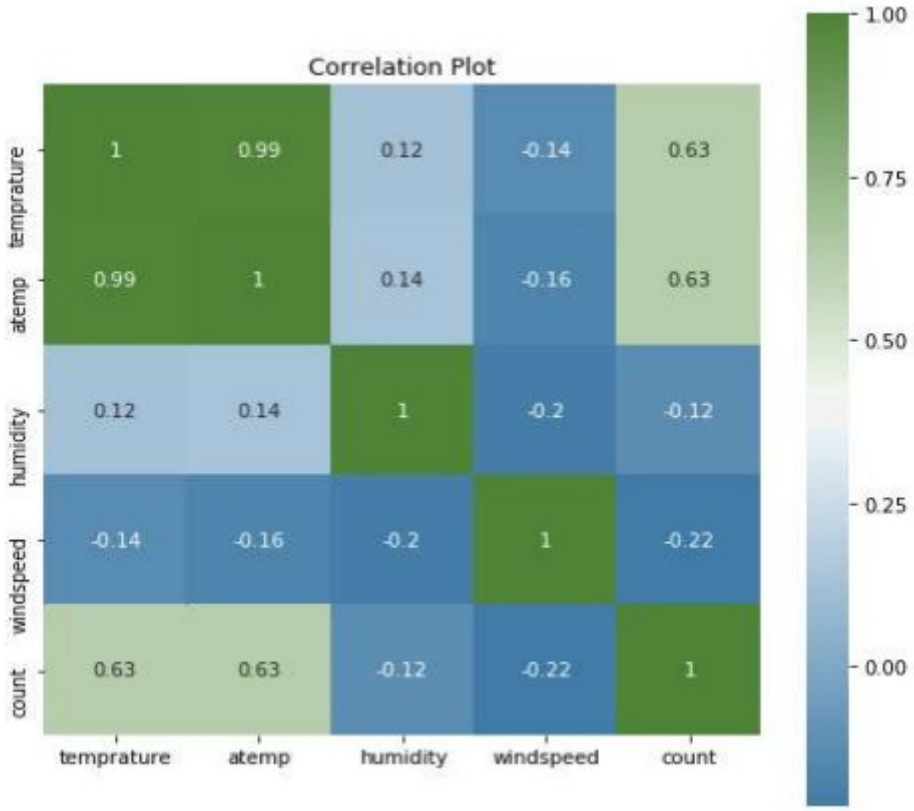


Fig: Collinearity test in train

The value for collinearity is between -1 to 1. So, any value close to -1/1 will result in high collinearity.

It seems not all right in our train and test data the situations nothing is more than 0.99 is positive direction and nothing is less than -0.1 is negative direction.

From correlation analysis we have found that temperature and atemp has high correlation (>0.9), so we have excluded the atemp column.

ANOVA Test For Categorical DATA

workingday	1.02E+07	1	0	2.736742	0.098495
Residual	2.73E+09	729	0	NaN	NaN

	sum_sq	df	F	PR(>F)
weather	2.42E+08	1	70.729298	2.15E-16
Residual	2.50E+09	729	NaN	NaN

	sum_sq	df	F	PR(>F)
season	4.52E+08	1	143.967653	2.13E-30
Residual	2.29E+09	729	NaN	NaN

	sum_sq	df	F	PR(>F)
year	8.80E+08	1	344.890586	2.48E-63
Residual	1.86E+09	729	NaN	NaN

	sum_sq	df	F	PR(>F)
month	2.15E+08	1	62.004625	1.24E-14
Residual	2.52E+09	729	NaN	NaN

	sum_sq	df	F	PR(>F)
holiday	1.28E+07	1	3.421441	0.064759
Residual	2.73E+09	729	NaN	NaN

	sum_sq	df	F	PR(>F)
weekday	1.25E+07	1	3.331091	0.068391
Residual	2.73E+09	729	NaN	NaN

	sum_sq	df	F	PR(>F)

from ANOVA analysis we have found that in categorical variables **Holiday, weekday and working day** have the **pr(>0.05)**, so we excluded them.

After Correlation Analysis we have remaining variables :

Continuous variables in dataset-

- **temperature float64**
- **humidity float64**
- **windspeed float64**
- **count int64**

Categorical variables in dataset-

- **season int64**
- **year int64**

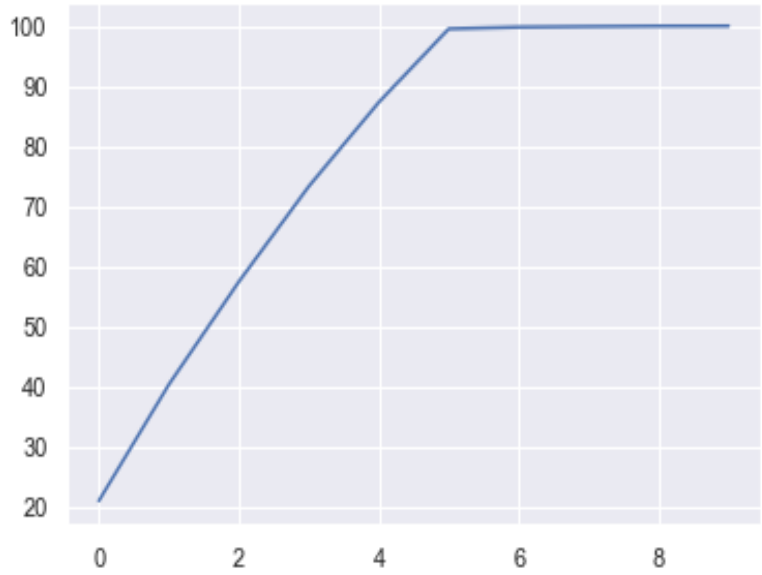
- **month int64**
- **weather int64**

Dimension Reduction :

Before performing any type of modeling we need to assess the importance of each predictor variable in our analysis. There is a possibility that many variables in our analysis are not important at all to the problem of class prediction. There are several methods of doing that. Here, we do this in two steps : Firstly, we find and remove the correlated features and then we use a

more advanced technique for dimensionality reduction called PCA .While doing this, we first plot a cumulative distribution function plot to observe how much percentage of variance is explained by how many variables (Principle Components).

The CDF plot for the same is plotted below :



It is very clear from the above CDF plot for 'Variance Explained' Vs 'Principle components' , that almost 99%+ variance is explained by just 5 variables (Principle components). We, can imagine how powerful PCA is, It just shrank down our feature space to just 5 from a total of 7 features. So, we will keep only 5 principle components in the data and will perform modeling on it.

2.2 Modeling

We always start our model building from the most simplest to more complex. Therefore we start with KNN Regressor.

2.2.1 KNN Regression

KNN regression is one of the simplest algorithm in the whole of Machine learning. It gives a weighted average of the regression function in a local space (k nearest points to a given point). So, we first try to implement and fit KNN regression to our Data and got following results after tuning the hyper-parameter k :

```
n_neighbours : 10  ----KNN rmse: 963.1413189164306
n_neighbours : 20  ----KNN rmse: 1103.7651249493354
n_neighbours : 30  ----KNN rmse: 1267.4708116342053
n_neighbours : 40  ----KNN rmse: 1352.9516793413773
n_neighbours : 50  ----KNN rmse: 1414.4452563547625
n_neighbours : 60  ----KNN rmse: 1452.8106744648467
n_neighbours : 70  ----KNN rmse: 1475.7028170063595
n_neighbours : 80  ----KNN rmse: 1502.5664561109309
n_neighbours : 90  ----KNN rmse: 1528.8631551009655
n_neighbours : 100  ----KNN rmse: 1527.0453864361905
```

So, n_neighbors =10 seems to have best RMSE value, Lets fit and predict it on our data

```
Train Data
n_neighbours : 100  ----KNN rmse: 884.3329652869164
Test Data
n_neighbours : 100  ----KNN rmse: 963.1413189164306
Accuracy :
Out[7]: 0.7920622261849743
```

2.2.2 Ordinary Least Squares

Now we will try to implement Multiple Linear Regression algorithm using Ordinary Least Squares, the simplest of all.Ordinary least squares (OLS) minimises the squared distances between the observed and the predicted dependent variable.

```
Train Data
Ordinary Least Squares rmse: 1217.9259815210899
Test Data
Ordinary Least Squares rmse: 1228.8143165506606
Accuracy :
Out[8]: 0.60559396519018
```

2.2.3 Ridge Regression

Ridge Regression essentially is an instance of Linear Regression with regularisation. Ridge regression is that it enforces the β coefficients to be lower, but it does not enforce them to be zero.

```
alpha : 0.1  ----Ridge rmse: 1228.9021249856148
alpha : 0.5  ----Ridge rmse: 1229.257689087821
alpha : 1.0  ----Ridge rmse: 1229.7117410822514
alpha : 3.0  ----Ridge rmse: 1231.6299961091233
alpha : 7.0  ----Ridge rmse: 1235.9101534993472
alpha : 10.0  ----Ridge rmse: 1239.4573878062963
```

That is, it will not get rid of irrelevant features but rather minimise their impact on the trained model.

```
Train Data
Ridge rmse: 1217.9262555308007
Test Data
Ridge rmse: 1228.9021249856148
Accuracy :
Out[10]: 0.6055937877227519
```

So, we can see from the above results that, our model gives a RMSE (Root Mean Square Value) of 1217.926 for the train data and a RMSE value of 1228.902 for the test data. Looking at the train and test error, we can say that the model doesn't fit at all and that might be because of the regularisation term involve in the cost function.

2.2.4 Lasso Regression

Least absolute shrinkage and selection operator, abbreviated as LASSO, is an Linear Regression technique which also performs regularisation on variables in consideration.

It sets the coefficients to zero thus reducing the errors completely. That is, It will get rid of irrelevant features completely.

```
alpha : 0.1  ---- Lasso rmse: 1228.8484240665598
alpha : 0.5  ---- Lasso rmse: 1228.9856412446593
alpha : 1.0  ---- Lasso rmse: 1229.1589854044041
alpha : 3.0  ---- Lasso rmse: 1229.8725974120973
alpha : 7.0  ---- Lasso rmse: 1231.396764629864
alpha : 10.0  ---- Lasso rmse: 1232.6244780956392
```

So, after we implement Lasso Regression on our data, we get the following results :

```
Train Data
Lasso rmse: 1217.9260238137874
Test Data
Lasso rmse: 1228.8484240665598
Accuracy :
Out[12]: 0.6055939377985392
```

2.2.5 Support Vector Regression

Support Vector Machine can be applied not only to classification problems but also to the case of regression.

```
C : 1 , gamma : 0.001 ----SVR rmse: 1921.587292887561
C : 1 , gamma : 0.0001 ----SVR rmse: 1922.2376870301678
C : 10 , gamma : 0.001 ----SVR rmse: 1914.812061428869
C : 10 , gamma : 0.0001 ----SVR rmse: 1921.6107643204737
C : 100 , gamma : 0.001 ----SVR rmse: 1875.799973317111
C : 100 , gamma : 0.0001 ----SVR rmse: 1915.0264379448229
C : 1000 , gamma : 0.001 ----SVR rmse: 1712.3568627278732
C : 1000 , gamma : 0.0001 ----SVR rmse: 1880.615245629411
```

Still it contains all the main features that characterise maximum margin algorithm: a non-linear function is leaned by linear learning machine mapping into high dimensional kernel induced feature space.

```
Train Data
Support Vector Regression rmse: 1937.934050204391
Test Data
Support Vector Regression rmse: 1921.587292887561
Accuracy :
Out[14]: 0.001427163492411121
```

2.2.6 Decision Tree Regression

Decision tree builds regression , models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed.

```
depth : 1 ----  Decision Tree rmse: 1605.7608914281689
depth : 2 ----  Decision Tree rmse: 1271.7778961252823
depth : 5 ----  Decision Tree rmse: 1115.004690167551
depth : 10 ----  Decision Tree rmse: 1126.4971906559128
depth : 20 ----  Decision Tree rmse: 1045.3689255359518
```

The final result is a tree with decision nodes and leaf nodes. So, after we implement Decision Tree Regression on our data, we get the following results :

```
Train Data
Decision Tree rmse: 1.847805065921142
Test Data
Decision Tree rmse: 1035.171098940476
Out[16]: 0.9999990921500627
```

2.2.7 Gradient Boosting Decision Tree

Regression 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 do, and it generalises them by allowing optimisation of an arbitrary differentiable loss function.

```
depth : 1, learning_rate0.001  ---- Gradient Boosting Regression rmse: 1863.8944324430588
depth : 1, learning_rate0.01  ---- Gradient Boosting Regression rmse: 1548.0363678074316
depth : 1, learning_rate0.1  ---- Gradient Boosting Regression rmse: 940.1956608274061
depth : 2, learning_rate0.001  ---- Gradient Boosting Regression rmse: 1817.7726785259672
depth : 2, learning_rate0.01  ---- Gradient Boosting Regression rmse: 1355.1948602818422
depth : 2, learning_rate0.1  ---- Gradient Boosting Regression rmse: 860.7308122073152
depth : 5, learning_rate0.001  ---- Gradient Boosting Regression rmse: 1786.7804330185918
depth : 5, learning_rate0.01  ---- Gradient Boosting Regression rmse: 1086.1474714791846
depth : 5, learning_rate0.1  ---- Gradient Boosting Regression rmse: 802.3830089307202
```

So, after we implement Gradient Boosting Decision Trees on our data, we get the following results :

```
Train Data
GBDT rmse: 1.847805065921142
Test Data
GBDT rmse: 1035.171098940476
Accuracy :
Out[18]: 0.9814856627410993
```

2.2.8 Random Forest Regression

Random Forest Regression or Regression Trees are know to be very unstable, in other words, a small change in our data may drastically change your model. The Random Forest uses this instability as an advantage through bagging resulting on a very stable model.

```
depth : 2, n_estimators : 100  ---- Random Forest Regression rmse: 1264.9116691553118
depth : 5, n_estimators : 100  ---- Random Forest Regression rmse: 859.0817472299184
depth : 10, n_estimators : 100  ---- Random Forest Regression rmse: 821.5446466817359
depth : 20, n_estimators : 100  ---- Random Forest Regression rmse: 815.344916565795
depth : 30, n_estimators : 100  ---- Random Forest Regression rmse: 817.5711204724713
depth : 2, n_estimators : 200  ---- Random Forest Regression rmse: 1261.953907889363
depth : 5, n_estimators : 200  ---- Random Forest Regression rmse: 855.6246950547284
depth : 10, n_estimators : 200  ---- Random Forest Regression rmse: 813.6231227522725
depth : 20, n_estimators : 200  ---- Random Forest Regression rmse: 812.7233295268747
depth : 30, n_estimators : 200  ---- Random Forest Regression rmse: 811.9080232302881
```

So, after we implement Random Forest Regression or Regression Trees on our data, we get the following results :

```
Train Data
Random Forest rmse: 313.96351555534534
Test Data
Random Forest rmse: 797.772995480028
Accuracy :
Out[20]: 0.9737904160016929
```

Chapter 3

Conclusion 3.1

Model Evaluation

Now that we have a few models for predicting the target variable, we need to decide which one to choose. There are several criteria that exist for evaluating and comparing models. We can compare the models using any of the following criteria:

1. Predictive Performance

2. Interpretability

3. Computational Efficiency

In our case of Cab fare , the latter two, Interpretability and Computation Efficiency, do not hold much significance. Therefore we will use Predictive performance as the criteria to compare and evaluate models. Predictive performance can be measured by comparing Predictions of the models with real values of the target variables, and calculating some average error measure.

3.1.1 Root Mean Square Value

Root Mean Square Error (RMSE) is the standard deviation of the residuals (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. Also, Since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors. So, RMSE becomes more useful when large errors are particularly undesirable. So, Root Mean Square value seems like a perfect choice for our problem at hand.

3.2 Model Selection

3.2.1 Cross Validation

we used two techniques of **Cross Validation** in our model-

- Random Search
- Grid Search

Random Search
Random search is a technique where random combinations of the hyperparameters are used to find the best solution for the built model. In this search pattern, random combinations of parameters are considered in every iteration. The chances of finding the optimal parameter are comparatively higher in random search because of the random search pattern where the model might end up being trained on the optimised parameters without any aliasing.

Grid Search
Grid search is a technique which tends to find the right set of hyperparameters for the particular model. Hyperparameters are not the model parameters and it is not possible to find the best set from the training data. Model parameters are learned during training when we optimise a loss function using something like a gradient descent. In this tuning technique, we simply build a model for every combination of various hyperparameters and evaluate each model. The model which gives the highest accuracy wins. The pattern followed here is similar to the grid, where all the values are placed in the form of a matrix. Each set of parameters is taken into consideration and the accuracy is noted. Once all the combinations are evaluated, the model with the set of parameters which give the top accuracy is considered to be the best.

	Model Name	MAPE_Train	MAPE_Test	R-squared_Train	R-squared_Test
0	Decision Tree	62.260133	36.948093	0.677563	0.646470
1	Decision Tree Random Search CV	14.180789	23.419816	0.874435	0.809361
2	Decision Tree Grid Search CV	14.180789	23.419816	0.874435	0.809361
3	Random Forest	16.776997	20.426067	0.979178	0.881801
4	Random Forest Random Search CV	21.445350	21.029355	0.978219	0.878929
5	Random Forest Grid Search CV	21.320742	20.567325	0.964826	0.875335
6	Linear Regression	44.444512	18.800696	0.832760	0.841110
7	Gradient Boosting	44.444512	19.899341	0.945385	0.864595
8	Gradient Boosting Random Search CV	1.732620	21.730096	0.998236	0.866549
9	Gradient Boosting Grid Search CV	18.833448	25.485646	0.922114	0.833746

From the observation of all MAPE and R-Squared Value we have concluded that **Random Forest** has minimum value of MAPE (20.42%) and it's R-Squared Value is also maximum (0.88).Means, By Random forest algorithm predictor are able to explain 88% to the target variable on the test data. The MAPE value of Test data and Train does not differs a lot this implies that it is not the case of overfitting

Appendix A - Extra Figures

R-Result-

Model	MAPE_Train	MAPE_Test	R.Squared_Train	R.Squared_Test
Decision Tree for Regression	56.30014552	23.70970208	0.793974257	0.752194789
Random Search in Decision Tree	56.30014552	23.70970208	0.793974257	0.752194789
Gird Search in Decision Tree	56.30014552	23.70970208	0.793974257	0.752194789
Random Forest	23.31578346	17.41229633	0.967674325	0.866706395
Random Search in Random Forest	25.44288679	17.52099336	0.96787762	0.865370081
Grid Search in Random Forest	24.88492838	17.61271901	0.964533357	0.866318841
Linear Regression	47.40023298	16.87102913	0.900758595	0.851246285
Gradient Boosting	37.02665525	17.24280795	0.900758595	0.851246285
Random Search in Gradient Boosting	25.00204653	17.60370181	0.968267213	0.863821245
Grid Search in Gradient Boosting	25.64630592	17.40282785	0.964533357	0.866318841

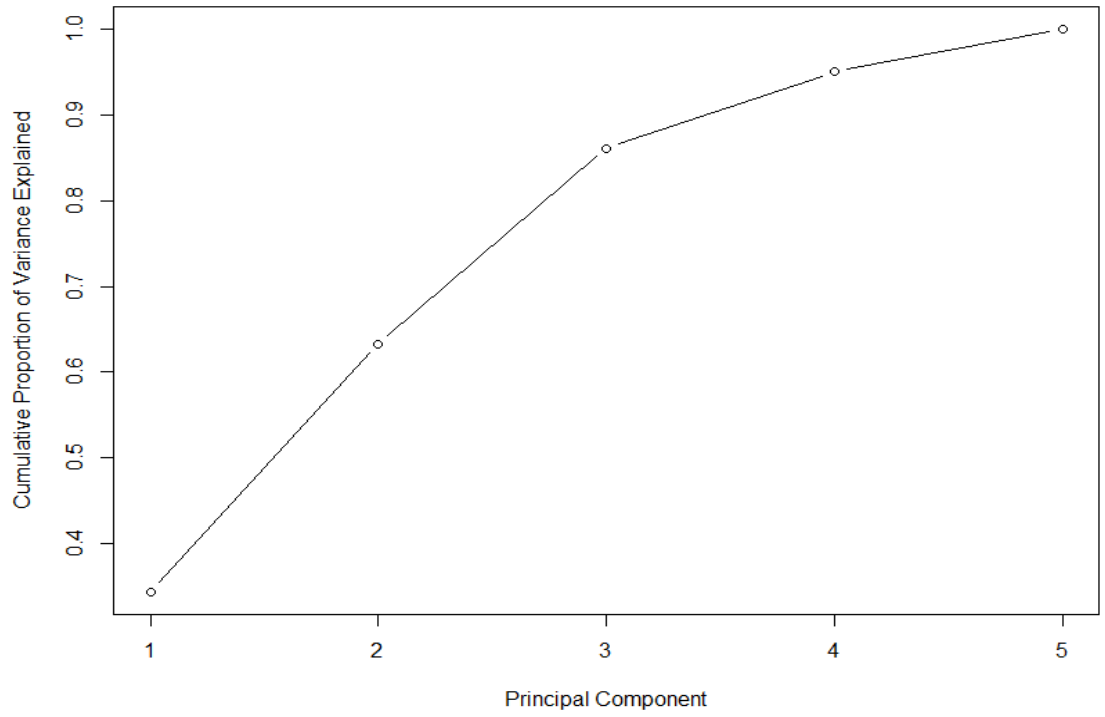


Fig : PCA Analysis in R

4. REFRENCES

- Edwisor Data Science Training Path
- Couresra Machine Learning, Stanford University.
- Udacity's Data Visualization
- Statistics from DataScienceCentral blog and TowardsDataScience blogs.
- Solved errors by searching same errors on Stackoverflow.

END OF THE REPORT