GREAT LEARNING

POST GRADUATE PROGRAM IN DATA SCIENCE & BUSINESS ANALYTICS

BUSINESS REPORT

CASE STUDIES ON:

Machine Learning – ABC Company and Travel Agency

Submitted By:

STEFFIN JOHN

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CASE STUDY

1. **Machine Learning – ABC Company and Travel Agency**

MACHINE LEARNING

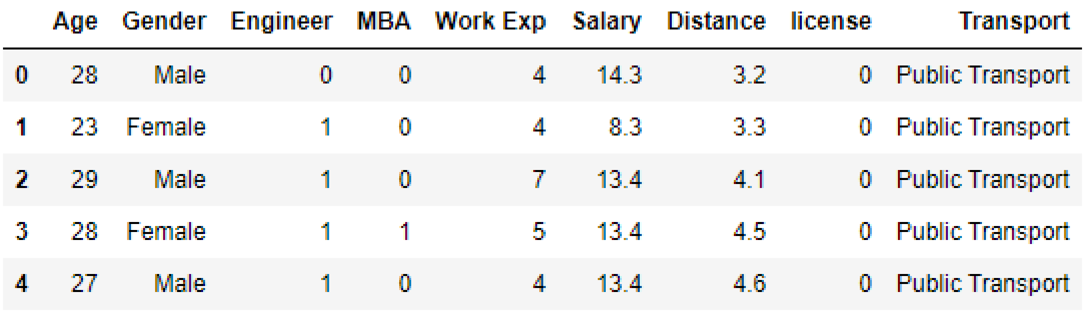
You work for an office transport company. You are in discussions with ABC Consulting company for providing transport for their employees. For this purpose, you are tasked with understanding how do the employees of ABC Consulting prefer to commute presently (between home and office). Based on
the parameters like age, salary, work experience etc. given in the data set 'Transport.csv', You are required to predict the preferred mode of transport. The project requires you to build several Machine Learning models and compare them so that the model can be finalised.

Data Ingestion:

1. Read the dataset. Do the descriptive statistics and do the null value condition check. Write an inference on it.

Exploratory Data Analysis (EDA)

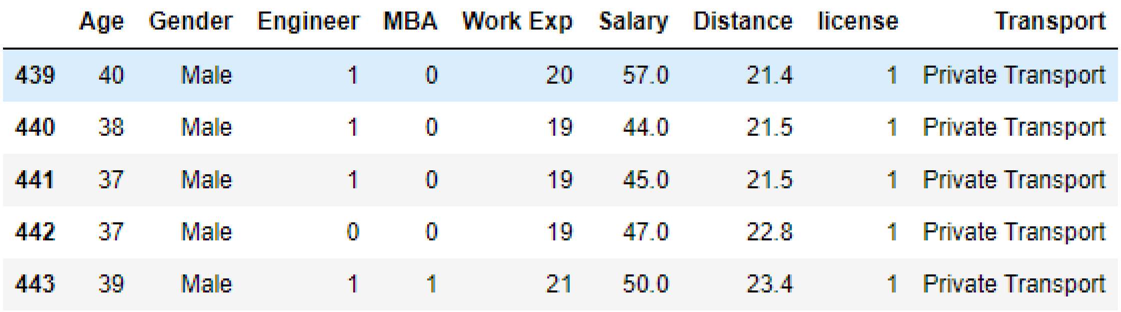
Head of the Dataset



	Age	Gender	Engineer	MBA	Work Exp	Salary	Distance	license	Transport
0	28	Male	0	0	4	14.3	3.2	0	Public Transport
1	23	Female	1	0	4	8.3	3.3	0	Public Transport
2	29	Male	1	0	7	13.4	4.1	0	Public Transport
3	28	Female	1	1	5	13.4	4.5	0	Public Transport
4	27	Male	1	0	4	13.4	4.6	0	Public Transport

Table 1: Dataset Head

Tail of the dataset



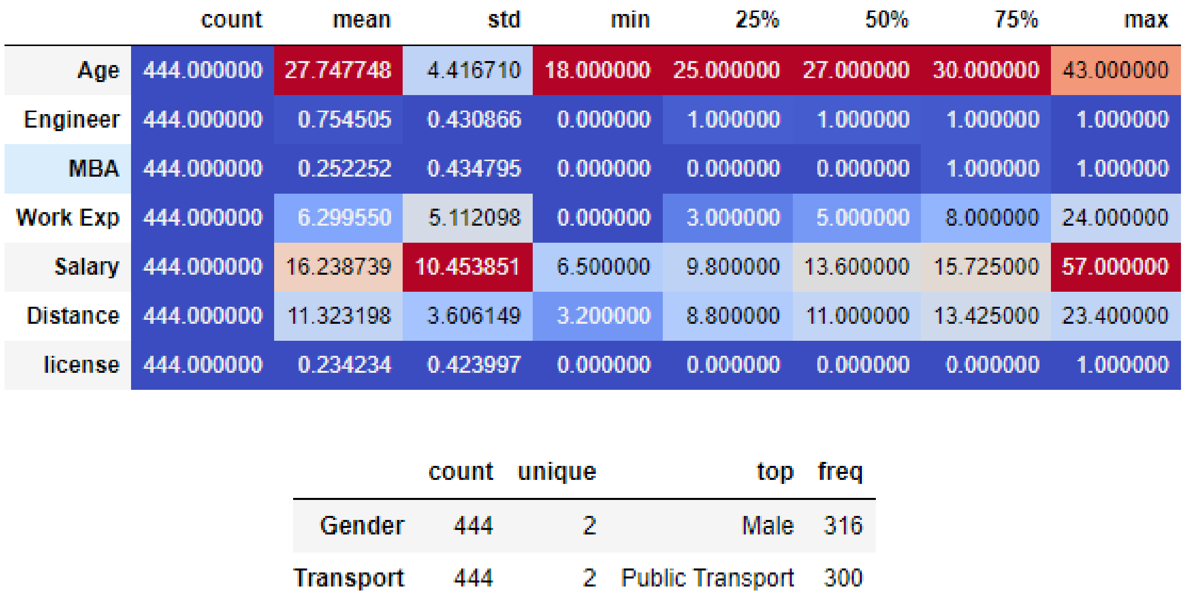
	Age	Gender	Engineer	MBA	Work Exp	Salary	Distance	license	Transport
439	40	Male	1	0	20	57.0	21.4	1	Private Transport
440	38	Male	1	0	19	44.0	21.5	1	Private Transport
441	37	Male	1	0	19	45.0	21.5	1	Private Transport
442	37	Male	0	0	19	47.0	22.8	1	Private Transport
443	39	Male	1	1	21	50.0	23.4	1	Private Transport

Table 2: Dataset Tail

Shape of the Dataset

- Total No. of Rows = 444
- Total No. of Columns = 9

Summary of the Dataset



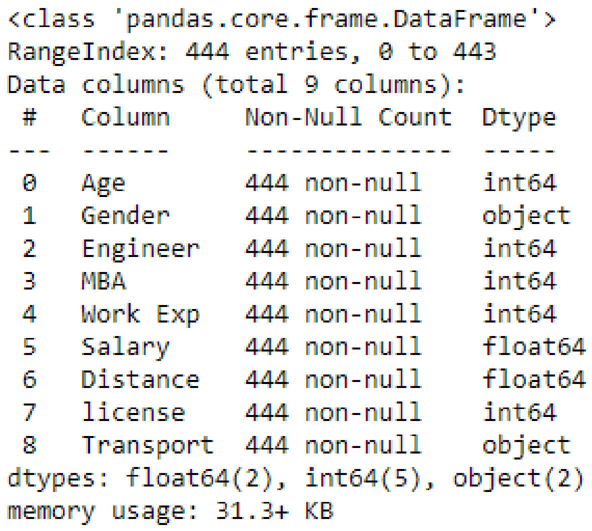
	count	mean	std	min	25%	50%	75%	max
Age	444.000000	27.747748	4.416710	18.000000	25.000000	27.000000	30.000000	43.000000
Engineer	444.000000	0.754505	0.430866	0.000000	1.000000	1.000000	1.000000	1.000000
MBA	444.000000	0.252252	0.434795	0.000000	0.000000	0.000000	1.000000	1.000000
Work Exp	444.000000	6.299550	5.112098	0.000000	3.000000	5.000000	8.000000	24.000000
Salary	444.000000	16.238739	10.453851	6.500000	9.800000	13.600000	15.725000	57.000000
Distance	444.000000	11.323198	3.606149	3.200000	8.800000	11.000000	13.425000	23.400000
license	444.000000	0.234234	0.423997	0.000000	0.000000	0.000000	0.000000	1.000000

	count	unique	top	freq
Gender	444	2	Male	316
Transport	444	2	Public Transport	300

Table 3: Dataset Describe

1. **Age**: Min age of Employee is 18 years where as Max age of Employee is 43 years in the ABC company.
2. **Engineer**: Yes = 1 and No = 0.
3. **MBA**: Yes = 1 and No = 0.
4. **Work Exp** : Min Working Experience of Employee is 0 years where as Max Working Experience of Employee is 24 years in the ABC company. It may contain outliers.
5. **Salary** : Min Salary of Employee is 6.5 whereas Max Working Salary of Employee is 57 in the ABC company. It may contain outliers.
6. **Distance**: Min distance of employee House from the office is 3.2 whereas Max distance of the employee house from the office is 23.4. It may contain outliers.
7. **License** : Yes = 1 and No = 0.
8. **Gender**: Frequency of Male are more compared to female in the ABC company.
9. **Transport** : Preferred Mode of transport for the employees of the ABC company is Public Transport.

Info of the Dataset



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 444 entries, 0 to 443
Data columns (total 9 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Age        444 non-null    int64
 1   Gender     444 non-null    object
 2   Engineer   444 non-null    int64
 3   MBA        444 non-null    int64
 4   Work Exp   444 non-null    int64
 5   Salary     444 non-null    float64
 6   Distance   444 non-null    float64
 7   license    444 non-null    int64
 8   Transport  444 non-null    object
dtypes: float64(2), int64(5), object(2)
memory usage: 31.3+ KB
```

Table 4: Dataset Info

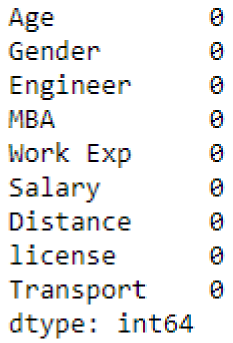
- Out of 9 columns total no. of float64(2), int64(5), object (2).
- Gender and Transport are our Object datatype.
- Transport is our Target Column.

Data Cleaning

Step-1: Checking for duplicate records in the data.

- Number of duplicate rows = 0

Step 2: Checking Missing value.



```
Age          0
Gender       0
Engineer     0
MBA          0
Work Exp     0
Salary       0
Distance     0
license      0
Transport    0
dtype: int64
```

Table 5: Dataset Null Values

- Number of Missing Values = 0

2. Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for Outliers.

Step 3 : Outlier Checks.

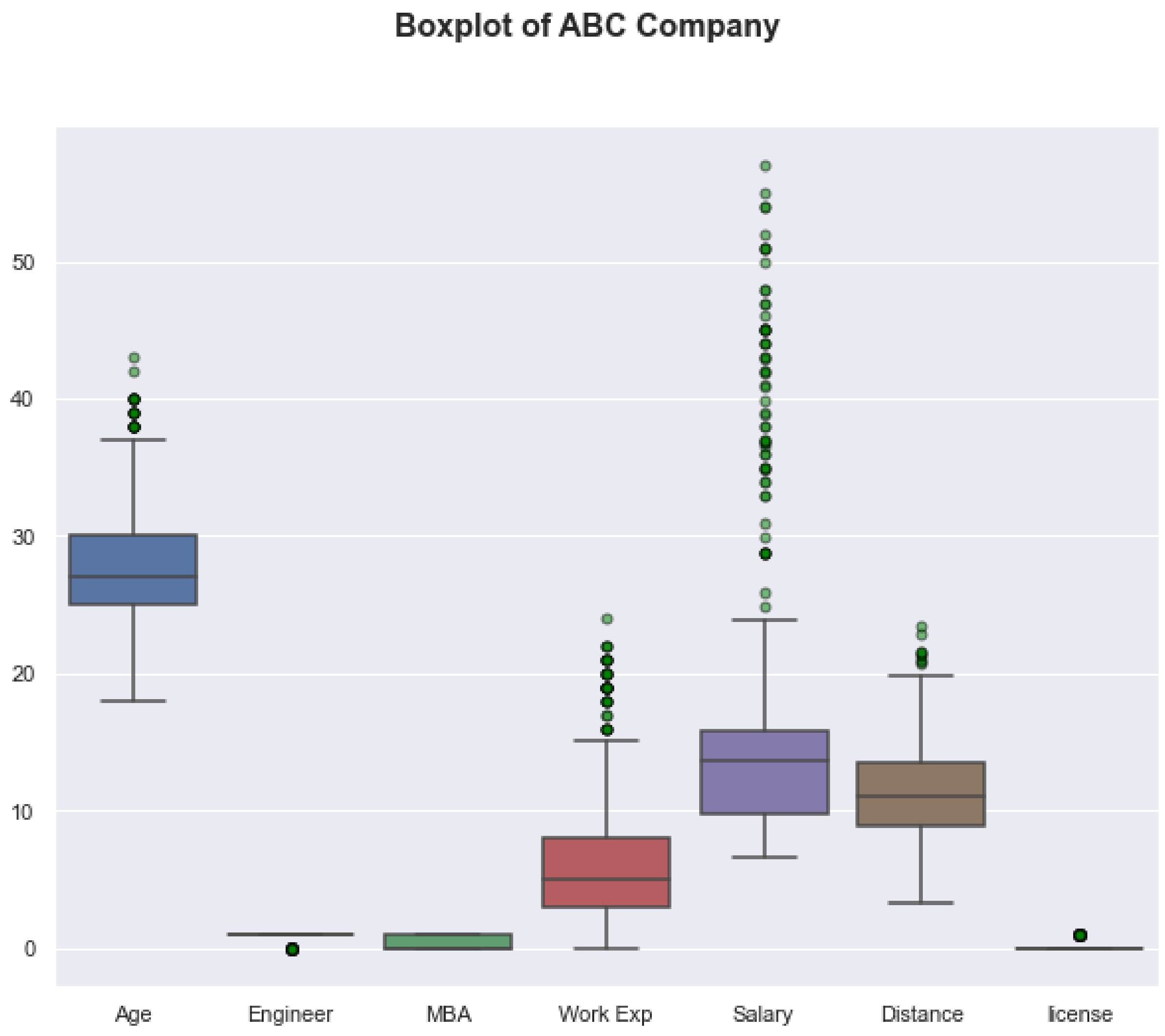


Fig 1: Boxplot before removing outliers

- Outliers can be easily seen for **Age, Work Exp, Salary** and **Distance**
- Thus we will treat these outliers

Removing Outliers

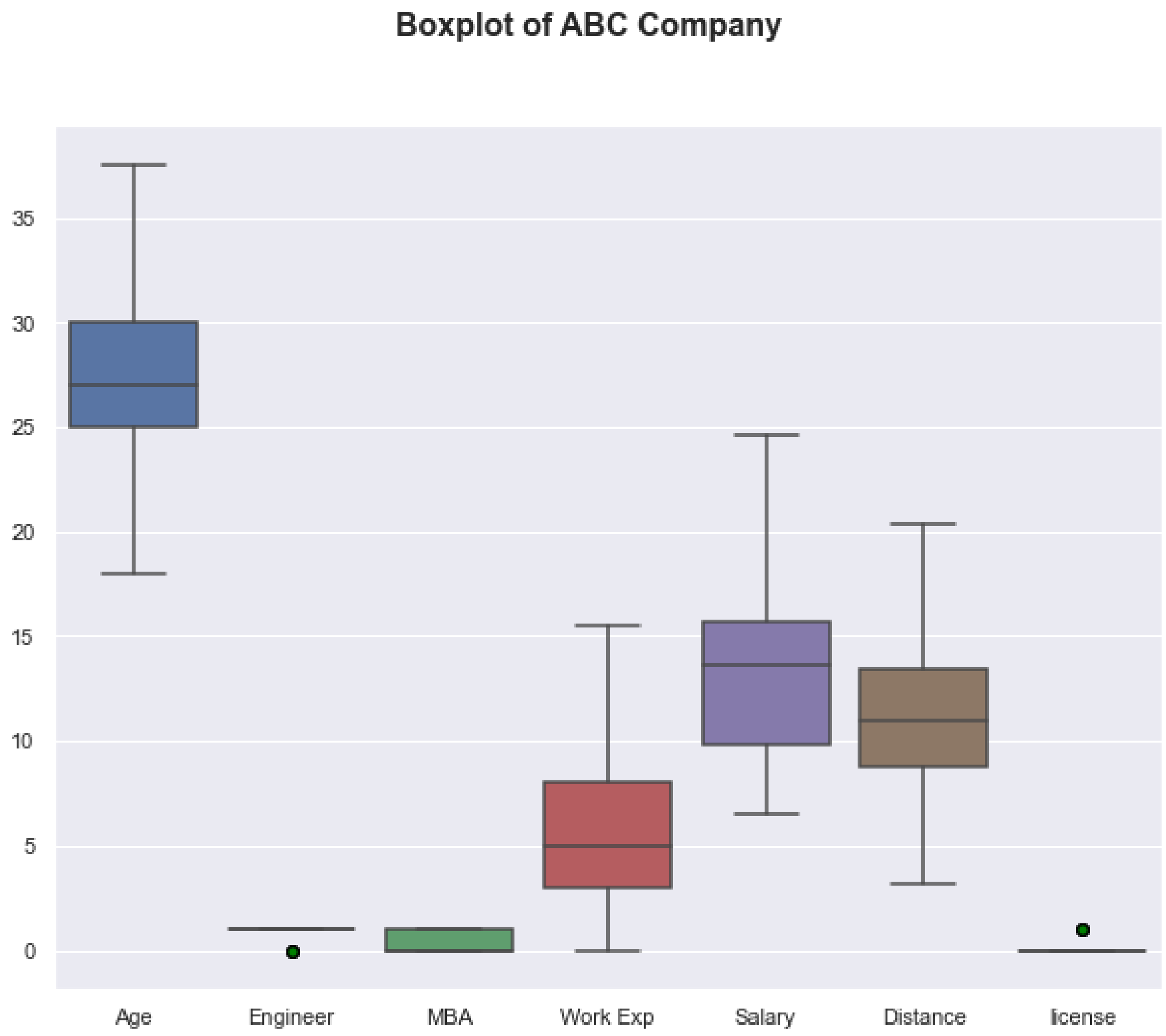


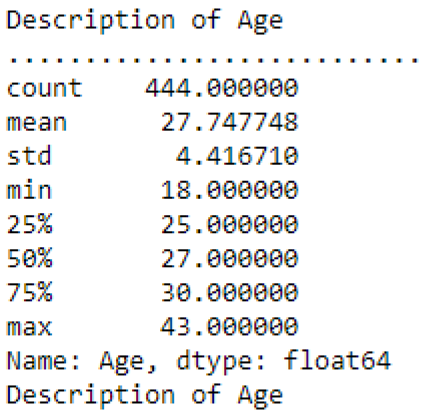
Fig 2: Boxplot after removing outliers

- As we can see the outliers are successfully removed

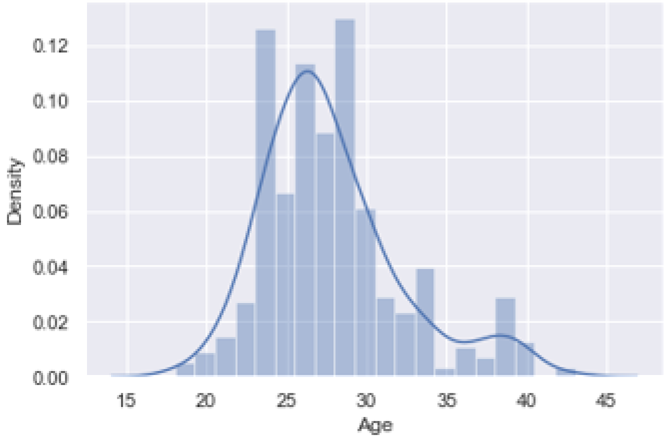
Step 4 - Univariate analysis, Bivariate Analysis and Multivariate Analysis

Numerical Variables:

1. Distribution of Age



```
Description of Age
...........................
count    444.000000
mean      27.747748
std        4.416710
min       18.000000
25%       25.000000
50%       27.000000
75%       30.000000
max       43.000000
Name: Age, dtype: float64
Description of Age
```



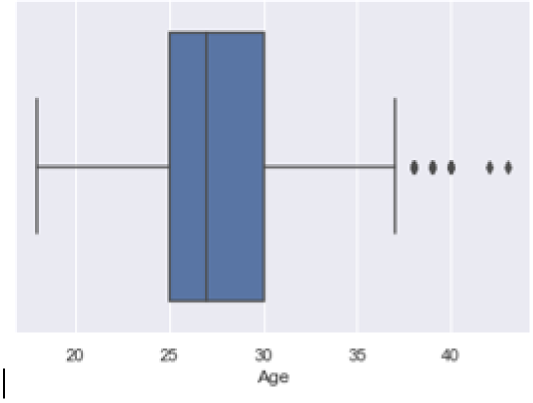
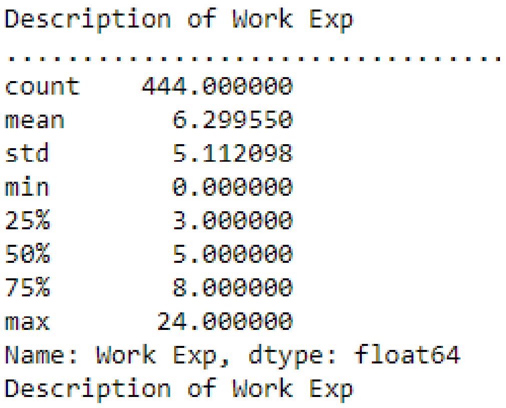


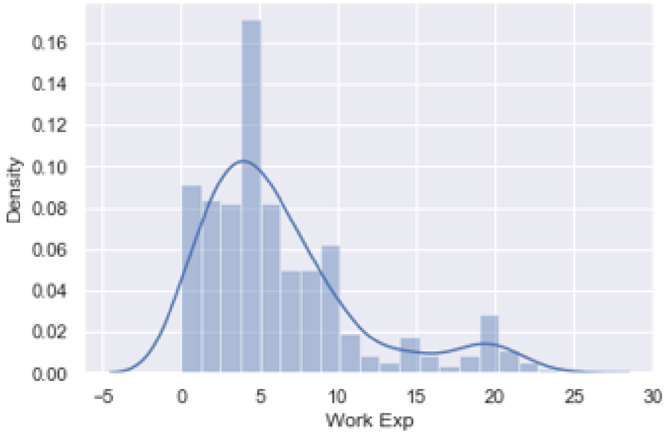
Fig 3: Distribution of Age

- The Distribution of Age of Employees lie between 18-43 years.
- The boxplot shows that data has outliers.
- There is Positive skewness in the Age column.

2. **Distribution of Work Exp**



```
Description of Work Exp
.................................
count     444.000000
mean        6.299550
std         5.112098
min         0.000000
25%         3.000000
50%         5.000000
75%         8.000000
max        24.000000
Name: Work Exp, dtype: float64
Description of Work Exp
```



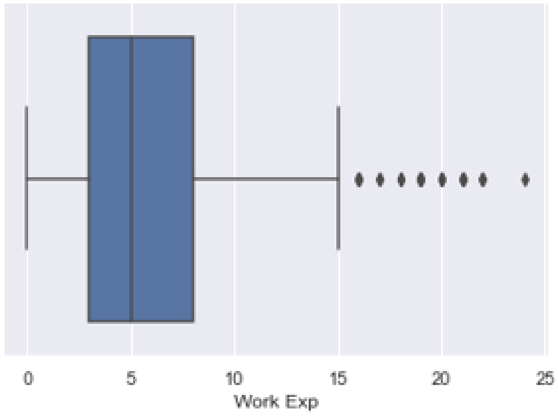
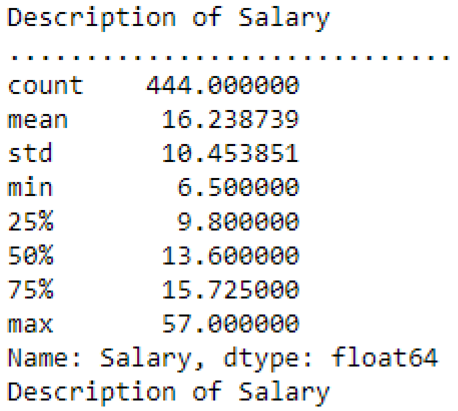


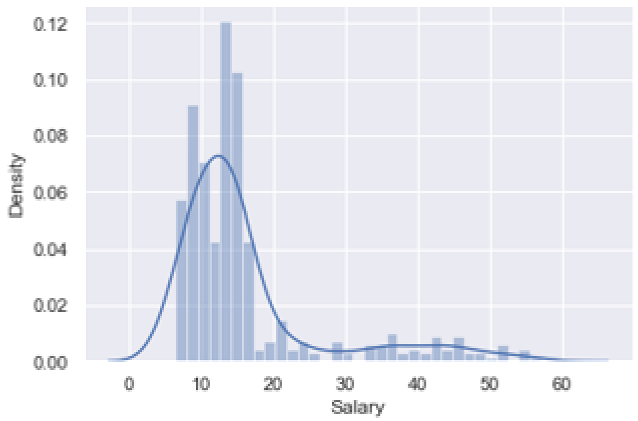
Fig 4: Distribution of Work Exp

- The Distribution of Employees Work Exp in the ABC company ranges from 0-24 years.
- The boxplot shows that data has outliers.
- There is Positive skewness in the Work Exp column towards right.

3. **Distribution of Salary**



```
Description of Salary
.............................
count    444.000000
mean      16.238739
std       10.453851
min        6.500000
25%        9.800000
50%       13.600000
75%       15.725000
max       57.000000
Name: Salary, dtype: float64
Description of Salary
```



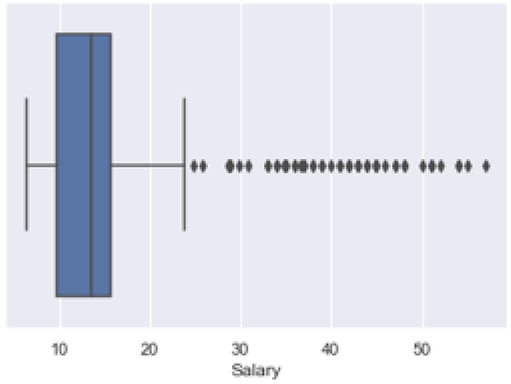
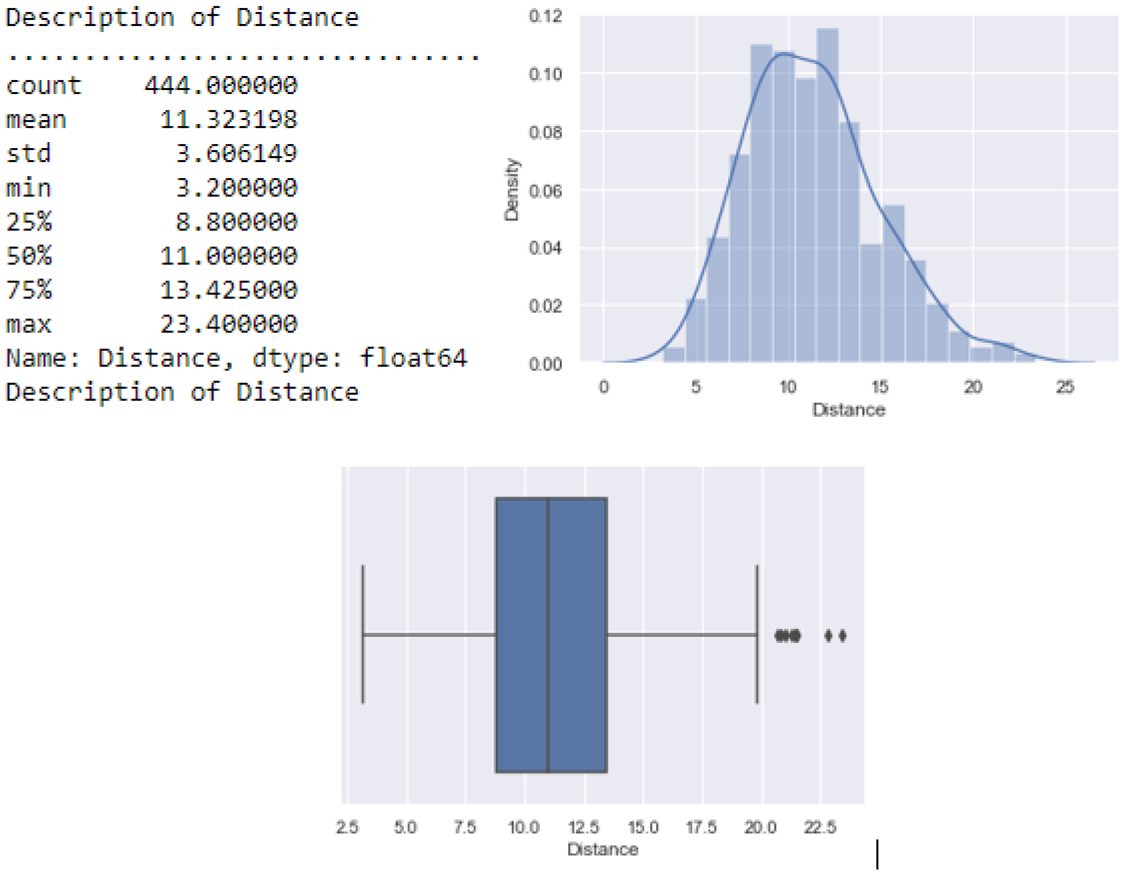


Fig 5: Distribution of Work Salary

- The Distribution of Employees Salary in the ABC company ranges from 3.2 - 23.4
- The boxplot shows that data has outliers.
- There is Positive skewness in the Salary column towards right.

4. **Distribution of Distance**



```
Description of Distance
................................
count    444.000000
mean      11.323198
std        3.606149
min        3.200000
25%        8.800000
50%       11.000000
75%       13.425000
max       23.400000
Name: Distance, dtype: float64
Description of Distance
```

Fig 6: Distribution of Distance

- The Distribution of Distance of office from home of each Employee in the ABC company ranges from 3-23.
- The boxplot shows that data has outliers.
- There is Positive skewness in the Distance column towards right.

Univariate and Bivariate Analysis Categorical Variables

1.Distribution of Gender

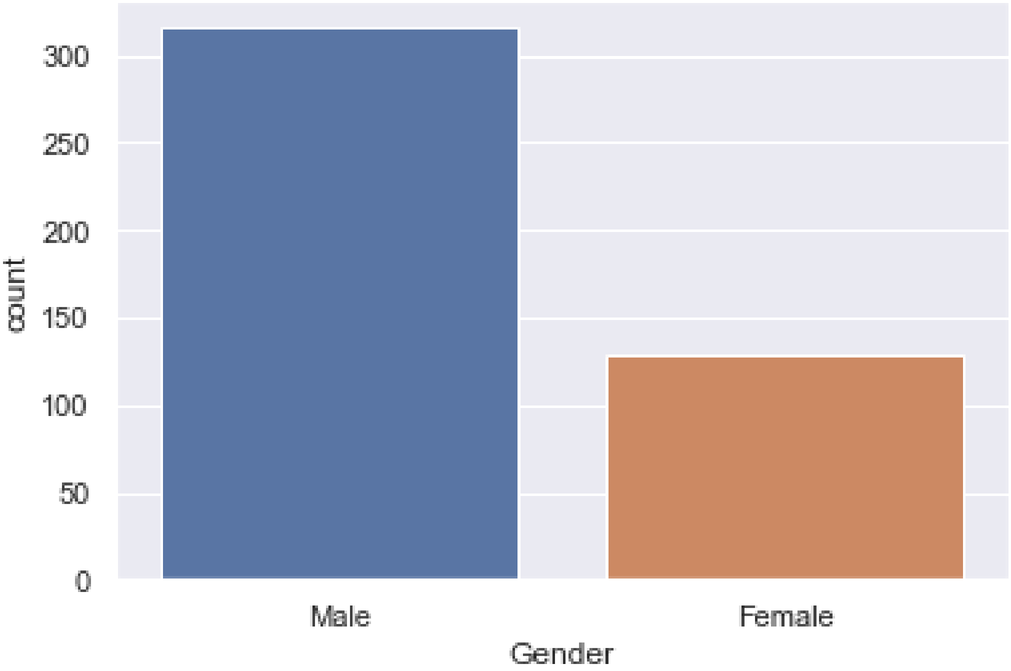


Fig 7: Distribution of Gender

- The count of Employees in ABC Company have more Males(316) compared to Females(218).

2.Distribution of Engineer

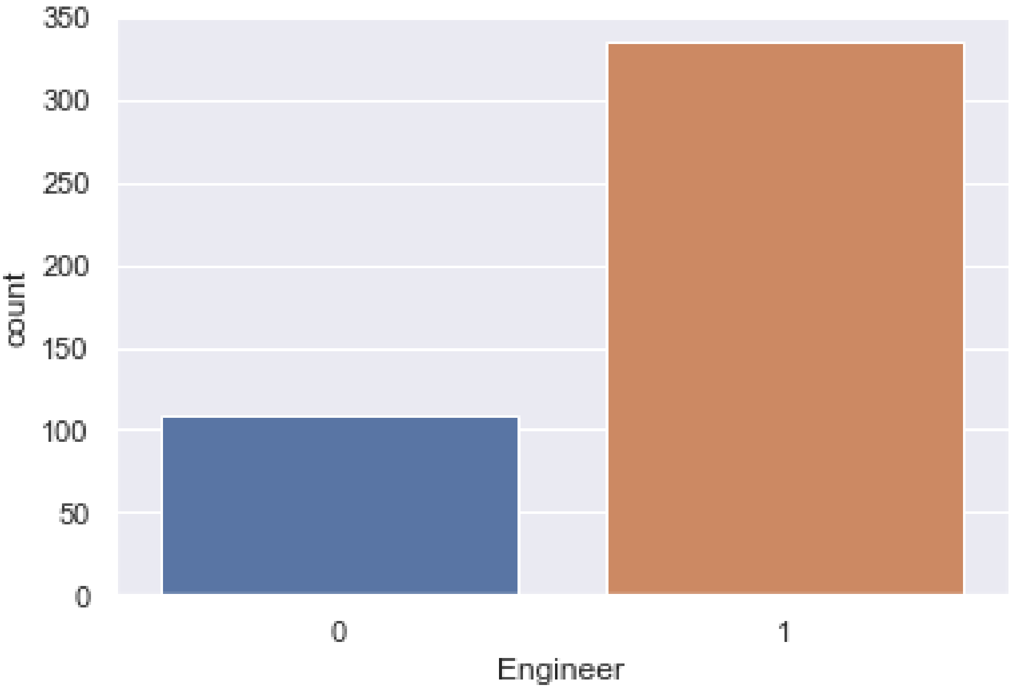


Fig 8: Distribution of Engineer

- The Distribution of Employees who are Engineer in the ABC company is between 0(No) and 1(Yes).
- Total 335 employees who are Engineer

3.Distribution of MBA

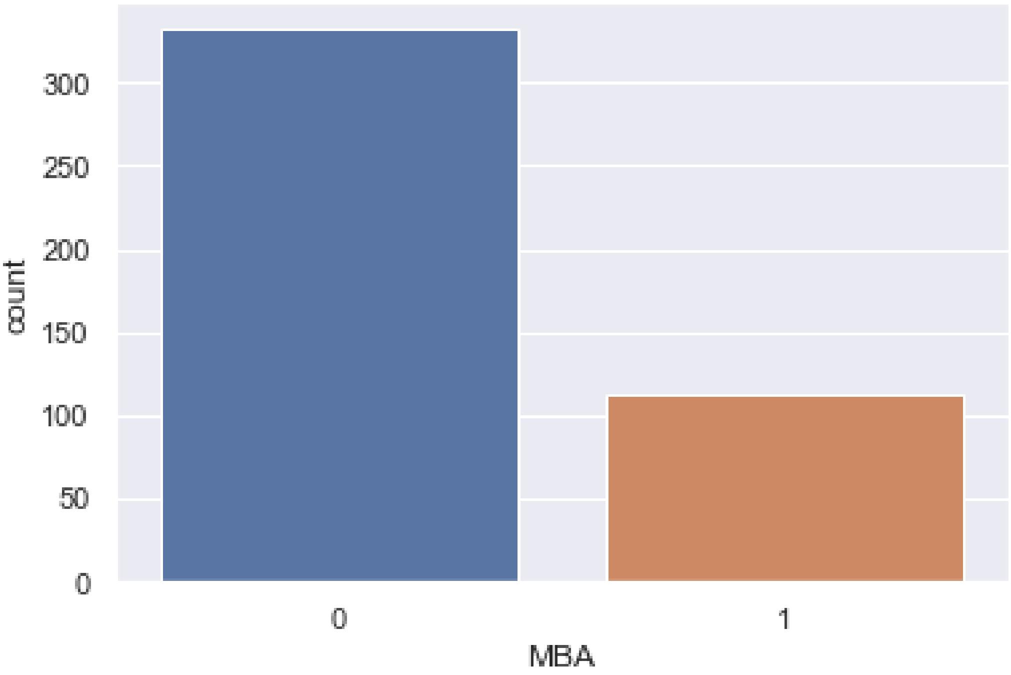


Fig 9: Distribution of MBA

- The Distribution of Employees who are MBA in the ABC company is between 0(No) and 1(Yes).
- Total 112 employees who are MBA.

4.Distribution of license

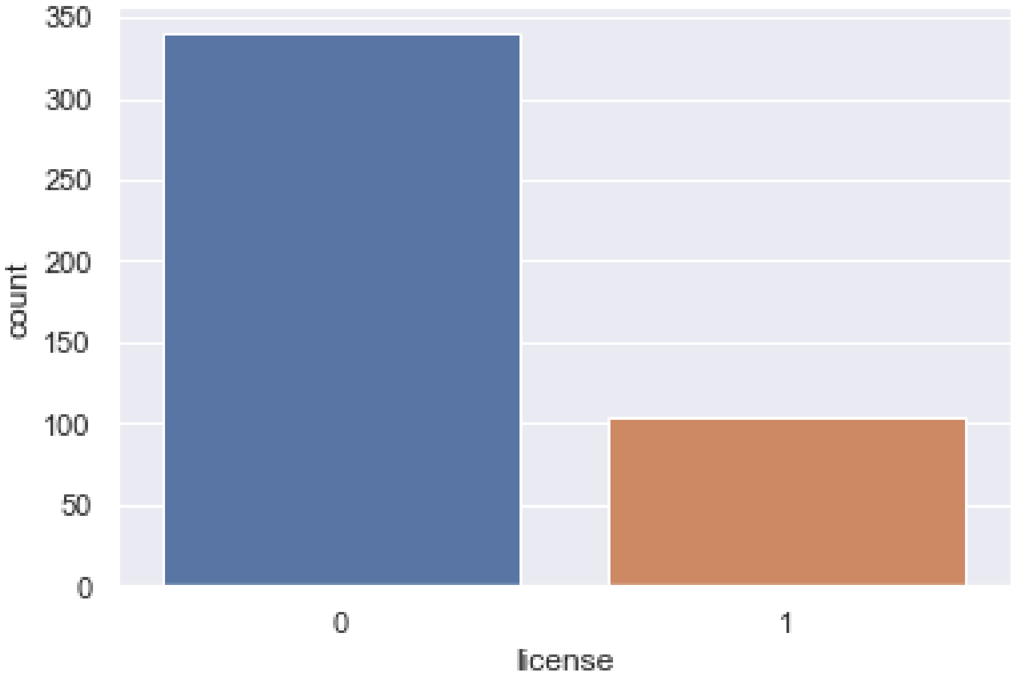


Fig 10: Distribution of License

- The Distribution of Employees who have license in the ABC company is between 0(No) and 1(Yes).
- Total 104 employees who are license.

5.Distribution of Transport

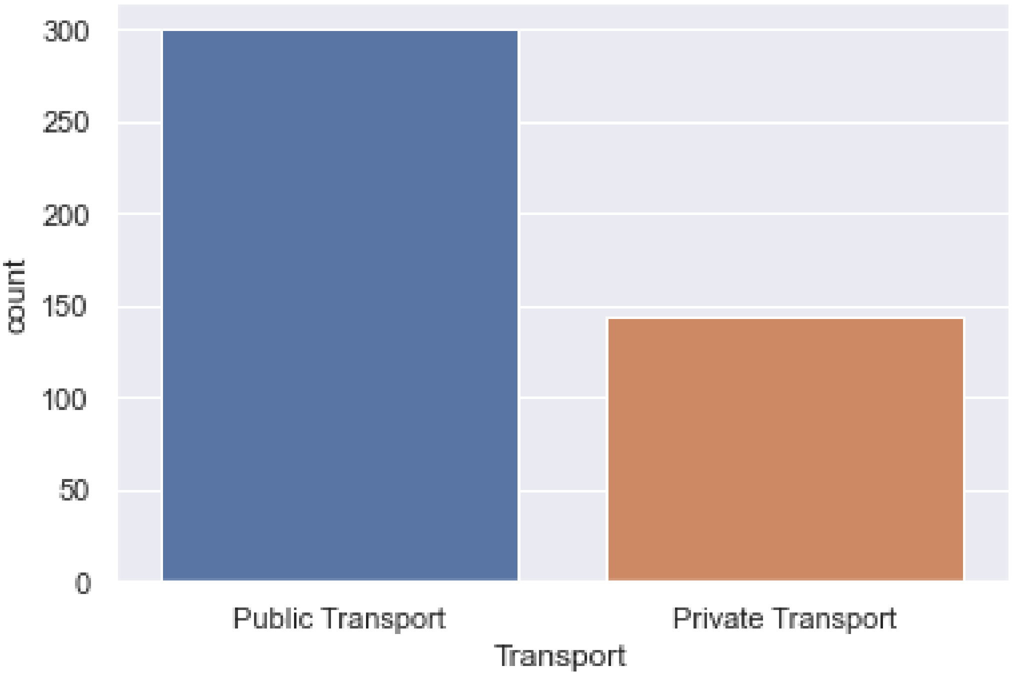


Fig 11: Distribution of Transport

- The count of Employees in ABC Company using Public Transport(300) are more compared to Private Transport (144).

Bi - Variate Analysis

Gender Vs License

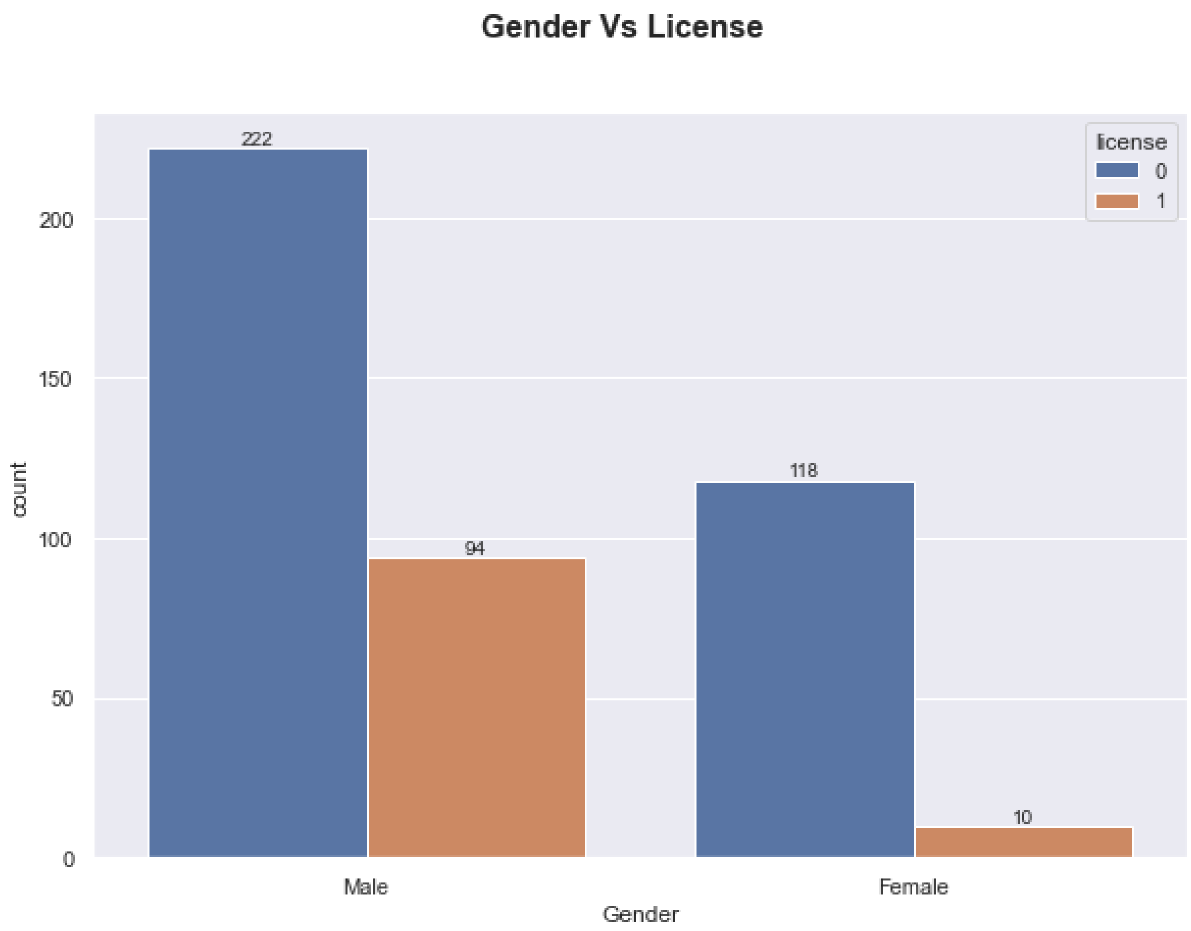


Fig 12: Gender Vs License

- The Count plot clearly states that most of the Employee in ABC Company doesn't have license.
- More no. of Male's(94) have license compared to Female's(10).

Gender Vs Transport

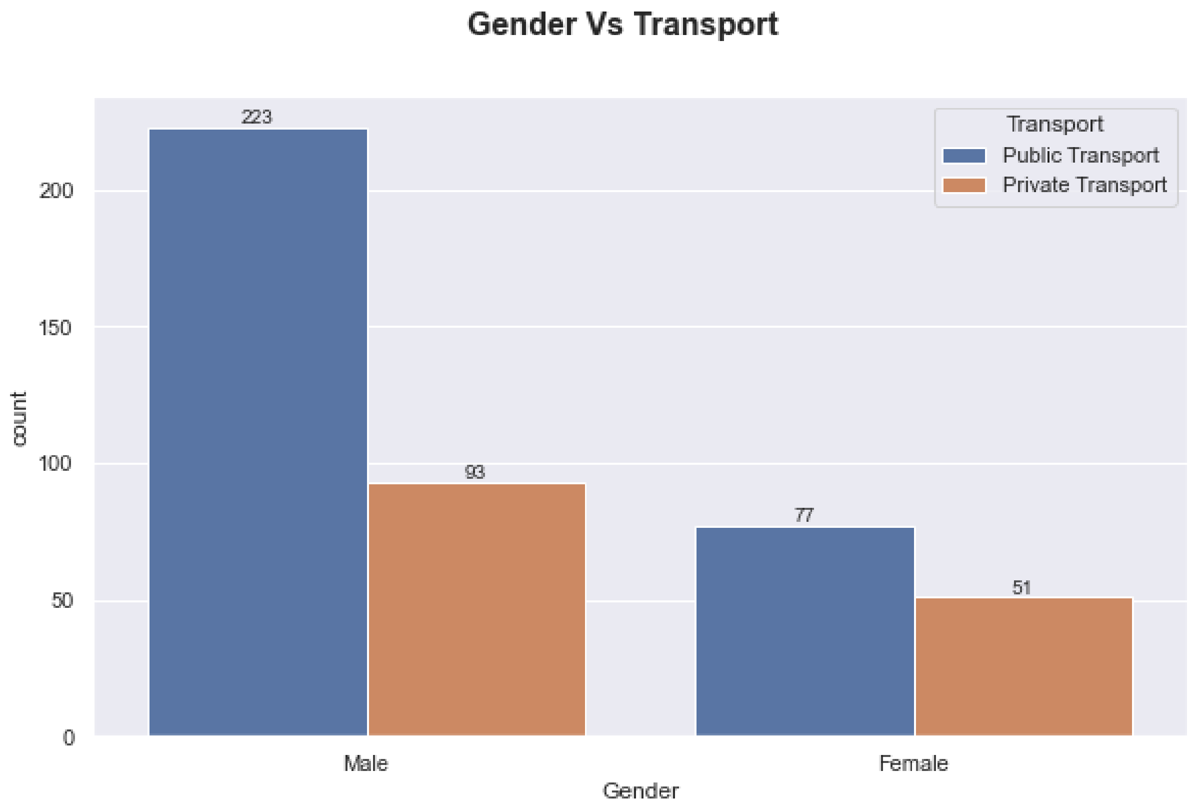


Fig 13: Gender Vs Transport

- Public Transport is the most preferred mode by the ABC Company Employees.
- Male(93) are more compared to Females(51) in Private Transport.
- Their is Imbalance of data between Public and Private Transport in Males (i.e 223:93).

Transport vs license

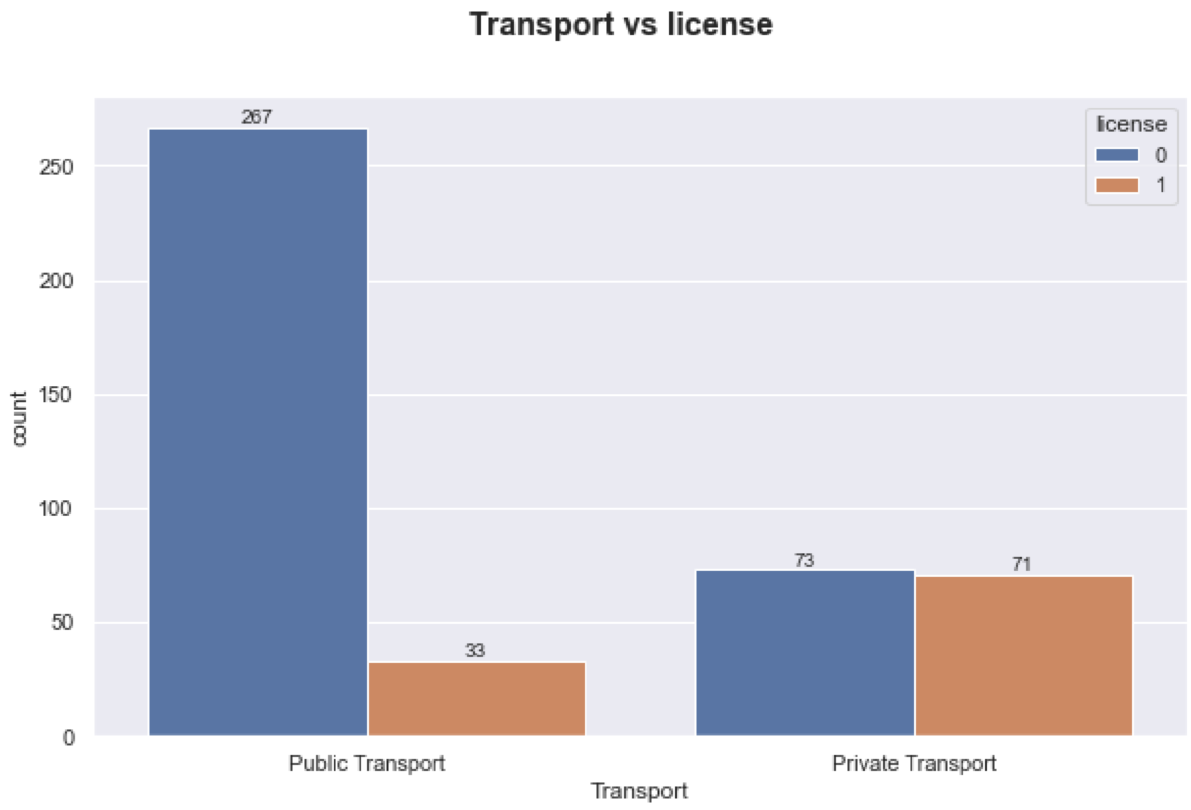


Fig 14: Transport Vs License

- Almost equal balance of having and not having license in the Private Transport.
- We can clearly observe here that there are Employees of ABC Company who prefer Private Transport even if they don't have license.
- There are some Employees who prefer to Public Transport even though they have license. We can assume that these employees live far from office.

Work Exp Vs Salary

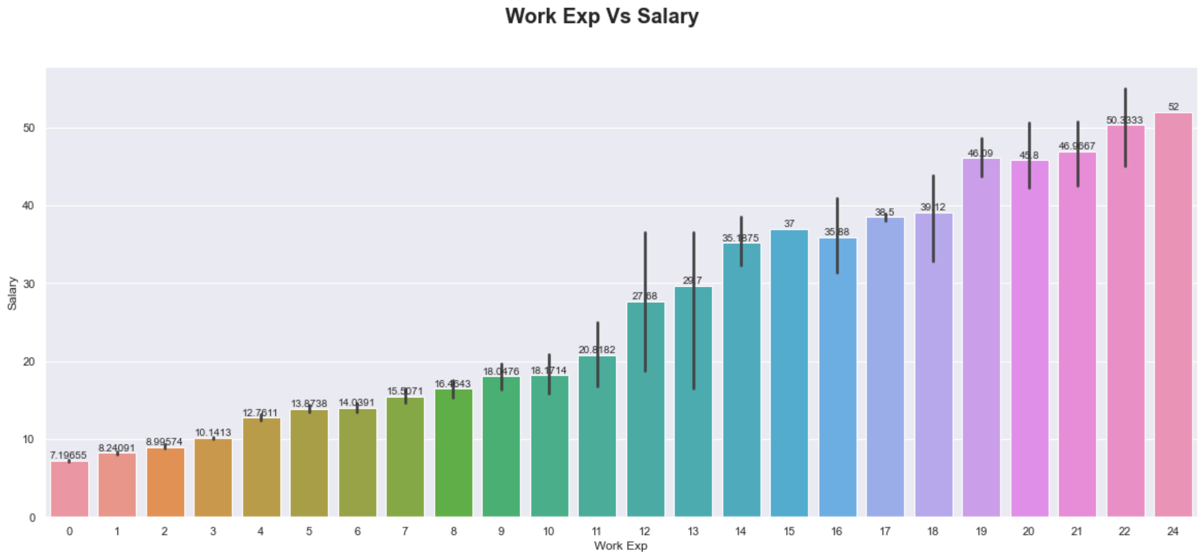


Fig 15: Work Exp Vs Salary

- There is a clear indication that as the no. of years of Work Experience of an Employee increases therefore his Salary also increases simultaneously.

Age Vs Salary

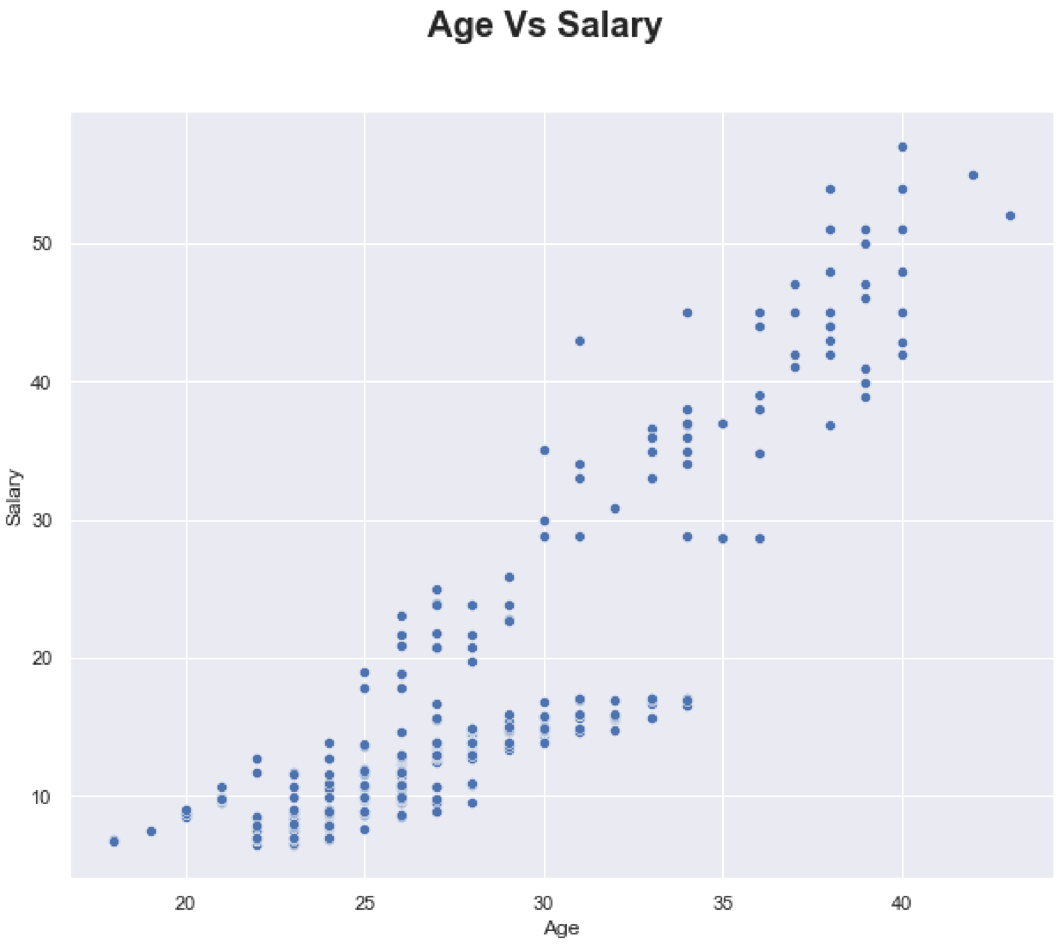


Fig 16: Age Vs Salary

- There is a clear indication that as the years of Age of an Employee increases therefore his Salary also increases simultaneously.

Multi-Variate Analysis

Gender Vs Salary Vs Transport

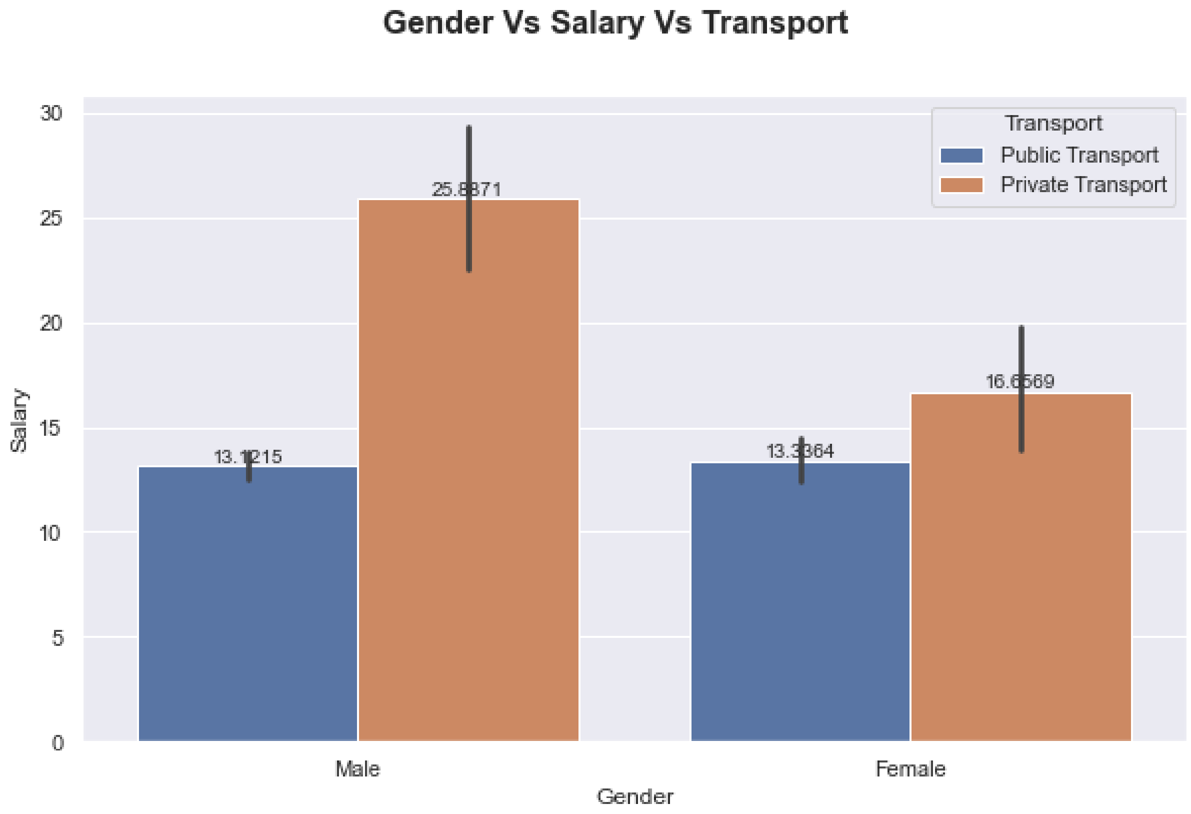


Fig 17: Gender Vs Salary Vs Transport

- The Barplot show that Employees(Male and Female) of salary upto 13 or 14 do prefer Public Transportation but as their salary increases which has increased the standard of living both Male and Female Employees prefer Private Transportation

Transport Vs Age Vs Gender

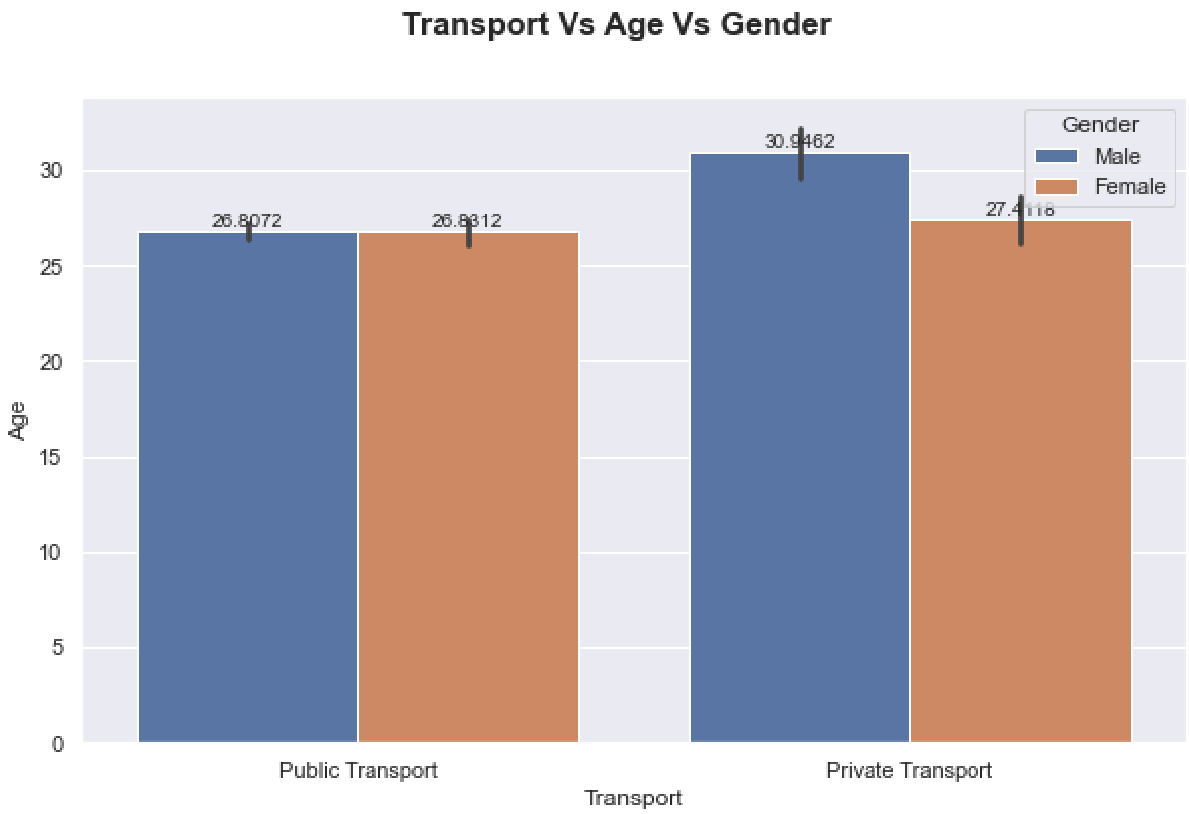


Fig 18: Transport Vs Age Vs Gender

- The Barplot Indicates that both Male and Female Employee upto age of 27 are comfortable with Public and Private Transport
- The dependency of on Private Transport increases for male as their age increases.

Pair Plot of Dataset

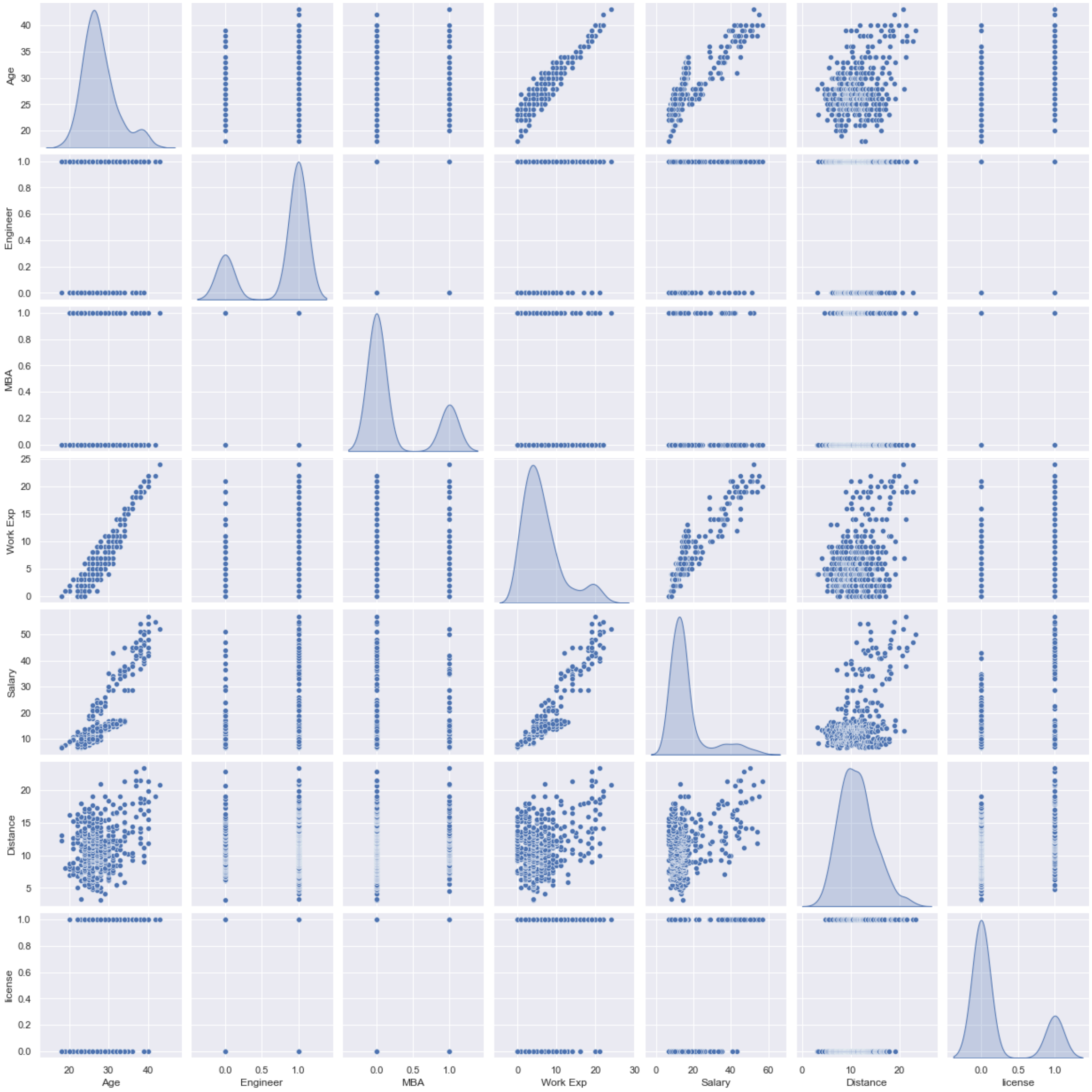
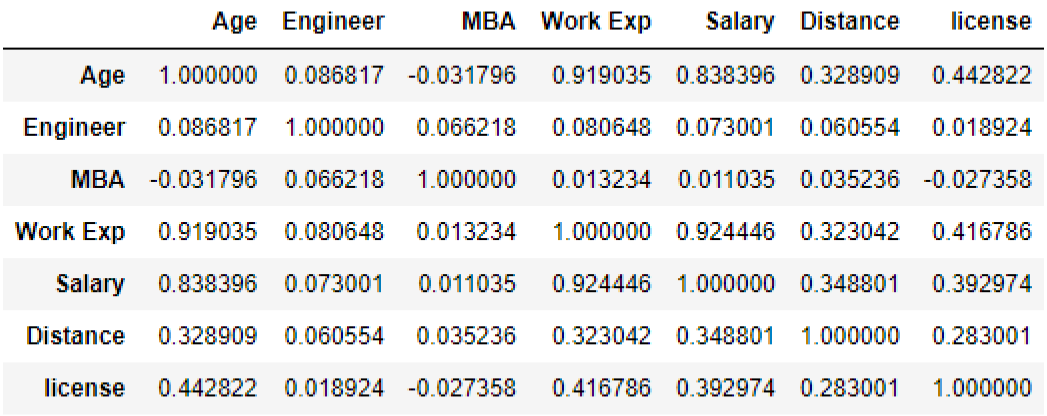


Fig 19: Pair Plot

- Age has high correlation with Work Experience, Salary and Distance.
- Work Exp have high correlation with Salary and Distance.

Correlation Matrix



	Age	Engineer	MBA	Work Exp	Salary	Distance	license
Age	1.000000	0.086817	-0.031796	0.919035	0.838396	0.328909	0.442822
Engineer	0.086817	1.000000	0.066218	0.080648	0.073001	0.060554	0.018924
MBA	-0.031796	0.066218	1.000000	0.013234	0.011035	0.035236	-0.027358
Work Exp	0.919035	0.080648	0.013234	1.000000	0.924446	0.323042	0.416786
Salary	0.838396	0.073001	0.011035	0.924446	1.000000	0.348801	0.392974
Distance	0.328909	0.060554	0.035236	0.323042	0.348801	1.000000	0.283001
license	0.442822	0.018924	-0.027358	0.416786	0.392974	0.283001	1.000000

Table 6: Correlation Matrix Table

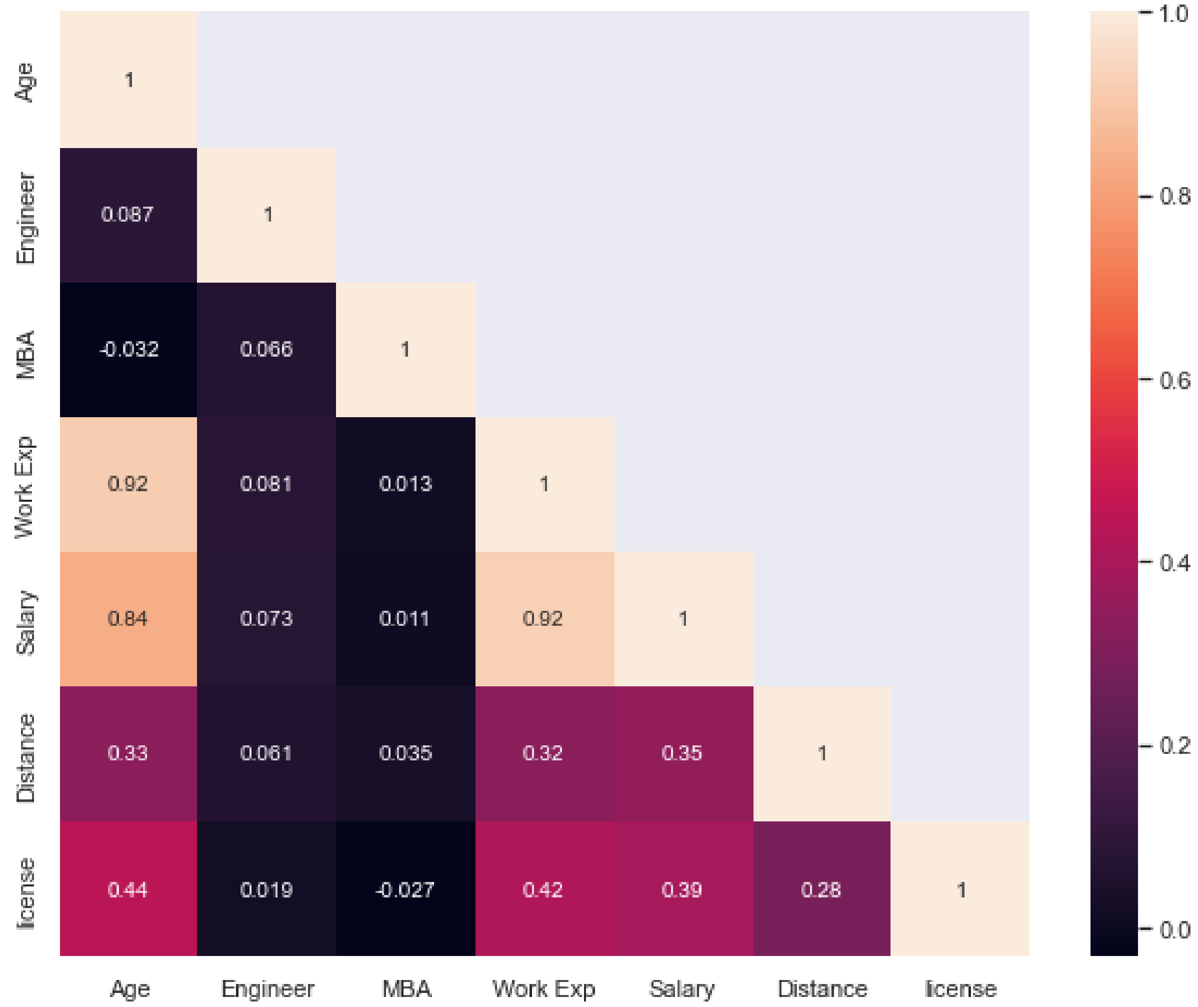


Fig 20: Heat Map

- The Heatmap clear indicates that their is high positive correlation between Age and Salary, also Age and Work Exp.
- There is high positive correlation between Work Exp and Salary

Data Preparation:

3. Encode the data (having string values) for Modelling. Is Scaling necessary here or not? Data Split: Split the data into train and test (70:30).

Yes, Scaling of data is necessary for K-NN. KNN which uses Euclidean distance is one such algorithm which essentially require scaling. In KNN if one of the feature has a broad range of values, the distance is governed by this (i.e. **Salary**) particular feature.

As we can see from the **Salary, Age and Distance** are in different values and this may get more weightage.

Scaling will help keep the values in relatively same range.

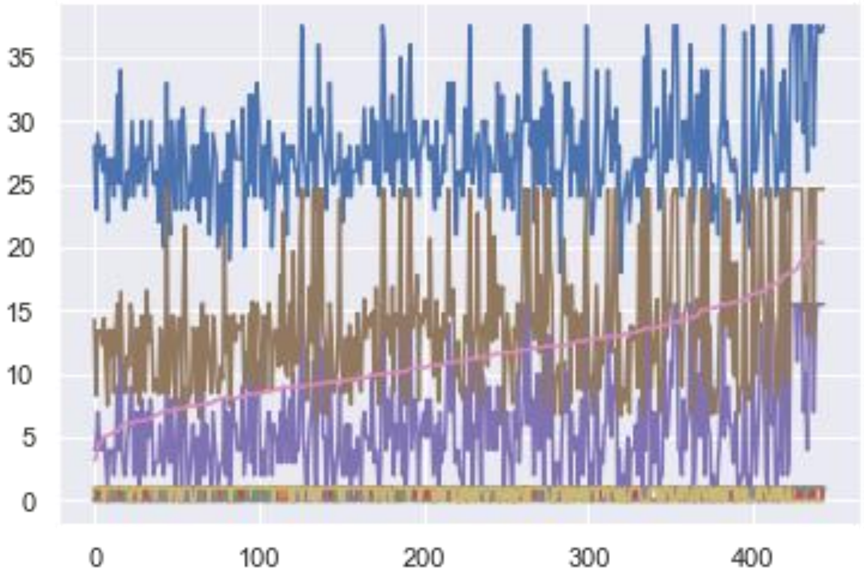


Fig 21: Data before Scaling

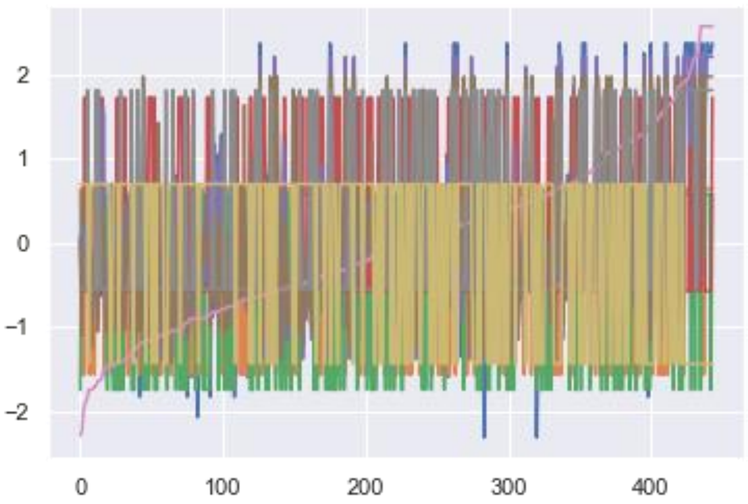
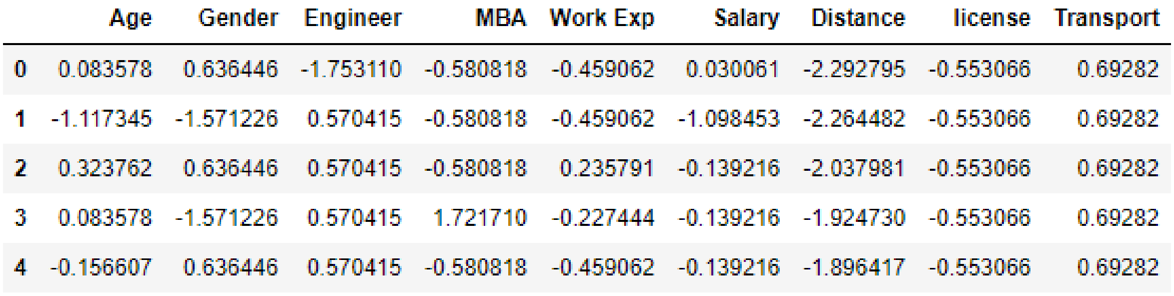


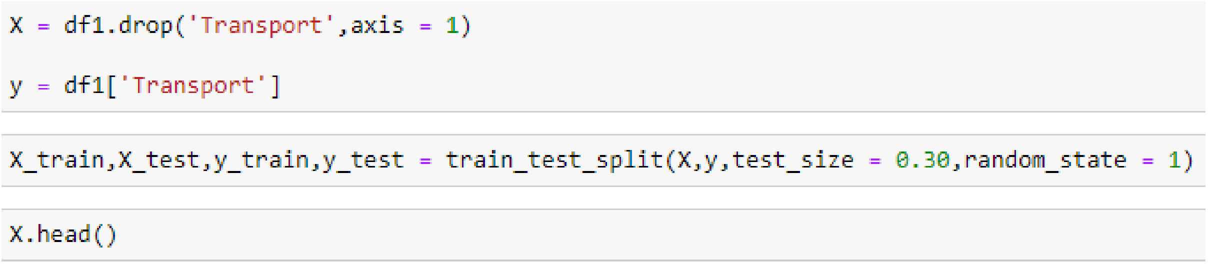
Fig 22: Data after Scaling



	Age	Gender	Engineer	MBA	Work Exp	Salary	Distance	license	Transport
0	0.083578	0.636446	-1.753110	-0.580818	-0.459062	0.030061	-2.292795	-0.553066	0.69282
1	-1.117345	-1.571226	0.570415	-0.580818	-0.459062	-1.098453	-2.264482	-0.553066	0.69282
2	0.323762	0.636446	0.570415	-0.580818	0.235791	-0.139216	-2.037981	-0.553066	0.69282
3	0.083578	-1.571226	0.570415	1.721710	-0.227444	-0.139216	-1.924730	-0.553066	0.69282
4	-0.156607	0.636446	0.570415	-0.580818	-0.459062	-0.139216	-1.896417	-0.553066	0.69282

Table 7: Scaled Table

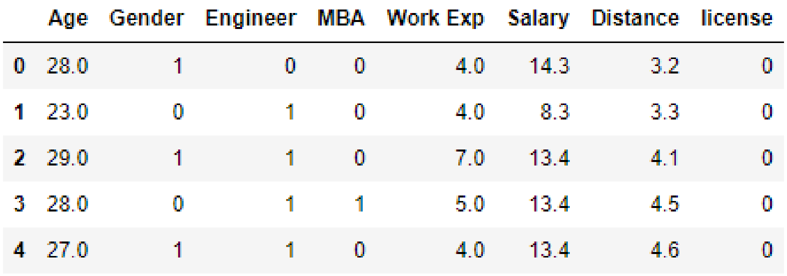
Split the data into train and test (70:30)



```
X = df1.drop('Transport',axis = 1)

y = df1['Transport']
```

```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.30,random_state = 1)
```

```
X.head()
```



	Age	Gender	Engineer	MBA	Work Exp	Salary	Distance	license
0	28.0	1	0	0	4.0	14.3	3.2	0
1	23.0	0	1	0	4.0	8.3	3.3	0
2	29.0	1	1	0	7.0	13.4	4.1	0
3	28.0	0	1	1	5.0	13.4	4.5	0
4	27.0	1	1	0	4.0	13.4	4.6	0

Modelling:

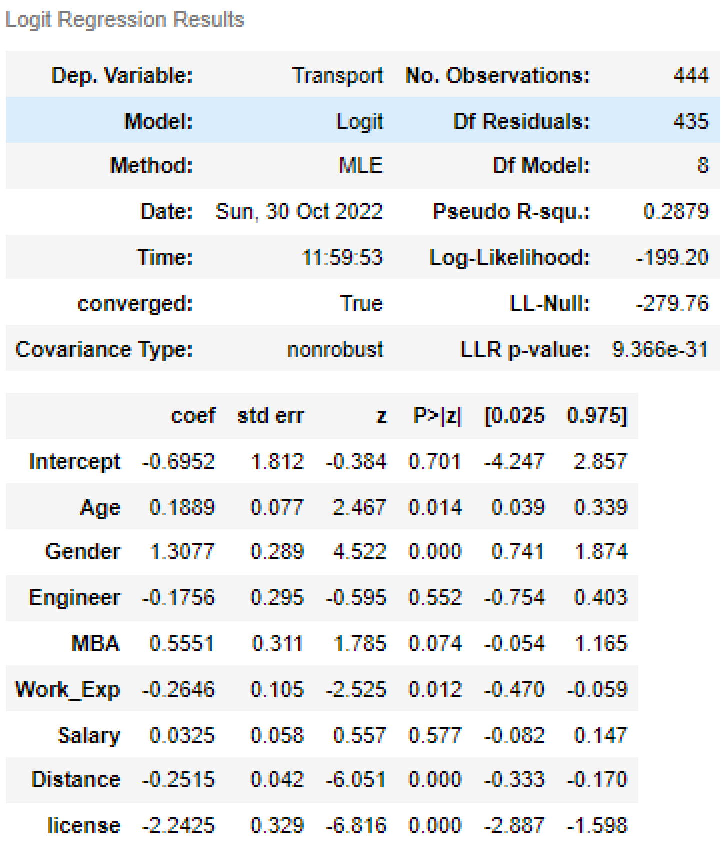
4. Apply Logistic Regression.

Logistic Regression VIF Values:

```
Age VIF =  6.8
Gender VIF =  1.08
Engineer VIF =  1.01
MBA VIF =  1.03
Work Exp VIF =  13.38
Salary VIF =  7.09
Distance VIF =  1.18
license VIF =  1.35
```

We can consider a rule of thumb that if vif is greater than 5, we can choose to drop the variable as there can be a problem of multicollinearity. This essentially means that we can choose to drop a predictor variable whose 80% variation is being explained by the other predictor variables.

Model 1



Logit Regression Results

Dep. Variable:	Transport	No. Observations:	444
Model:	Logit	Df Residuals:	435
Method:	MLE	Df Model:	8
Date:	Sun, 30 Oct 2022	Pseudo R-squ.:	0.2879
Time:	11:59:53	Log-Likelihood:	-199.20
converged:	True	LL-Null:	-279.76
Covariance Type:	nonrobust	LLR p-value:	9.366e-31

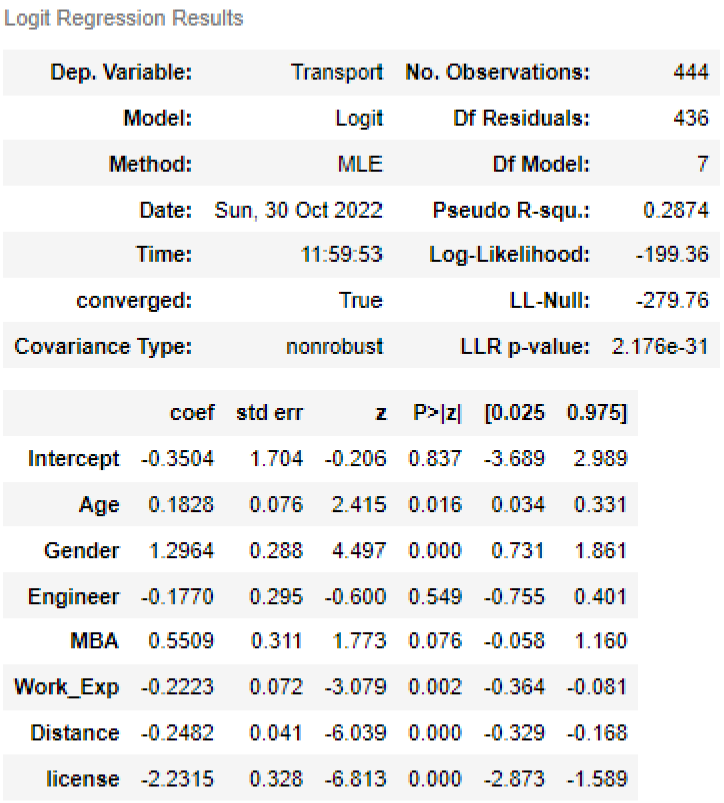
	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-0.6952	1.812	-0.384	0.701	-4.247	2.857
Age	0.1889	0.077	2.467	0.014	0.039	0.339
Gender	1.3077	0.289	4.522	0.000	0.741	1.874
Engineer	-0.1756	0.295	-0.595	0.552	-0.754	0.403
MBA	0.5551	0.311	1.785	0.074	-0.054	1.165
Work_Exp	-0.2646	0.105	-2.525	0.012	-0.470	-0.059
Salary	0.0325	0.058	0.557	0.577	-0.082	0.147
Distance	-0.2515	0.042	-6.051	0.000	-0.333	-0.170
license	-2.2425	0.329	-6.816	0.000	-2.887	-1.598

Table 8: Logistic Regression Model 1

Observation Logistic Regression Model 1:

- We can clearly see that Salary has highest p-value (0.557) thus it is greater than 0.05
- Salary has VIF = 7.09 which means it has Multi-collinearity
- Hence it confirms that Salary attribute is not significant
- The adjusted pseudo R-square value is 0.2593510805530256

Model 2 (Dropping 'Salary ')



Logit Regression Results

Dep. Variable:	Transport	No. Observations:	444
Model:	Logit	Df Residuals:	436
Method:	MLE	Df Model:	7
Date:	Sun, 30 Oct 2022	Pseudo R-squ.:	0.2874
Time:	11:59:53	Log-Likelihood:	-199.36
converged:	True	LL-Null:	-279.76
Covariance Type:	nonrobust	LLR p-value:	2.176e-31

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-0.3504	1.704	-0.206	0.837	-3.689	2.989
Age	0.1828	0.076	2.415	0.016	0.034	0.331
Gender	1.2964	0.288	4.497	0.000	0.731	1.861
Engineer	-0.1770	0.295	-0.600	0.549	-0.755	0.401
MBA	0.5509	0.311	1.773	0.076	-0.058	1.160
Work_Exp	-0.2223	0.072	-3.079	0.002	-0.364	-0.081
Distance	-0.2482	0.041	-6.039	0.000	-0.329	-0.168
license	-2.2315	0.328	-6.813	0.000	-2.873	-1.589

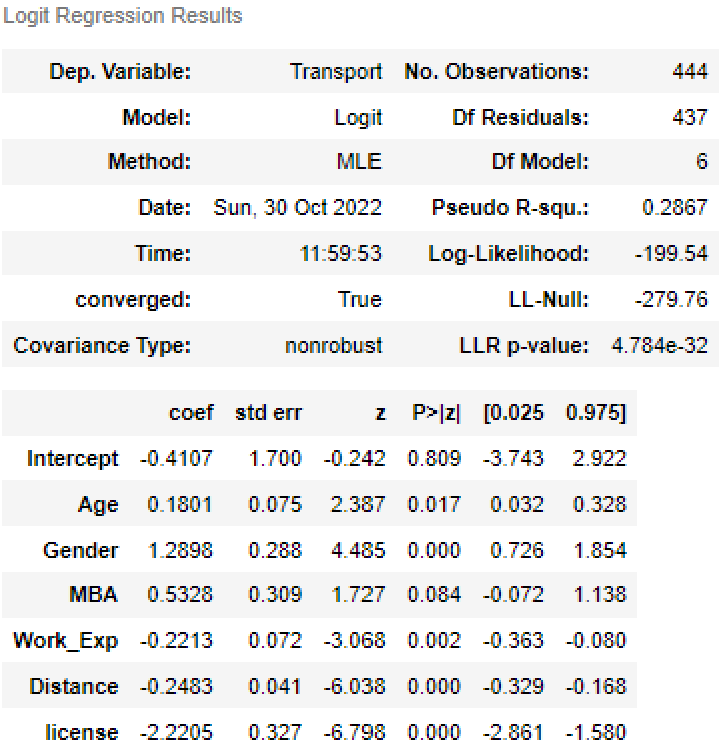
Table 9: Logistic Regression Model 2(Dropping Salary)

Observation Logistic Regression Model 2:

- We can clearly see that Engineer has highest p-value (0.549) thus it is greater than 0.05

- Hence it confirms that Engineer attribute is not significant
- The adjusted pseudo R-square value is 0.262370680495439

Model 3 (Dropping 'Engineer ')



Logit Regression Results

Dep. Variable:	Transport	**No. Observations:**	444
Model:	Logit	**Df Residuals:**	437
Method:	MLE	**Df Model:**	6
Date:	Sun, 30 Oct 2022	**Pseudo R-squ.:**	0.2867
Time:	11:59:53	**Log-Likelihood:**	-199.54
converged:	True	**LL-Null:**	-279.76
Covariance Type:	nonrobust	**LLR p-value:**	4.784e-32

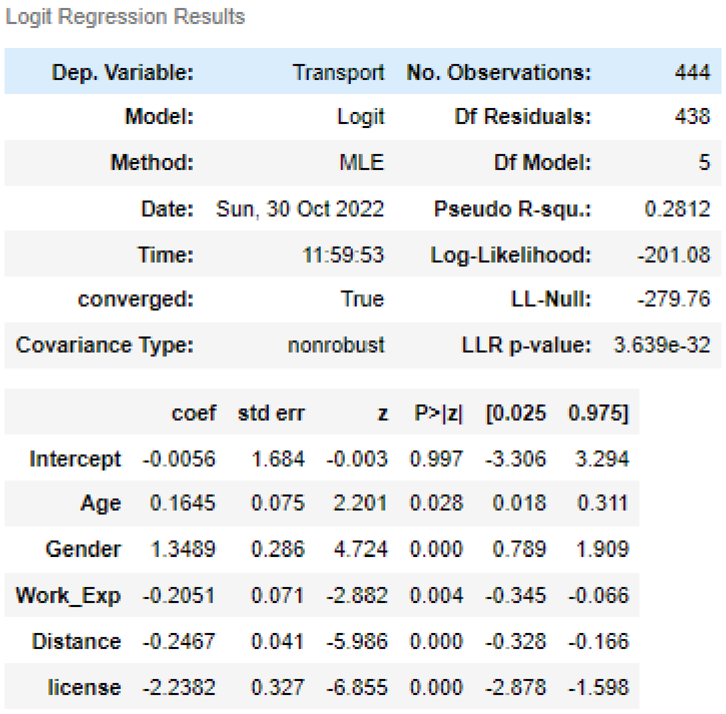
	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
Intercept	-0.4107	1.700	-0.242	0.809	-3.743	2.922
Age	0.1801	0.075	2.387	0.017	0.032	0.328
Gender	1.2898	0.288	4.485	0.000	0.726	1.854
MBA	0.5328	0.309	1.727	0.084	-0.072	1.138
Work_Exp	-0.2213	0.072	-3.068	0.002	-0.363	-0.080
Distance	-0.2483	0.041	-6.038	0.000	-0.329	-0.168
license	-2.2205	0.327	-6.798	0.000	-2.861	-1.580

Table 10: Logistic Regression Model 3(Dropping Engineer)

Observation Logistic Regression Model 3:

- We can clearly see that MBA has highest p-value (0.084) thus it is greater than 0.05
- Hence it confirms that MBA attribute is not significant
- The adjusted pseudo R-square value is 0.26529616067759454

Model 4 (Dropping 'MBA ')



Logit Regression Results

Dep. Variable:	Transport	No. Observations:	444
Model:	Logit	Df Residuals:	438
Method:	MLE	Df Model:	5
Date:	Sun, 30 Oct 2022	Pseudo R-squ.:	0.2812
Time:	11:59:53	Log-Likelihood:	-201.08
converged:	True	LL-Null:	-279.76
Covariance Type:	nonrobust	LLR p-value:	3.639e-32

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-0.0056	1.684	-0.003	0.997	-3.306	3.294
Age	0.1645	0.075	2.201	0.028	0.018	0.311
Gender	1.3489	0.286	4.724	0.000	0.789	1.909
Work_Exp	-0.2051	0.071	-2.882	0.004	-0.345	-0.066
Distance	-0.2467	0.041	-5.986	0.000	-0.328	-0.166
license	-2.2382	0.327	-6.855	0.000	-2.878	-1.598

Table 11: Logistic Regression Model 3(Dropping MBA)

Observation Logistic Regression Model 4:

- Now all p values are less than 0.05. Hence all these attributes and their coefficients have importance in deciding the target variable Transport.
- Also, we can see that co-ef value is highest for license followed by Gender, Distance, Work_Exp and Age.
- 'p values' indicate that all the variable are significant at 95% confidence level
- The adjusted pseudo R-square value is 0.26334633603351165

Summary of Logistic Regression :

Logistic regression equation is as shown below :-

Log(odd) = (-0.01)+ (0.16)Age + (1.35)Gender + (-0.21)Work_Exp + (-0.25)Distance + (-2.24)license

Most important attribute here is license followed by Distance,Gender, Work_Exp and Age

Prediction on Data:

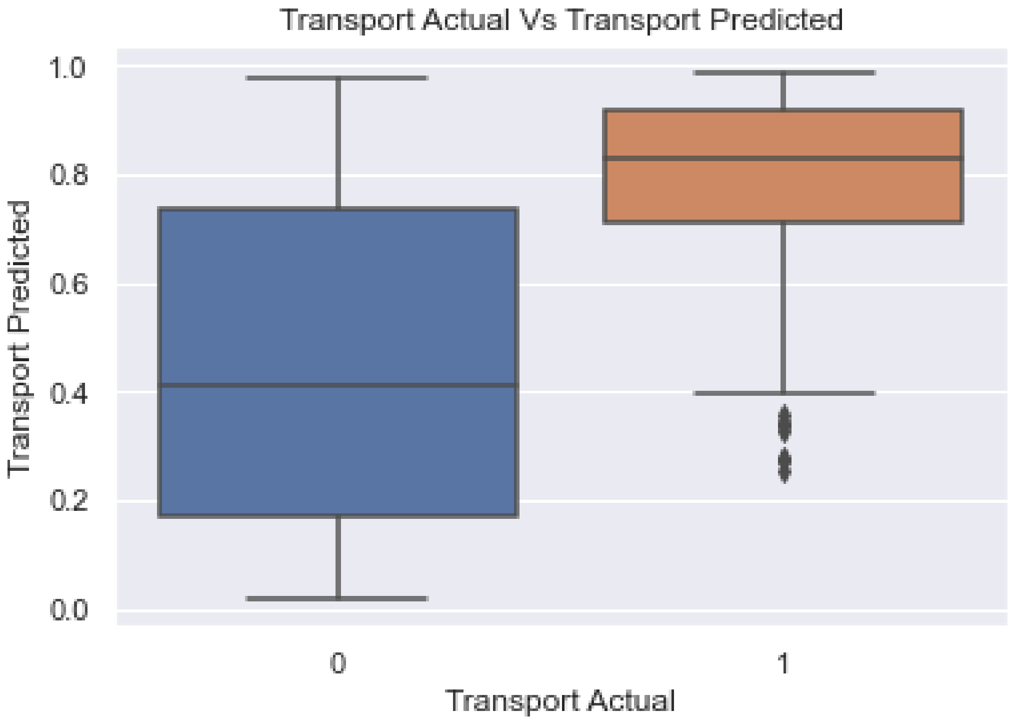
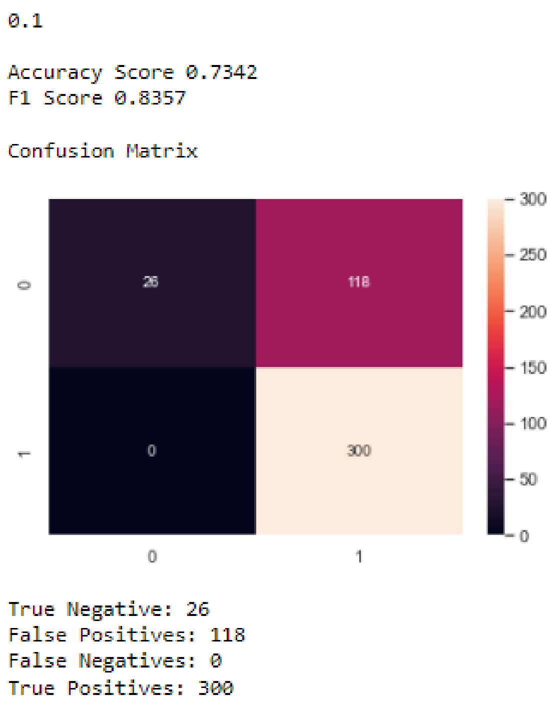


Fig 23: Boxplot of Transport Actual Vs Predicted

Observation:

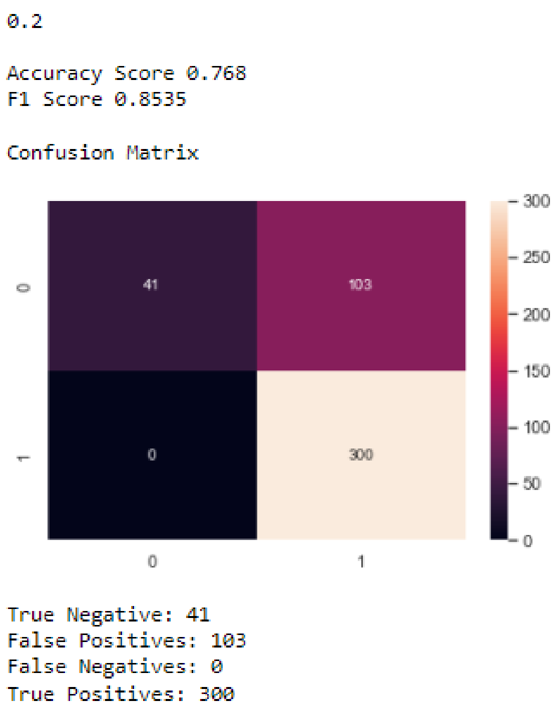
From the above boxplot, we need to decide on one such value of a cut-off which gives most reasonable power of the model

Choosing a different cut-off method for the predictions on the Probability Predictions Data



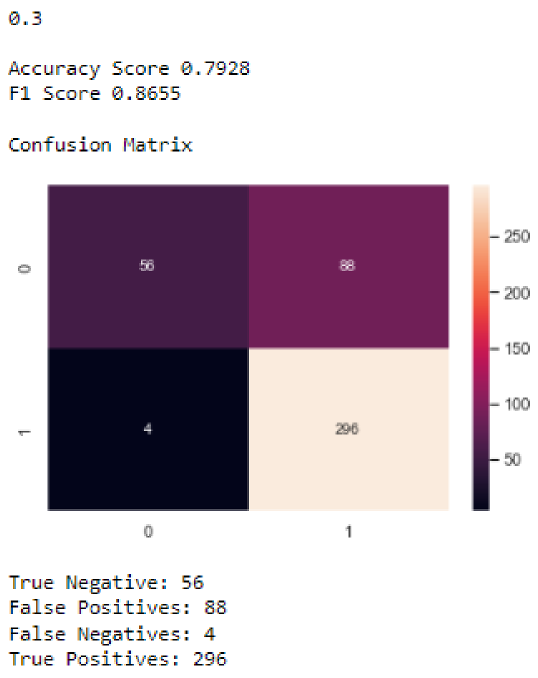
```
0.1

Accuracy Score 0.7342
F1 Score 0.8357

Confusion Matrix
```

```
True Negative: 26
False Positives: 118
False Negatives: 0
True Positives: 300
```



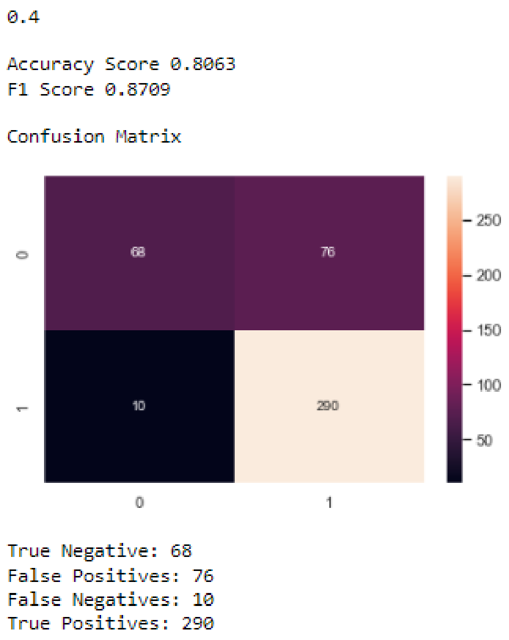
```
0.2

Accuracy Score 0.768
F1 Score 0.8535

Confusion Matrix
```

```
True Negative: 41
False Positives: 103
False Negatives: 0
True Positives: 300
```



```
0.3

Accuracy Score 0.7928
F1 Score 0.8655

Confusion Matrix
```

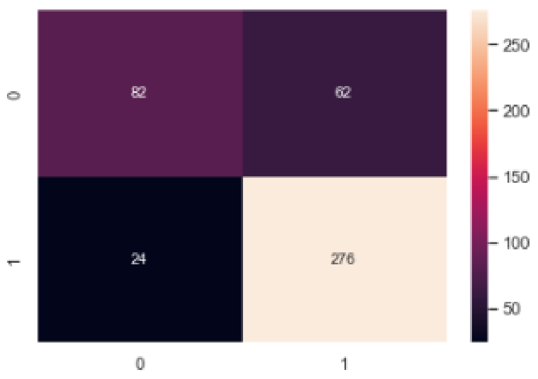
```
True Negative: 56
False Positives: 88
False Negatives: 4
True Positives: 296
```



```
0.4

Accuracy Score 0.8063
F1 Score 0.8709

Confusion Matrix
```

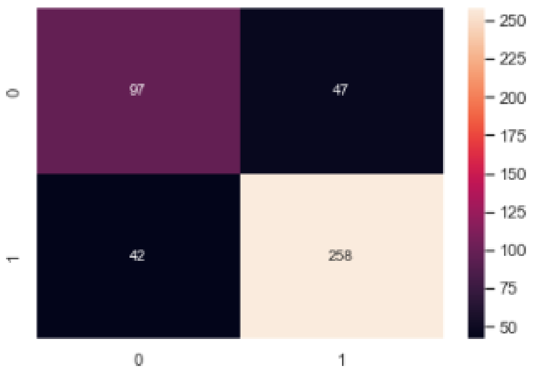
```
True Negative: 68
False Positives: 76
False Negatives: 10
True Positives: 290
```

```
0.5

Accuracy Score 0.8063
F1 Score 0.8652

Confusion Matrix
```



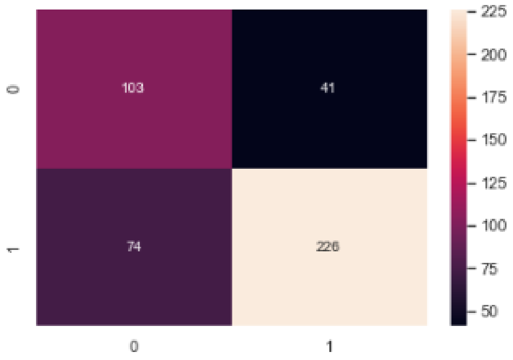
```
True Negative: 82
False Positives: 62
False Negatives: 24
True Positives: 276
```

```
0.6

Accuracy Score 0.7995
F1 Score 0.8529

Confusion Matrix
```



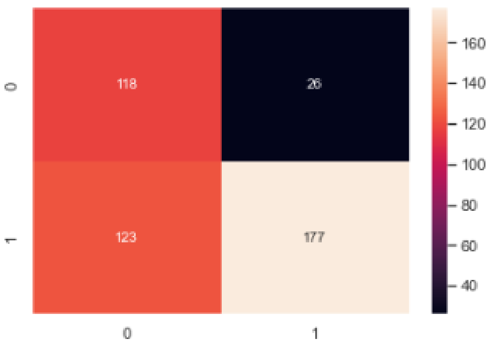
```
True Negative: 97
False Positives: 47
False Negatives: 42
True Positives: 258
```

```
0.7

Accuracy Score 0.741
F1 Score 0.7972

Confusion Matrix
```

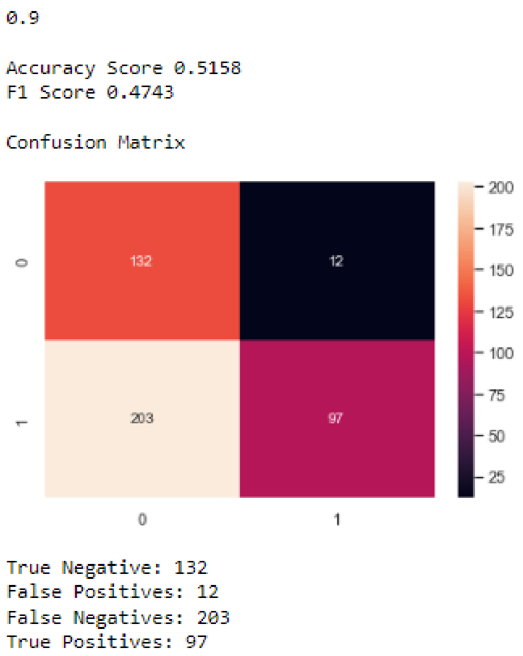


```
True Negative: 103
False Positives: 41
False Negatives: 74
True Positives: 226
```

```
0.8

Accuracy Score 0.6644
F1 Score 0.7038

Confusion Matrix
```



```
True Negative: 118
False Positives: 26
False Negatives: 123
True Positives: 177
```



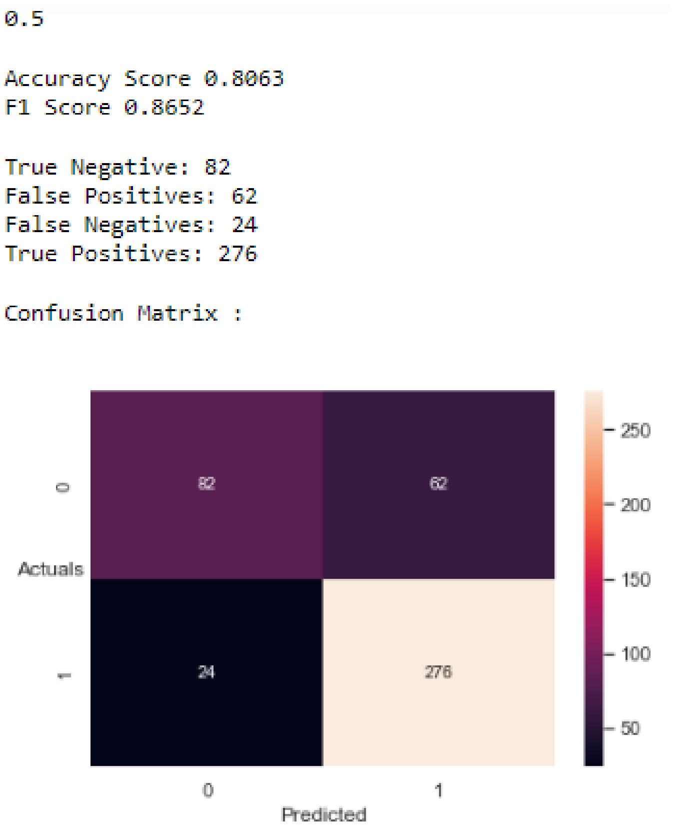
```
0.9

Accuracy Score 0.5158
F1 Score 0.4743

Confusion Matrix
```

```
True Negative: 132
False Positives: 12
False Negatives: 203
True Positives: 97
```

Boxplot's at different Cut-offs

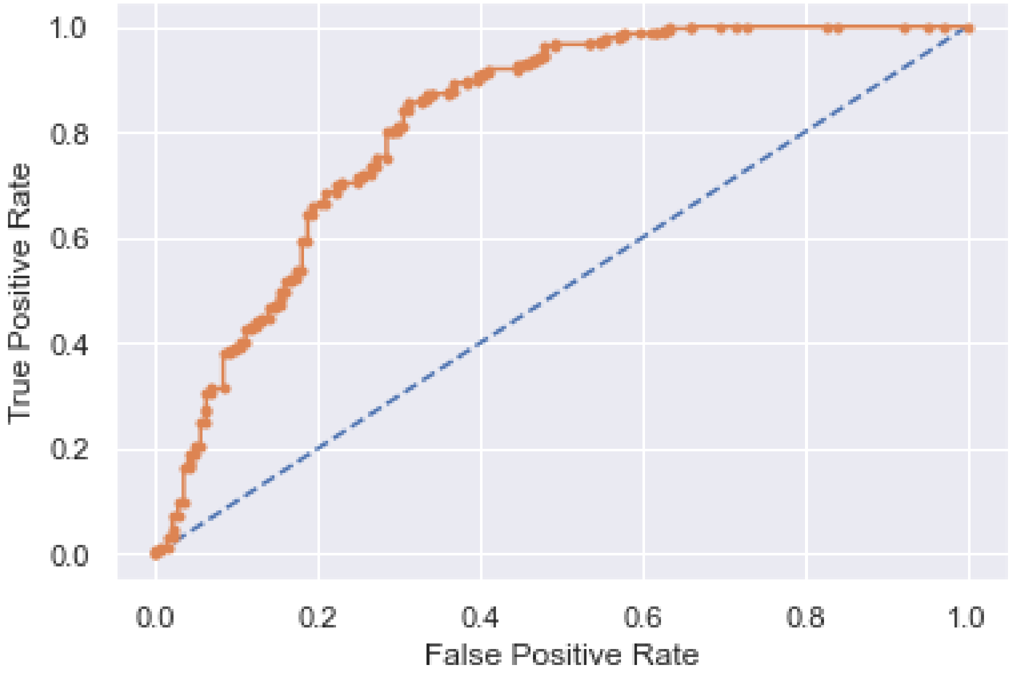
Let us take a cut-off at 0.5 and check power of the model which looks balanced with very good scores

Model Evaluation on Data at cut-off 0.5 using Confusion Matrix heatmap and AUC-ROC curve



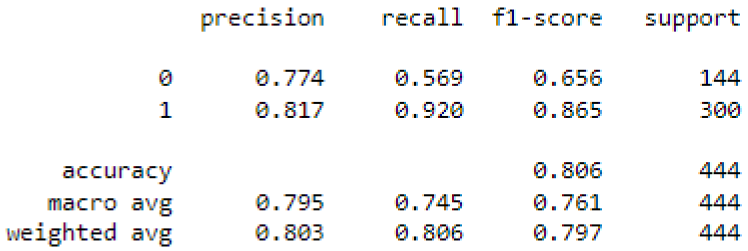
```
0.5

Accuracy Score 0.8063
F1 Score 0.8652

True Negative: 82
False Positives: 62
False Negatives: 24
True Positives: 276

Confusion Matrix :
```

Boxplot's at 0.5 Cut-offs



AUC -ROC Curve at 0.5 Cut-offs

Classification report at cut-off 0.05



```
              precision    recall  f1-score   support

           0      0.774     0.569     0.656       144
           1      0.817     0.920     0.865       300

    accuracy                          0.806       444
   macro avg      0.795     0.745     0.761       444
weighted avg      0.803     0.806     0.797       444
```

With accuracy of 80.6% and recall rate of 92%, model is able to predict 92% of Public Transport which were actually claimed as claimed.

Precision is 81.7% of data which means, out of total employees predicted by model as opt for Public Transport, 81.7% employees actually opted for the Public Transport.

F1-score is the harmonic mean of precision and recall, it takes into the effect of both the scores and this value is low if any of these 2 values is low.

Since we are building a model to predict if whether employee will opt for Public Transport or Private Transport, for practical purposes, we will be more interested in correctly classifying 1 (Public Transport) than 0(Private Transport).

If a employee Private Transport is incorrectly predicted to be "Public Transport" by the model, then the impact on cost for the travel company would be bare minimum. But if an employee opted for Public Transport is incorrectly predicted to be Private Transport by the model, then the cost impact would be very high for the Transport company. Its a loss of potential lead for the company. Hence recall rate (actual data point identified as True by model) is very important in this scenario.

As Recall rate of test dataset is very good around 92% plus Precision rate is also 81.7% for opting Public transport thus this does looks good enough for classification

Thus our Transport Company can target these Employee's for providing services.

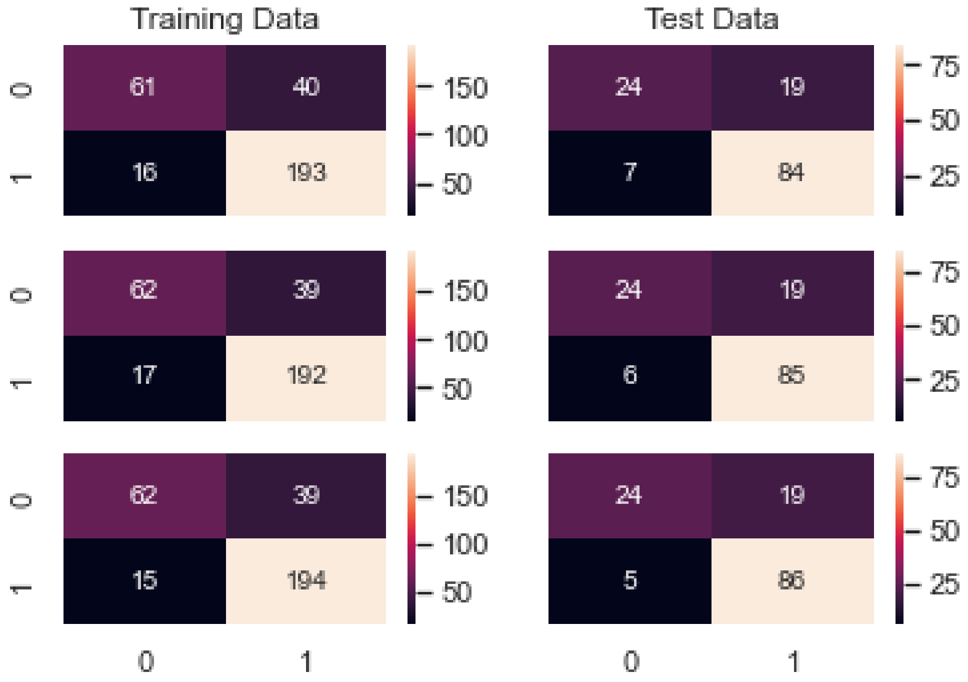
Only for Predictive purposes of Logistic Regression

If we only wanted to predict using Logistic Regression and was not looking for the model building aspect of it, we can do that as well. For this, we will use the same variables as of Model 2, Model 3 and Model 4.

First we will split the data into train and test. We will build the model on the training data and check the the model performance metrics on the test data.

We will take the default cut-off of 0.5 to get the class predictions.

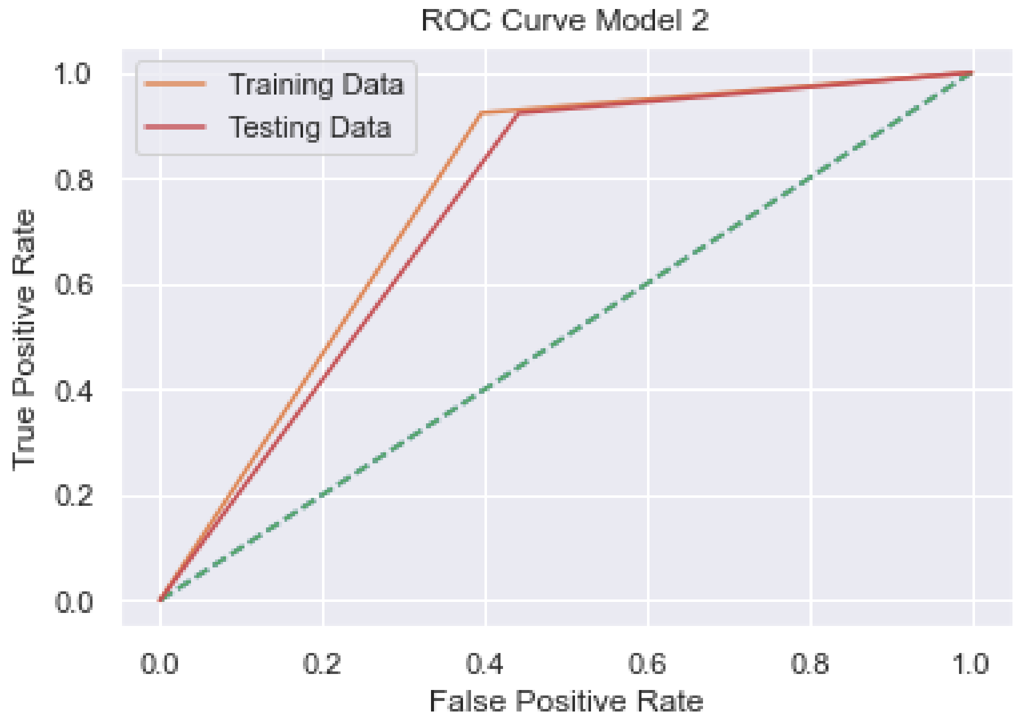
Confusion Matrix check for all the models built Train and Test



Confusion Matrix of all model Train and Test

Model 2 (at Cut off 0.5)

AUC and ROC for the Training and Testing data Model 2



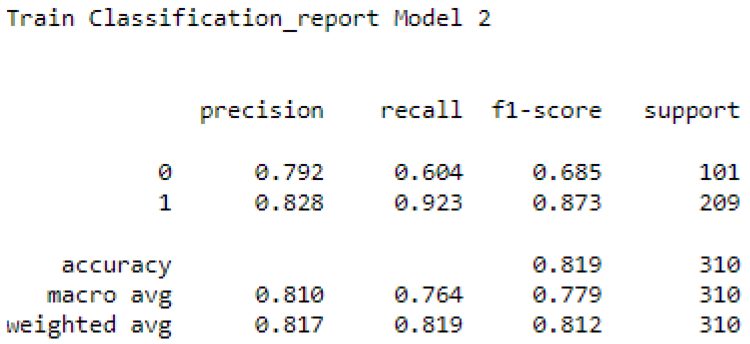
AUC ROC Curve Model 2 Train and Test

Observation AUC for Model 2:

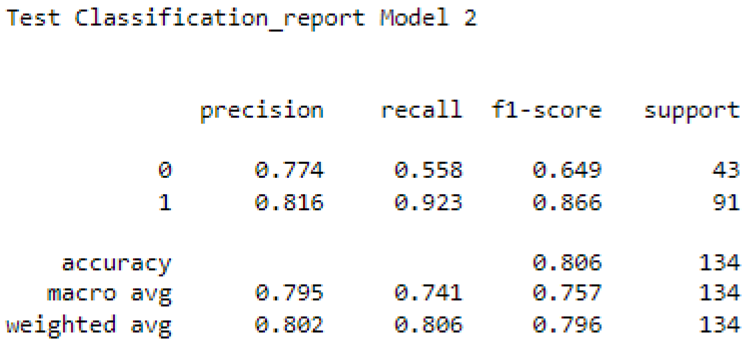
AUC for Train dataset: 0.764

AUC for test dataset: 0.741

Classification_report Model 2



```
Train Classification_report Model 2

              precision    recall  f1-score   support

           0      0.792     0.604     0.685       101
           1      0.828     0.923     0.873       209

    accuracy                          0.819       310
   macro avg      0.810     0.764     0.779       310
weighted avg      0.817     0.819     0.812       310
```



```
Test Classification_report Model 2

              precision    recall  f1-score   support

           0      0.774     0.558     0.649        43
           1      0.816     0.923     0.866        91

    accuracy                          0.806       134
   macro avg      0.795     0.741     0.757       134
weighted avg      0.802     0.806     0.796       134
```

With accuracy of 80.6% and recall rate of 92.3%, model is able to predict 92.3% of Public Transport which were actually claimed as claimed.

Precision is 81.6% of data which means, out of total employees predicted by model as opt for Public Transport , 81.6% employees actually opted for the Public Transport.

F1-score is the harmonic mean of precision and recall, it takes into the effect of both the scores and this value is low if any of these 2 value is low.

Since we are building a model to predict if whether employee will opt for Public Transport or Private Transport, for practical purposes, we will be more interested in correctly classifying 1 (Public Transport) than 0(Private Transport).

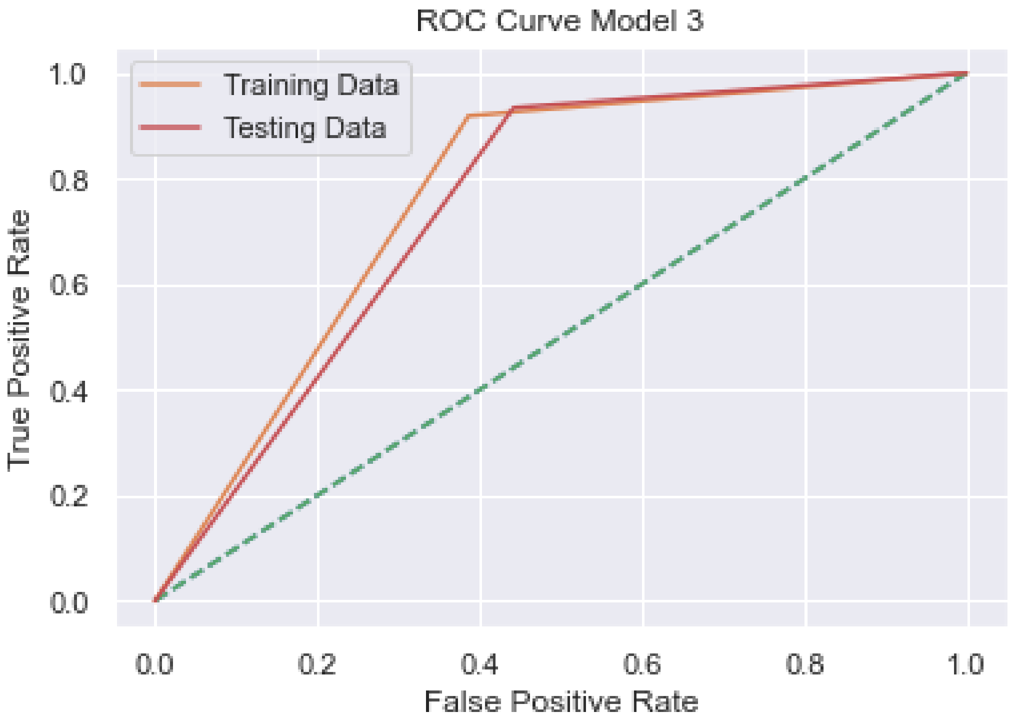
If a employee Private Transport is incorrectly predicted to be "Public Transport" by the model, then the impact on cost for the travel company would be bare minimum. But if am employee opted for Public Transport is incorrectly predicted to be Private Transport by the model, then the cost impact would be very high for the Transport company. Its a loss of potential lead for the company. Hence recall rate (actual data point identified as True by model) is very important in this scenario.

As Recall rate of test dataset is very good around 92.3% plus Precision rate is also 81.6% for opting Public transport thus this does looks good enough for classification

Thus our Transport Company can target these Employee's for providing services.

Model 3 (at Cut off 0.5)

AUC and ROC for the Training and Testing data Model 3



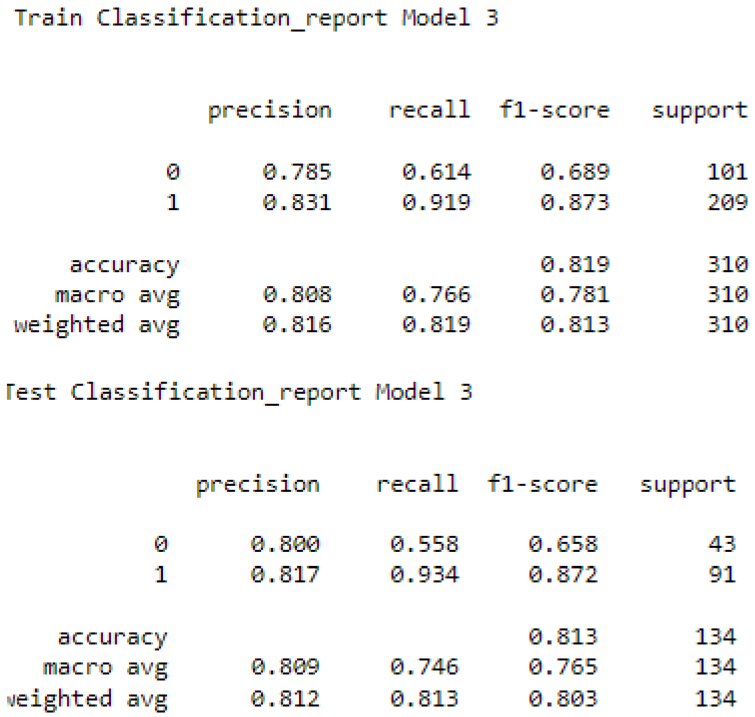
AUC ROC Curve Model 3 Train and Test

Observation AUC for Model 3:

AUC for Train dataset: 0.766

AUC for test dataset: 0.746

Classification_report Model 3



```
Train Classification_report Model 3

              precision    recall  f1-score   support

           0      0.785     0.614     0.689       101
           1      0.831     0.919     0.873       209

    accuracy                          0.819       310
   macro avg      0.808     0.766     0.781       310
weighted avg      0.816     0.819     0.813       310

Test Classification_report Model 3

              precision    recall  f1-score   support

           0      0.800     0.558     0.658        43
           1      0.817     0.934     0.872        91

    accuracy                          0.813       134
   macro avg      0.809     0.746     0.765       134
weighted avg      0.812     0.813     0.803       134
```

Observation :

With accuracy of 81.3% and recall rate of 93.4%, model is able to predict 91% of Public Transport which were actually claimed as claimed.

Precision is 81.7% of data which means, out of total employees predicted by model as opt for Public Transport , 81.7% employees actually opted for the Public Transport.

F1-score is the harmonic mean of precision and recall, it takes into the effect of both the scores and this value is low if any of these 2 value is low.

Since we are building a model to predict if whether employee will opt for Public Transport or Private Transport, for practical purposes, we will be more interested in correctly classifying 1 (Public Transport) than 0(Private Transport).

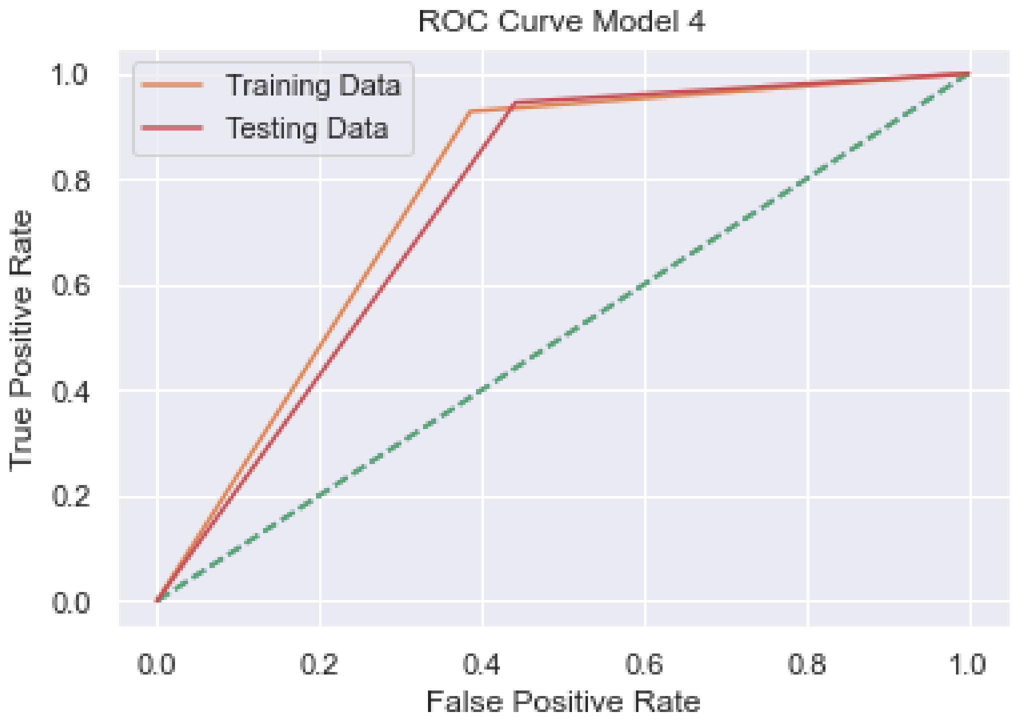
If a employee Private Transport is incorrectly predicted to be "Public Transport" by the model, then the impact on cost for the travel company would be bare minimum. But if am employee opted for Public Transport is incorrectly predicted to be Private Transport by the model, then the cost impact would be very high for the Transport company. Its a loss of potential lead for the company. Hence recall rate (actual data point identified as True by model) is very important in this scenario.

As Recall rate of test dataset is very good around 93.4% plus Precision rate is also 81.7% for opting Public transport thus this does looks good enough for classification

Thus our Transport Company can target these Employee's for providing services.

Model 4 (at Cut off 0.5)

AUC and ROC for the Training and Testing data Model 4



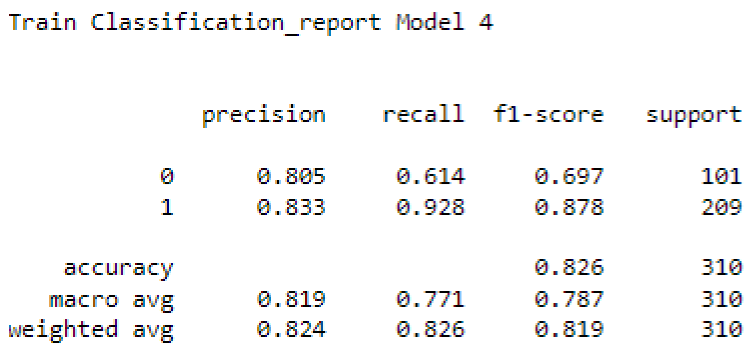
AUC ROC Curve Model 4 Train and Test

Observation AUC for Model 4:

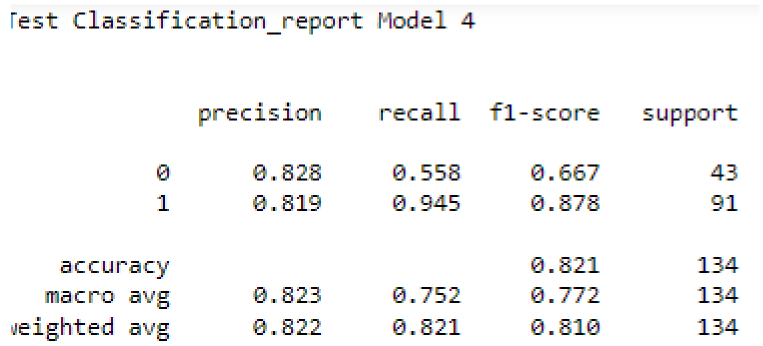
AUC for Train dataset: 0.771

AUC for test dataset: 0.752

Classification_report Model 4



```
Train Classification_report Model 4

              precision    recall  f1-score   support

           0      0.805     0.614     0.697       101
           1      0.833     0.928     0.878       209

    accuracy                          0.826       310
   macro avg      0.819     0.771     0.787       310
weighted avg      0.824     0.826     0.819       310
```



```
Test Classification_report Model 4

              precision    recall  f1-score   support

           0      0.828     0.558     0.667        43
           1      0.819     0.945     0.878        91

    accuracy                          0.821       134
   macro avg      0.823     0.752     0.772       134
weighted avg      0.822     0.821     0.810       134
```

Observation :

With accuracy of 82.1% and recall rate of 94.5%, model is able to predict 94.5% of Public Transport which were actually claimed as claimed.

Precision is 81.9% of data which means, out of total employees predicted by model as opt for Public Transport , 81.9% employees actually opted for the Public Transport.

F1-score is the harmonic mean of precision and recall, it takes into the effect of both the scores and this value is low if any of these 2 value is low.

Since we are building a model to predict if whether employee will opt for Public Transport or Private Transport, for practical purposes, we will be more interested in correctly classifying 1 (Public Transport) than 0(Private Transport).

If a employee Private Transport is incorrectly predicted to be "Public Transport" by the model, then the impact on cost for the travel company would be bare minimum. But if am employee opted for Public Transport is incorrectly predicted to be Private Transport by the model, then the cost impact would be very high for the Transport company. Its a loss of potential lead for the company. Hence recall rate (actual data point identified as True by model) is very important in this scenario.

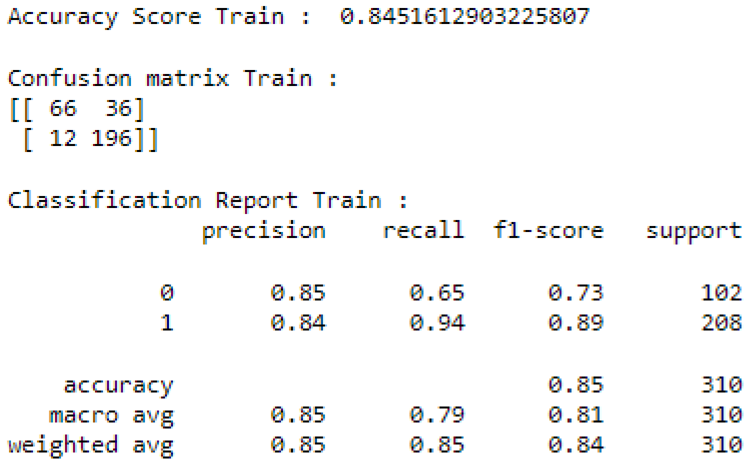
As Recall rate of test dataset is very good around 94.5% plus Precision rate is also 81.9% for opting Public transport thus this does looks good enough for classification

Thus our Transport Company can target these Employee's for providing services.

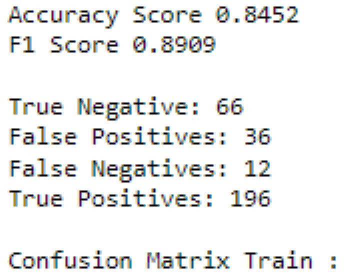
5. Apply KNN Model. Interpret the results.

KNN Model

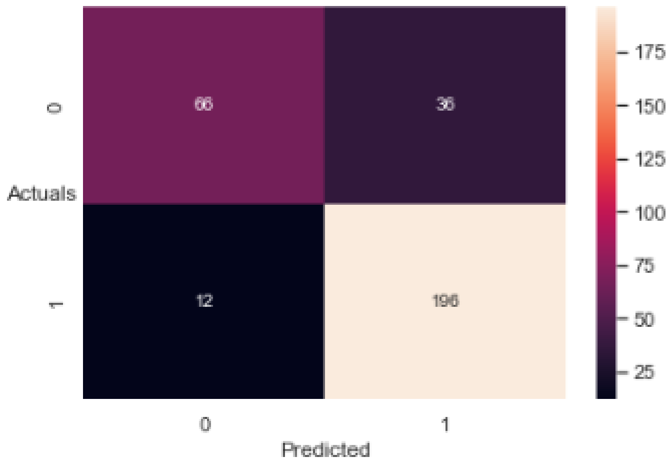
Performance Matrix and Heat Map on the train data:



```
Accuracy Score Train :  0.8451612903225807

Confusion matrix Train :
[[ 66  36]
 [ 12 196]]

Classification Report Train :
              precision    recall  f1-score   support

           0       0.85      0.65      0.73       102
           1       0.84      0.94      0.89       208

    accuracy                           0.85       310
   macro avg       0.85      0.79      0.81       310
weighted avg       0.85      0.85      0.84       310
```

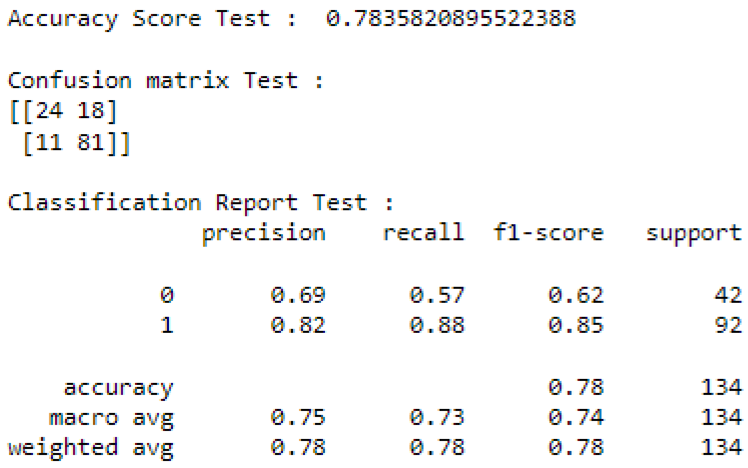


```
Accuracy Score 0.8452
F1 Score 0.8909

True Negative: 66
False Positives: 36
False Negatives: 12
True Positives: 196

Confusion Matrix Train :
```

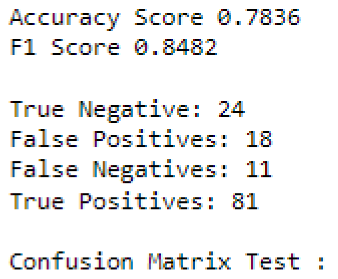


KNN heatmap Train

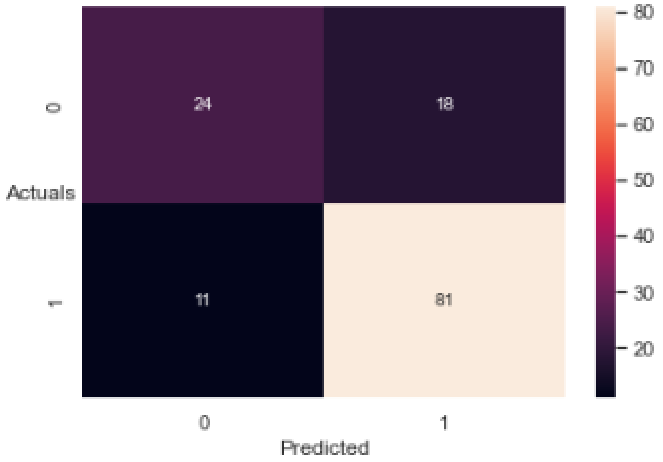
Performance Matrix and Heat Map on the test data:



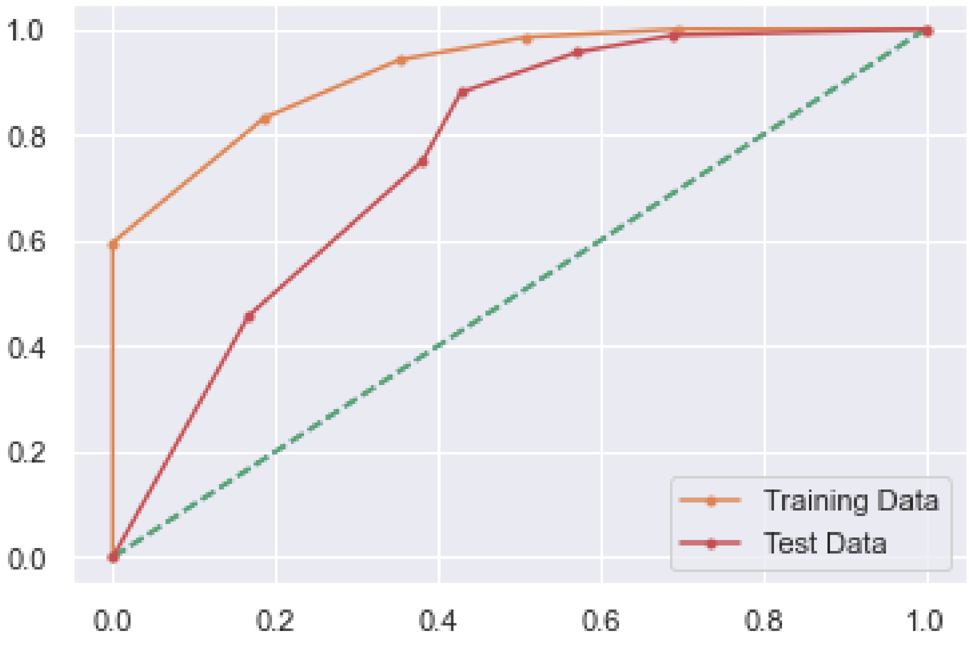
```
Accuracy Score Test :  0.7835820895522388

Confusion matrix Test :
[[24 18]
 [11 81]]

Classification Report Test :
              precision    recall  f1-score   support

           0       0.69      0.57      0.62        42
           1       0.82      0.88      0.85        92

    accuracy                           0.78       134
   macro avg       0.75      0.73      0.74       134
weighted avg       0.78      0.78      0.78       134
```



```
Accuracy Score 0.7836
F1 Score 0.8482

True Negative: 24
False Positives: 18
False Negatives: 11
True Positives: 81

Confusion Matrix Test :
```



KNN heatmap Test



KNN AUC ROC Train and Test

```
AUC for the Training Data: 0.921
AUC for the Test Data: 0.761
```

Observation :

With accuracy of 78% and recall rate of 88%, model is able to predict 88% of Public Transport which were actually claimed as claimed.

Precision is 82% of data which means, out of total employees predicted by model as opt for Public Transport , 82% employees actually opted for the Public Transport.

F1-score is the harmonic mean of precision and recall, it takes into the effect of both the scores and this value is low if any of these 2 value is low.

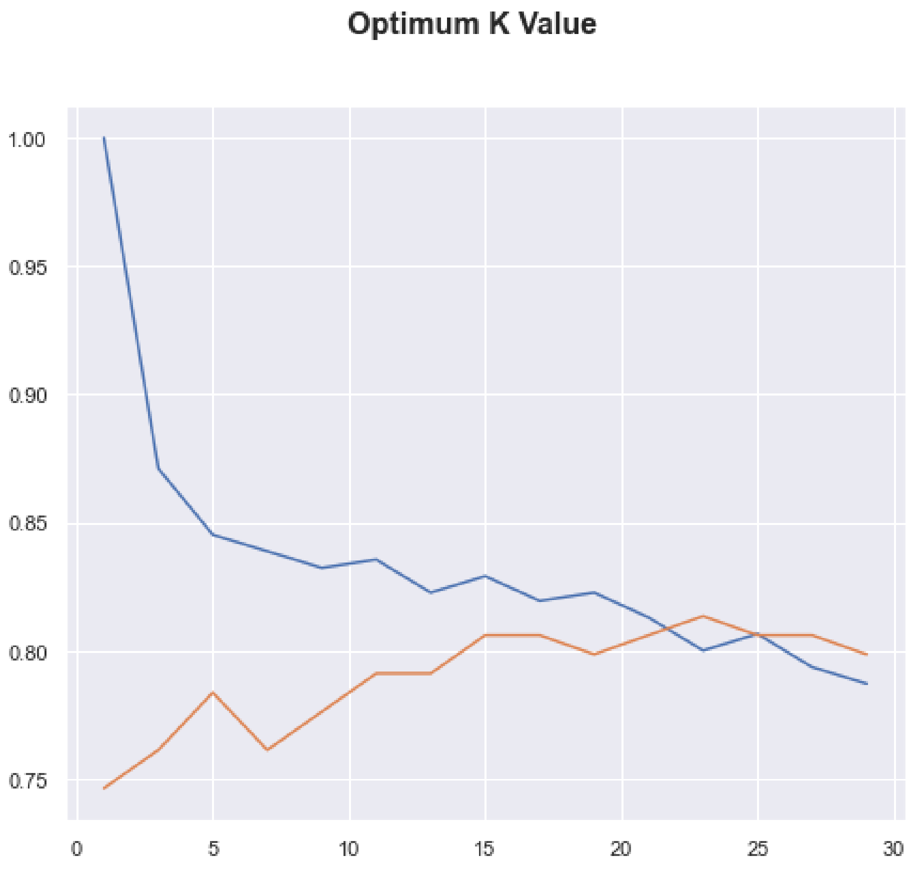
Since we are building a model to predict if whether employee will opt for Public Transport or Private Transport, for practical purposes, we will be more interested in correctly classifying 1 (Public Transport) than 0(Private Transport).

If a employee Private Transport is incorrectly predicted to be "Public Transport" by the model, then the impact on cost for the travel company would be bare minimum. But if am employee opted for Public Transport is incorrectly predicted to be Private Transport by the model, then the cost impact would be very high for the Transport company. Its a loss of potential lead for the company. Hence recall rate (actual data point identified as True by model) is very important in this scenario.

As Recall rate of test dataset is very good around 88% plus Precision rate is also 82% for opting Public transport thus this does looks good enough for classification

Thus our Transport Company can target these Employee's for providing services.

Finding the Optimum K Value using Graphical Presentation

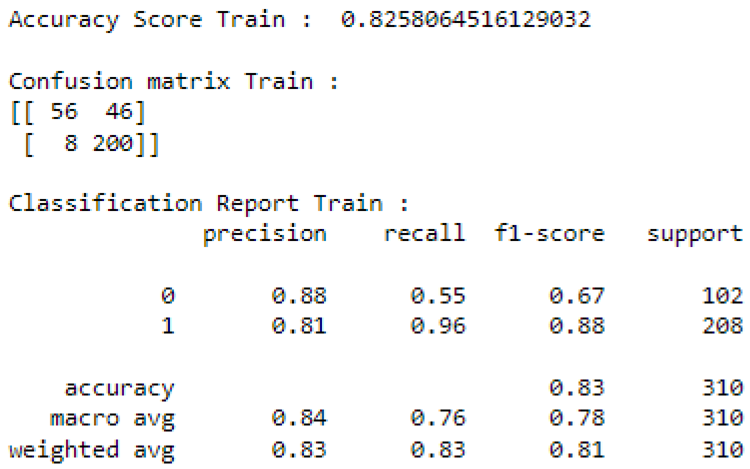


Optimum K Value

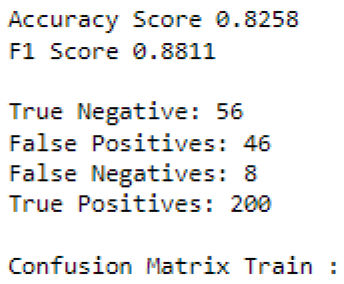
Observation:

- Form the above Graph we can clearly see that **Optimum K value lie in the range between 19-22**

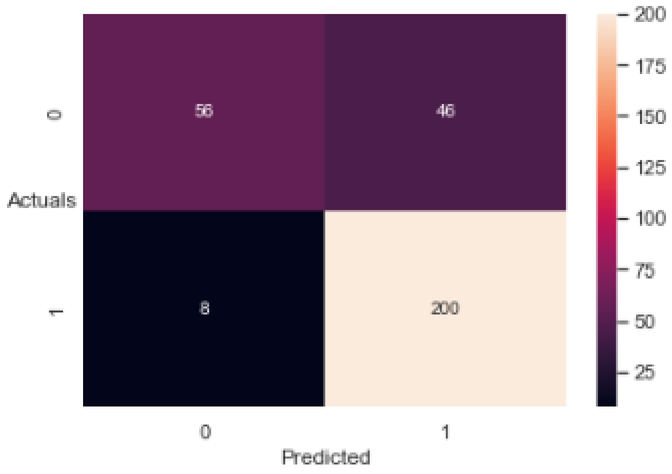
Performance Matrix and Heat Map on the Train data (Assume K = 20):



```
Accuracy Score Train :  0.8258064516129032

Confusion matrix Train :
[[ 56  46]
 [  8 200]]

Classification Report Train :
              precision    recall  f1-score   support

           0       0.88      0.55      0.67       102
           1       0.81      0.96      0.88       208

    accuracy                           0.83       310
   macro avg       0.84      0.76      0.78       310
weighted avg       0.83      0.83      0.81       310
```

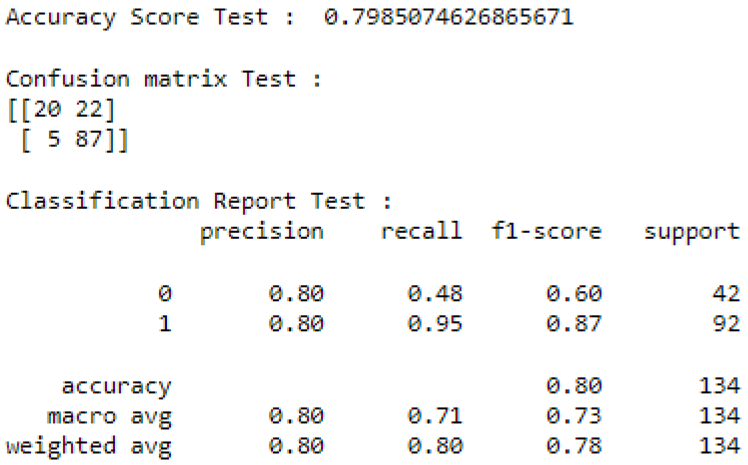


```
Accuracy Score 0.8258
F1 Score 0.8811

True Negative: 56
False Positives: 46
False Negatives: 8
True Positives: 200

Confusion Matrix Train :
```

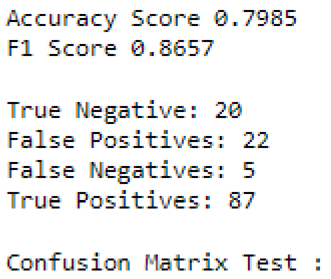


Heat Map K Value = 20 Train

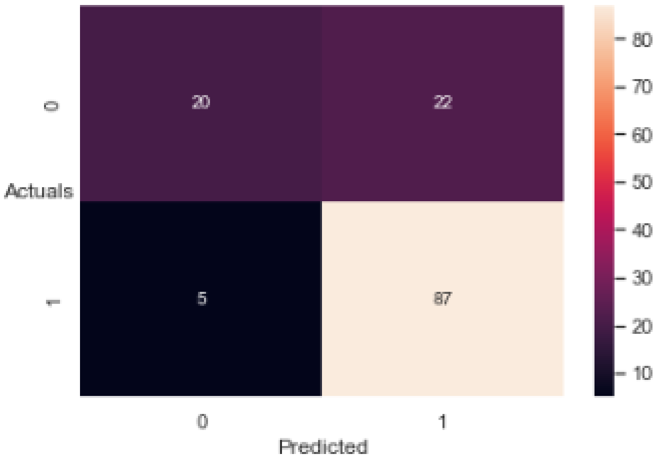
Performance Matrix and Heat Map on the Test data (Assume K = 20):



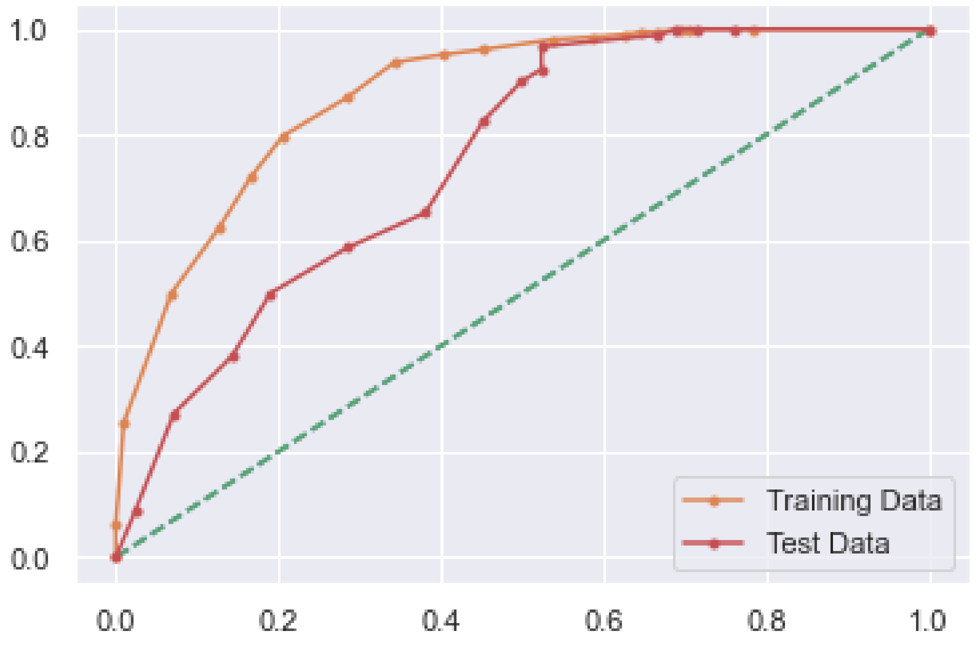
```
Accuracy Score Test :  0.7985074626865671

Confusion matrix Test :
[[20 22]
 [ 5 87]]

Classification Report Test :
              precision    recall  f1-score   support

           0       0.80      0.48      0.60        42
           1       0.80      0.95      0.87        92

    accuracy                           0.80       134
   macro avg       0.80      0.71      0.73       134
weighted avg       0.80      0.80      0.78       134
```



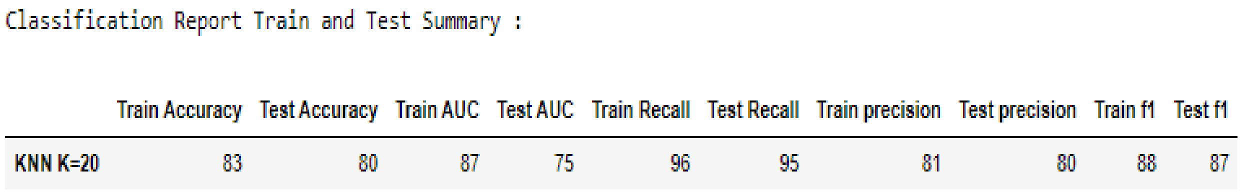
```
Accuracy Score 0.7985
F1 Score 0.8657

True Negative: 20
False Positives: 22
False Negatives: 5
True Positives: 87

Confusion Matrix Test :
```



Heat Map K Value = 20 Test



AUC ROC Curve Train Test K Value = 20



Classification Report Train and Test Summary :

	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
KNN K=20	83	80	87	75	96	95	81	80	88	87

Observation :

When we took **K-value = 20**

Then with accuracy of 80% and recall rate of 95%, model is able to predict 95% of Public Transport which were actually claimed as claimed.

Precision is 80% of data which means, out of total employees predicted by model as opt for Public Transport , 80% employees actually opted for the Public Transport.

F1-score is the harmonic mean of precision and recall, it takes into the effect of both the scores and this value is low if any of these 2 value is low.

Since we are building a model to predict if whether employee will opt for Public Transport or Private Transport, for practical purposes, we will be more interested in correctly classifying 1 (Public Transport) than 0(Private Transport).

If a employee Private Transport is incorrectly predicted to be "Public Transport" by the model, then the impact on cost for the travel company would be bare minimum. But if am employee opted for Public Transport is incorrectly predicted to be Private Transport by the model, then the cost impact would be very high for the Transport company. Its a loss of potential lead for the company. Hence recall rate (actual data point identified as True by model) is very important in this scenario.

As Recall rate of test dataset is very good around 95% plus Precision rate is also 80% for opting Public transport thus this does looks good enough for classification

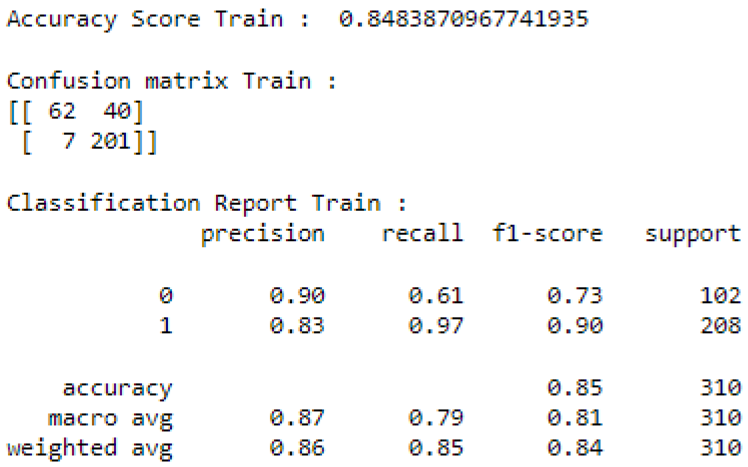
Thus our Transport Company can target these Employee's for providing services.

Hyperparameters on KNN Model

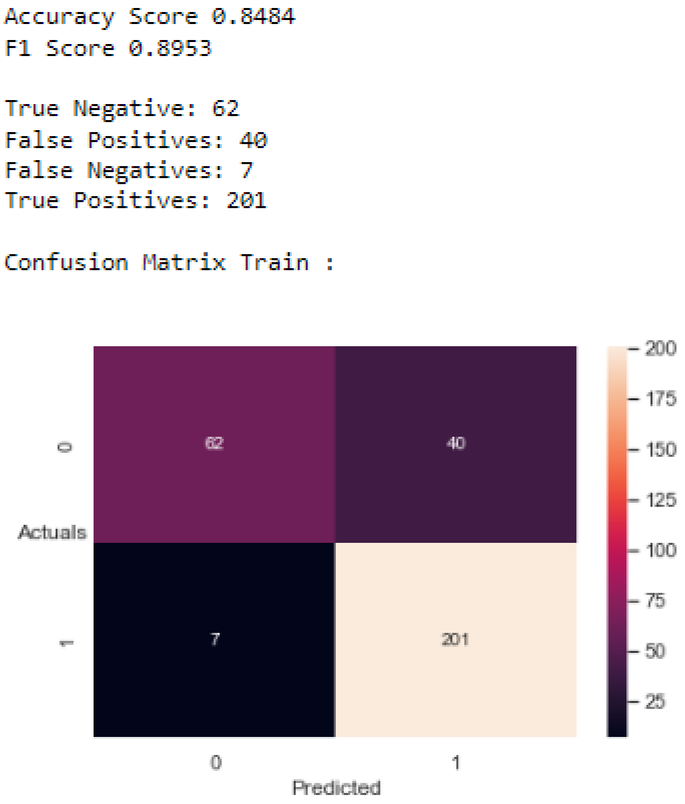
After doing the Hyperparameters on KNN we found K value 20 as best optimum fit.

Now let's check whether it's true or not

Performance Matrix and Heat Map on the Train data (Assume K = 15):



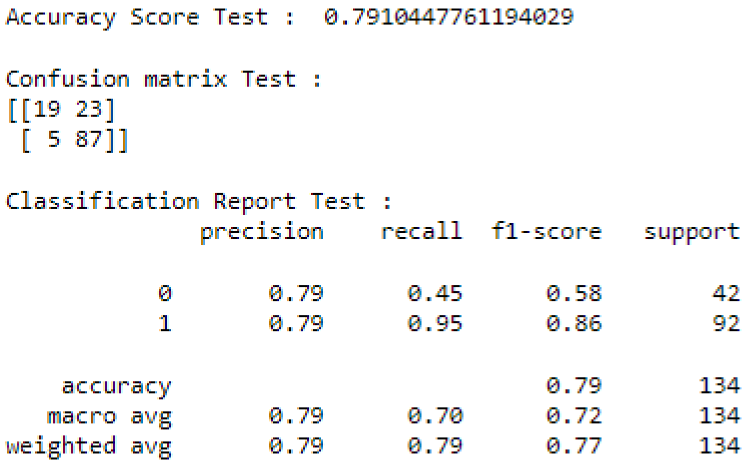
```
Accuracy Score Train :  0.8483870967741935

Confusion matrix Train :
[[ 62  40]
 [  7 201]]

Classification Report Train :
              precision    recall  f1-score   support

           0       0.90      0.61      0.73       102
           1       0.83      0.97      0.90       208

    accuracy                           0.85       310
   macro avg       0.87      0.79      0.81       310
weighted avg       0.86      0.85      0.84       310
```



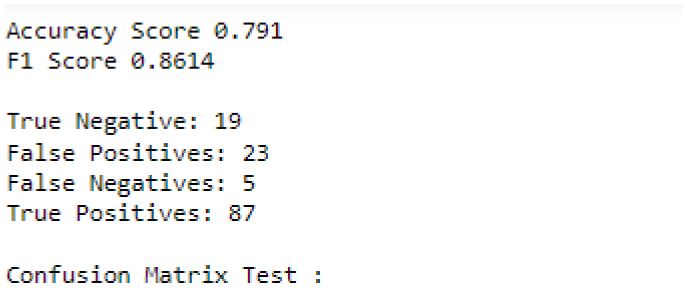
```
Accuracy Score 0.8484
F1 Score 0.8953

True Negative: 62
False Positives: 40
False Negatives: 7
True Positives: 201

Confusion Matrix Train :
```

Heat Map K Value = 15 Train

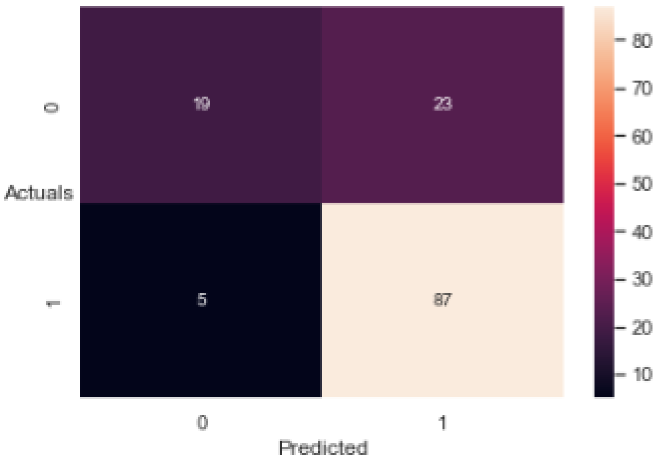
Performance Matrix and Heat Map on the Test data (Assume K = 15):



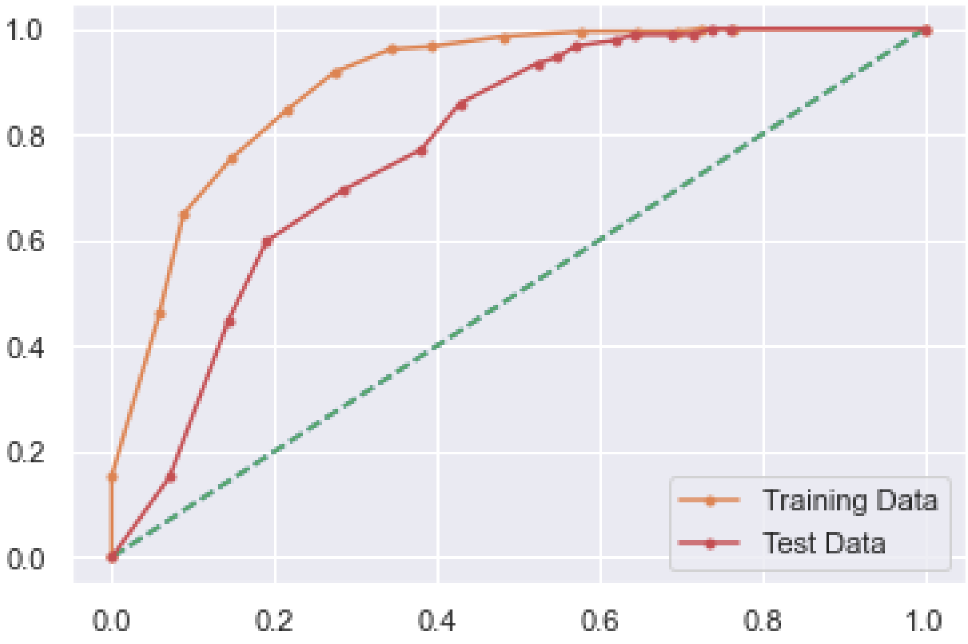
```
Accuracy Score Test :  0.7910447761194029

Confusion matrix Test :
[[19 23]
 [ 5 87]]

Classification Report Test :
              precision    recall  f1-score   support

           0       0.79      0.45      0.58        42
           1       0.79      0.95      0.86        92

    accuracy                           0.79       134
   macro avg       0.79      0.70      0.72       134
weighted avg       0.79      0.79      0.77       134
```



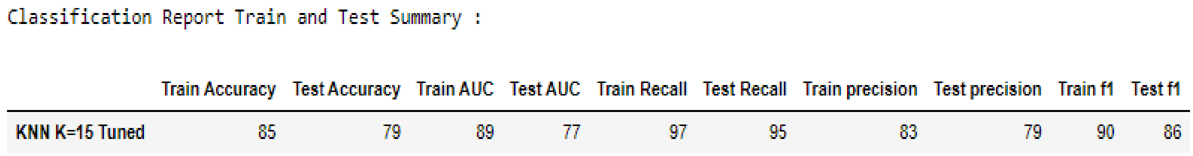
```
Accuracy Score 0.791
F1 Score 0.8614

True Negative: 19
False Positives: 23
False Negatives: 5
True Positives: 87

Confusion Matrix Test :
```



Heat Map K Value = 15 Test



AUC ROC Curve Train Test K Value = 15



```
Classification Report Train and Test Summary :
```

	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
KNN K=15 Tuned	85	79	89	77	97	95	83	79	90	86

Observation :

When we took **K-value = 15**

Then with accuracy of 79% and recall rate of 95%, model is able to predict 95% of Public Transport which were actually claimed as claimed.

Precision is 79% of data which means, out of total employees predicted by model as opt for Public Transport , 79% employees actually opted for the Public Transport.

F1-score is the harmonic mean of precision and recall, it takes into the effect of both the scores and this value is low if any of these 2 value is low.

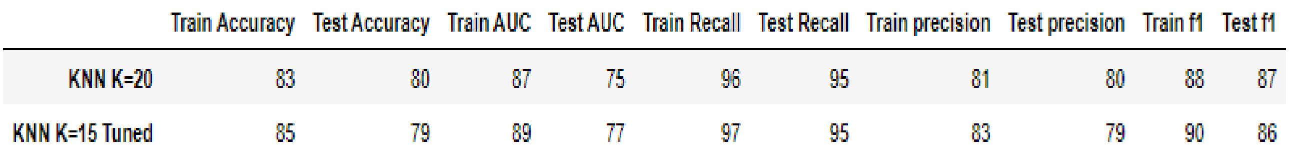
Since we are building a model to predict if whether employee will opt for Public Transport or Private Transport, for practical purposes, we will be more interested in correctly classifying 1 (Public Transport) than 0(Private Transport).

If a employee Private Transport is incorrectly predicted to be "Public Transport" by the model, then the impact on cost for the travel company would be bare minimum. But if am employee opted for Public Transport is incorrectly predicted to be Private Transport by the model, then the cost impact would be very high for the Transport company. Its a loss of potential lead for the company. Hence recall rate (actual data point identified as True by model) is very important in this scenario.

As Recall rate of test dataset is very good around 95% plus Precision rate is also 79% for opting Public transport thus this does looks good enough for classification

Thus our Transport Company can target these Employee's for providing services.

KNN Summary Report:



	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
KNN K=20	83	80	87	75	96	95	81	80	88	87
KNN K=15 Tuned	85	79	89	77	97	95	83	79	90	86

Observation:

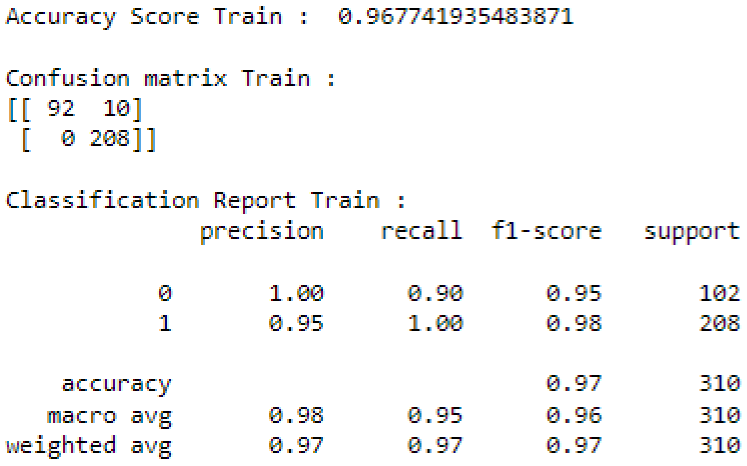
We can clearly see that even after Hyperparameter tunning of KNN Model(K=15) there is a decrease in the model performance.

Thus K=20 gives the best output for KNN Model.

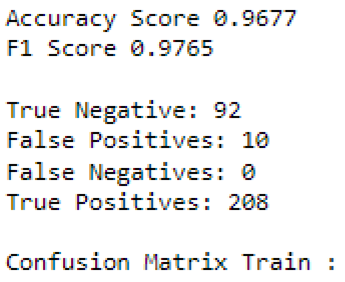
6. Model Tuning, Bagging (Random Forest should be applied for Bagging), and Boosting.

Bagging Classifier (Random Forest)

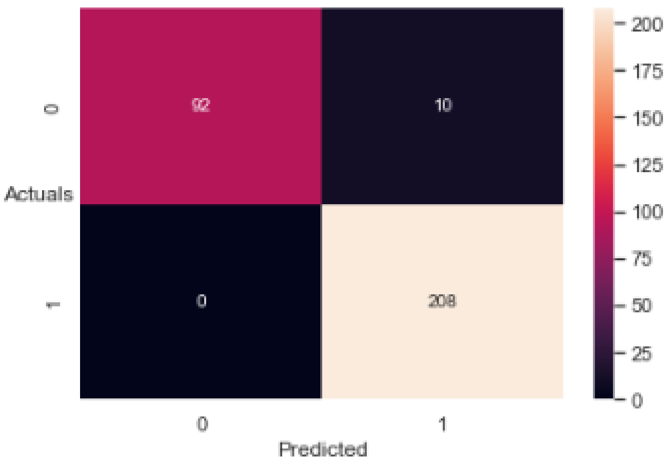
Performance Matrix and Heat Map on the Train data



```
Accuracy Score Train :  0.967741935483871

Confusion matrix Train :
[[ 92  10]
 [  0 208]]

Classification Report Train :
              precision    recall  f1-score   support

           0       1.00      0.90      0.95       102
           1       0.95      1.00      0.98       208

    accuracy                           0.97       310
   macro avg       0.98      0.95      0.96       310
weighted avg       0.97      0.97      0.97       310
```

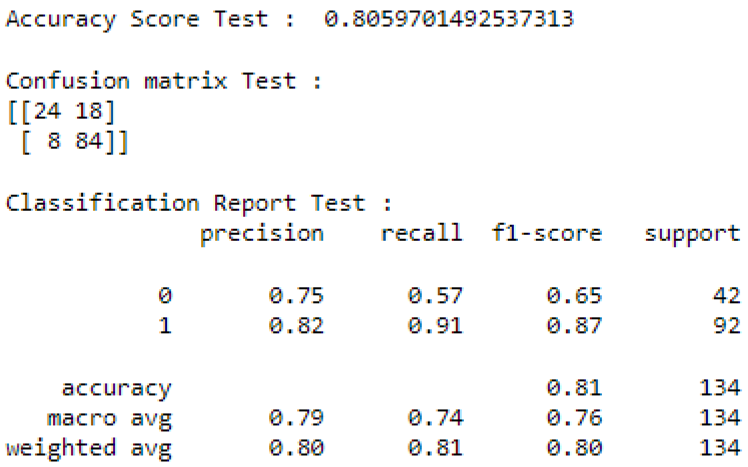


```
Accuracy Score 0.9677
F1 Score 0.9765

True Negative: 92
False Positives: 10
False Negatives: 0
True Positives: 208

Confusion Matrix Train :
```

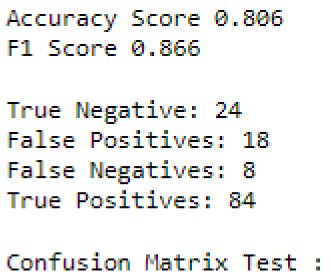


Heat Map Bagging (Random Forest) Train

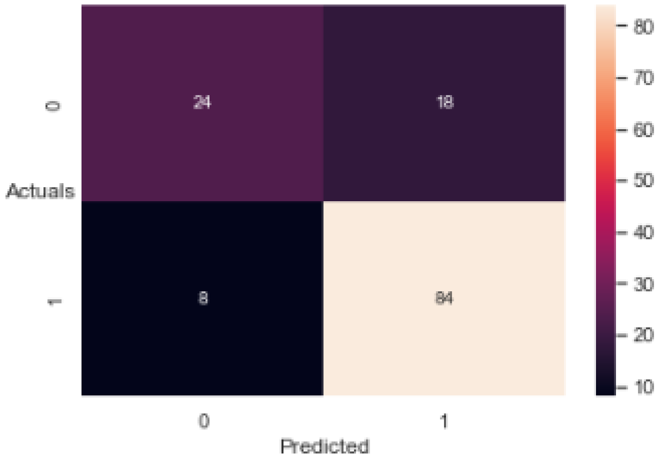
Performance Matrix and Heat Map on the Train data



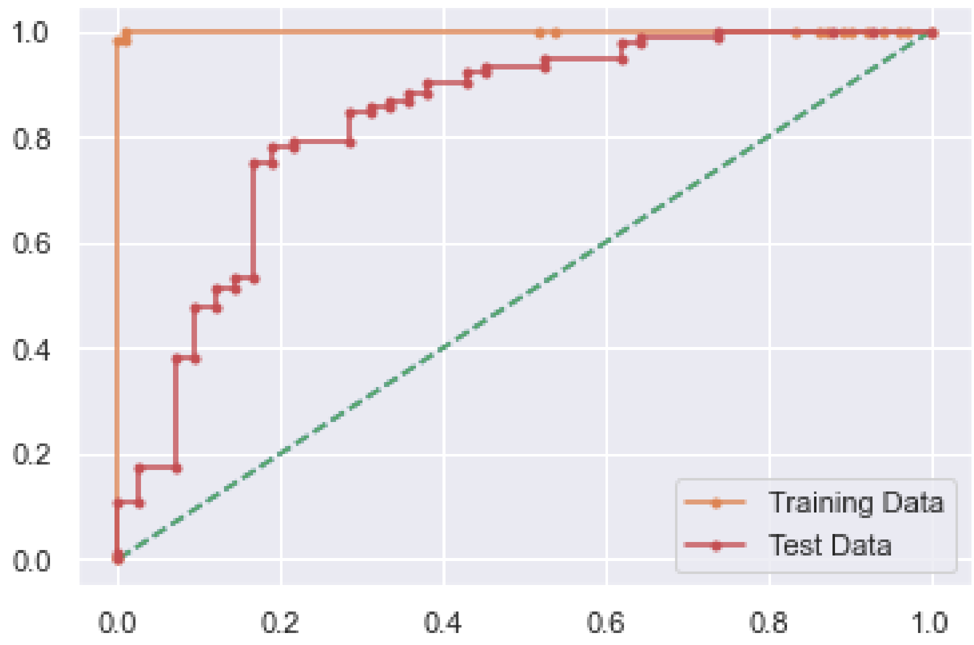
```
Accuracy Score Test :  0.8059701492537313

Confusion matrix Test :
[[24 18]
 [ 8 84]]

Classification Report Test :
              precision    recall  f1-score   support

           0       0.75      0.57      0.65        42
           1       0.82      0.91      0.87        92

    accuracy                           0.81       134
   macro avg       0.79      0.74      0.76       134
weighted avg       0.80      0.81      0.80       134
```



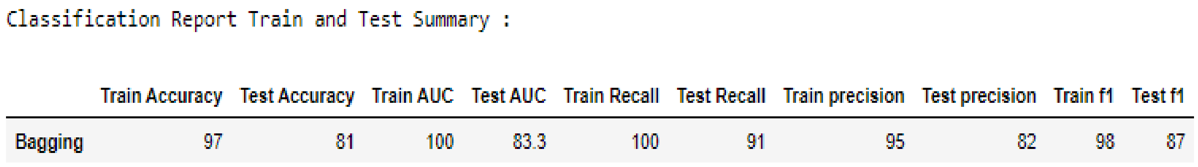
```
Accuracy Score 0.806
F1 Score 0.866

True Negative: 24
False Positives: 18
False Negatives: 8
True Positives: 84

Confusion Matrix Test :
```



Heat Map Bagging (Random Forest) Test



AUC ROC Curve Bagging (Random Forest) Train and Test



```
Classification Report Train and Test Summary :
```

	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
Bagging	97	81	100	83.3	100	91	95	82	98	87

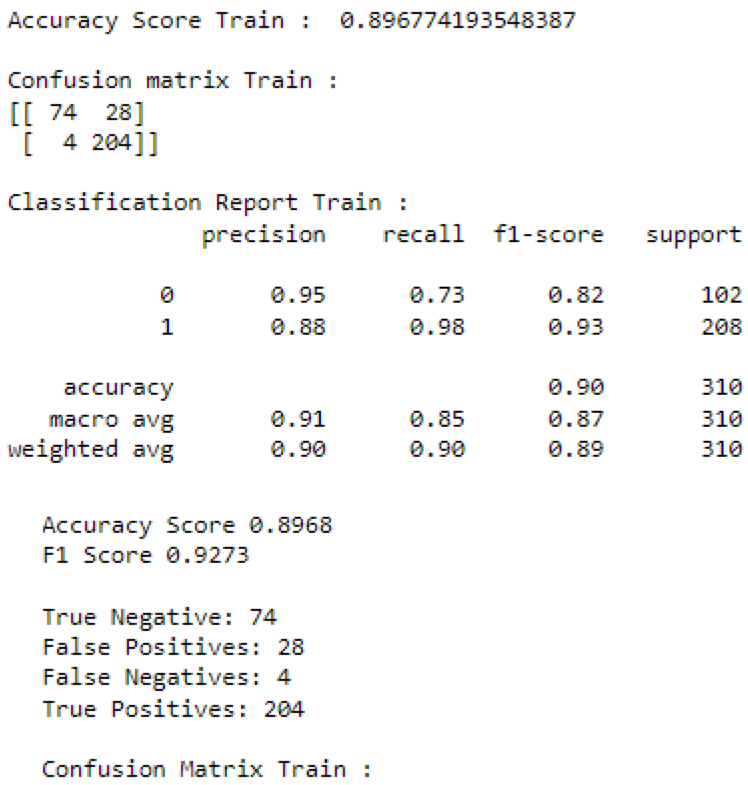
Observation

- The overall accuracy (train 97% and test 81%) seems to be decent using bagging but , there is a huge difference between test/train accuracy value.
- Similarly Model Recall score shows 100% in train and 91% Test which is decent.
- Precision also have huge difference which means the model is not able to precisely tell who all the employee opting for 1(Public Transport)

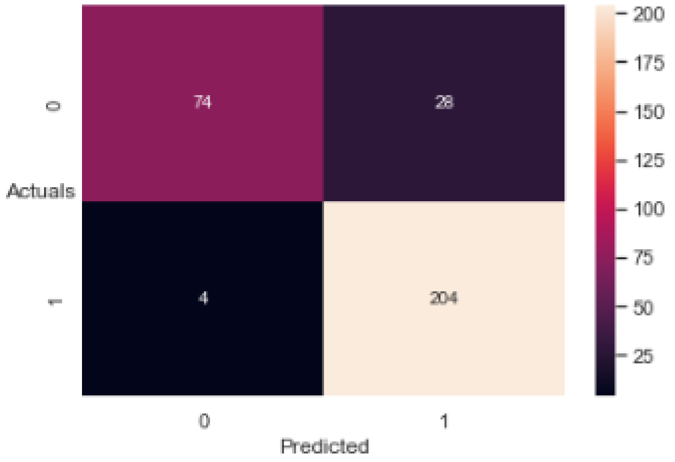
Therefore there seems to be Overfitting which can be rectified by Hyperparameter tuning of the Random Forest Classifier.

Hyperparameter tuning of Bagging Classifier (Random Forest)

Performance Matrix and Heat Map on the Train data

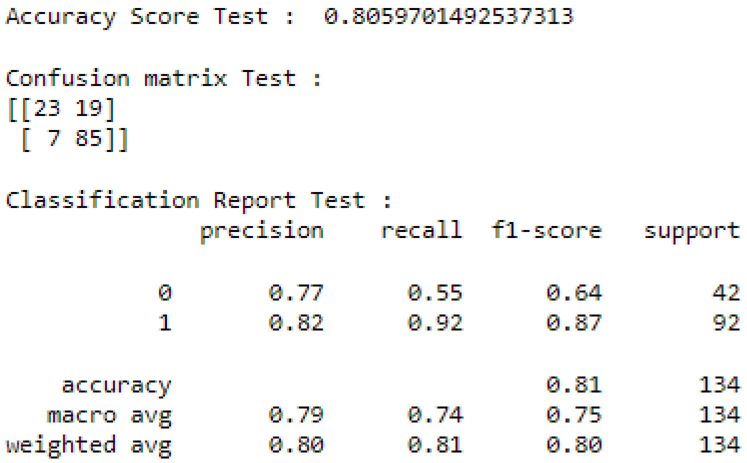


```
Accuracy Score Train :  0.896774193548387

Confusion matrix Train :
[[ 74  28]
 [  4 204]]

Classification Report Train :
              precision    recall  f1-score   support

           0       0.95      0.73      0.82       102
           1       0.88      0.98      0.93       208

    accuracy                           0.90       310
   macro avg       0.91      0.85      0.87       310
weighted avg       0.90      0.90      0.89       310

  Accuracy Score 0.8968
  F1 Score 0.9273

  True Negative: 74
  False Positives: 28
  False Negatives: 4
  True Positives: 204

  Confusion Matrix Train :
```

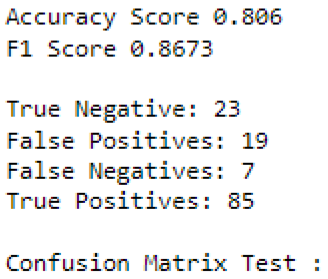


Heat Map Bagging Tuned (Random Forest) Train

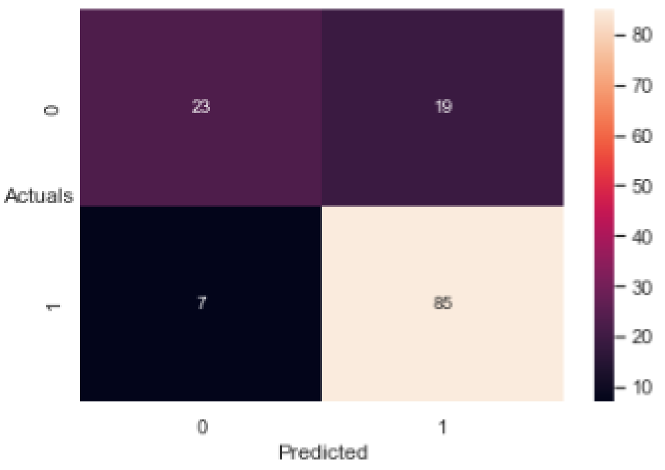
Performance Matrix and Heat Map on the Test data



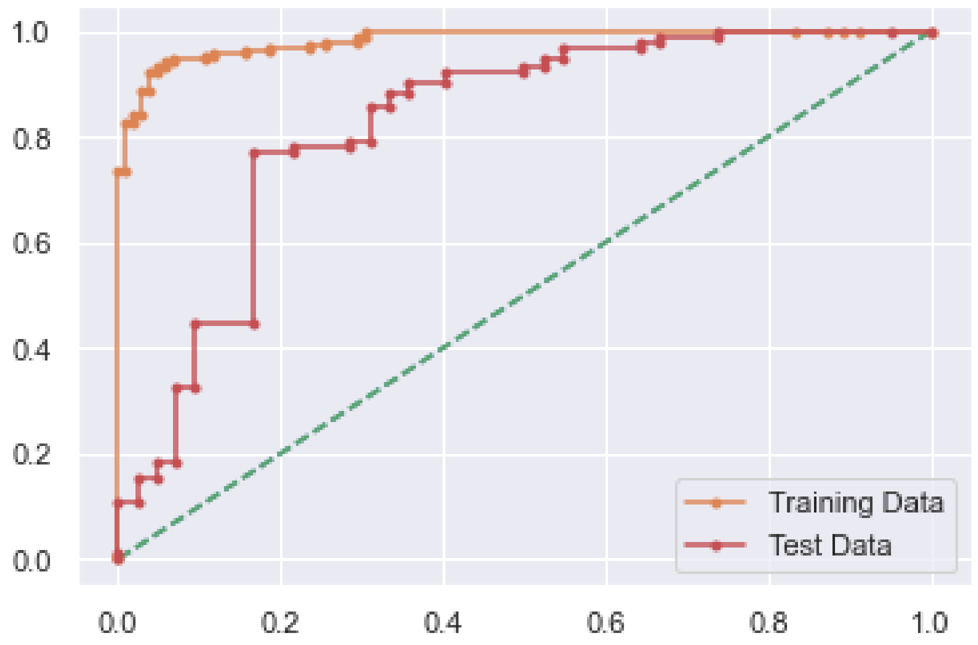
```
Accuracy Score Test :  0.8059701492537313

Confusion matrix Test :
[[23 19]
 [ 7 85]]

Classification Report Test :
              precision    recall  f1-score   support

           0       0.77      0.55      0.64        42
           1       0.82      0.92      0.87        92

    accuracy                           0.81       134
   macro avg       0.79      0.74      0.75       134
weighted avg       0.80      0.81      0.80       134
```



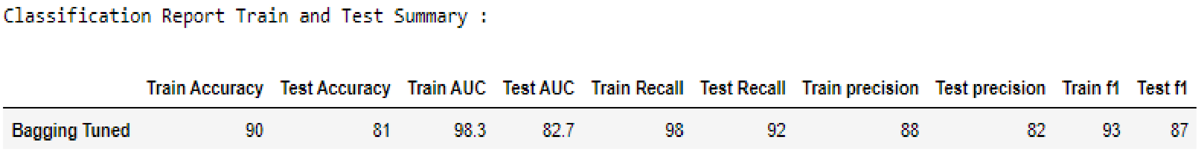
```
Accuracy Score 0.806
F1 Score 0.8673

True Negative: 23
False Positives: 19
False Negatives: 7
True Positives: 85

Confusion Matrix Test :
```



Heat Map Bagging Tuned (Random Forest) Test



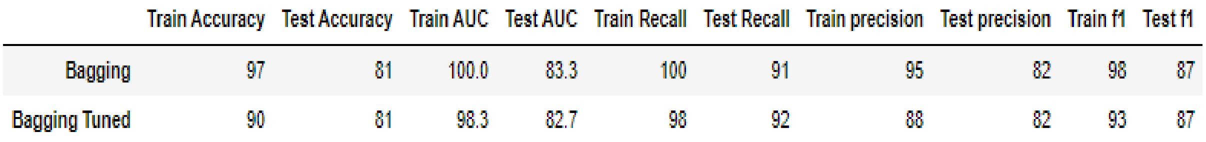
AUC ROC Curve Bagging Tuned (Random Forest) Train and Test



```
Classification Report Train and Test Summary :
```

	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
Bagging Tuned	90	81	98.3	82.7	98	92	88	82	93	87

Bagging Summary Report:



	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
Bagging	97	81	100.0	83.3	100	91	95	82	98	87
Bagging Tuned	90	81	98.3	82.7	98	92	88	82	93	87

Observation:

Here we can clearly interpret an improvement in the model.

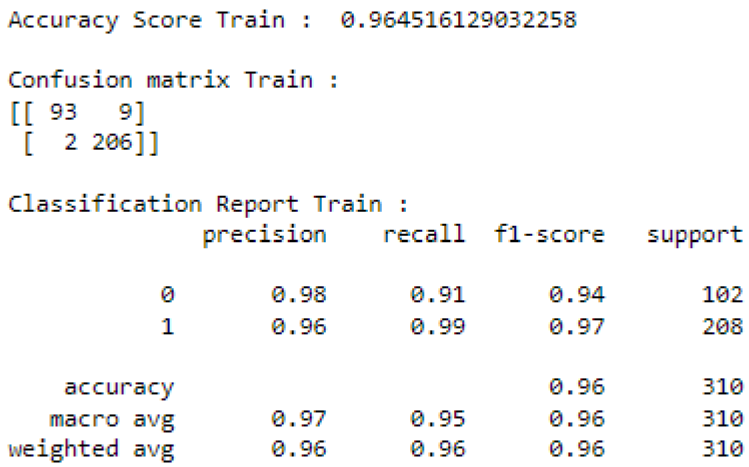
- Accuracy score seems decent(train 90% and test 81%) not much improvement.
- There seems to be a good improvement in Recall score(train 98% and test 92%) which means model is able to properly recall the no. of employees opting for 1(Public Transport).

- Also a very a good improvement in the Precision rate(train 88% and test 82%) which means model is able to precisely tell who are the employee that are opting for 1(Public Transport)
- A good improvement in the AUC also.

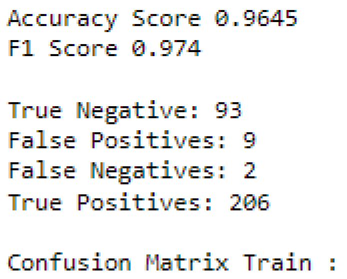
Boosting

a) Gradient Boosting:

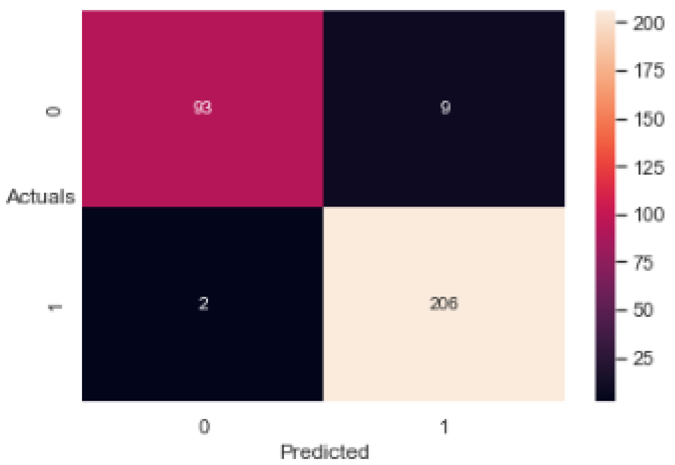
Performance Matrix and Heat Map on the Train data



```
Accuracy Score Train :  0.964516129032258

Confusion matrix Train :
[[ 93   9]
 [  2 206]]

Classification Report Train :
              precision    recall  f1-score   support

           0       0.98      0.91      0.94       102
           1       0.96      0.99      0.97       208

    accuracy                           0.96       310
   macro avg       0.97      0.95      0.96       310
weighted avg       0.96      0.96      0.96       310
```

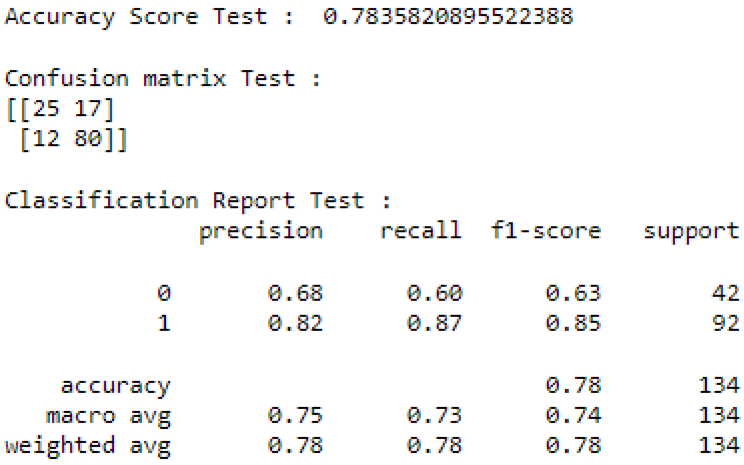


```
Accuracy Score 0.9645
F1 Score 0.974

True Negative: 93
False Positives: 9
False Negatives: 2
True Positives: 206

Confusion Matrix Train :
```

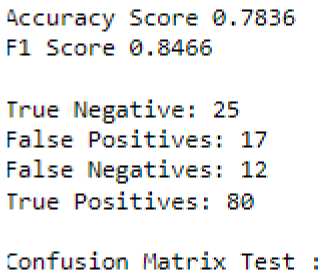


Heat Map Gradient Boost Train

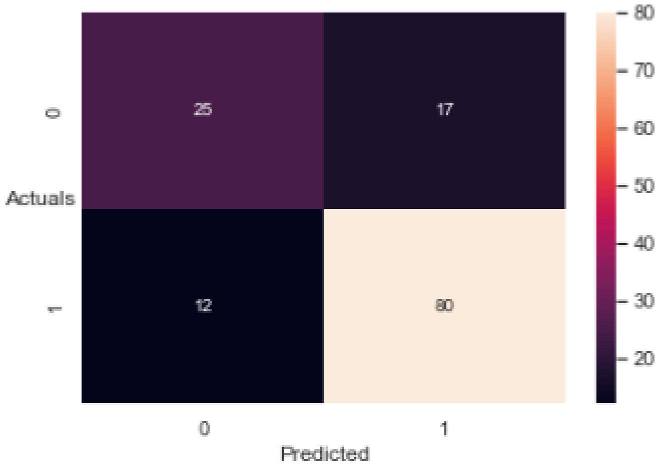
Performance Matrix and Heat Map on the Test data



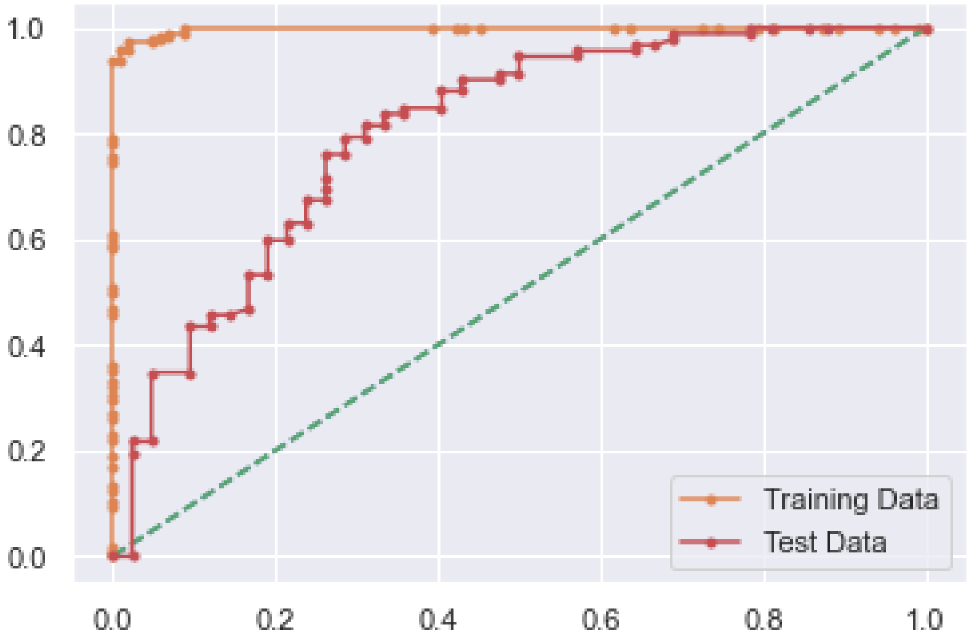
```
Accuracy Score Test :  0.7835820895522388

Confusion matrix Test :
[[25 17]
 [12 80]]

Classification Report Test :
              precision    recall  f1-score   support

           0       0.68      0.60      0.63        42
           1       0.82      0.87      0.85        92

    accuracy                           0.78       134
   macro avg       0.75      0.73      0.74       134
weighted avg       0.78      0.78      0.78       134
```



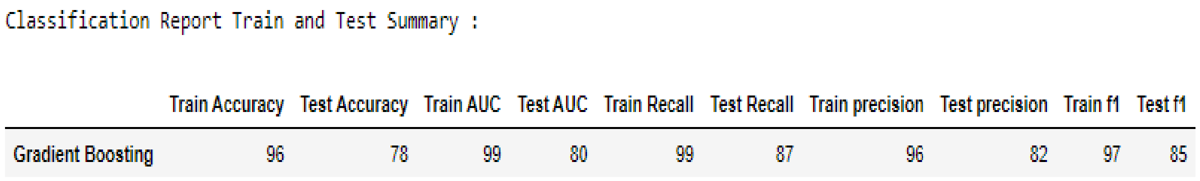
```
Accuracy Score 0.7836
F1 Score 0.8466

True Negative: 25
False Positives: 17
False Negatives: 12
True Positives: 80

Confusion Matrix Test :
```



Heat Map Gradient Boost Test



AUC ROC Curve Gradient Boost Train and Test



```
Classification Report Train and Test Summary :
```

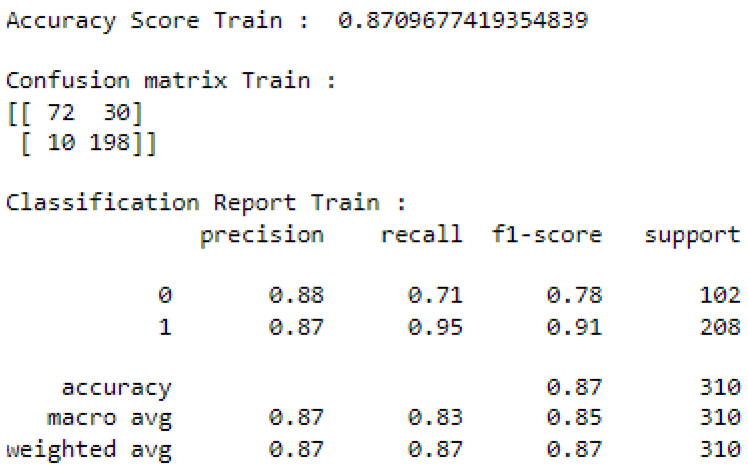
	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
Gradient Boosting	96	78	99	80	99	87	96	82	97	85

Observation :

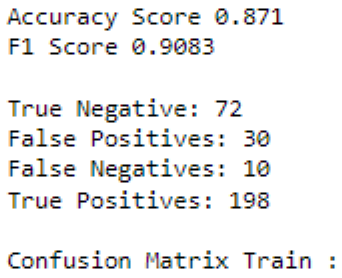
- Their seems to be clear case of Overfitting of model.
- Huge difference between train and test for Accuracy, Recall, Precision, and F1 score
- Gradient Boosting model is not properly grasp the data need to do proper Hyperparameter Tuning.

Hyperparameter tuning of Gradient Boosting

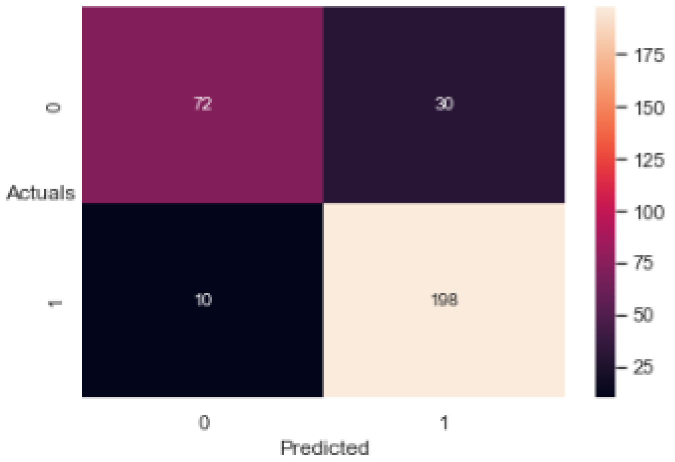
Performance Matrix and Heat Map on the Train data



```
Accuracy Score Train :  0.8709677419354839

Confusion matrix Train :
[[ 72  30]
 [ 10 198]]

Classification Report Train :
              precision    recall  f1-score   support

           0       0.88      0.71      0.78       102
           1       0.87      0.95      0.91       208

    accuracy                           0.87       310
   macro avg       0.87      0.83      0.85       310
weighted avg       0.87      0.87      0.87       310
```

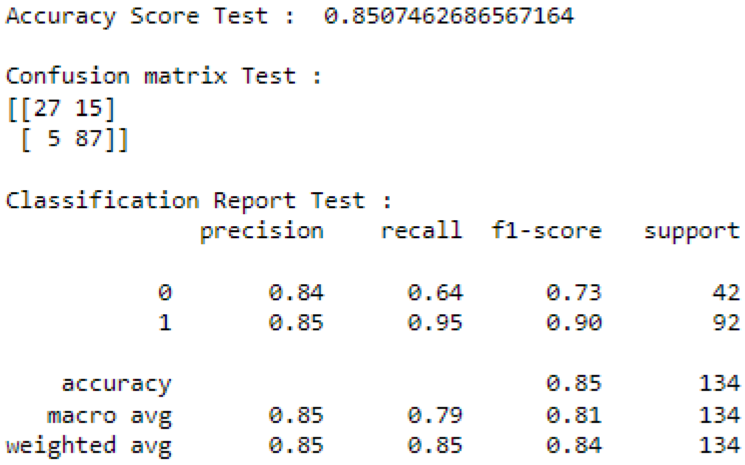


```
Accuracy Score 0.871
F1 Score 0.9083

True Negative: 72
False Positives: 30
False Negatives: 10
True Positives: 198

Confusion Matrix Train :
```

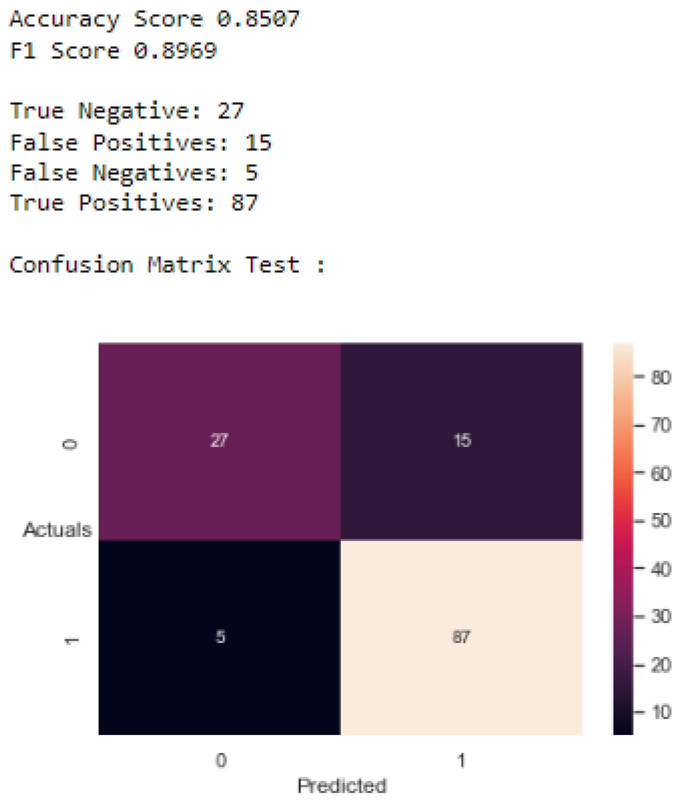


Heat Map Gradient Tuned Boost Train

Performance Matrix and Heat Map on the Test data

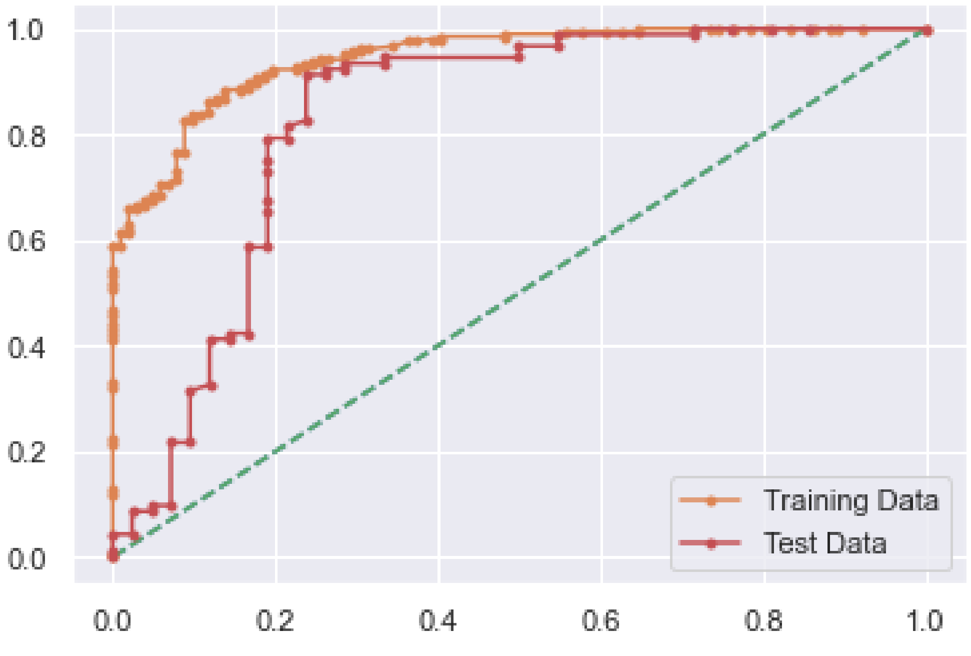


```
Accuracy Score Test :  0.8507462686567164

Confusion matrix Test :
[[27 15]
 [ 5 87]]

Classification Report Test :
              precision    recall  f1-score   support

           0       0.84      0.64      0.73        42
           1       0.85      0.95      0.90        92

    accuracy                           0.85       134
   macro avg       0.85      0.79      0.81       134
weighted avg       0.85      0.85      0.84       134
```

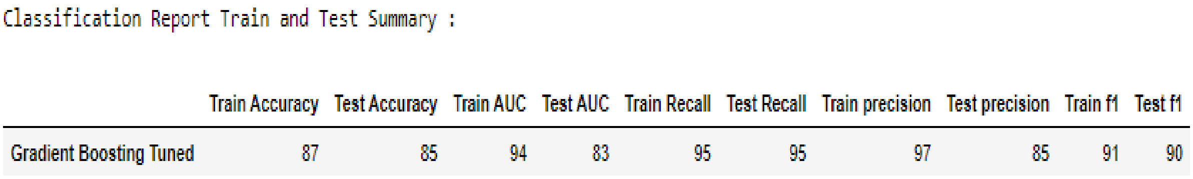


```
Accuracy Score 0.8507
F1 Score 0.8969

True Negative: 27
False Positives: 15
False Negatives: 5
True Positives: 87

Confusion Matrix Test :
```

Heat Map Gradient Tuned Boost Test



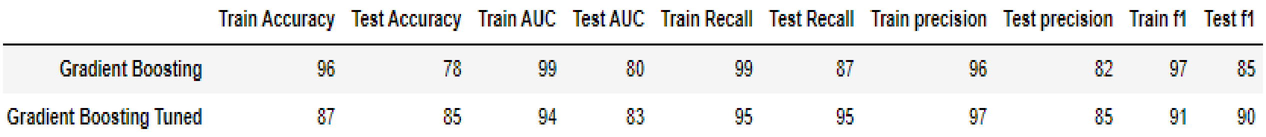
AUC ROC Curve Gradient Boost Tuned Train and Test



Classification Report Train and Test Summary :

	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
Gradient Boosting Tuned	87	85	94	83	95	95	97	85	91	90

Gradient Boosting Summary Report:



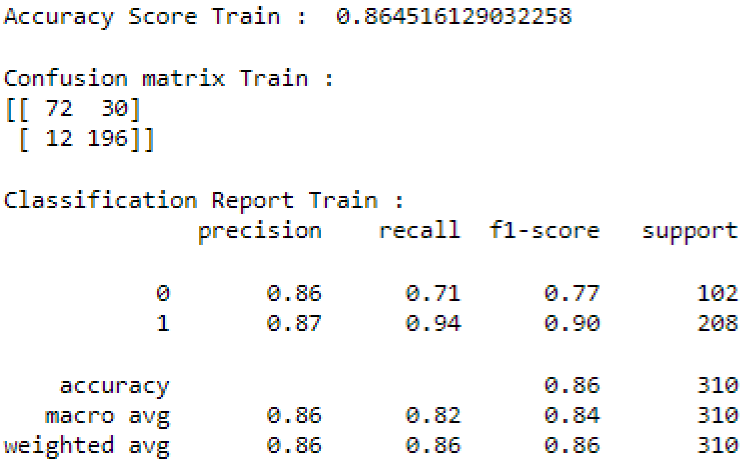
	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
Gradient Boosting	96	78	99	80	99	87	96	82	97	85
Gradient Boosting Tuned	87	85	94	83	95	95	97	85	91	90

Observation

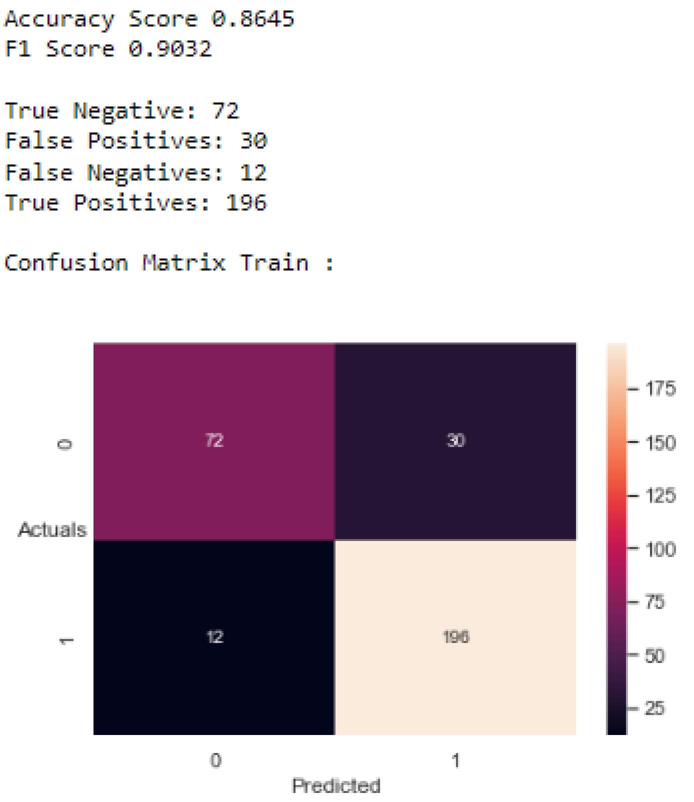
- Their is an over all improvement in the model
- Recall seems to be prefect for both train and test
- Accuracy score also improved
- Precision and Recall also improved

b) ADA Boost

Performance Matrix and Heat Map on the Train data



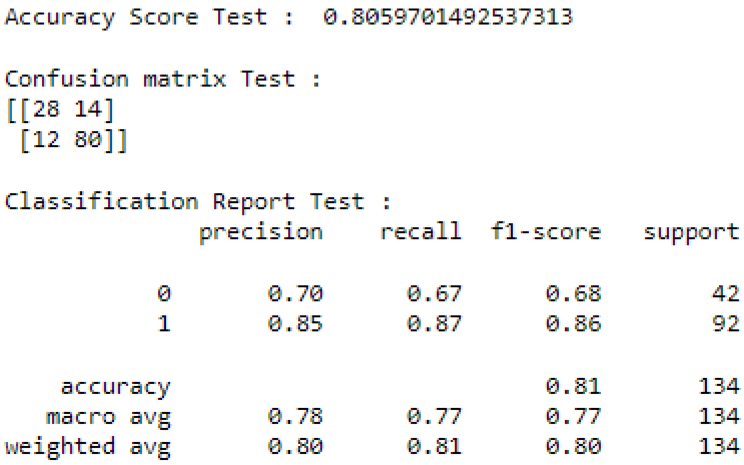
```
Accuracy Score Train :  0.864516129032258

Confusion matrix Train :
[[ 72  30]
 [ 12 196]]

Classification Report Train :
              precision    recall  f1-score   support

           0       0.86      0.71      0.77       102
           1       0.87      0.94      0.90       208

    accuracy                           0.86       310
   macro avg       0.86      0.82      0.84       310
weighted avg       0.86      0.86      0.86       310
```



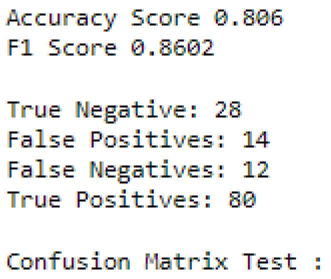
```
Accuracy Score 0.8645
F1 Score 0.9032

True Negative: 72
False Positives: 30
False Negatives: 12
True Positives: 196

Confusion Matrix Train :
```

Heat Map ADA Boost Train

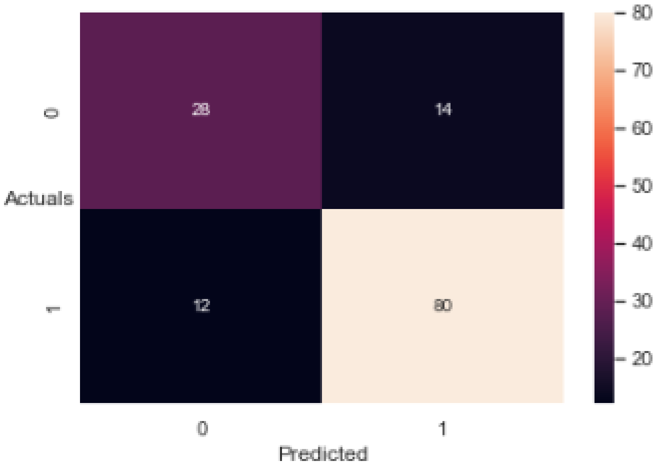
Performance Matrix and Heat Map on the Test data



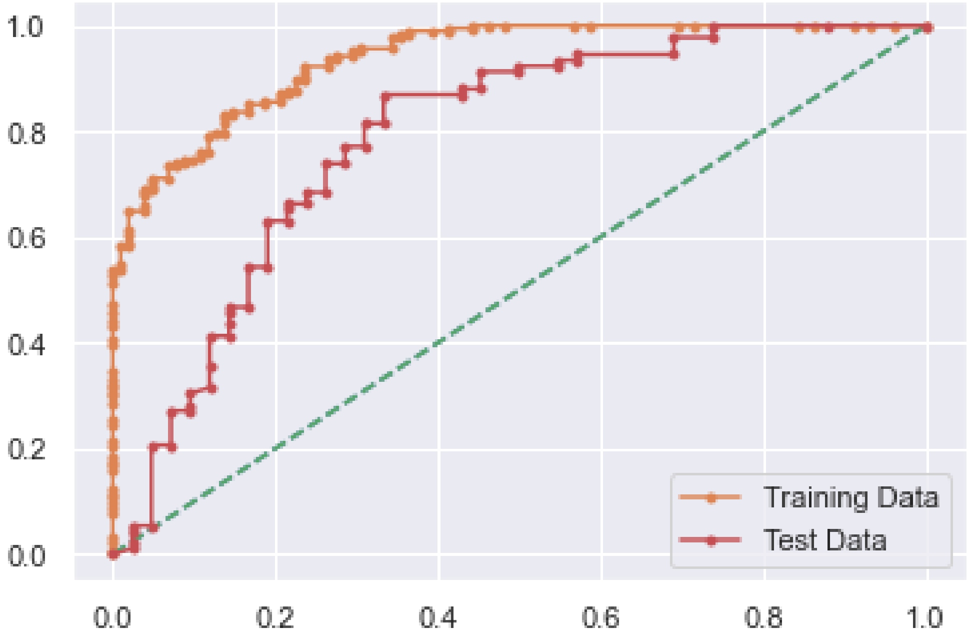
```
Accuracy Score Test :  0.8059701492537313

Confusion matrix Test :
[[28 14]
 [12 80]]

Classification Report Test :
              precision    recall  f1-score   support

           0       0.70      0.67      0.68        42
           1       0.85      0.87      0.86        92

    accuracy                           0.81       134
   macro avg       0.78      0.77      0.77       134
weighted avg       0.80      0.81      0.80       134
```



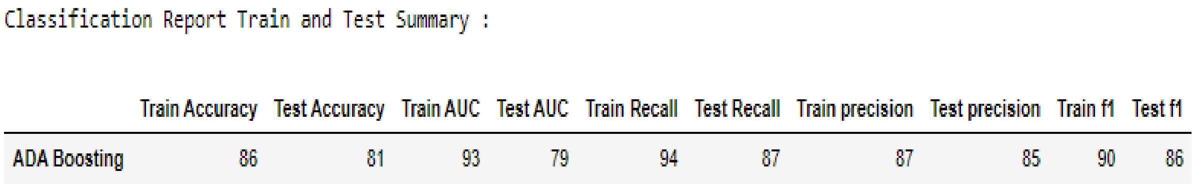
```
Accuracy Score 0.806
F1 Score 0.8602

True Negative: 28
False Positives: 14
False Negatives: 12
True Positives: 80

Confusion Matrix Test :
```



Heat Map ADA Boost Test



AUC ROC Curve ADA Boost Train and Test



Classification Report Train and Test Summary :

	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
ADA Boosting	86	81	93	79	94	87	87	85	90	86

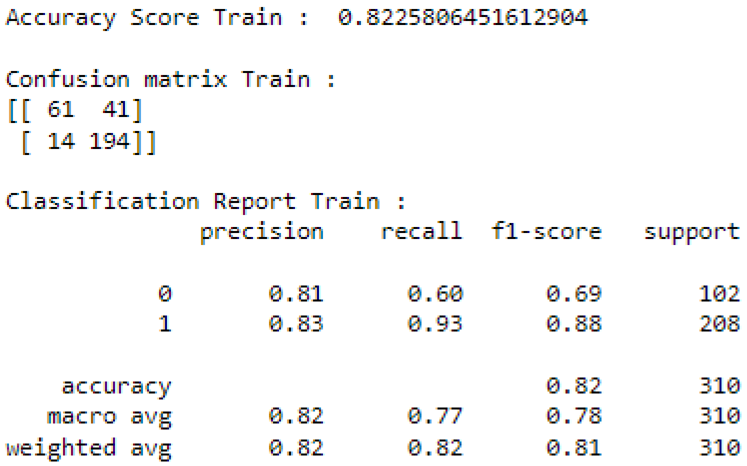
Observation:

- Their seems to be a case of Overfitting of model.
- Huge difference between train and test for Recall and F1 score
- Accuracy and Precision seems decent

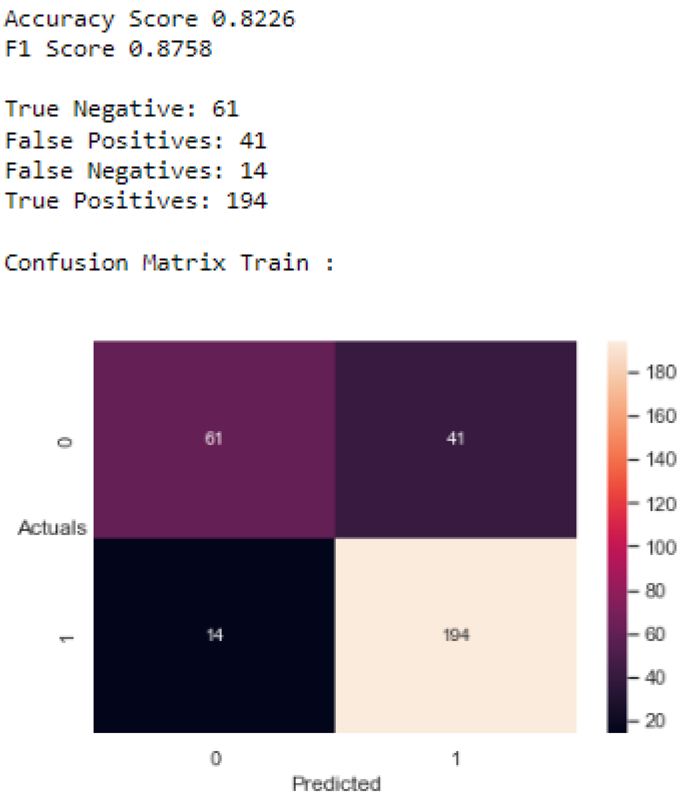
Let’s do some hyperparameter tuning to see if there is any improvement

<u>Hyperparameter tuning of ADA Boosting</u>

Performance Matrix and Heat Map on the Train data



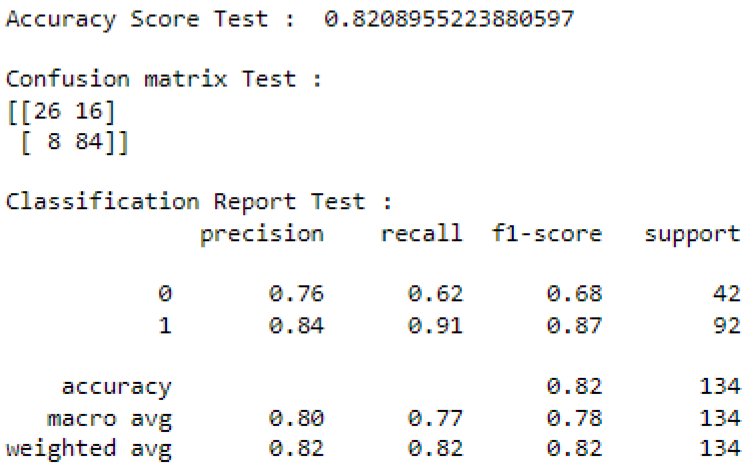
```
Accuracy Score Train :  0.8225806451612904

Confusion matrix Train :
[[ 61  41]
 [ 14 194]]

Classification Report Train :
              precision    recall  f1-score   support

           0       0.81      0.60      0.69       102
           1       0.83      0.93      0.88       208

    accuracy                           0.82       310
   macro avg       0.82      0.77      0.78       310
weighted avg       0.82      0.82      0.81       310
```



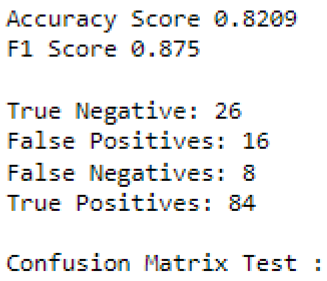
```
Accuracy Score 0.8226
F1 Score 0.8758

True Negative: 61
False Positives: 41
False Negatives: 14
True Positives: 194

Confusion Matrix Train :
```

Heat Map ADA Boost Tuned Train

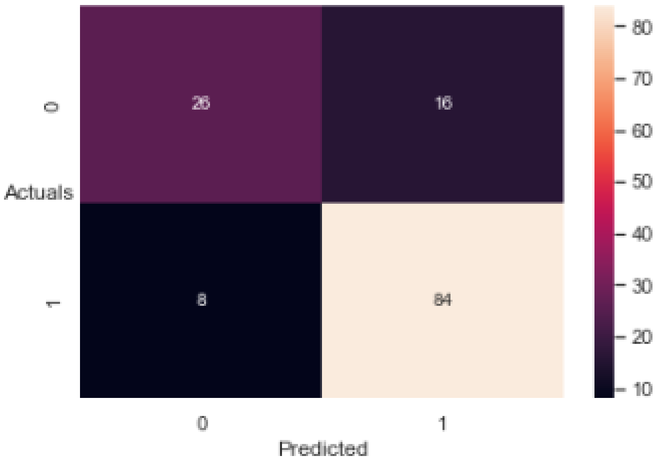
Performance Matrix and Heat Map on the Test data



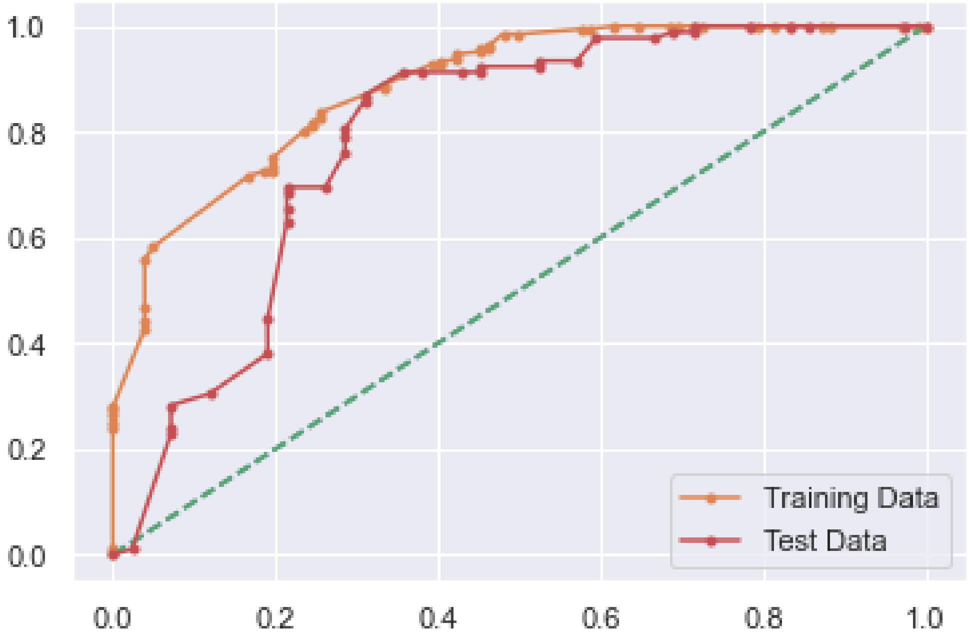
```
Accuracy Score Test :  0.8208955223880597

Confusion matrix Test :
[[26 16]
 [ 8 84]]

Classification Report Test :
              precision    recall  f1-score   support

           0       0.76      0.62      0.68        42
           1       0.84      0.91      0.87        92

    accuracy                           0.82       134
   macro avg       0.80      0.77      0.78       134
weighted avg       0.82      0.82      0.82       134
```



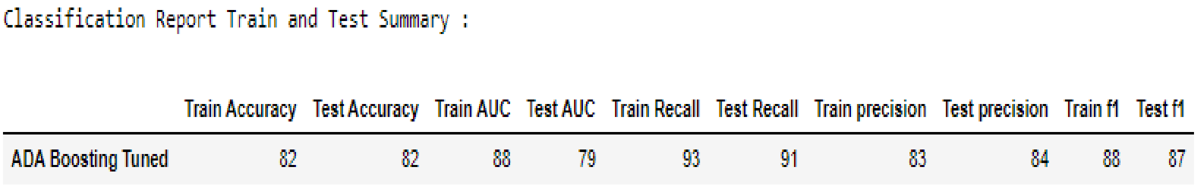
```
Accuracy Score 0.8209
F1 Score 0.875

True Negative: 26
False Positives: 16
False Negatives: 8
True Positives: 84

Confusion Matrix Test :
```



Heat Map ADA Boost Tuned Test



AUC ROC Curve ADA Boos Tuned Train and Test



```
Classification Report Train and Test Summary :
```

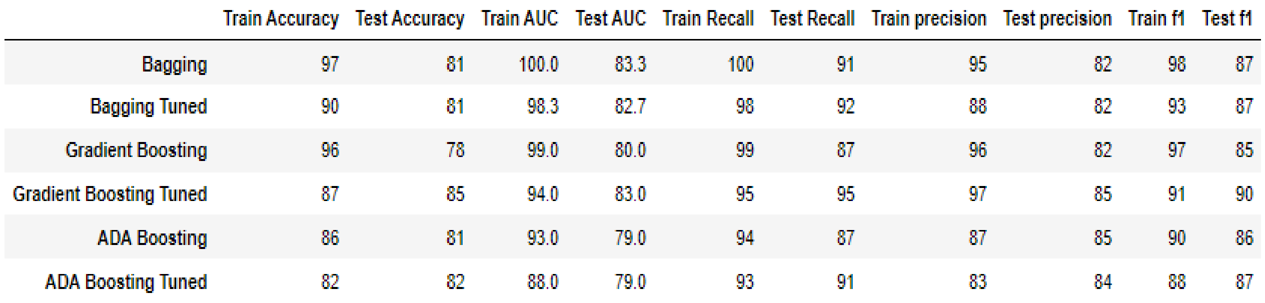
	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
ADA Boosting Tuned	82	82	88	79	93	91	83	84	88	87

ADA Boost Summary Report:

Observation:

- Their is an over all improvement in the model
- Accuracy seems to be prefect for both train and test
- Recall score also improved
- Precision and Recall also improved

Bagging and Boosting Models Summary Report:



	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
Bagging	97	81	100.0	83.3	100	91	95	82	98	87
Bagging Tuned	90	81	98.3	82.7	98	92	88	82	93	87
Gradient Boosting	96	78	99.0	80.0	99	87	96	82	97	85
Gradient Boosting Tuned	87	85	94.0	83.0	95	95	97	85	91	90
ADA Boosting	86	81	93.0	79.0	94	87	87	85	90	86
ADA Boosting Tuned	82	82	88.0	79.0	93	91	83	84	88	87

Observation

- ADA Boosting Model seems to be preforming better than Bagging and Gradient Boosting

In the Ada Boosting Tuned Model:

With accuracy of 82% and recall rate of 91%, model is able to predict 91% of Public Transport which were actually claimed as claimed.

Precision is 84% of data which means, out of total employees predicted by model as opt for Public Transport , 84% employees actually opted for the Public Transport.

F1-score is the harmonic mean of precision and recall, it takes into the effect of both the scores and this value is low if any of these 2 value is low.

Since we are building a model to predict if whether employee will opt for Public Transport or Private Transport, for practical purposes, we will be more interested in correctly classifying 1 (Public Transport) than 0(Private Transport).

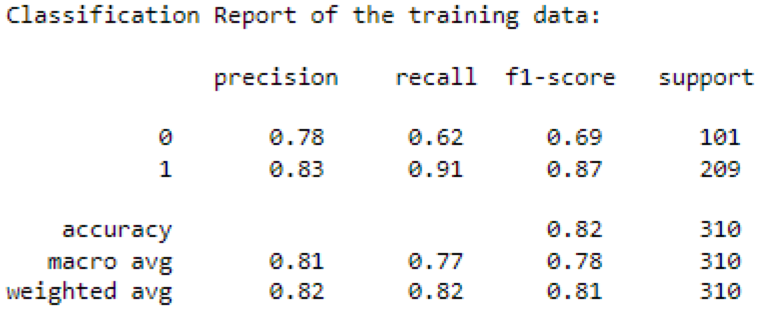
If a employee Private Transport is incorrectly predicted to be "Public Transport" by the model, then the impact on cost for the travel company would be bare minimum. But if am employee opted for Public Transport is incorrectly predicted to be Private Transport by the model, then the cost impact would be very high for the Transport company. Its a loss of potential lead for the company. Hence recall rate (actual data point identified as True by model) is very important in this scenario.

As Recall rate of test dataset is very good around 91% plus Precision rate is also 84% for opting Public transport thus this does looks good enough for classification

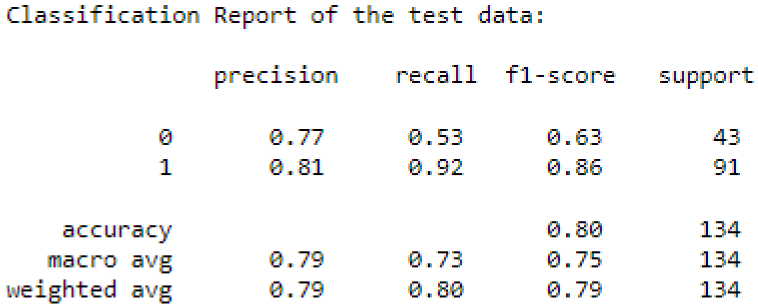
7. Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model. Final Model: Compare the models and write inference which model is best/optimized.

Logistic Regression model

Predictions on Train and Test:

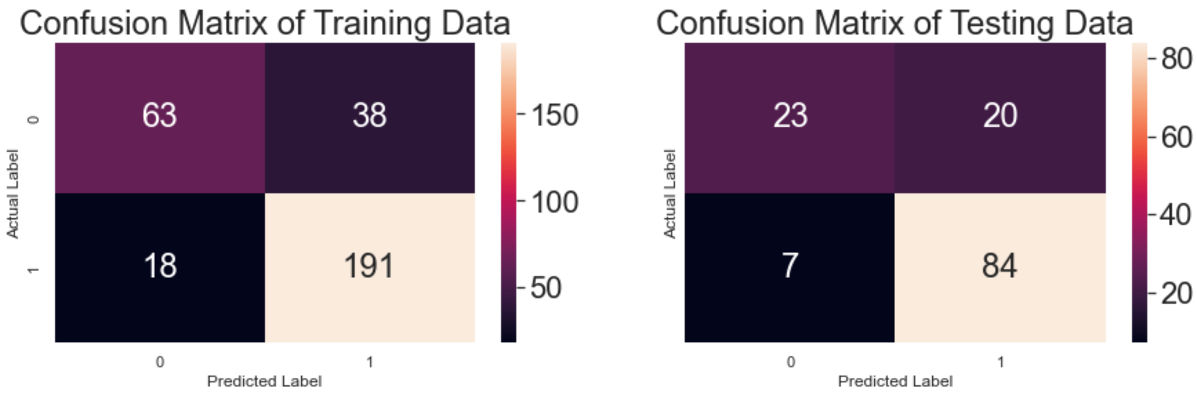


```
Classification Report of the training data:

              precision    recall  f1-score   support

           0       0.78      0.62      0.69       101
           1       0.83      0.91      0.87       209

    accuracy                           0.82       310
   macro avg       0.81      0.77      0.78       310
weighted avg       0.82      0.82      0.81       310
```

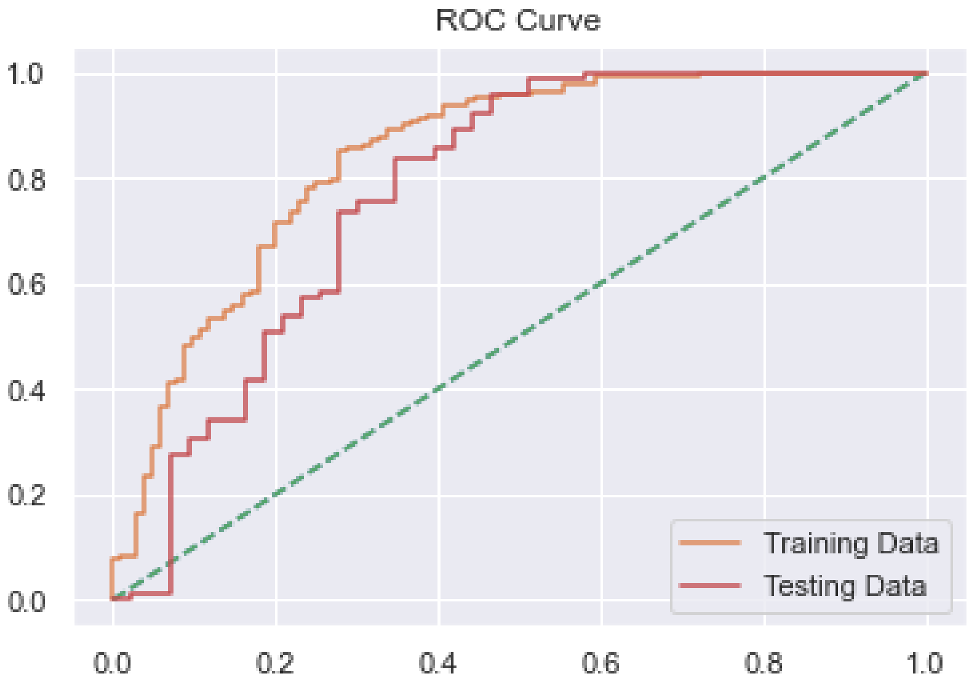


```
Classification Report of the test data:

              precision    recall  f1-score   support

           0       0.77      0.53      0.63        43
           1       0.81      0.92      0.86        91

    accuracy                           0.80       134
   macro avg       0.79      0.73      0.75       134
weighted avg       0.79      0.80      0.79       134
```

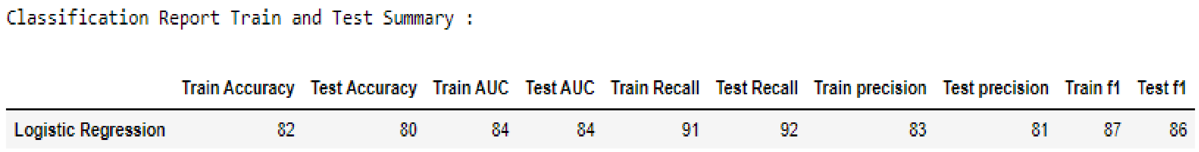
Confusion Matrix Train and Test:



AUC ROC on Train and Test:



Summary Train and Test:



```
Classification Report Train and Test Summary :
```

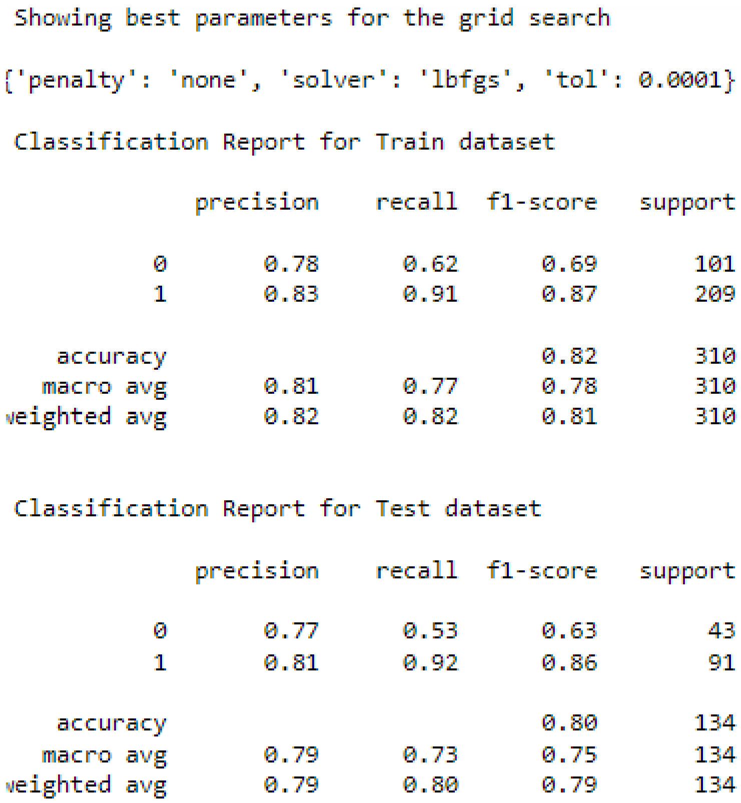
	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
Logistic Regression	82	80	84	84	91	92	83	81	87	86

Observation:

Logistic regression Model seems good with Good Accuracy, Recall , Precision and F1 Scores

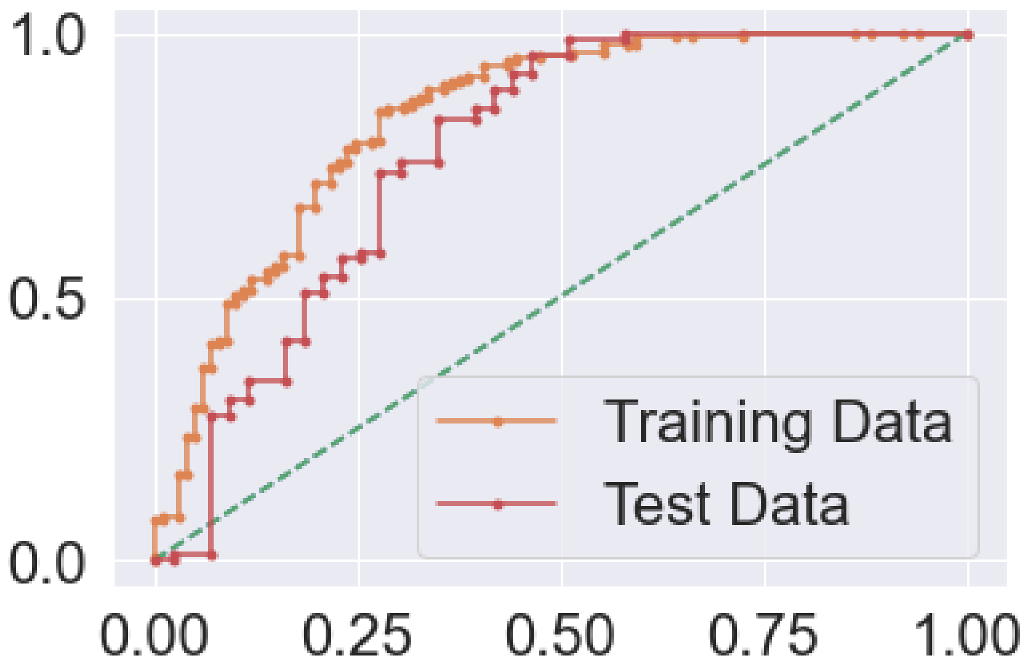
Applying GridSearch CV for Logistic Regression Tuned:

Predictions on Train and Test:

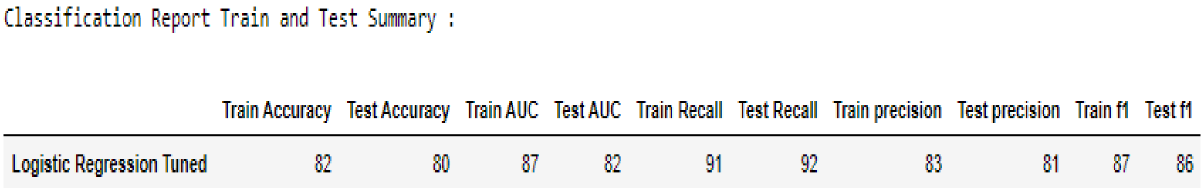


```
 Showing best parameters for the grid search

{'penalty': 'none', 'solver': 'lbfgs', 'tol': 0.0001}

 Classification Report for Train dataset

              precision    recall  f1-score   support

           0       0.78      0.62      0.69       101
           1       0.83      0.91      0.87       209

    accuracy                           0.82       310
   macro avg       0.81      0.77      0.78       310
weighted avg       0.82      0.82      0.81       310


 Classification Report for Test dataset

              precision    recall  f1-score   support

           0       0.77      0.53      0.63        43
           1       0.81      0.92      0.86        91

    accuracy                           0.80       134
   macro avg       0.79      0.73      0.75       134
weighted avg       0.79      0.80      0.79       134
```

AUC ROC Train and Test:



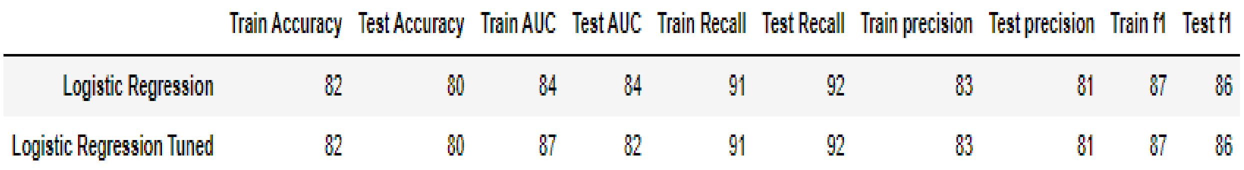
Summary Train and Test:



```
Classification Report Train and Test Summary :
```

	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
Logistic Regression Tuned	82	80	87	82	91	92	83	81	87	86

Logistic Regression Summary:



	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
Logistic Regression	82	80	84	84	91	92	83	81	87	86
Logistic Regression Tuned	82	80	87	82	91	92	83	81	87	86

Observation:

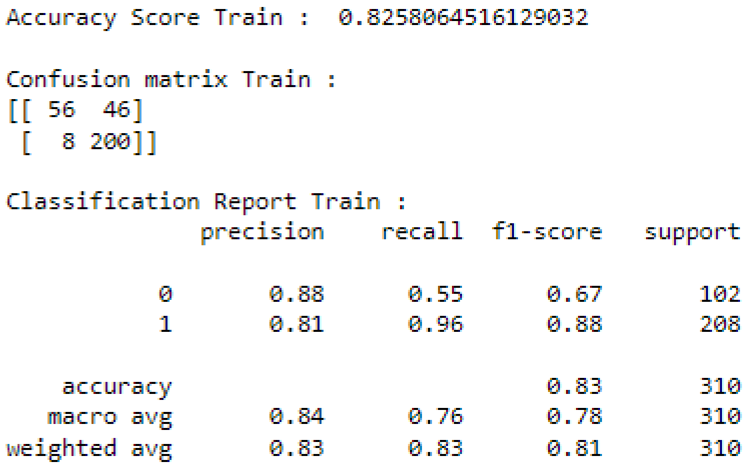
- Even after Logistic regression model tuning results seems similar no change

- Logistic Regression Model seem good

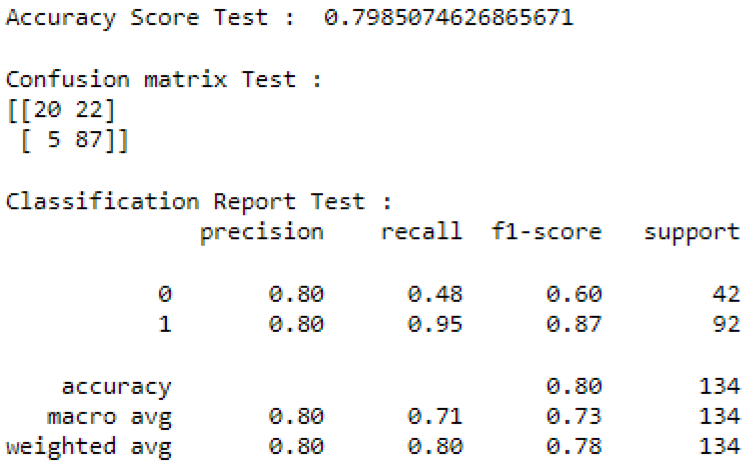
Note: For KNN, Bagging and Boosting full detailed information please check Questions 4 ,5 and 6

<u>KNN Model</u>

Predictions on Train and Test:

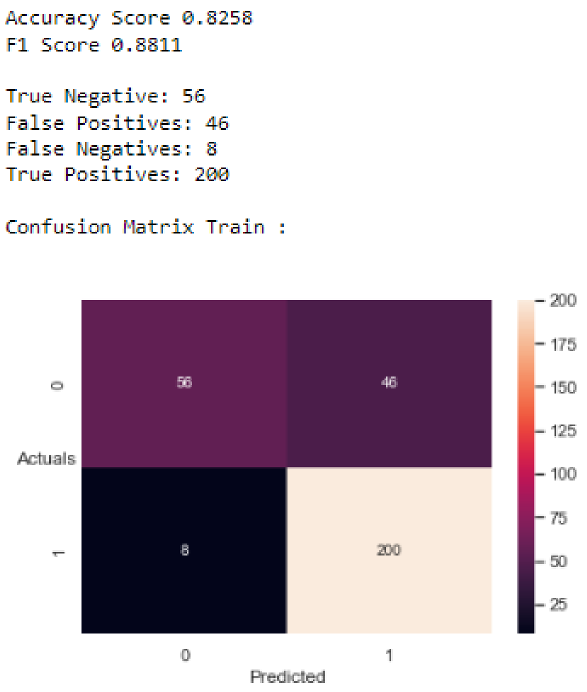


```
Accuracy Score Train :  0.8258064516129032

Confusion matrix Train :
[[ 56  46]
 [  8 200]]

Classification Report Train :
              precision    recall  f1-score   support

           0       0.88      0.55      0.67       102
           1       0.81      0.96      0.88       208

    accuracy                           0.83       310
   macro avg       0.84      0.76      0.78       310
weighted avg       0.83      0.83      0.81       310
```

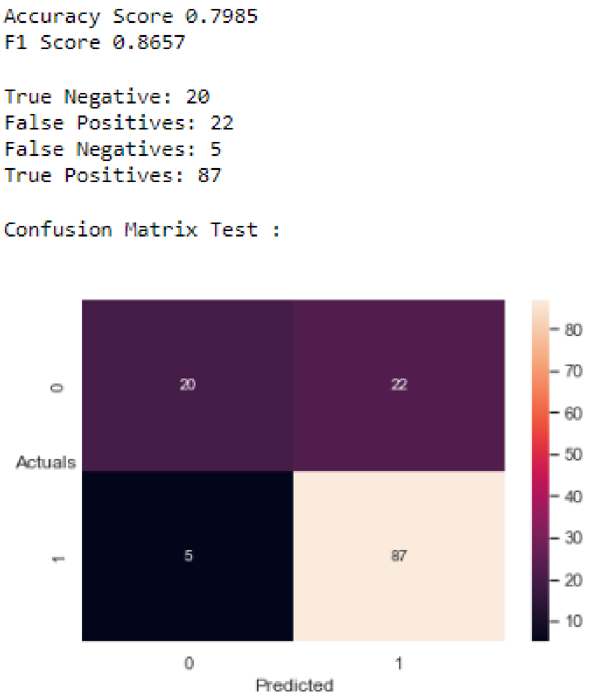


```
Accuracy Score Test :  0.7985074626865671

Confusion matrix Test :
[[20 22]
 [ 5 87]]

Classification Report Test :
              precision    recall  f1-score   support

           0       0.80      0.48      0.60        42
           1       0.80      0.95      0.87        92

    accuracy                           0.80       134
   macro avg       0.80      0.71      0.73       134
weighted avg       0.80      0.80      0.78       134
```

Confusion Matrix Train and Test:



```
Accuracy Score 0.8258
F1 Score 0.8811

True Negative: 56
False Positives: 46
False Negatives: 8
True Positives: 200

Confusion Matrix Train :
```

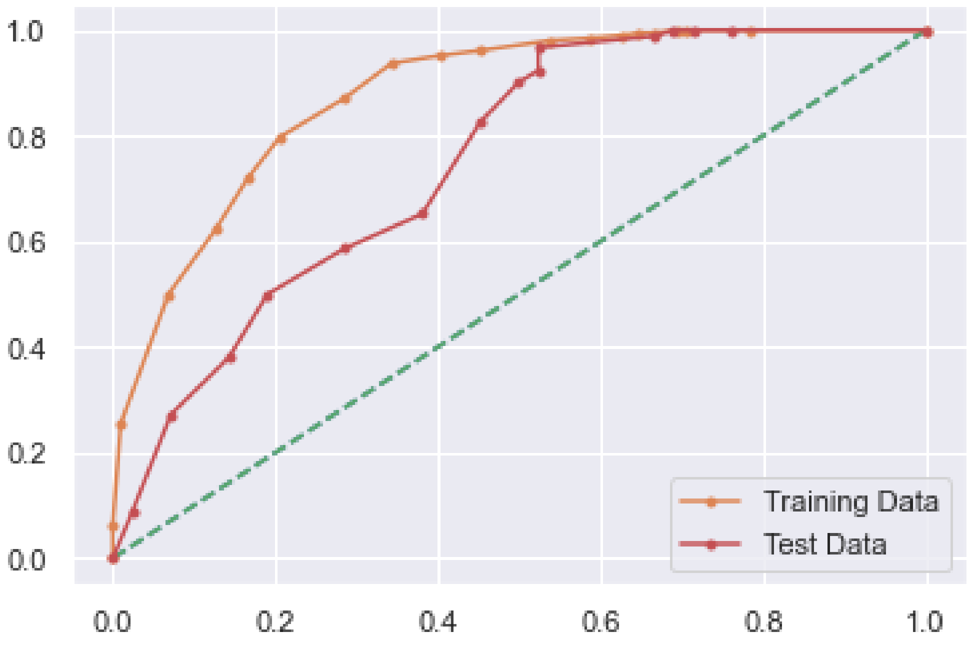


```
Accuracy Score 0.7985
F1 Score 0.8657

True Negative: 20
False Positives: 22
False Negatives: 5
True Positives: 87

Confusion Matrix Test :
```

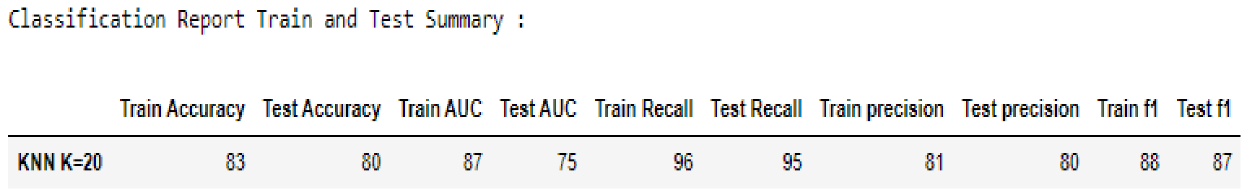
Confusion Matrix Train

Confusion Matrix test

AUC ROC on Train and Test:



Summary Train and Test:



```
Classification Report Train and Test Summary :
```

	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
KNN K=20	83	80	87	75	96	95	81	80	88	87

Observation :

When we took **K-value = 20**

Then with accuracy of 80% and recall rate of 95%, model is able to predict 95% of Public Transport which were actually claimed as claimed.

Precision is 80% of data which means, out of total employees predicted by model as opt for Public Transport , 80% employees actually opted for the Public Transport.

F1-score is the harmonic mean of precision and recall, it takes into the effect of both the scores and this value is low if any of these 2 value is low.

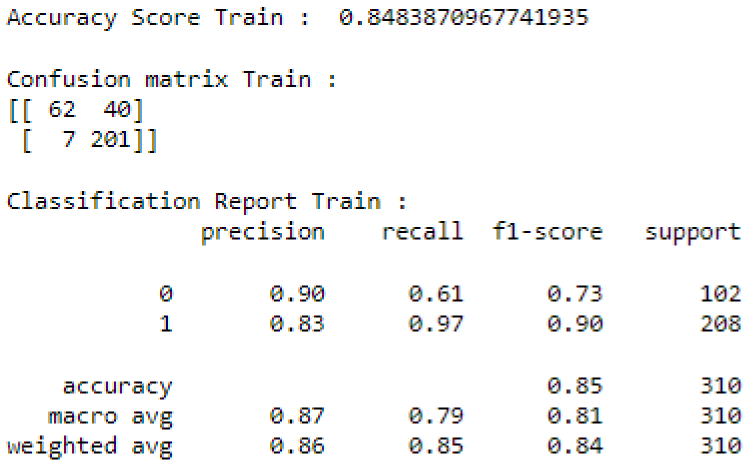
Since we are building a model to predict if whether employee will opt for Public Transport or Private Transport, for practical purposes, we will be more interested in correctly classifying 1 (Public Transport) than 0(Private Transport).

If a employee Private Transport is incorrectly predicted to be "Public Transport" by the model, then the impact on cost for the travel company would be bare minimum. But if am employee opted for Public Transport is incorrectly predicted to be Private Transport by the model, then the cost impact would be very high for the Transport company. Its a loss of potential lead for the company. Hence recall rate (actual data point identified as True by model) is very important in this scenario.

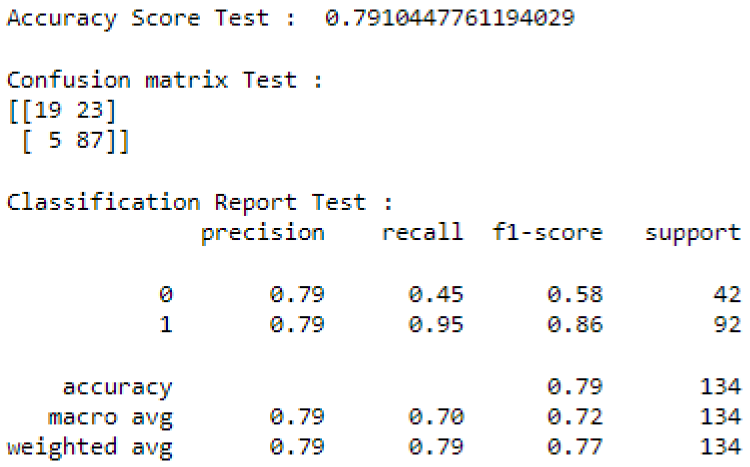
As Recall rate of test dataset is very good around 95% plus Precision rate is also 80% for opting Public transport thus this does looks good enough for classification

Applying GridSearch CV on KNN:

Predictions on Train and Test:

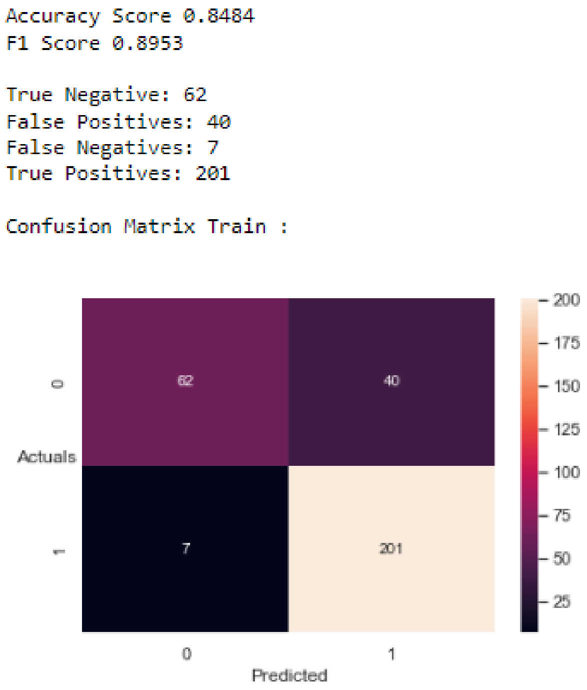


```
Accuracy Score Train :  0.8483870967741935

Confusion matrix Train :
[[ 62  40]
 [  7 201]]

Classification Report Train :
              precision    recall  f1-score   support

           0       0.90      0.61      0.73       102
           1       0.83      0.97      0.90       208

    accuracy                           0.85       310
   macro avg       0.87      0.79      0.81       310
weighted avg       0.86      0.85      0.84       310
```

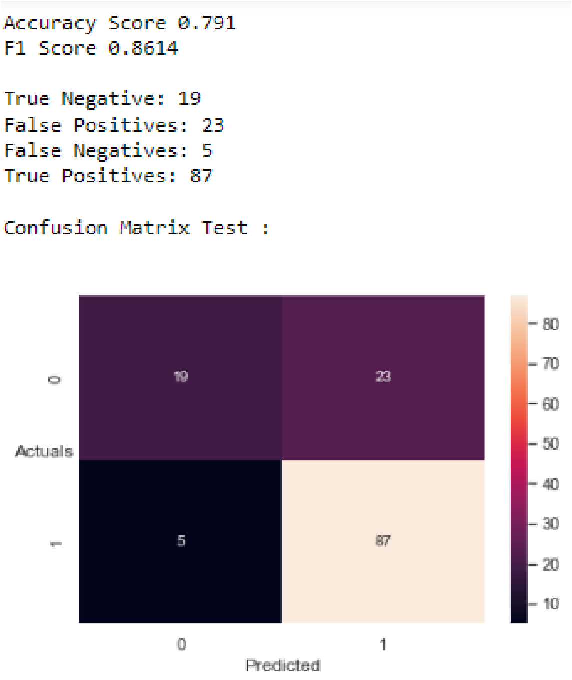


```
Accuracy Score Test :  0.7910447761194029

Confusion matrix Test :
[[19 23]
 [ 5 87]]

Classification Report Test :
              precision    recall  f1-score   support

           0       0.79      0.45      0.58        42
           1       0.79      0.95      0.86        92

    accuracy                           0.79       134
   macro avg       0.79      0.70      0.72       134
weighted avg       0.79      0.79      0.77       134
```

Confusion Matrix Train and Test:



```
Accuracy Score 0.8484
F1 Score 0.8953

True Negative: 62
False Positives: 40
False Negatives: 7
True Positives: 201

Confusion Matrix Train :
```

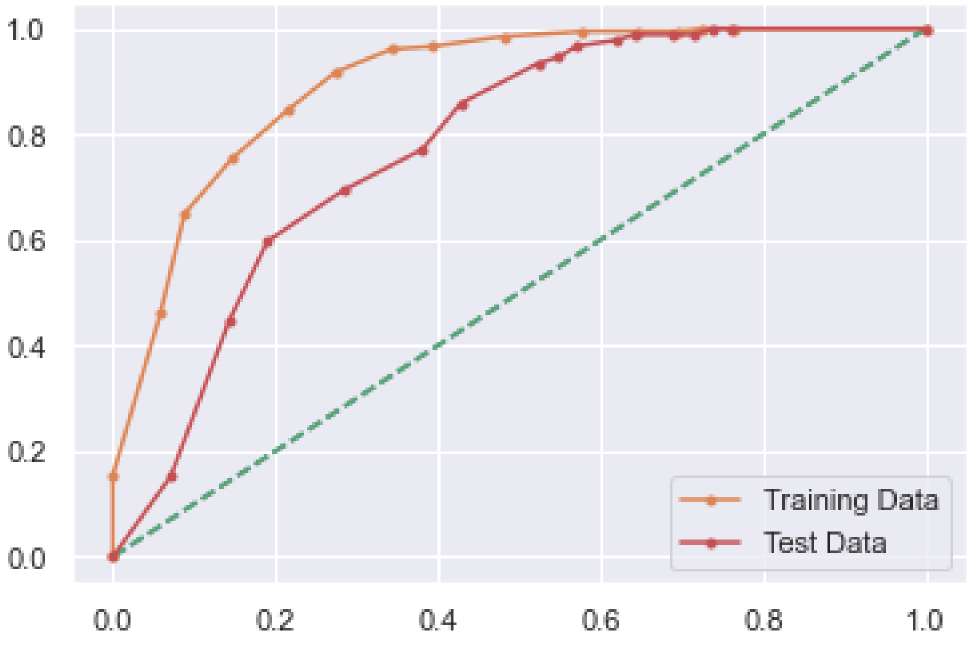


```
Accuracy Score 0.791
F1 Score 0.8614

True Negative: 19
False Positives: 23
False Negatives: 5
True Positives: 87

Confusion Matrix Test :
```

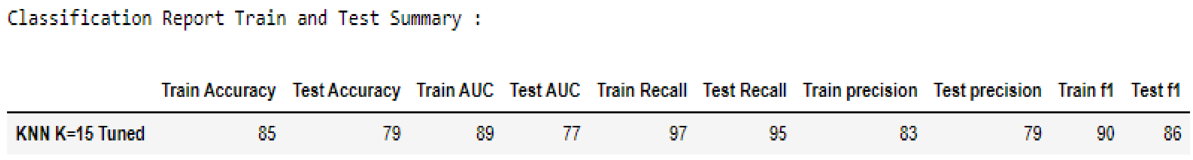
Confusion Matrix Train

Confusion Matrix test

AUC ROC on Train and Test:



Summary Train and Test:



```
Classification Report Train and Test Summary :
```

	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
KNN K=15 Tuned	85	79	89	77	97	95	83	79	90	86

Observation :

When we took **K-value = 15**

Then with accuracy of 79% and recall rate of 95%, model is able to predict 95% of Public Transport which were actually claimed as claimed.

Precision is 79% of data which means, out of total employees predicted by model as opt for Public Transport , 79% employees actually opted for the Public Transport.

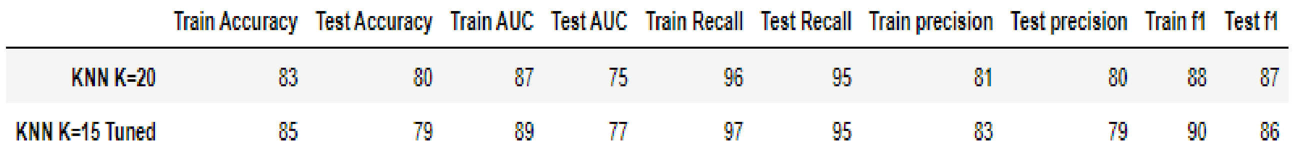
F1-score is the harmonic mean of precision and recall, it takes into the effect of both the scores and this value is low if any of these 2 value is low.

Since we are building a model to predict if whether employee will opt for Public Transport or Private Transport, for practical purposes, we will be more interested in correctly classifying 1 (Public Transport) than 0(Private Transport).

If a employee Private Transport is incorrectly predicted to be "Public Transport" by the model, then the impact on cost for the travel company would be bare minimum. But if am employee opted for Public Transport is incorrectly predicted to be Private Transport by the model, then the cost impact would be very high for the Transport company. Its a loss of potential lead for the company. Hence recall rate (actual data point identified as True by model) is very important in this scenario.

As Recall rate of test dataset is very good around 95% plus Precision rate is also 79% for opting Public transport thus this does looks good enough for classification

KNN Summary:



	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
KNN K=20	83	80	87	75	96	95	81	80	88	87
KNN K=15 Tuned	85	79	89	77	97	95	83	79	90	86

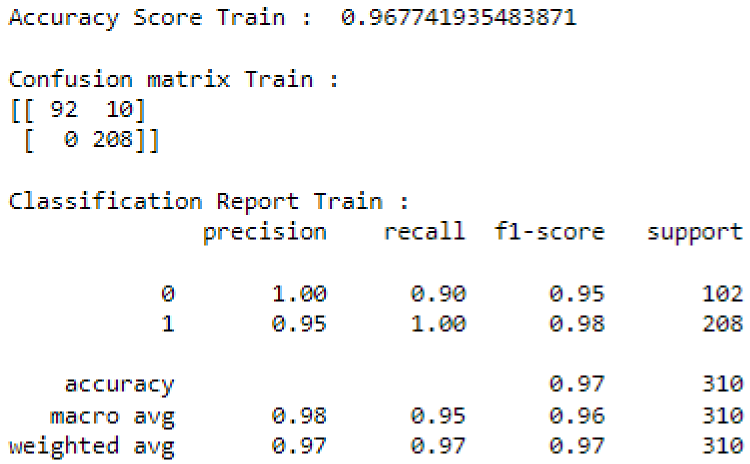
Observation:

We can clearly see that even after Hyperparameter tunning of KNN Model(K=15) there is a decrease in the model performance.

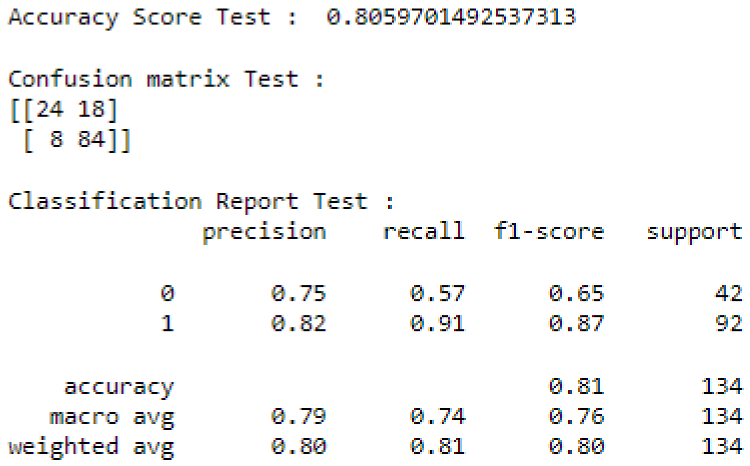
Thus K=20 gives the best output for KNN Model.

BAGGING (Random Forest)

Predictions on Train and Test:

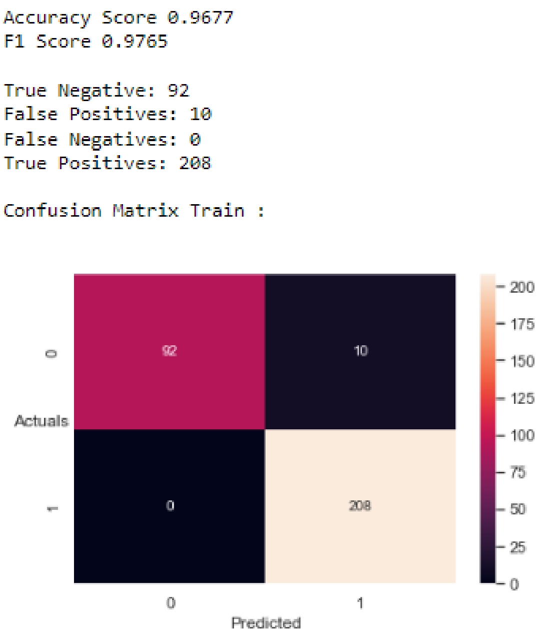


```
Accuracy Score Train :  0.967741935483871

Confusion matrix Train :
[[ 92  10]
 [  0 208]]

Classification Report Train :
              precision    recall  f1-score   support

           0       1.00      0.90      0.95       102
           1       0.95      1.00      0.98       208

    accuracy                           0.97       310
   macro avg       0.98      0.95      0.96       310
weighted avg       0.97      0.97      0.97       310
```

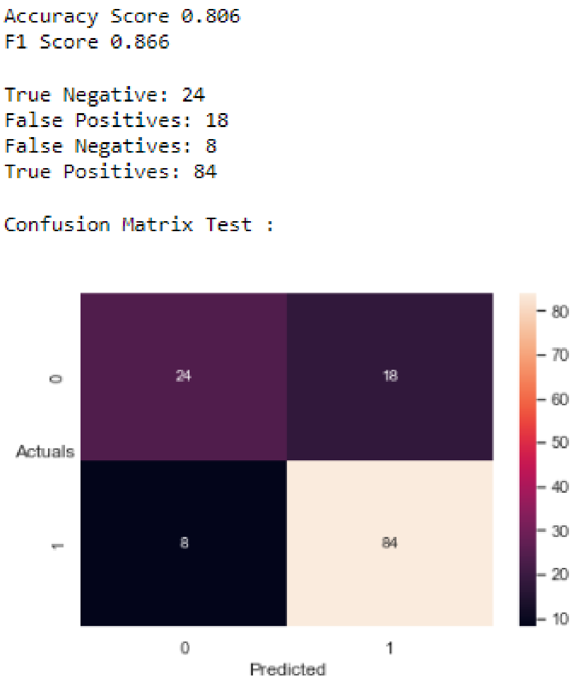


```
Accuracy Score Test :  0.8059701492537313

Confusion matrix Test :
[[24 18]
 [ 8 84]]

Classification Report Test :
              precision    recall  f1-score   support

           0       0.75      0.57      0.65        42
           1       0.82      0.91      0.87        92

    accuracy                           0.81       134
   macro avg       0.79      0.74      0.76       134
weighted avg       0.80      0.81      0.80       134
```

Confusion Matrix Train and Test:



```
Accuracy Score 0.9677
F1 Score 0.9765

True Negative: 92
False Positives: 10
False Negatives: 0
True Positives: 208

Confusion Matrix Train :
```

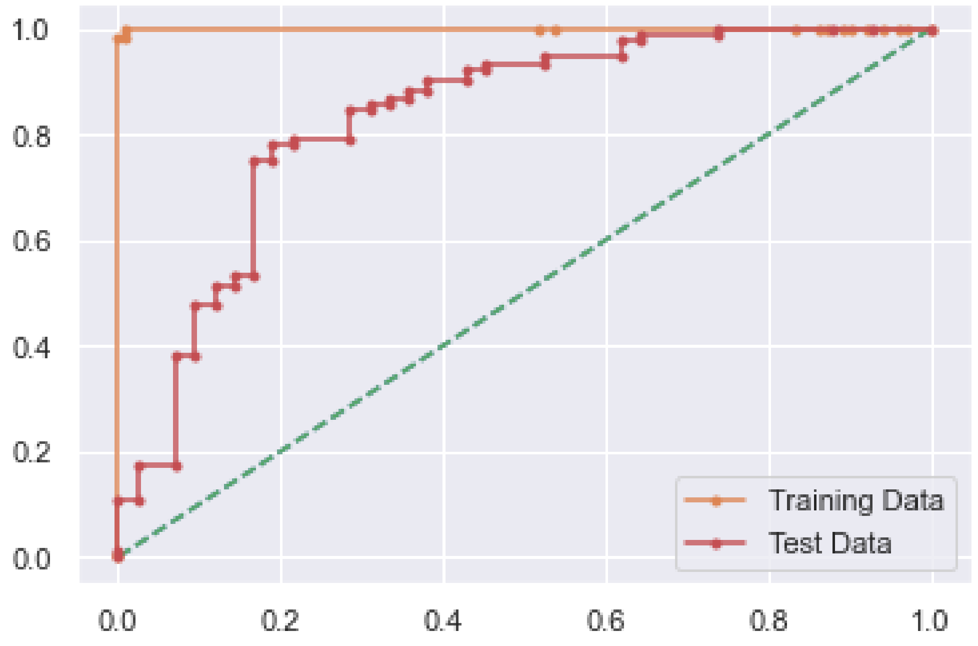


```
Accuracy Score 0.806
F1 Score 0.866

True Negative: 24
False Positives: 18
False Negatives: 8
True Positives: 84

Confusion Matrix Test :
```

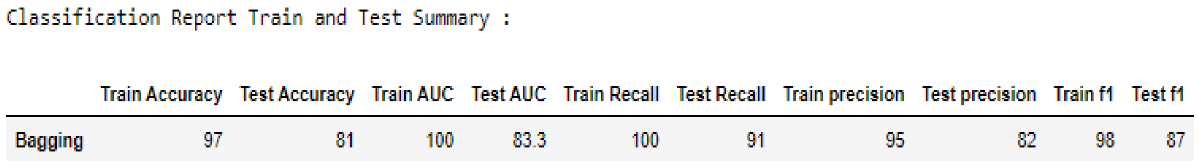
Confusion Matrix Train

Confusion Matrix test

AUC ROC on Train and Test:



Summary Train and Test:



```
Classification Report Train and Test Summary :
```

	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
Bagging	97	81	100	83.3	100	91	95	82	98	87

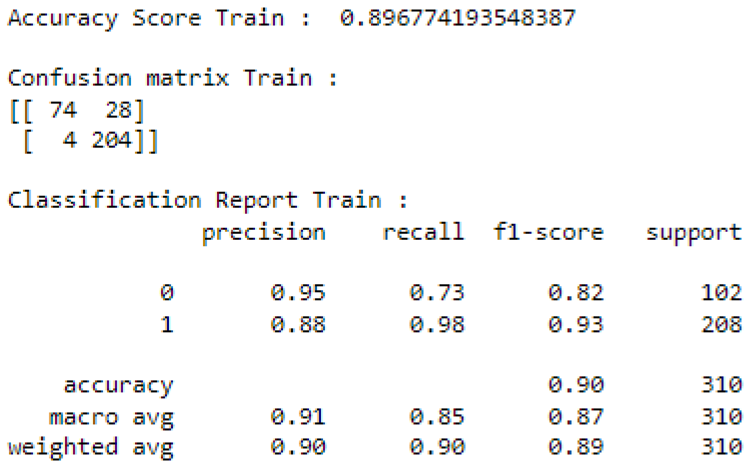
Observation

- The overall accuracy (train 97% and test 81%) seems to be decent using bagging but , there is a huge difference between test/train accuracy value.
- Similarly Model Recall score shows 100% in train and 91% Test which is decent.
- Precision also have huge difference which means the model is not able to precisely tell who all the employee opting for 1(Public Transport)

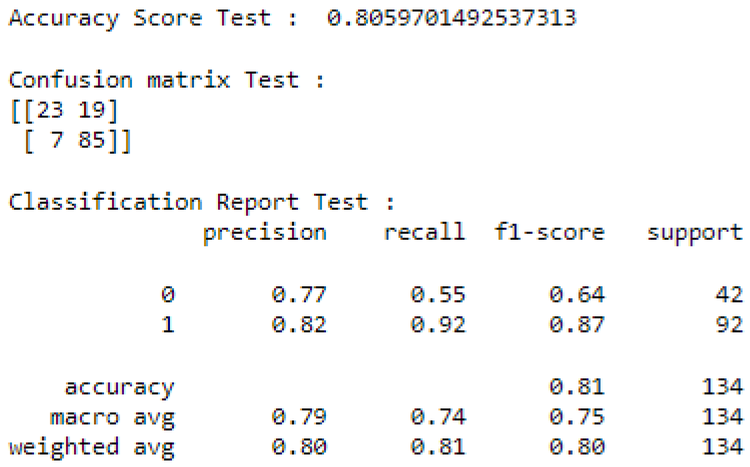
Therefore there seems to be Overfitting which can be rectified by Hyperparameter tuning of the Random Forest Classifier.

Applying GridSearch CV on Bagging(Random Forest):

Predictions on Train and Test:

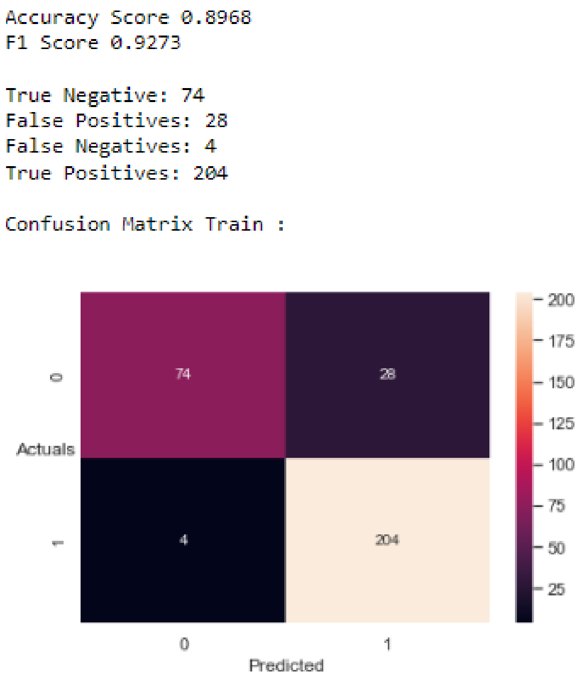


```
Accuracy Score Train :  0.896774193548387

Confusion matrix Train :
[[ 74  28]
 [  4 204]]

Classification Report Train :
              precision    recall  f1-score   support

           0       0.95      0.73      0.82       102
           1       0.88      0.98      0.93       208

    accuracy                           0.90       310
   macro avg       0.91      0.85      0.87       310
weighted avg       0.90      0.90      0.89       310
```



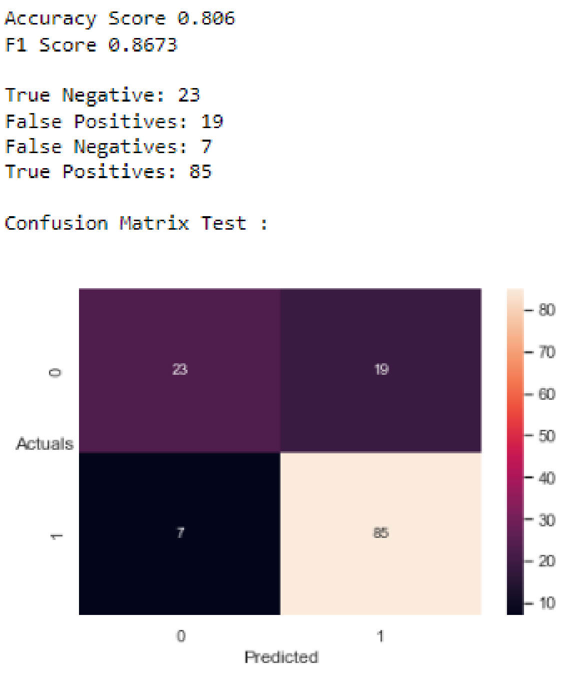
```
Accuracy Score Test :  0.8059701492537313

Confusion matrix Test :
[[23 19]
 [ 7 85]]

Classification Report Test :
              precision    recall  f1-score   support

           0       0.77      0.55      0.64        42
           1       0.82      0.92      0.87        92

    accuracy                           0.81       134
   macro avg       0.79      0.74      0.75       134
weighted avg       0.80      0.81      0.80       134
```

Confusion Matrix Train and Test:



```
Accuracy Score 0.8968
F1 Score 0.9273

True Negative: 74
False Positives: 28
False Negatives: 4
True Positives: 204

Confusion Matrix Train :
```

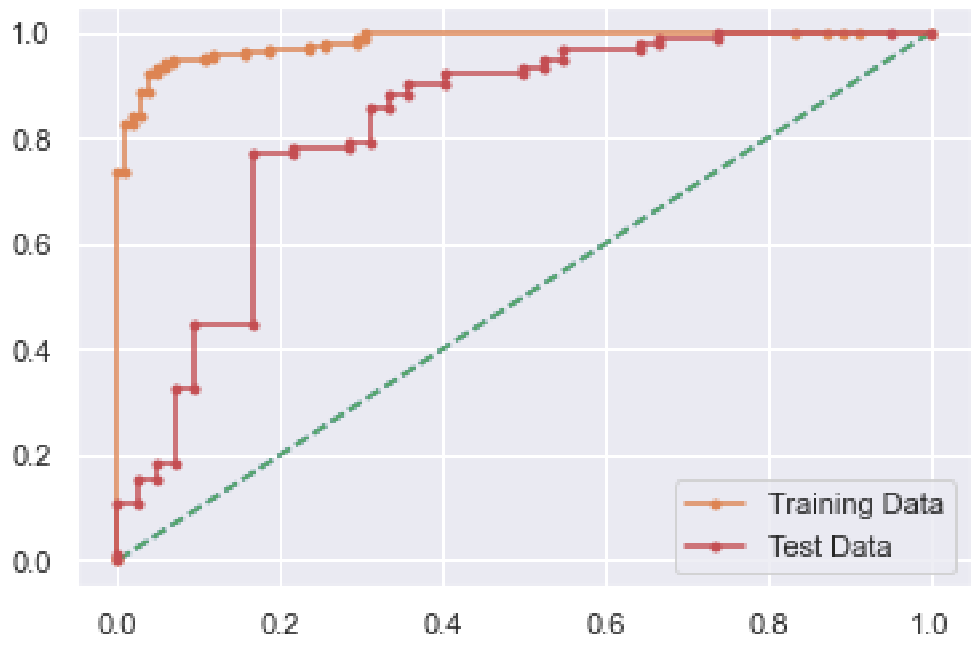
Confusion Matrix Train



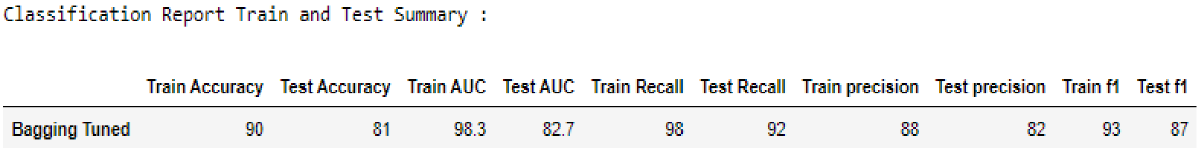
```
Accuracy Score 0.806
F1 Score 0.8673

True Negative: 23
False Positives: 19
False Negatives: 7
True Positives: 85

Confusion Matrix Test :
```

Confusion Matrix test

AUC ROC on Train and Test:



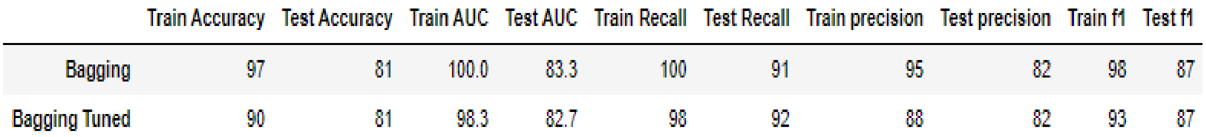
Summary Train and Test:



```
Classification Report Train and Test Summary :
```

	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
Bagging Tuned	90	81	98.3	82.7	98	92	88	82	93	87

Bagging Summary:



	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
Bagging	97	81	100.0	83.3	100	91	95	82	98	87
Bagging Tuned	90	81	98.3	82.7	98	92	88	82	93	87

Observation:

Here we can clearly interpret an improvement in the model.

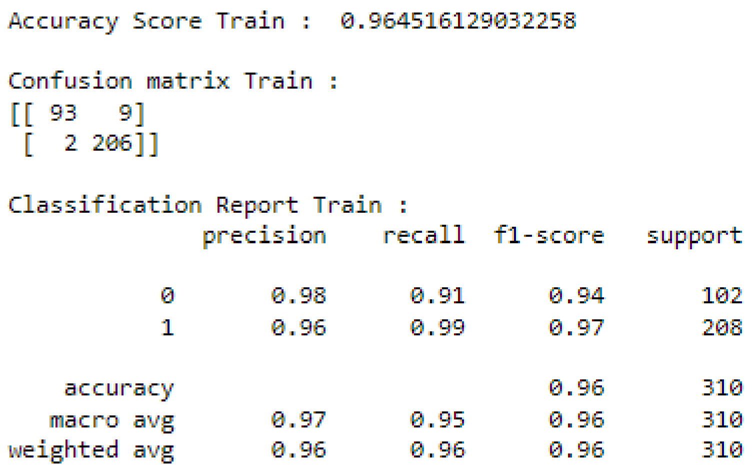
- Accuracy score seems decent(train 90% and test 81%) not much improvement.
- There seems to be a good improvement in Recall score(train 98% and test 92%) which means model is able to properly recall the no. of employees opting for 1(Public Transport).

- Also a very a good improvement in the Precision rate(train 88% and test 82%) which means model is able to precisely tell who are the employee that are opting for 1(Public Transport)

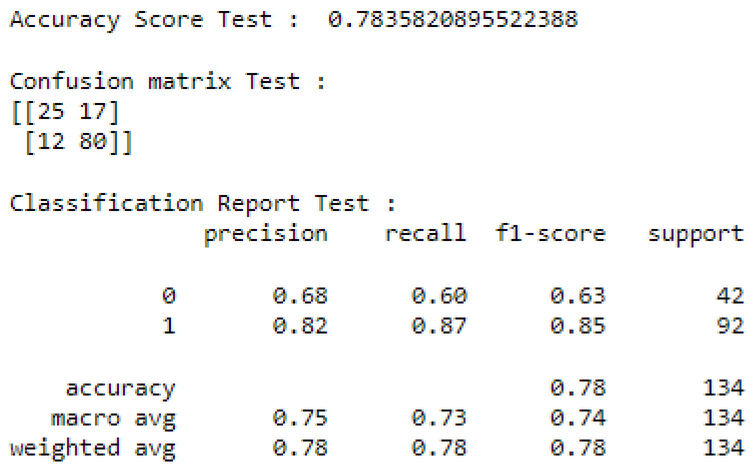
A good improvement in the AUC also

Gradient Boost (Boosting)

Predictions on Train and Test:

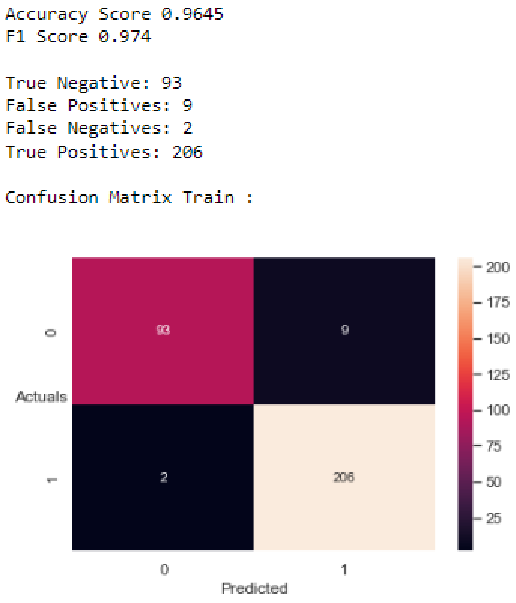


```
Accuracy Score Train :  0.964516129032258

Confusion matrix Train :
[[ 93   9]
 [  2 206]]

Classification Report Train :
              precision    recall  f1-score   support

           0       0.98      0.91      0.94       102
           1       0.96      0.99      0.97       208

    accuracy                           0.96       310
   macro avg       0.97      0.95      0.96       310
weighted avg       0.96      0.96      0.96       310
```

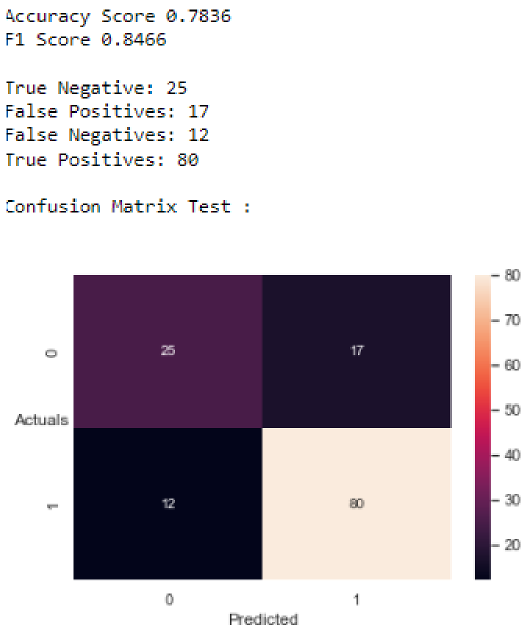


```
Accuracy Score Test :  0.7835820895522388

Confusion matrix Test :
[[25 17]
 [12 80]]

Classification Report Test :
              precision    recall  f1-score   support

           0       0.68      0.60      0.63        42
           1       0.82      0.87      0.85        92

    accuracy                           0.78       134
   macro avg       0.75      0.73      0.74       134
weighted avg       0.78      0.78      0.78       134
```

Confusion Matrix Train and Test:



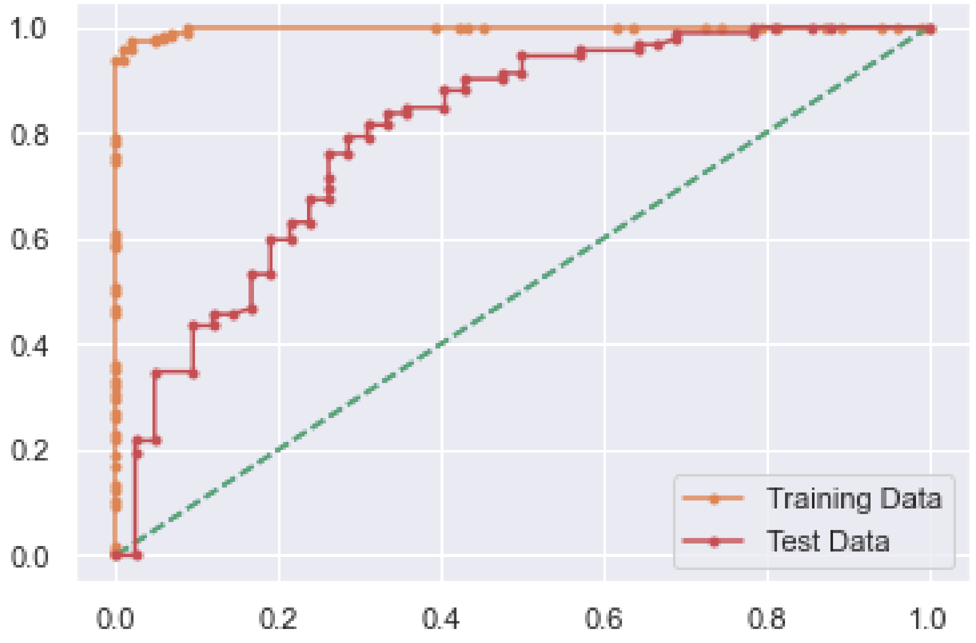
```
Accuracy Score 0.9645
F1 Score 0.974

True Negative: 93
False Positives: 9
False Negatives: 2
True Positives: 206

Confusion Matrix Train :
```



```
Accuracy Score 0.7836
F1 Score 0.8466

True Negative: 25
False Positives: 17
False Negatives: 12
True Positives: 80

Confusion Matrix Test :
```

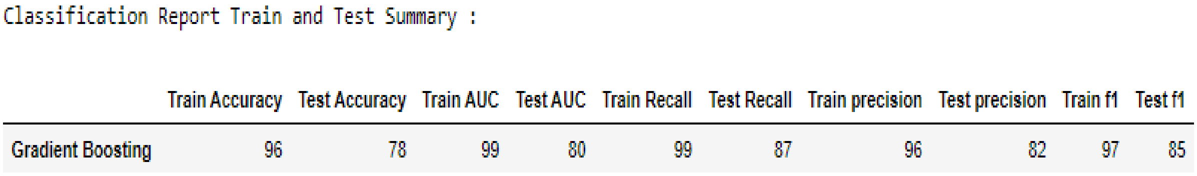
Confusion Matrix Train *Confusion Matrix test*

AUC ROC on Train and Test:



AUC ROC Curve Gradient Boost Train and Test

Summary Train and Test:



```
Classification Report Train and Test Summary :
```

	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
Gradient Boosting	96	78	99	80	99	87	96	82	97	85

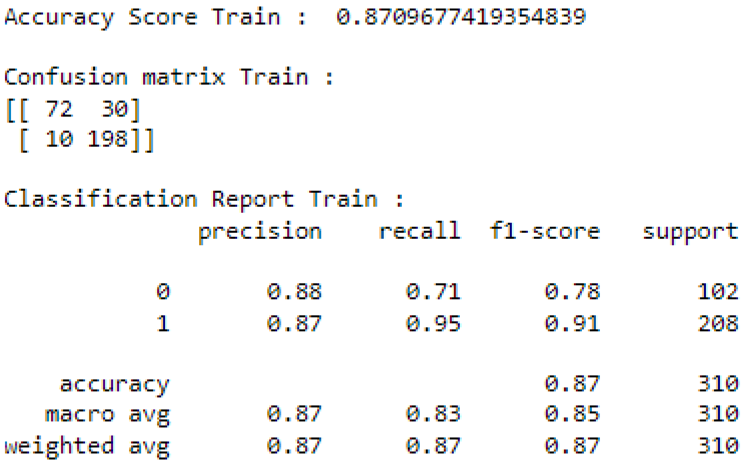
Observation :

- Their seems to be clear case of Overfitting of model.
- Huge difference between train and test for Accuracy, Recall, Precision, and F1 score

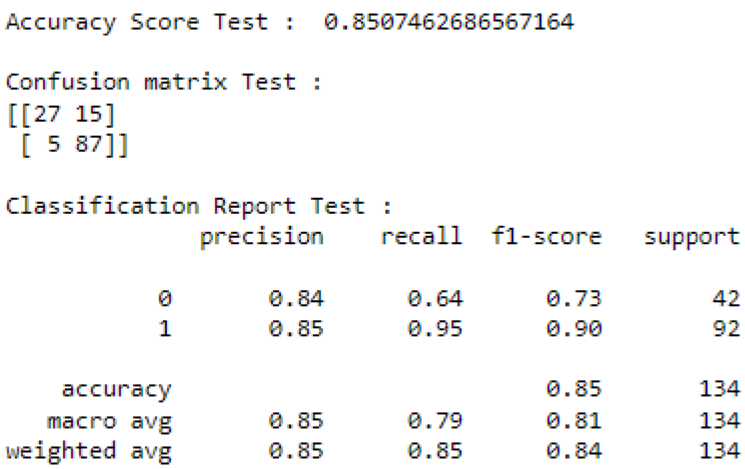
Gradient Boosting model is not properly grasp the data need to do proper Hyperparameter Tuning.

Applying GridSearch CV on Gradient Boost:

Predictions on Train and Test:

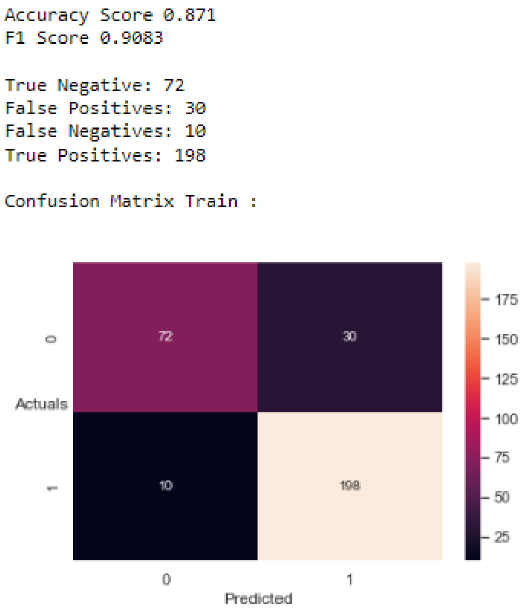


```
Accuracy Score Train :  0.8709677419354839

Confusion matrix Train :
[[ 72  30]
 [ 10 198]]

Classification Report Train :
              precision    recall  f1-score   support

           0       0.88      0.71      0.78       102
           1       0.87      0.95      0.91       208

    accuracy                           0.87       310
   macro avg       0.87      0.83      0.85       310
weighted avg       0.87      0.87      0.87       310
```

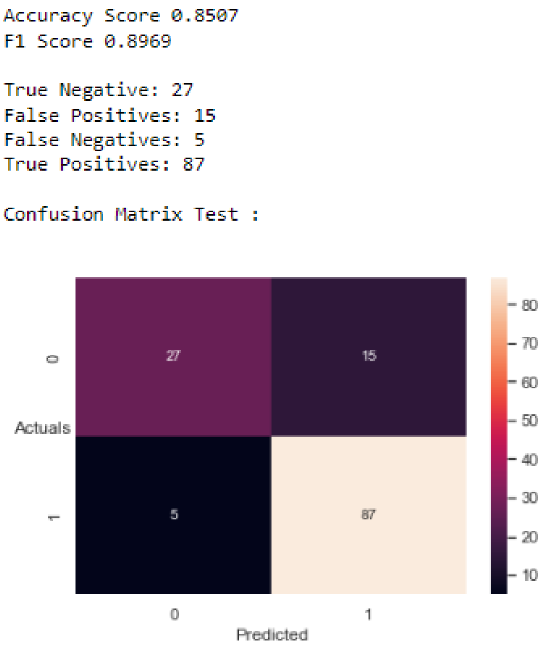


```
Accuracy Score Test :  0.8507462686567164

Confusion matrix Test :
[[27 15]
 [ 5 87]]

Classification Report Test :
              precision    recall  f1-score   support

           0       0.84      0.64      0.73        42
           1       0.85      0.95      0.90        92

    accuracy                           0.85       134
   macro avg       0.85      0.79      0.81       134
weighted avg       0.85      0.85      0.84       134
```

Confusion Matrix Train and Test:



```
Accuracy Score 0.871
F1 Score 0.9083

True Negative: 72
False Positives: 30
False Negatives: 10
True Positives: 198

Confusion Matrix Train :
```

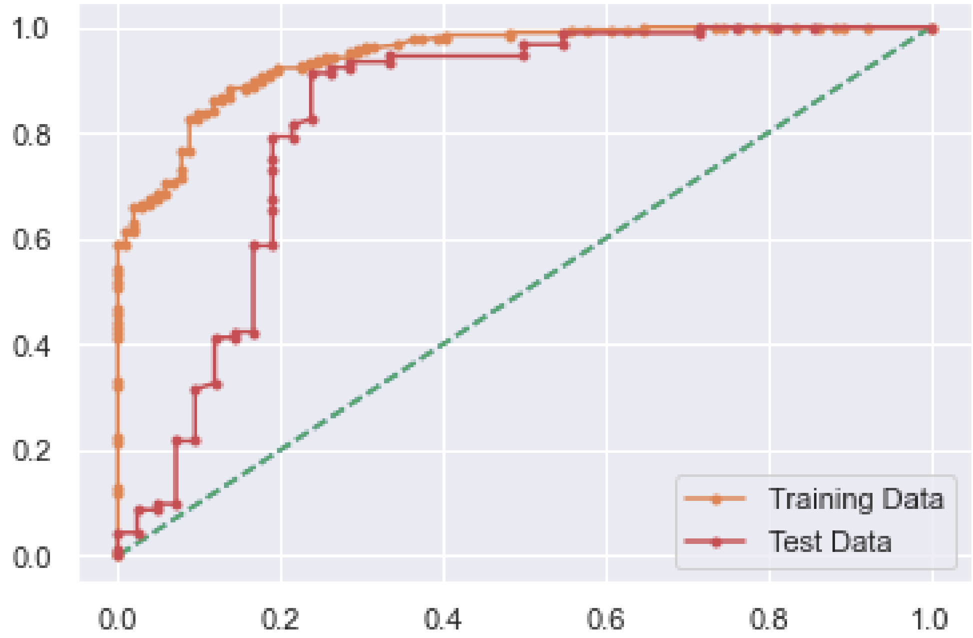


```
Accuracy Score 0.8507
F1 Score 0.8969

True Negative: 27
False Positives: 15
False Negatives: 5
True Positives: 87

Confusion Matrix Test :
```

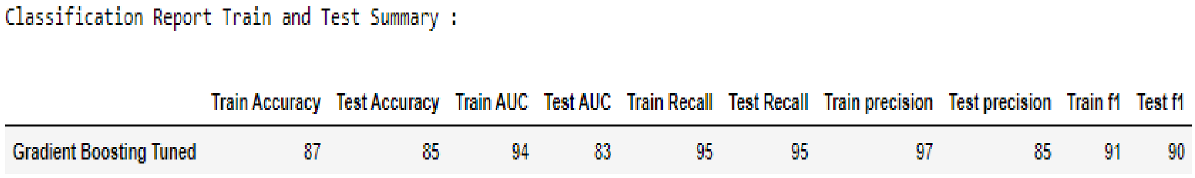
Confusion Matrix Train

Confusion Matrix test

AUC ROC on Train and Test:

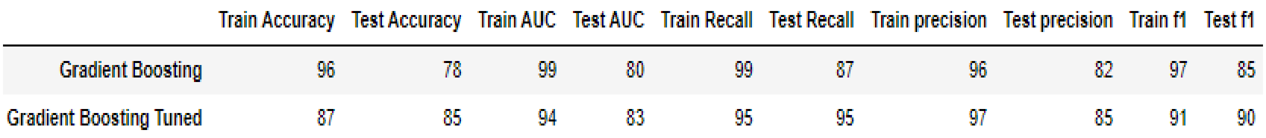


Summary Train and Test:


```
Classification Report Train and Test Summary :
```

	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
Gradient Boosting Tuned	87	85	94	83	95	95	97	85	91	90

Gradient Boost Summary:



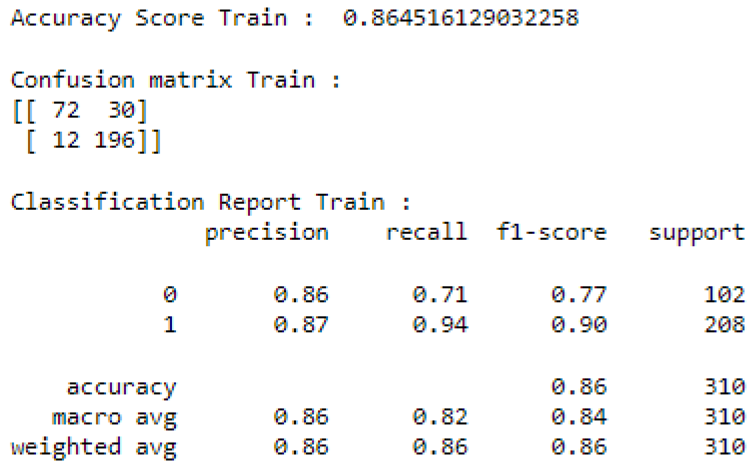
	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
Gradient Boosting	96	78	99	80	99	87	96	82	97	85
Gradient Boosting Tuned	87	85	94	83	95	95	97	85	91	90

Observation

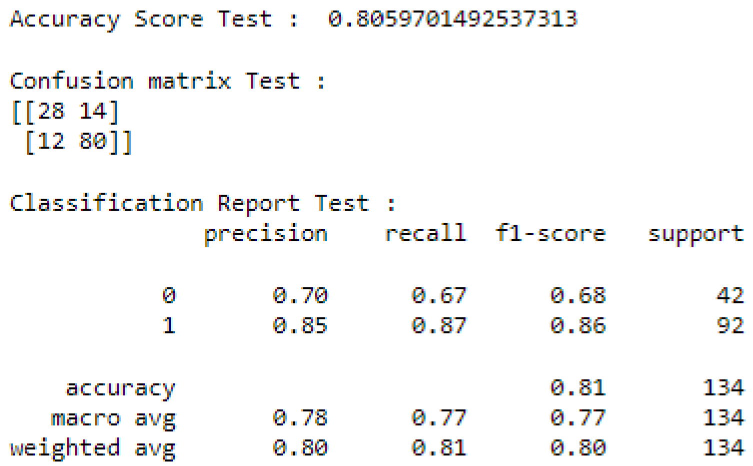
- Their is an over all improvement in the model
- Recall seems to be prefect for both train and test
- Accuracy score also improved
- Precision and Recall also improved

ADA Boost (Boosting)

Predictions on Train and Test:

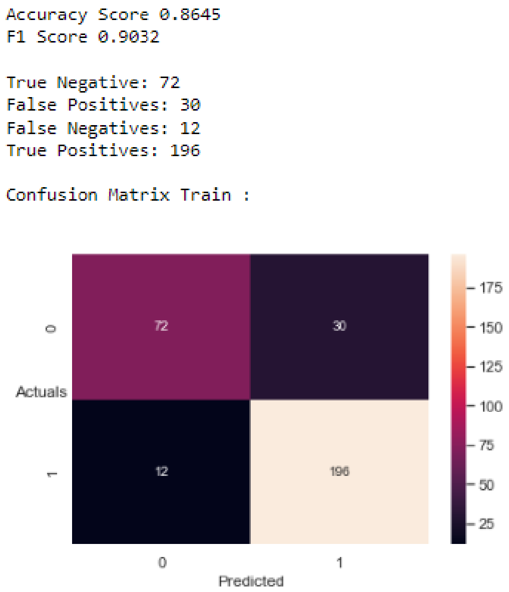


```
Accuracy Score Train :  0.864516129032258

Confusion matrix Train :
[[ 72  30]
 [ 12 196]]

Classification Report Train :
              precision    recall  f1-score   support

           0       0.86      0.71      0.77       102
           1       0.87      0.94      0.90       208

    accuracy                           0.86       310
   macro avg       0.86      0.82      0.84       310
weighted avg       0.86      0.86      0.86       310
```



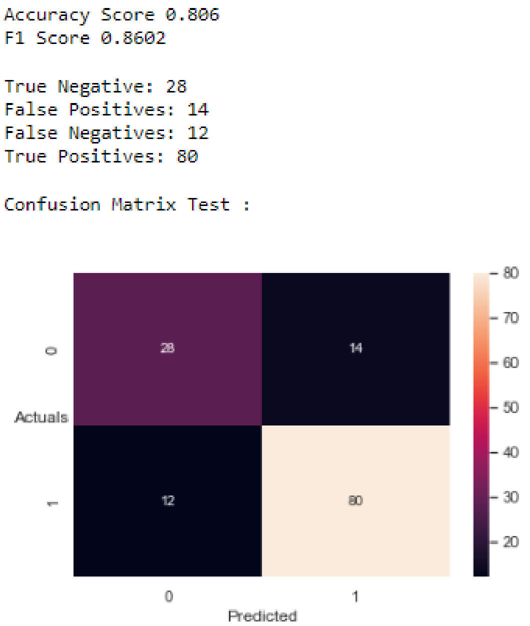
```
Accuracy Score Test :  0.8059701492537313

Confusion matrix Test :
[[28 14]
 [12 80]]

Classification Report Test :
              precision    recall  f1-score   support

           0       0.70      0.67      0.68        42
           1       0.85      0.87      0.86        92

    accuracy                           0.81       134
   macro avg       0.78      0.77      0.77       134
weighted avg       0.80      0.81      0.80       134
```

Confusion Matrix Train and Test:



```
Accuracy Score 0.8645
F1 Score 0.9032

True Negative: 72
False Positives: 30
False Negatives: 12
True Positives: 196

Confusion Matrix Train :
```

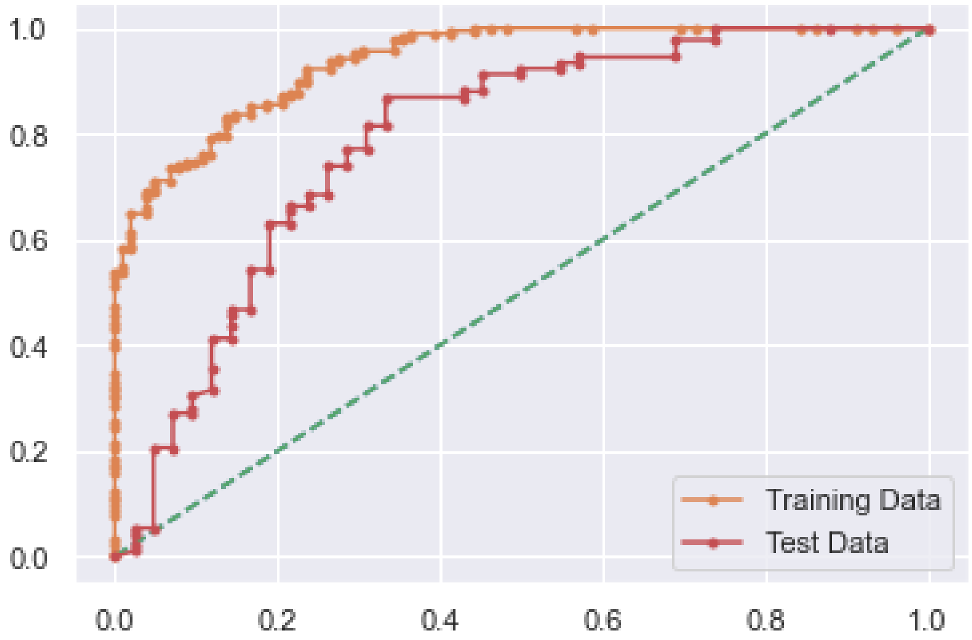
Confusion Matrix Train



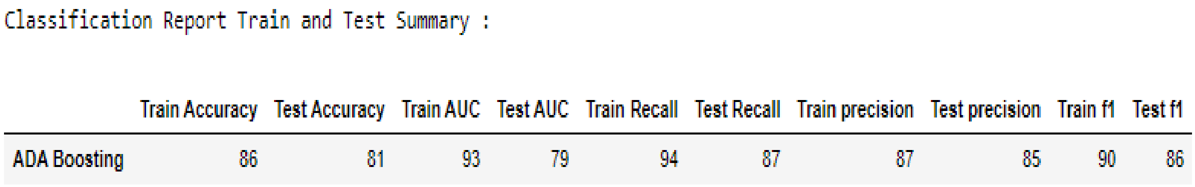
```
Accuracy Score 0.806
F1 Score 0.8602

True Negative: 28
False Positives: 14
False Negatives: 12
True Positives: 80

Confusion Matrix Test :
```

Confusion Matrix test

AUC ROC on Train and Test:



Summary Train and Test:



Classification Report Train and Test Summary :

	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
ADA Boosting	86	81	93	79	94	87	87	85	90	86

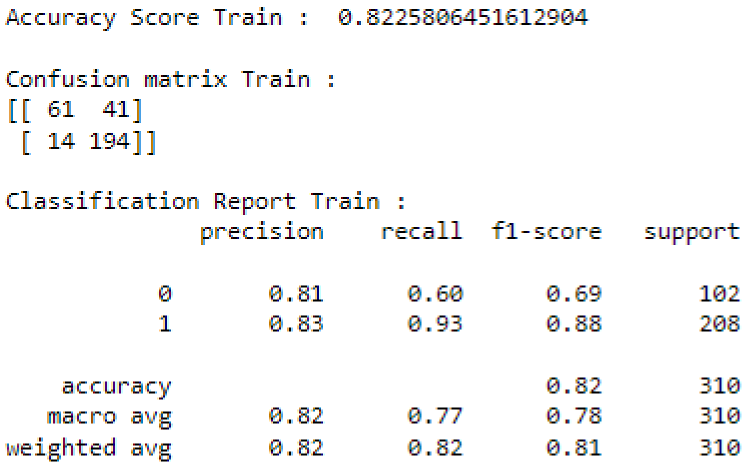
Observation:

- Their seems to be a case of Overfitting of model.
- Huge difference between train and test for Recall and F1 score
- Accuracy and Precision seems decent

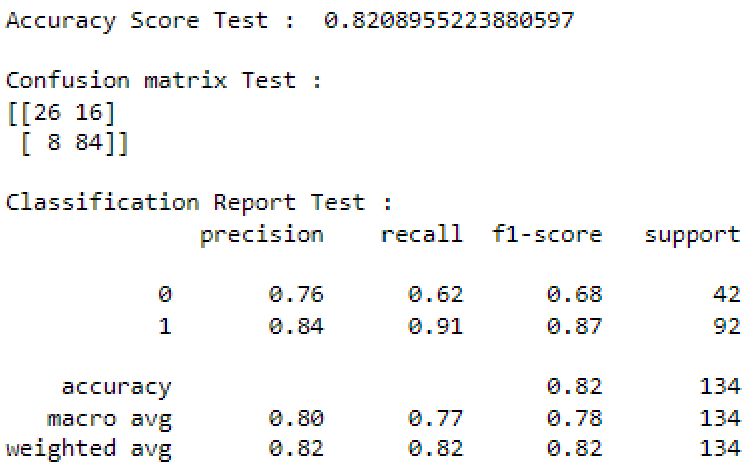
Let’s do some hyperparameter tuning to see if there is any improvement

Applying GridSearch CV on ADA Boost:

Predictions on Train and Test:

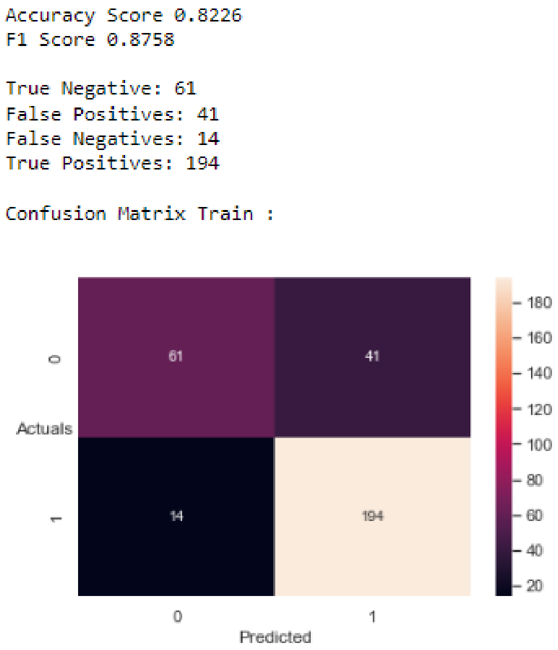


```
Accuracy Score Train :  0.8225806451612904

Confusion matrix Train :
[[ 61  41]
 [ 14 194]]

Classification Report Train :
              precision    recall  f1-score   support

           0       0.81      0.60      0.69       102
           1       0.83      0.93      0.88       208

    accuracy                           0.82       310
   macro avg       0.82      0.77      0.78       310
weighted avg       0.82      0.82      0.81       310
```



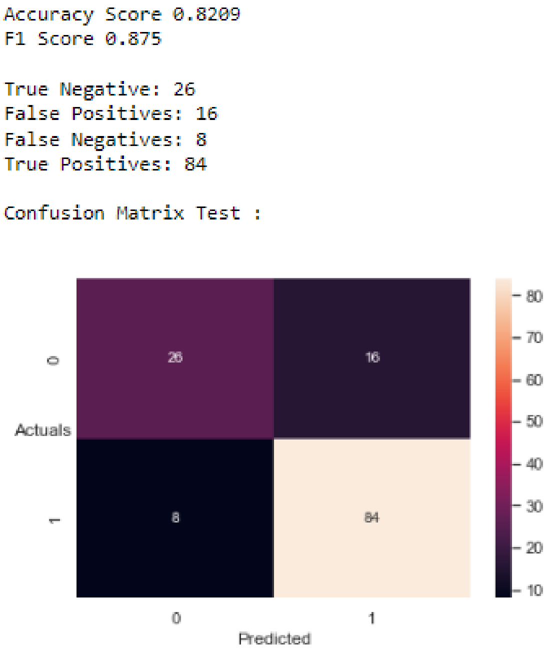
```
Accuracy Score Test :  0.8208955223880597

Confusion matrix Test :
[[26 16]
 [ 8 84]]

Classification Report Test :
              precision    recall  f1-score   support

           0       0.76      0.62      0.68        42
           1       0.84      0.91      0.87        92

    accuracy                           0.82       134
   macro avg       0.80      0.77      0.78       134
weighted avg       0.82      0.82      0.82       134
```

Confusion Matrix Train and Test:



```
Accuracy Score 0.8226
F1 Score 0.8758

True Negative: 61
False Positives: 41
False Negatives: 14
True Positives: 194

Confusion Matrix Train :
```

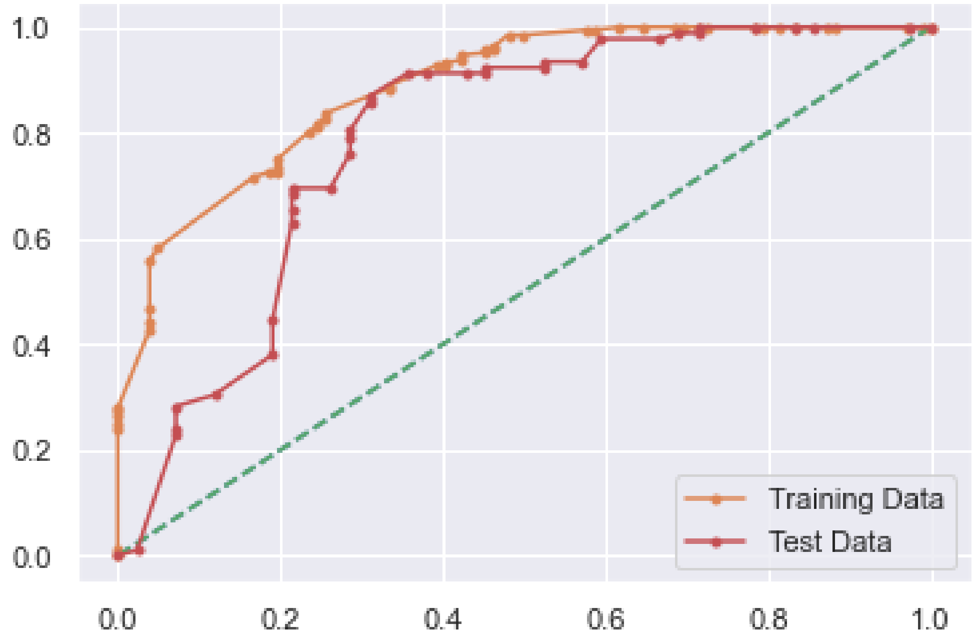
Confusion Matrix Train



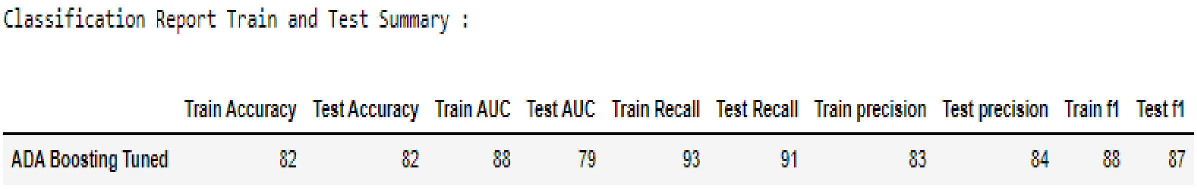
```
Accuracy Score 0.8209
F1 Score 0.875

True Negative: 26
False Positives: 16
False Negatives: 8
True Positives: 84

Confusion Matrix Test :
```

Confusion Matrix test

AUC ROC on Train and Test:



Summary Train and Test:



```
Classification Report Train and Test Summary :
```

	Train Accuracy	Test Accuracy	Train AUC	Test AUC	Train Recall	Test Recall	Train precision	Test precision	Train f1	Test f1
ADA Boosting Tuned	82	82	88	79	93	91	83	84	88	87

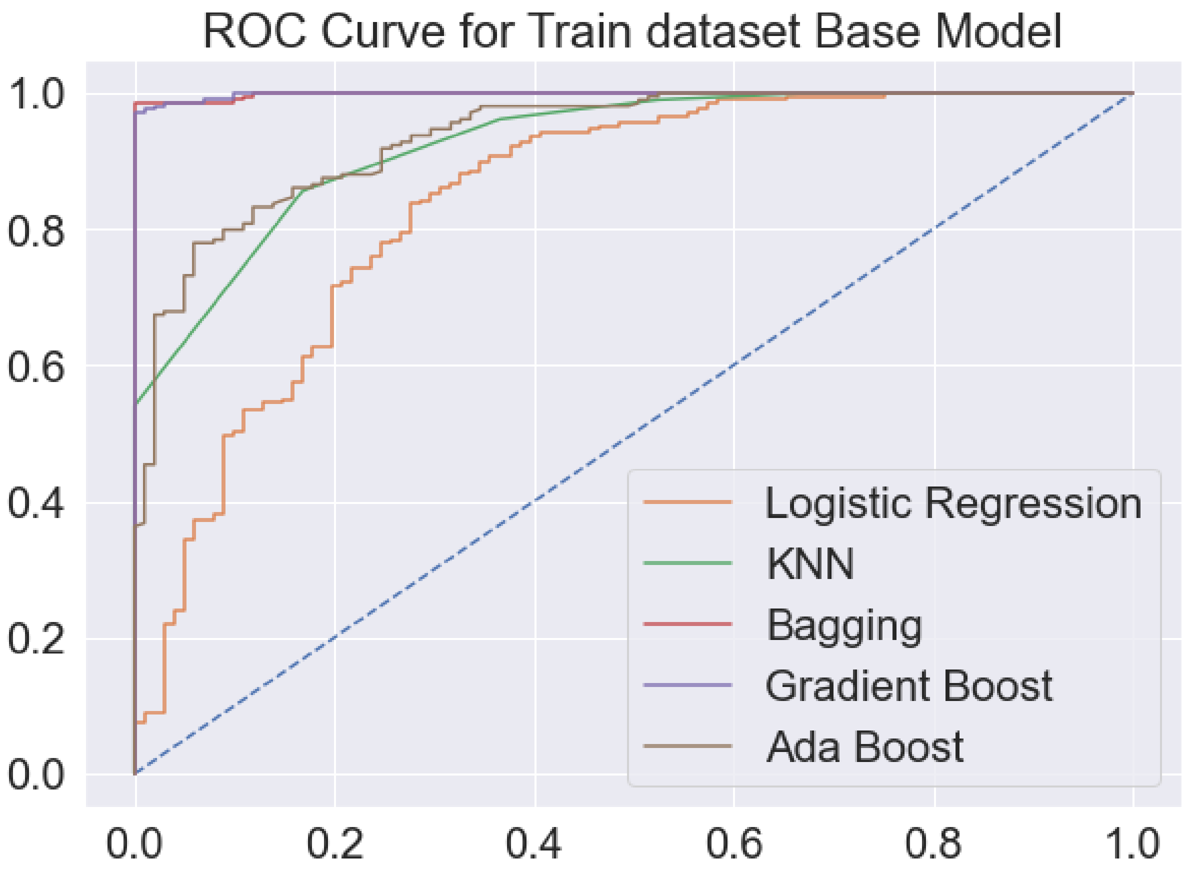
ADA Boost Summary:

Observation:

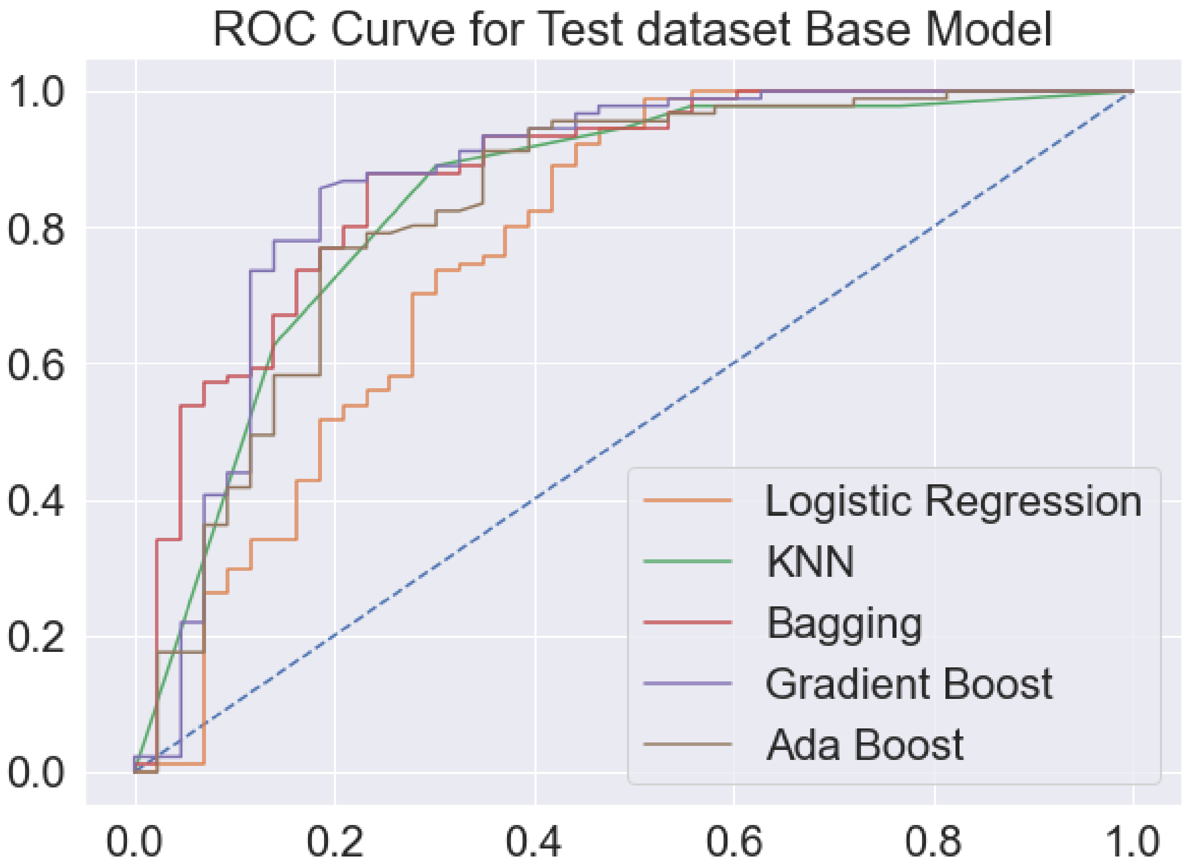
- Their is an over all improvement in the model
- Accuracy seems to be prefect for both train and test
- Recall score also improved
- Precision and Recall also improved

Here we care now checking all the 5 models performance together with and without Tuning in order to find which model performed better

Checking all the 5 Models without Tuning

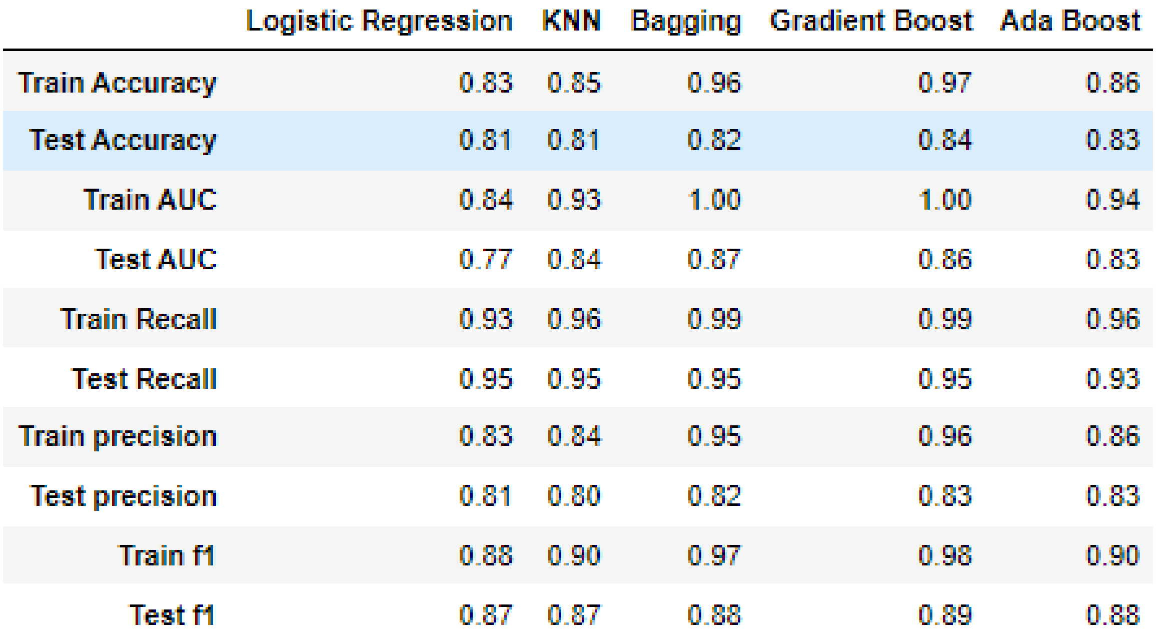


- AUC for Logistic Regression is: 0.84
- AUC for KNN is: 0.93
- AUC for Bagging is: 1.0
- AUC for Gradient Boost is: 1.0
- AUC for Ada Boost is: 0.94



- AUC for Logistic Regression is: 0.77
- AUC for KNN is: 0.84
- AUC for Bagging is: 0.87
- AUC for Gradient Boost is: 0.86
- AUC for Ada Boost is: 0.83

Summary Of Train and Test Accuracy on the 5 Models (Base Model)



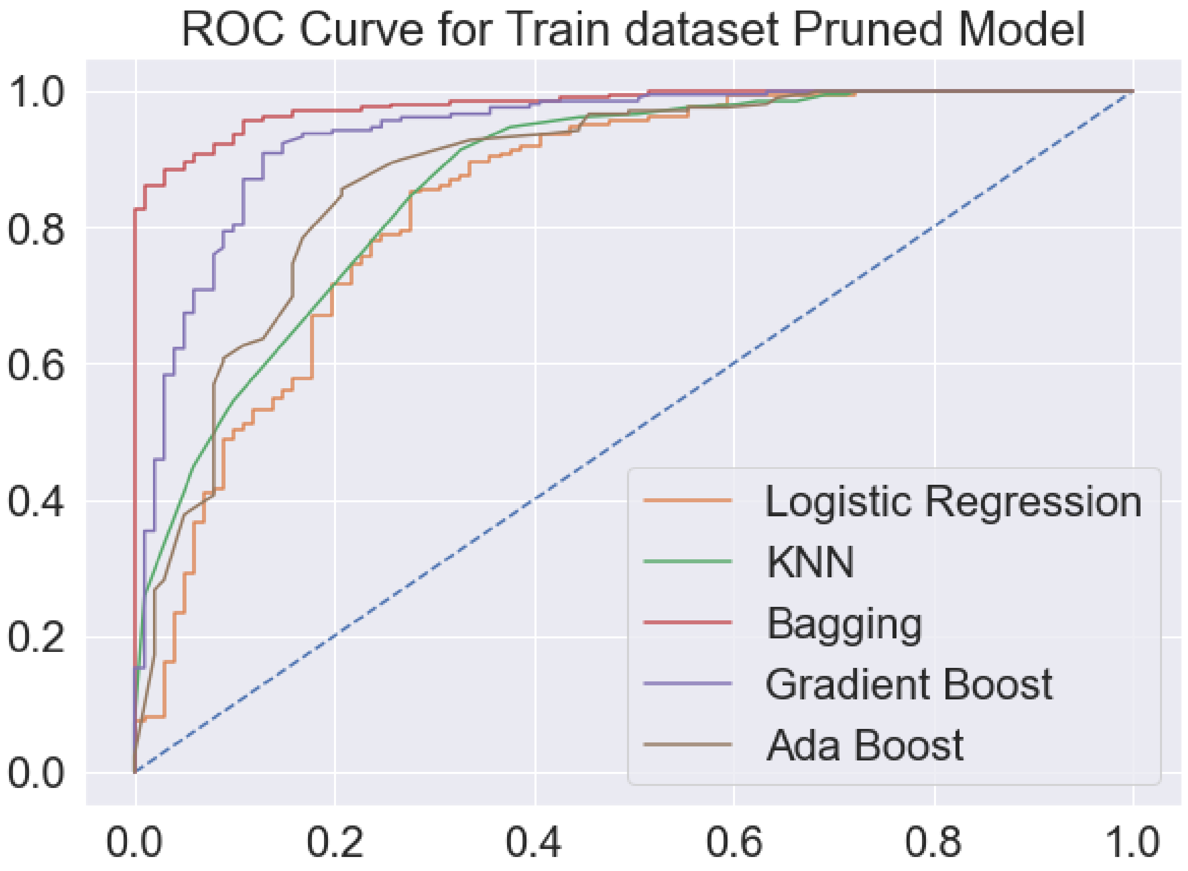
	Logistic Regression	KNN	Bagging	Gradient Boost	Ada Boost
Train Accuracy	0.83	0.85	0.96	0.97	0.86
Test Accuracy	0.81	0.81	0.82	0.84	0.83
Train AUC	0.84	0.93	1.00	1.00	0.94
Test AUC	0.77	0.84	0.87	0.86	0.83
Train Recall	0.93	0.96	0.99	0.99	0.96
Test Recall	0.95	0.95	0.95	0.95	0.93
Train precision	0.83	0.84	0.95	0.96	0.86
Test precision	0.81	0.80	0.82	0.83	0.83
Train f1	0.88	0.90	0.97	0.98	0.90
Test f1	0.87	0.87	0.88	0.89	0.88

Observation

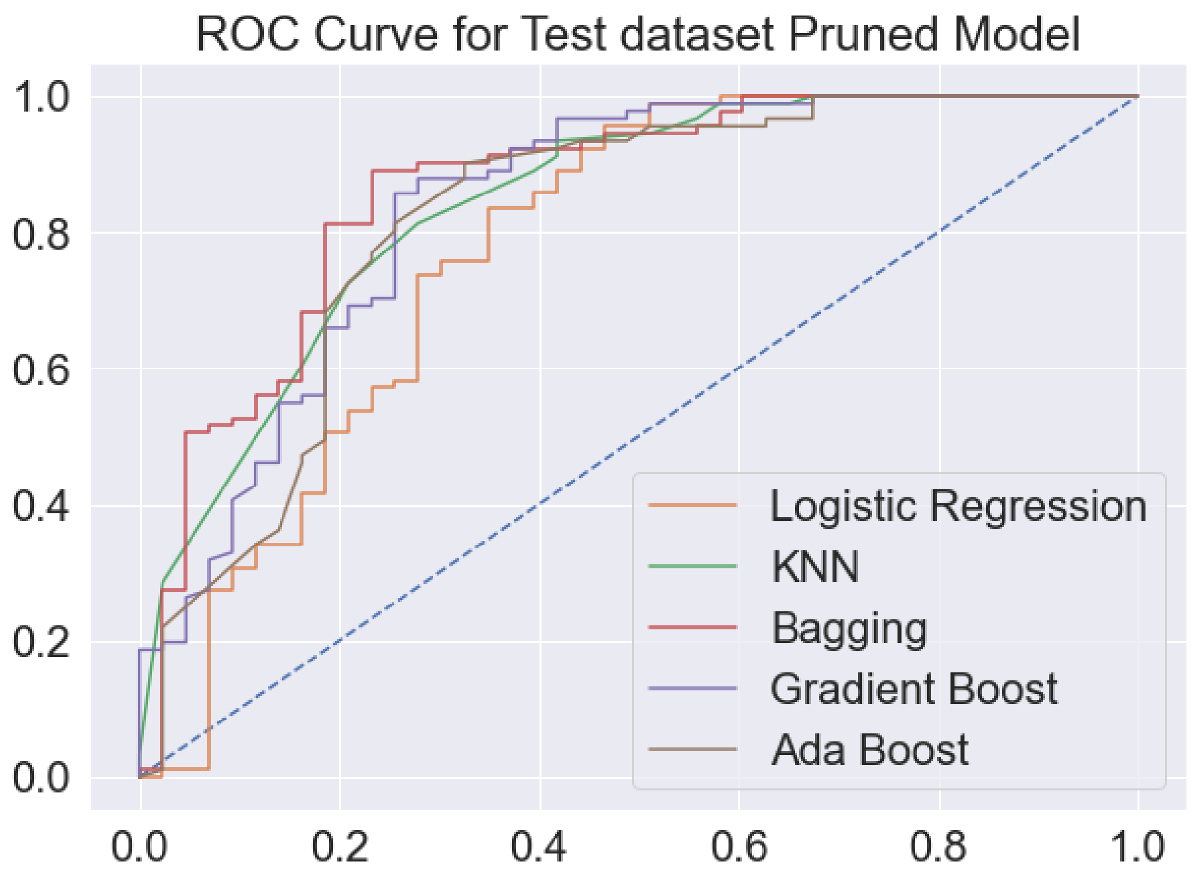
- Logistic Regression Model has show better performance out of all 5 models

Lets check what happens after Pruning of all the 5 models

Putting pruned parameters to the 5 models

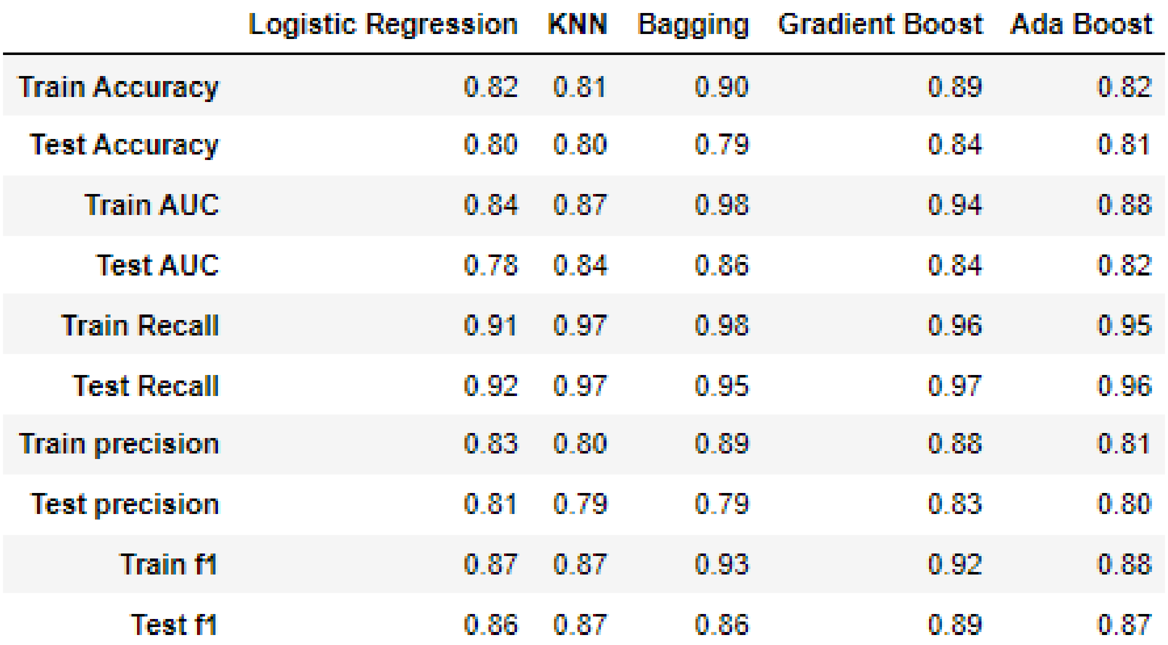


- AUC for Logistic Regression is: 0.84
- AUC for KNN is: 0.87
- AUC for Bagging is: 0.98
- AUC for Gradient Boost is: 0.94
- AUC for Ada Boost is: 0.88



- AUC for Logistic Regression is: 0.78
- AUC for KNN is: 0.84
- AUC for Bagging is: 0.86
- AUC for Gradient Boost is: 0.84
- AUC for Ada Boost is: 0.82

Summary Of Train and Test Accuracy on the 5 Models after Pruning



	Logistic Regression	KNN	Bagging	Gradient Boost	Ada Boost
Train Accuracy	0.82	0.81	0.90	0.89	0.82
Test Accuracy	0.80	0.80	0.79	0.84	0.81
Train AUC	0.84	0.87	0.98	0.94	0.88
Test AUC	0.78	0.84	0.86	0.84	0.82
Train Recall	0.91	0.97	0.98	0.96	0.95
Test Recall	0.92	0.97	0.95	0.97	0.96
Train precision	0.83	0.80	0.89	0.88	0.81
Test precision	0.81	0.79	0.79	0.83	0.80
Train f1	0.87	0.87	0.93	0.92	0.88
Test f1	0.86	0.87	0.86	0.89	0.87

Observation

- AdaBoost has preformed the best out of all 5 models after pruning i.e Logistic regression, KNN, Bagging, Gradient Boost, Ada Boost.
- AdaBoost is the most Balance model out of all % models.
- After Pruning Logistic Regression performance have gone down significantly which actually performed best before pruning.

Thus we can conclude that Ada Boost is the best model:

With accuracy of 81% and recall rate of 96%, model is able to predict 95% of Public Transport which were actually claimed as claimed.

Precision is 80% of data which means, out of total employees predicted by model as opt for Public Transport , 80% employees actually opted for the Public Transport.

F1-score(87%) is the harmonic mean of precision and recall, it takes into the effect of both the scores and this value is low if any of these 2 value is low.

Since we are building a model to predict if whether employee will opt for Public Transport or Private Transport, for practical purposes, we will be more interested in correctly classifying 1 (Public Transport) than 0(Private Transport).

If a employee Private Transport is incorrectly predicted to be "Public Transport" by the model, then the impact on cost for the travel company would be bare minimum. But if an employee opted for Public Transport is incorrectly predicted to be Private Transport by the model, then the cost impact would be very high for the Transport company. Its a loss of potential lead for the company. Hence recall rate (actual data point identified as True by model) is very important in this scenario.

As Recall rate of test dataset is very good around 96% plus Precision rate is also 80% for opting Public transport thus this does looks good enough for classification

Thus our Transport Company can target these Employee's for providing services.

Inference:

8. Based on these predictions, what are the insights?

Insights:

1. The count of Employees in ABC Company using Public Transport(300) are more compared to Private Transport (144).
2. The Count plot clearly states that most of the Employee in ABC Company doesn't have license. More no. of Male's(94) have license compared to Female's(10).
3. Public Transport is the most preferred mode by the ABC Company Employees. Male(93) are more compared to Females(51) in Private Transport. There is Imbalance of data between Public and Private Transport in Males (i.e 223:93). Thus our travel company can travel employees who use public transport.
4. There are some Employees who prefer to Public Transport even though they have license. We can assume that these employees live far from office.

5. As per Ada Boost model:

With accuracy of 81% and recall rate of 96%, model is able to predict 95% of Public Transport which were actually claimed as claimed.

Precision is 80% of data which means, out of total employees predicted by model as opt for Public Transport , 80% employees actually opted for the Public Transport.

F1-score(87%) is the harmonic mean of precision and recall, it takes into the effect of both the scores and this value is low if any of these 2 value is low.

Since we are building a model to predict if whether employee will opt for Public Transport or Private Transport, for practical purposes, we will be more interested in correctly classifying 1 (Public Transport) than 0(Private Transport).

If a employee Private Transport is incorrectly predicted to be "Public Transport" by the model, then the impact on cost for the travel company would be bare minimum. But if am employee opted for Public Transport is incorrectly predicted to be Private Transport by the model, then the cost impact would be very high for the Transport company. Its a loss of potential lead for the company. Hence recall rate (actual data point identified as True by model) is very important in this scenario.

As Recall rate of test dataset is very good around 96% plus Precision rate is also 80% for opting Public transport thus this does looks good enough for classification

Thus our Transport Company can target these Employee's for providing services.

Recommendations for Transport Company:

1. We can target employee's who use Public Transport as most of employee's in ABC Company uses Public Transport to come to office.
2. Our Transport company can provide pick and drop from home to office and vice versa facilities which helps saves times for the employee's.
3. Employees safety and comfort can also be maintained which is less compared to Public Transport.
4. Our travel agency can provide AC buses with proper hygiene maintained.
5. Timely pick and drop facilities for the employee's which will also help in their work life balance and also saves time which is usually very difficult in Public Transport.
6. Money spend on Public Transport can be saved by the employee's as they using our Travel Company facilities.