MACHINE LEARNING PROJECT REPORT 2023

Sangeeth A

PGP-DSBA Online

July - 2022

CONTENTS

LIST OF FIGURES

LIST OF TABLES

PROBLEM 1 – SUMMARY

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 finalized.

INTRODUCTION

The purpose of this exercise is 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 finalized.

DATA DESCRIPTION

1. Age – Age of the employee
2. Gender – Gender of the employee
3. Engineer – Employee is engineer or not
4. MBA – Employee is MBA or not.
5. Work Exp. – Work experience of the employee.
6. Salary - Salary of the employee.
7. Distance – Distance travelling the employee.
8. License – Whether the employee is having the license or not.
9. Transport – Transport system employees using for travelling.

SAMPLE 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 – Sample dataset

The dataset contains 9 features, our dependent variable is “Transport”, we have to create different models based on other 8 independent features.

EXPLORATORY DATA ANALYSIS

The dataset contains 444 entries and 9 columns, there are no null value present in the dataset.

NO.	Column	Non – Null content	Data Type
1	Age	444	Int64
2	Gender	444	Object
3	Engineer	444	Int64
4	MBA	444	Int64
5	Work Exp.	444	Int64
6	Salary	444	Float64
7	Distance	444	Float64
8	LIcense	444	Int64
9	Transport	444	Object

Table 2 – Exploratory Data Analysis

DESCRIPTIVE DATA ANALYSIS

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Age	444.0	NaN	NaN	NaN	27.747748	4.41671	18.0	25.0	27.0	30.0	43.0
Gender	444	2	Male	316	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Engineer	444.0	NaN	NaN	NaN	0.754505	0.430866	0.0	1.0	1.0	1.0	1.0
MBA	444.0	NaN	NaN	NaN	0.252252	0.434795	0.0	0.0	0.0	1.0	1.0
Work Exp	444.0	NaN	NaN	NaN	6.29955	5.112098	0.0	3.0	5.0	8.0	24.0
Salary	444.0	NaN	NaN	NaN	16.238739	10.453851	6.5	9.8	13.6	15.725	57.0
Distance	444.0	NaN	NaN	NaN	11.323198	3.606149	3.2	8.8	11.0	13.425	23.4
license	444.0	NaN	NaN	NaN	0.234234	0.423997	0.0	0.0	0.0	0.0	1.0
Transport	444	2	Public Transport	300	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Table 3 – Descriptive Data Analysis

1. Most of the employees working in the company are male
2. Age of the employees ranges from 18 to 43.
3. Most of the employees uses public transport for travelling.

Problem 1 – DATA INGESTION

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 finalized.

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

The problem statement is to read the data and do the initial necessary steps and conduct exploratory data analysis on the dataset provided.

	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 1.1.1 – Sample dataset

The above shown is the head of the dataset, the dataset contains 9 columns and they are,

1. Age – Age of the employee
2. Gender – Gender of the employee
3. Engineer – Employee is engineer or not
4. MBA – Employee is MBA or not.
5. Work Exp. – Work experience of the employee.
6. Salary - Salary of the employee.
7. Distance – Distance travelling the employee.
8. License – Whether the employee is having the license or not.
9. Transport – Transport system employees using for travelling.

All the variables in the dataset is numerical except gender and transport. Transport column is the dependent variable and based on other 8 independent variables we have to built a different models.

Now let us check the data types of the different features present in the dataset.

NO.	Column	Non – Null content	Data Type
1	Age	444	Int64
2	Gender	444	Object
3	Engineer	444	Int64
4	MBA	444	Int64
5	Work Exp.	444	Int64
6	Salary	444	Float64
7	Distance	444	Float64
8	LIcense	444	Int64
9	Transport	444	Object

Table 1.1.2 – Data info

There are total 444 entries and 9 columns in the dataset.

Now we can check for the null values present in the dataset,

NO.	Column	Null Value Present
1	Age	0
2	Gender	0
3	Engineer	0
4	MBA	0
5	Work Exp.	0
6	Salary	0
7	Distance	0
8	License	0
9	Transport	0

Table 1.1.3 – Null value check

There is no null value present in the dataset.

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Age	444.0	NaN	NaN	NaN	27.747748	4.41671	18.0	25.0	27.0	30.0	43.0
Gender	444	2	Male	316	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Engineer	444.0	NaN	NaN	NaN	0.754505	0.430866	0.0	1.0	1.0	1.0	1.0
MBA	444.0	NaN	NaN	NaN	0.252252	0.434795	0.0	0.0	0.0	1.0	1.0
Work Exp	444.0	NaN	NaN	NaN	6.29955	5.112098	0.0	3.0	5.0	8.0	24.0
Salary	444.0	NaN	NaN	NaN	16.238739	10.453851	6.5	9.8	13.6	15.725	57.0
Distance	444.0	NaN	NaN	NaN	11.323198	3.606149	3.2	8.8	11.0	13.425	23.4
license	444.0	NaN	NaN	NaN	0.234234	0.423997	0.0	0.0	0.0	0.0	1.0
Transport	444	2	Public Transport	300	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Table 1.1.4 – Summary statistics

The above shown table shows the summary statistics of the dataset.

1.1Most of the employees working in the company are male
2.1Age of the employees ranges from 18 to 43.
3.1Most of the employees uses public transport for travelling.

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

EXPLORATORY DATA ANALYSIS

Now let us do the exploratory data analysis of the given data.

UNIVARIATE ANALYSIS

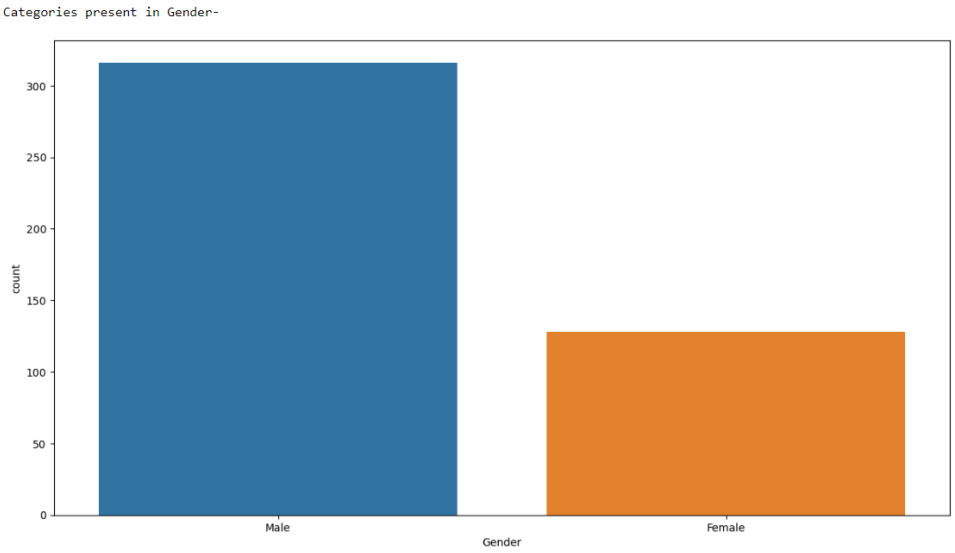

Fig 1.2.1 –Univariate analysis (HIst Plot) – Gender

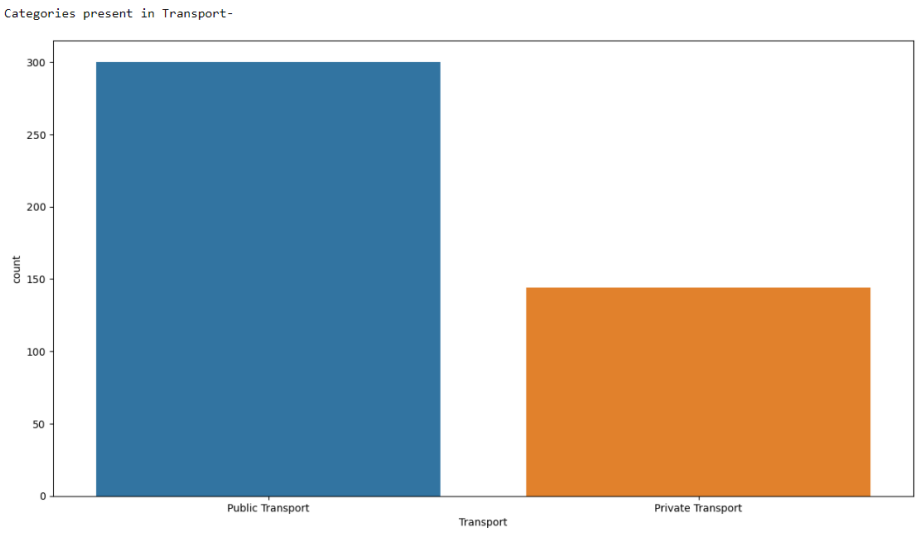

Fig 1.2.2–Univariate analysis (HIst Plot) – Transport

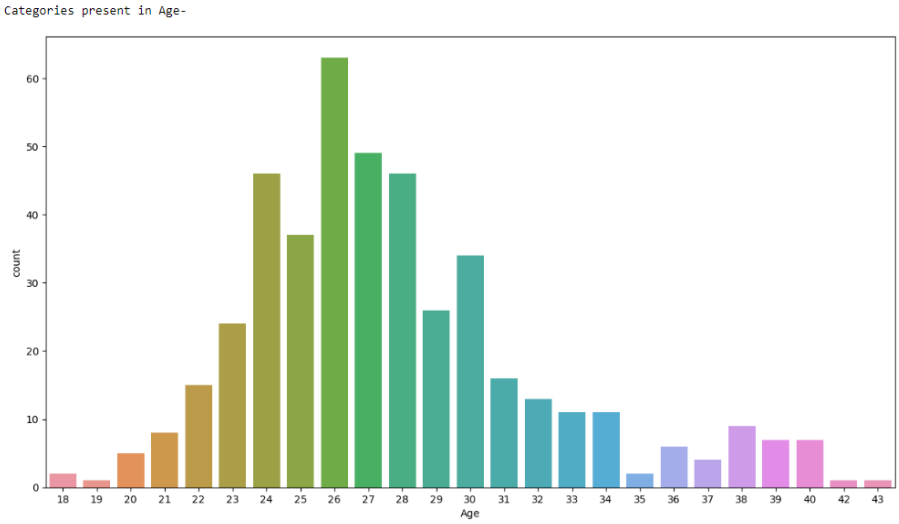

Fig 1.2.3–Univariate analysis (HIst Plot) – Age

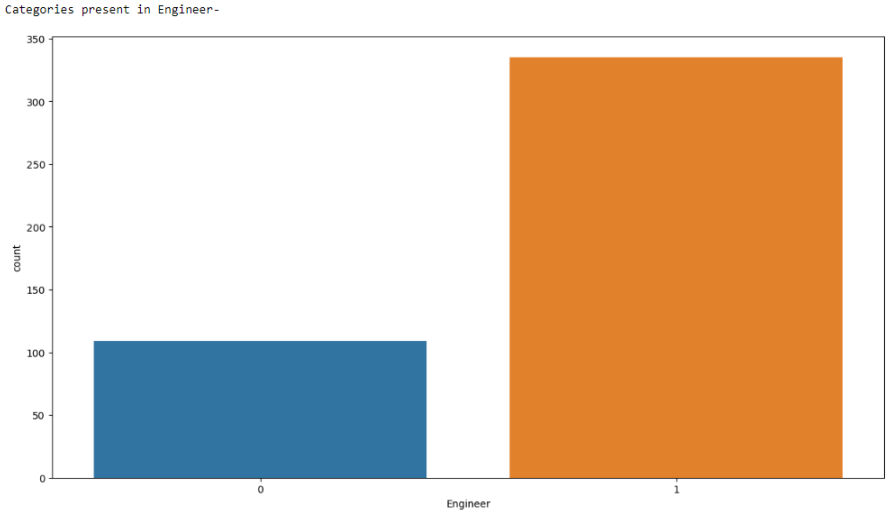

Fig 1.2.4–Univariate analysis (HIst Plot) – Engineer

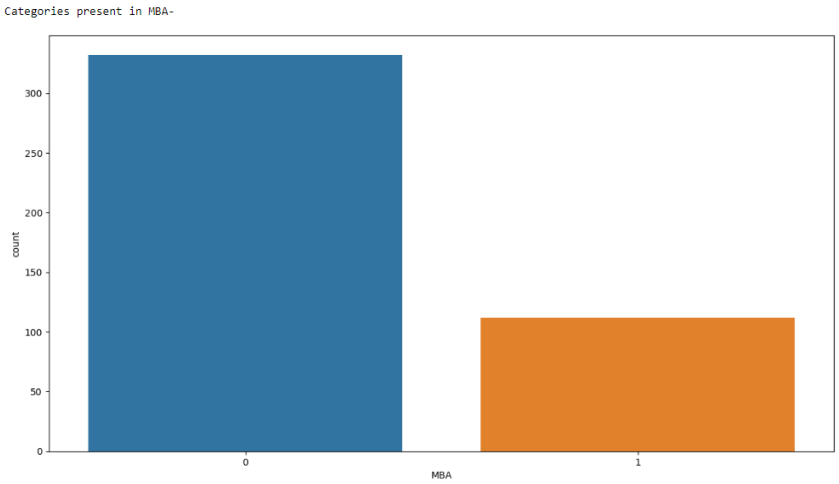

Fig 1.2.5–Univariate analysis (HIst Plot) – MBA

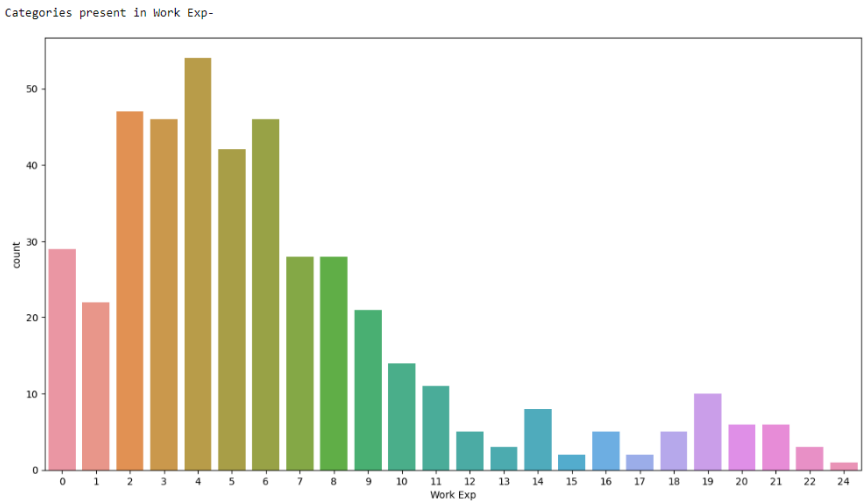

Fig 1.2.6–Univariate analysis (HIst Plot) – Work Exp

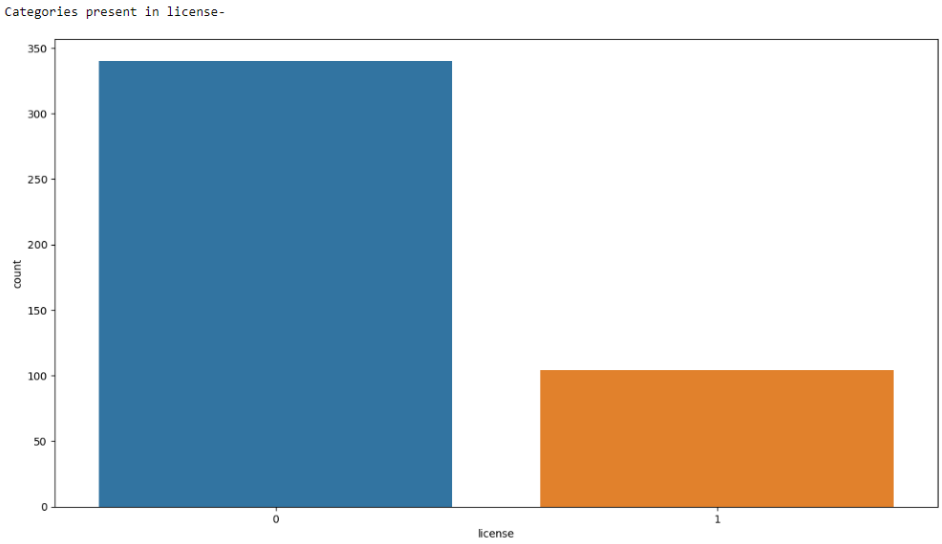

Fig 1.2.7–Univariate analysis (HIst Plot) – License

The above shown is the hist plot of the different features of the dataset. The observations from the hist plot is given below.

1. Most of the employees doesn't have license
2. Most of the employees have a work experience in the range 2 to 6.
3. Most of the employees in the company are engineers.
4. Most of the employees depends public transport for travelling.
5. Most of the employees working in the company are male.

Now let us check the boxplot of the numerical features present in the dataset.

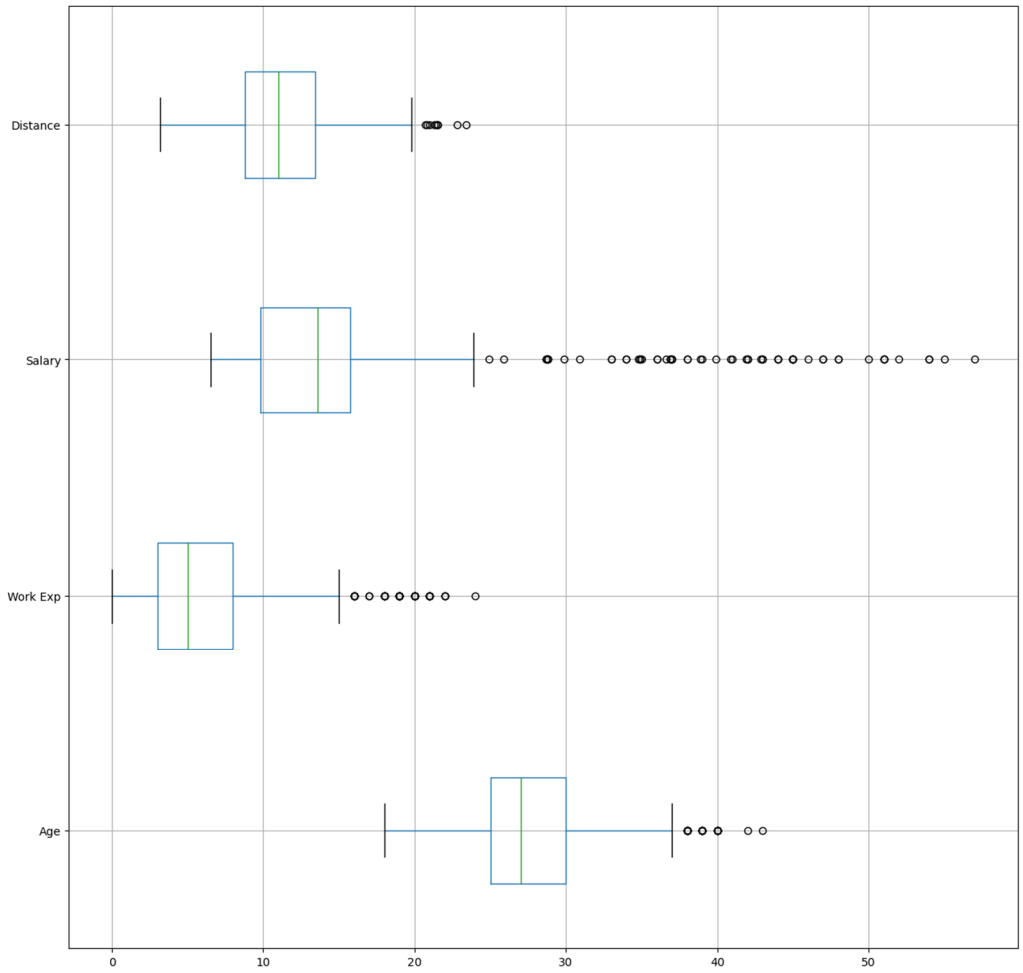

Fig 1.2.8–Univariate analysis (Box Plot)

The above shown is the box plot of different features in the dataset. The observations from the dataset is below,

1. All the features in the dataset contains outliers.
2. From the boxplot it is clear that the salary dataset is left skewed.

Since the dataset contains very less information and we don't have a opinion of domain expert we decide not to treat the outliers.

BIVARIATE ANALYSIS

Now let us do the bivariate analysis of the data.

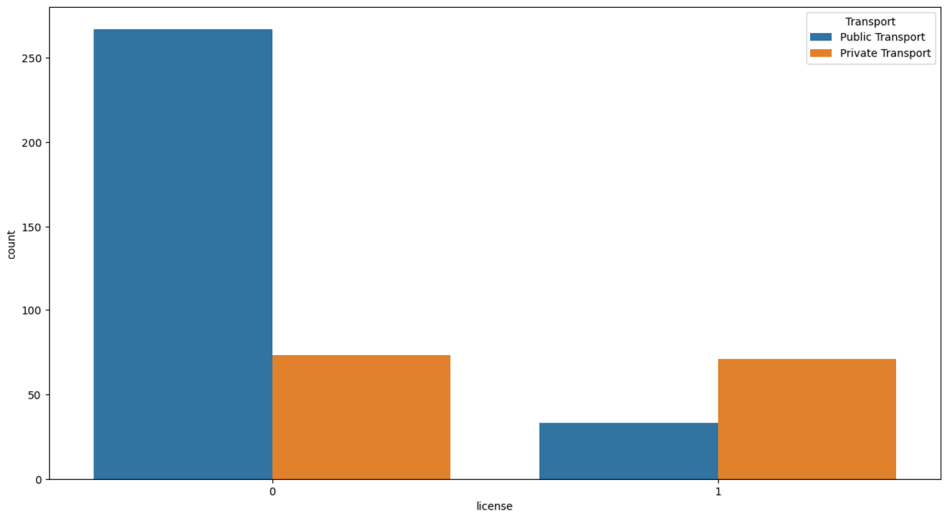

Fig 1.2.9 –Bivariate analysis (License v/s Transport)

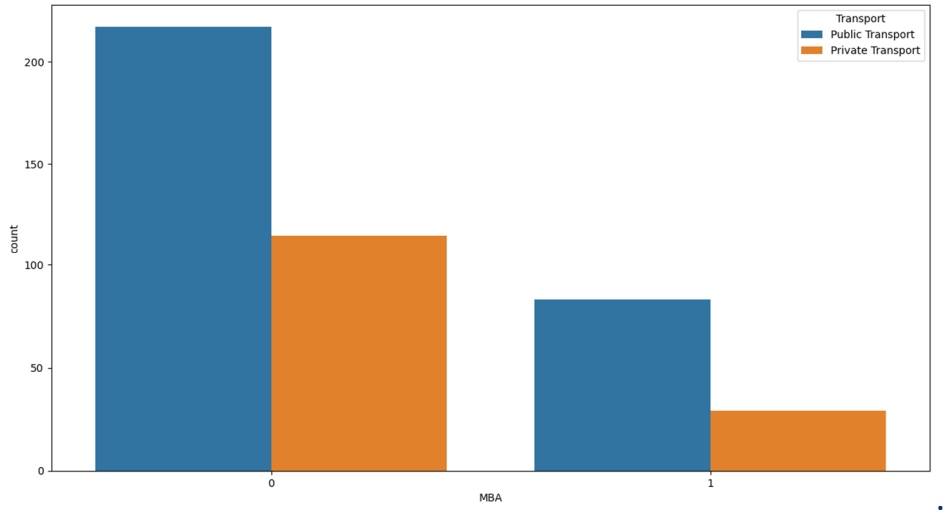

Fig 1.2.10 –Bivariate analysis (MBA v/s Transport)

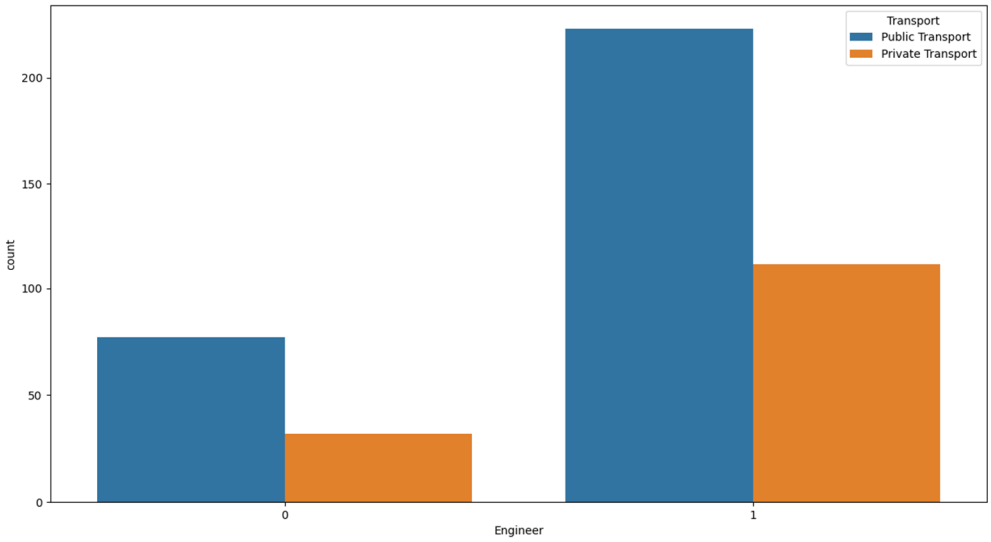

Fig 1.2.11 –Bivariate analysis (Engineer v/s Transport)

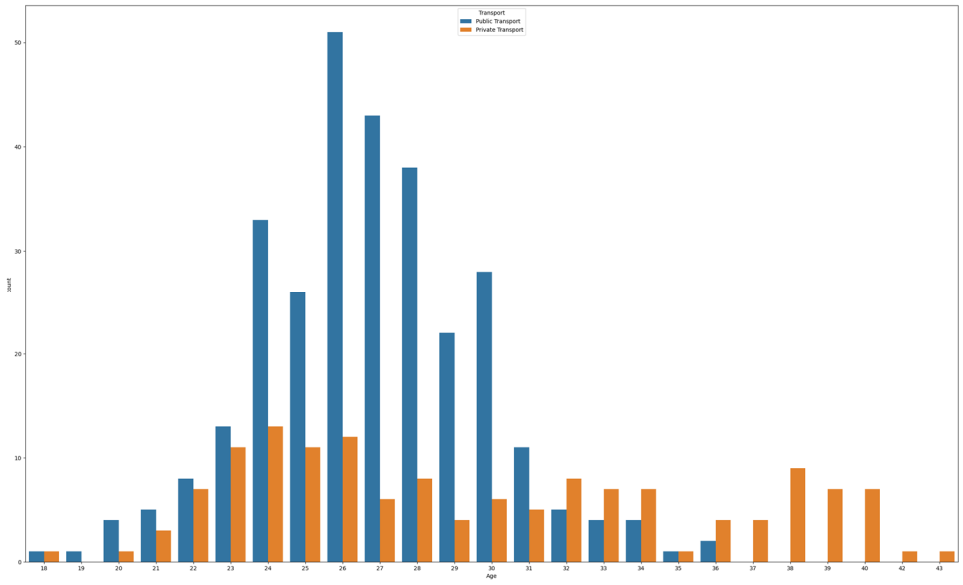

Fig 1.2.12 –Bivariate analysis (Age v/s Transport)

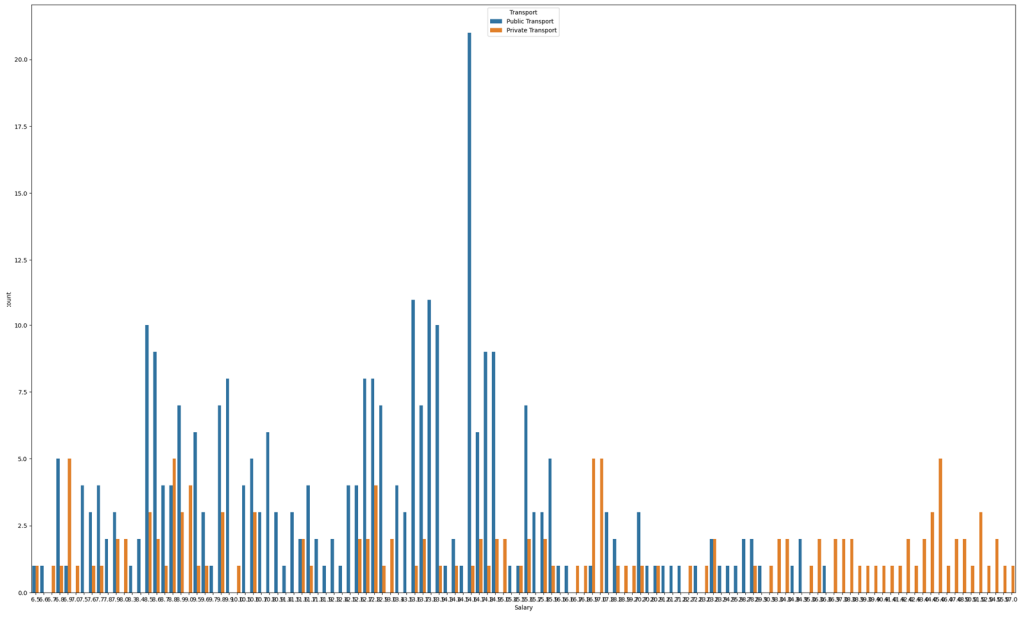

Fig 1.2.13 –Bivariate analysis (Salary v/s Transport)

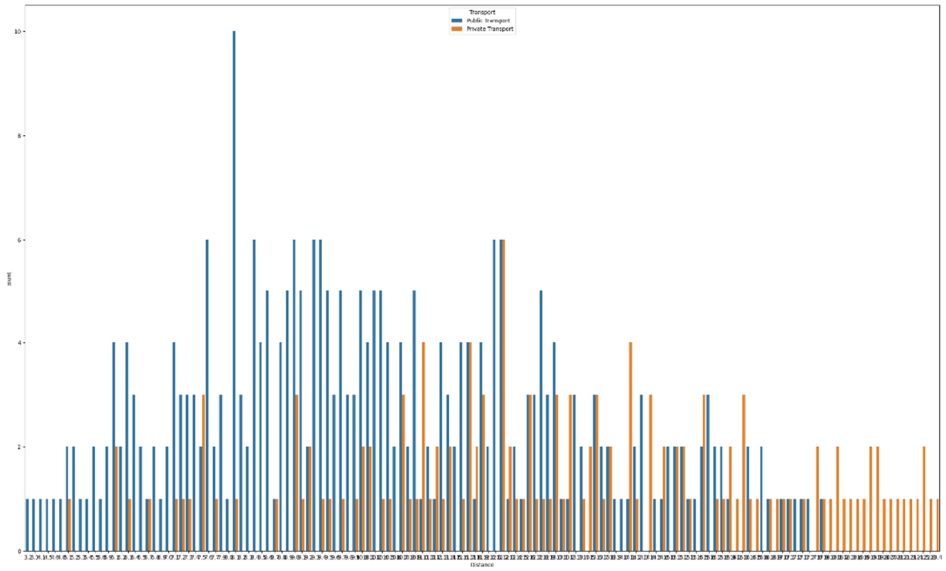

Fig 1.2.14 –Bivariate analysis (Distance v/s Transport)

MULTIVARIATE ANALYSIS

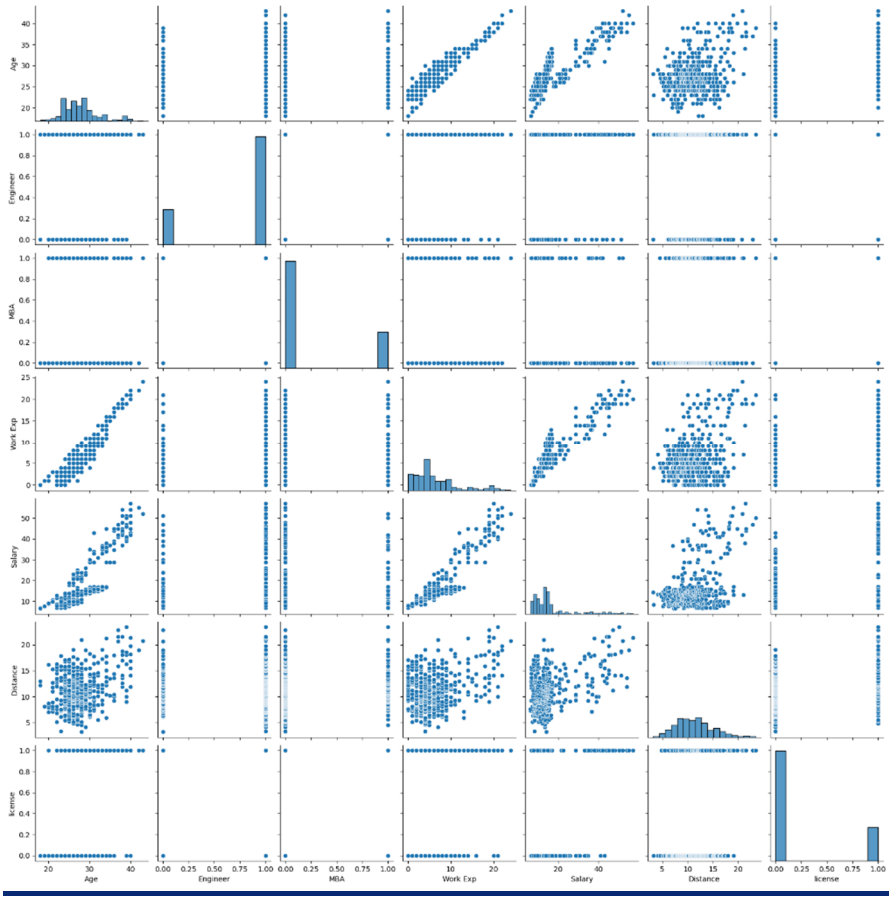

Fig 1.2.15 –Multivariate analysis (Pairplot)

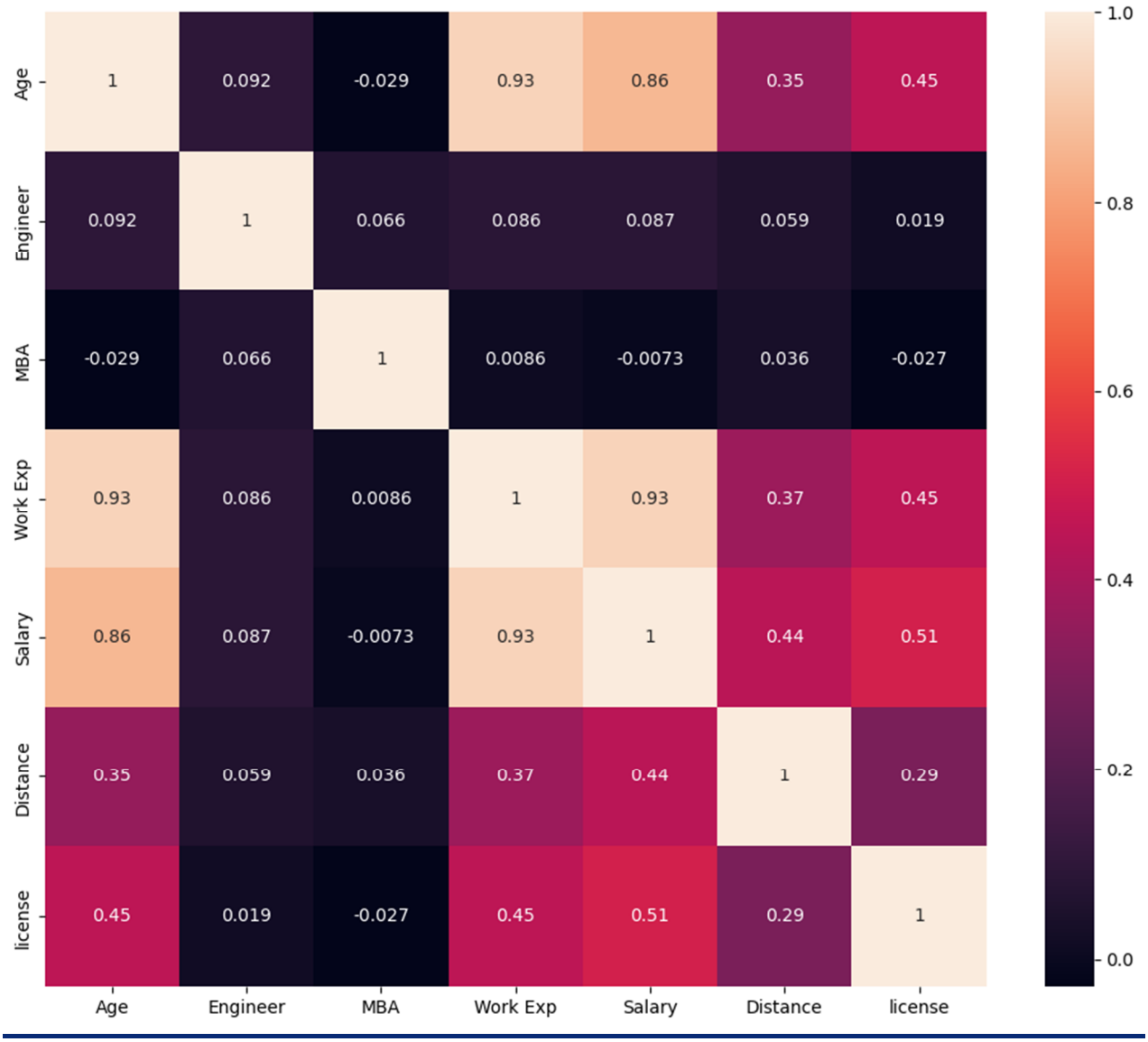

Fig 1.2.16 –Multivariate analysis (Heatmap)

The figure represents the correlation heatmap between different features of the dataset.

Pairplot is similar to heatmap, it shows the correlation between the features of the dataset.

Next let us check the outliers of the dataset,

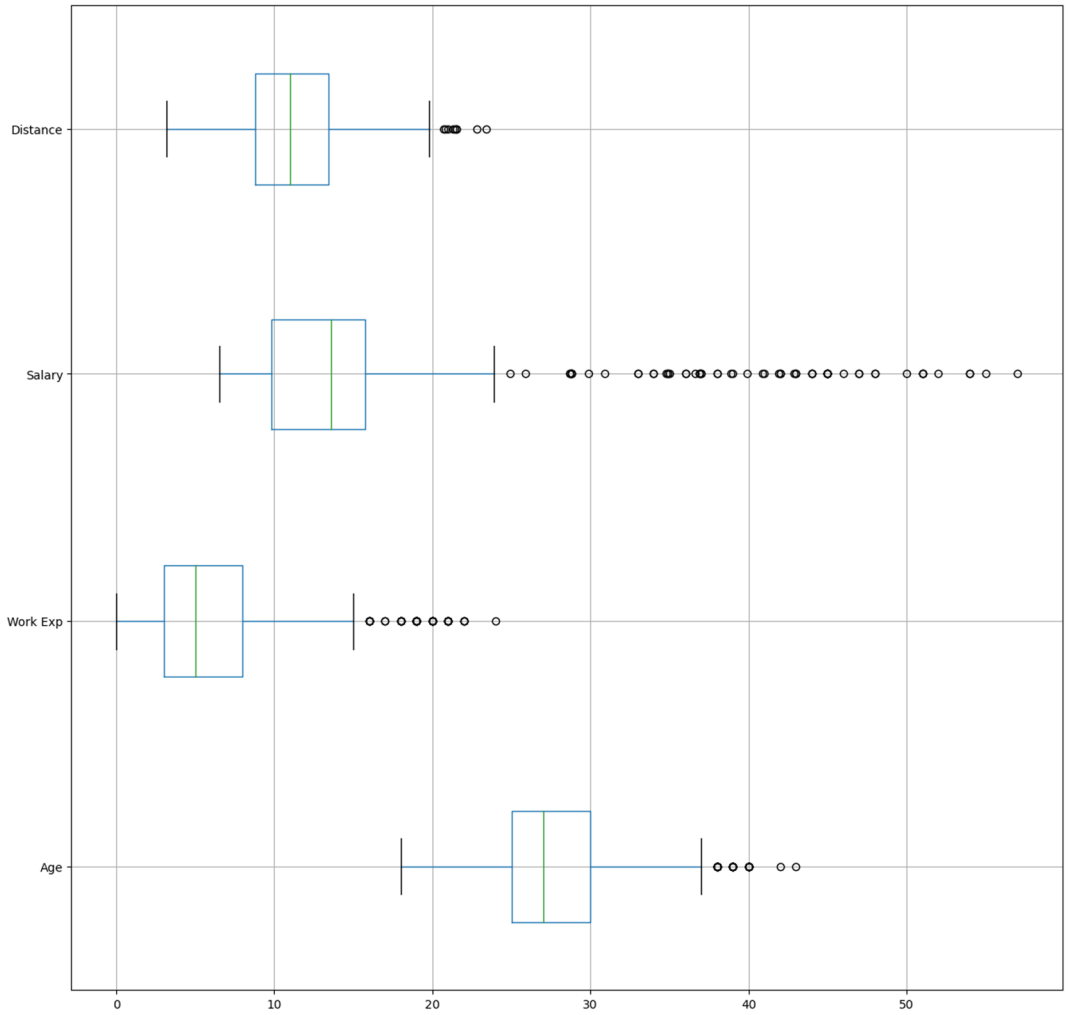

Fig 1.2.17–Univariate analysis (Box Plot)

All the numerical variable in the dataset contains outliers, let us check the outlier percentage of the variables,

	Outliers %
Age	5.63
Distance	2.03
Engineer	24.55
Gender	0.00
MBA	0.00
Salary	13.29
Transport	0.00
Work Exp	8.56
license	23.42

Table 1.2.1 Outlier Percentage

Engineer and License are actually categorical variables, so no need to consider those columns, among the numerical variables salary column contains highest percentage of outliers. Since the information we received is very less and we don't have any domain expert advice on this we decide not to treat the outliers.

PROBLEM 2 - DATA PREPARATION

2.1 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)

	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 2.1.1 –Dataset before encoding.

In the dataset gender and transport have string values, we have to encode those columns, and using pd.categorical method we can encode the columns having string values.

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

Table 2.1.2 –Dataset after encoding.

Next let us check the data type of the encoded 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    int8
 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    int8
dtypes: float64(2), int64(5), int8(2)
memory usage: 25.3 KB
```

Table 2.1.3 –Data type after encoding

Now all the categorical columns are converted to numerical columns.

Next let us check about the scaling of the dataset, when we building the KNN model scaling is required, We do scaling of the dataset to bring all features onto the same scale or range of values. Using standardscaler we can scale the data. Before scaling we can split the data in for train and test in the ratio 70:30 using train test split funct ion. The shape of the train and test data shown below,

X_train (310, 8)
X_test (134, 8)
y_train (310,)
y_test (134,)

Now we can scale the independent variables, after scaling train and test dataset is shown below.

	Age	Gender	Engineer	MBA	Work Exp	Salary	Distance	license
0	0.331618	0.659686	0.0	-0.594737	-0.218276	0.337996	-0.213637	0.0
1	-0.152154	0.659686	0.0	1.681416	0.013853	-0.223344	1.213093	0.0
2	-0.152154	0.659686	0.0	-0.594737	0.013853	-0.223344	0.569666	0.0
3	-1.119699	0.659686	0.0	-0.594737	-1.378917	-1.346022	0.122064	0.0
4	0.573504	0.659686	0.0	-0.594737	0.245981	0.150883	0.765491	0.0

Table 2.1.4 –Train data after scaling.

	Age	Gender	Engineer	MBA	Work Exp	Salary	Distance	license
0	-0.394040	-1.515873	0.0	-0.594737	0.478109	0.094749	0.094089	0.0
1	-0.152154	0.659686	0.0	1.681416	-0.218276	-0.036231	-0.353513	0.0
2	1.782935	-1.515873	0.0	-0.594737	2.219072	1.968218	-0.297563	0.0
3	-0.877813	0.659686	0.0	1.681416	-0.914661	-1.027930	-1.360616	0.0
4	1.299163	0.659686	0.0	-0.594737	1.174494	0.487686	-0.101737	0.0

Table 2.1.5 –Test data after scaling.

PROBLEM 3 – MODELING

3.1 Apply Logistic Regression

	Age	Engineer	MBA	Work_Exp	Salary	Distance	license	Transport	Gender_Male
0	28	0	0	4	14.3	3.2	0	1	1
1	23	1	0	4	8.3	3.3	0	1	0
2	29	1	0	7	13.4	4.1	0	1	1
3	28	1	1	5	13.4	4.5	0	1	0
4	27	1	0	4	13.4	4.6	0	1	1

Table 3.1.1 –Encoded data

We encoded the data before applying logistic regression. Now using stats model we can build the basic logistic regression model.

Logit Regression Results

Dep. Variable:	Transport	No. Observations:	444
Model:	Logit	Df Residuals:	435
Method:	MLE	Df Model:	8
Date:	Sat, 18 Mar 2023	Pseudo R-squ.:	0.3049
Time:	10:55:35	Log-Likelihood:	-194.45
converged:	True	LL-Null:	-279.76
Covariance Type:	nonrobust	LLR p-value:	9.551e-33

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-0.8340	1.788	-0.466	0.641	-4.339	2.671
Age	0.2083	0.077	2.706	0.007	0.057	0.359
Engineer	-0.1543	0.296	-0.521	0.603	-0.735	0.427
MBA	0.5601	0.314	1.782	0.075	-0.056	1.176
Work_Exp	-0.1005	0.100	-1.001	0.317	-0.297	0.096
Salary	-0.0805	0.040	-2.003	0.045	-0.159	-0.002
Distance	-0.2248	0.043	-5.290	0.000	-0.308	-0.142
license	-2.0463	0.334	-6.135	0.000	-2.700	-1.393
Gender_Male	1.2810	0.288	4.441	0.000	0.716	1.846

Table 3.1.2 –Basic logistic regression model.

Let us check the accuracy score of the train and test set of the basic model,

Accuracy Score of train set : 0.8193548387096774
Accuracy Score of test set : 0.8059701492537313

The parameters used for logistic regression are,

Solver: The algorithm used to optimize the weights. The default is 'lbfgs', but other o ptions include 'newton-cg', 'sag', and 'saga'

Penalty: Specifies the type of regularization used to prevent overfitting. The default i s 'l2' (ridge regularization), but 'l1' (lasso regularization) can also be used.

Now let us check the confusion matrix of the train and test set of the basic model.

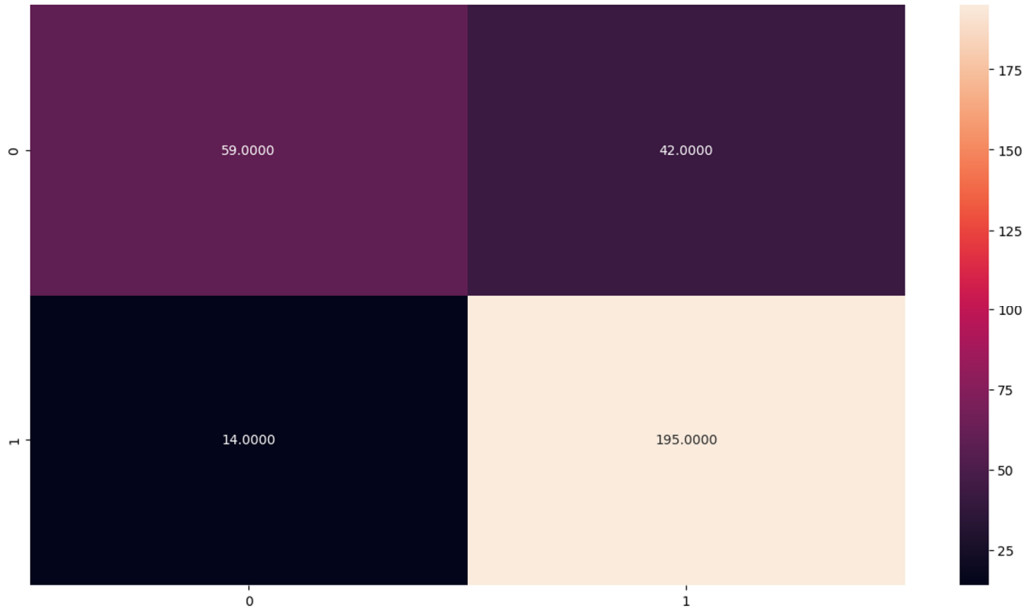

Table 3.1.3 –Confusion matrix of train set

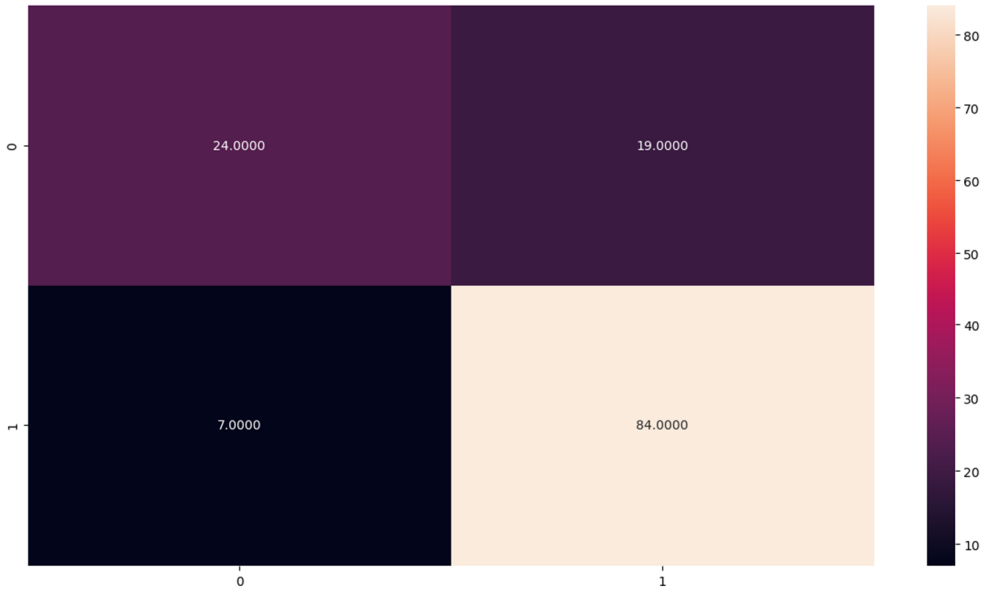

Table 3.1.4 –Confusion matrix of test set

Now let us check the classification report of the test and train set of the basic model.

	precision	recall	f1-score	support
0	0.81	0.58	0.68	101
1	0.82	0.93	0.87	209
accuracy			0.82	310
macro avg	0.82	0.76	0.78	310
weighted avg	0.82	0.82	0.81	310

Table 3.1.5 –Classification report of the train set

	precision	recall	f1-score	support
0	0.77	0.56	0.65	43
1	0.82	0.92	0.87	91
accuracy			0.81	134
macro avg	0.79	0.74	0.76	134
weighted avg	0.80	0.81	0.80	134

Table 3.1.6 –Classification report of the test set

Next let us check the ROC-AUC curve and ROC-AUC score of the test and train set,

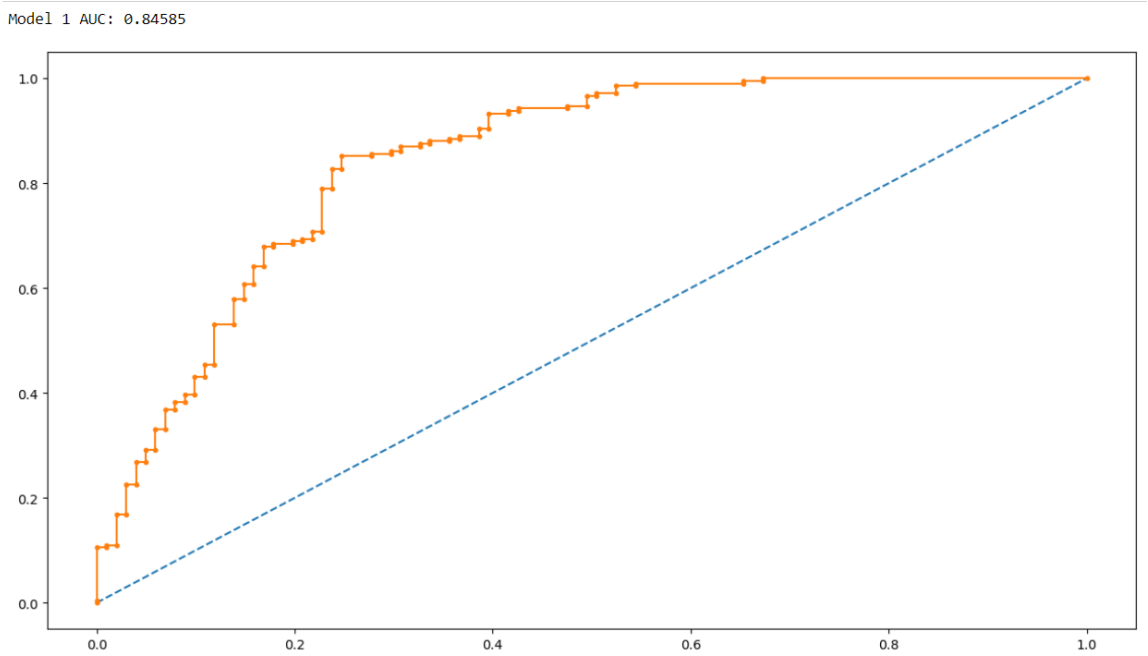

Fig 3.1.1 –ROC-AUC curve of train set

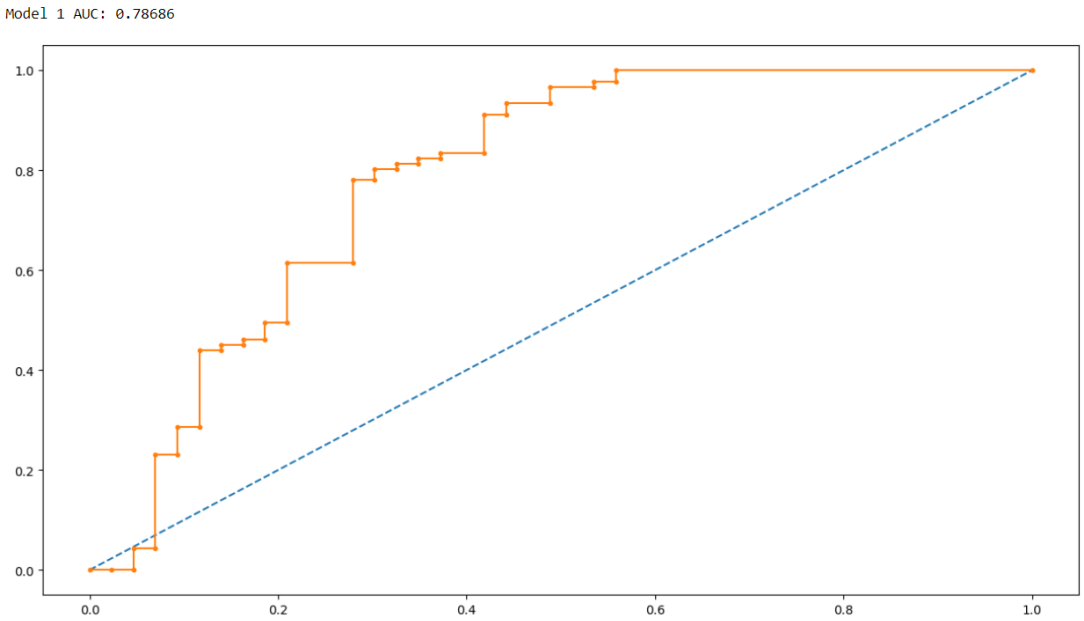

Fig 3.1.2 –ROC-AUC curve of test set

The accuracy score of the train and test set of the basic model of the logistic regression didn't have much difference, so the basic logistic regression model is acceptable.

3.2 Apply KNN Model. Interpret the results.

Next let us create KNN model on the scaled dataset we created before using kneighbourclassifier algorithm.

The accuracy score of the basic KNN model of the train and test data is shown below,

Train Accuracy is : 0.8419354838709677

Test Accuracy is : 0.7835820895522388

Train ROC-AUC score is : 0.9224877450980392

Test ROC-AUC score is : 0.7496118012422359

Next let us check the confusion matrix of the train and test data of the basic KNN model,

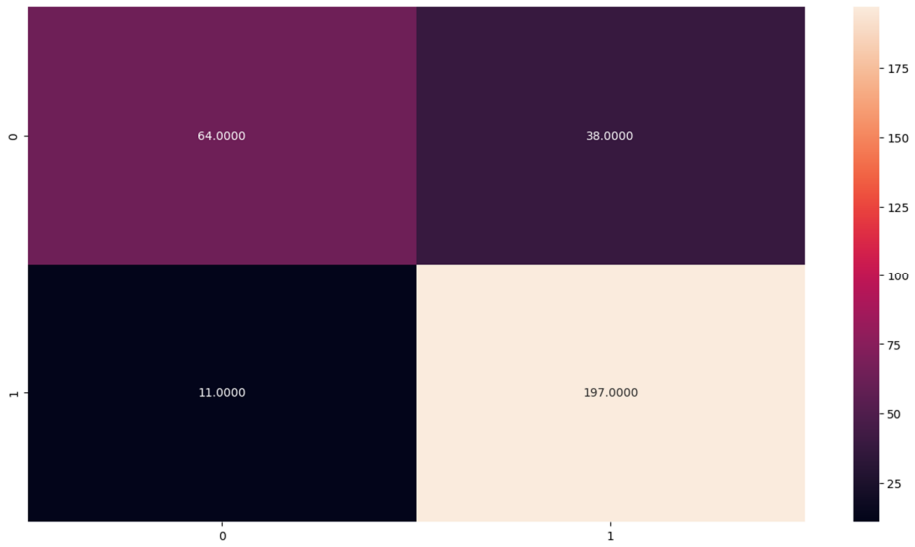

Table 3.2.1 –Confusion matrix of train set

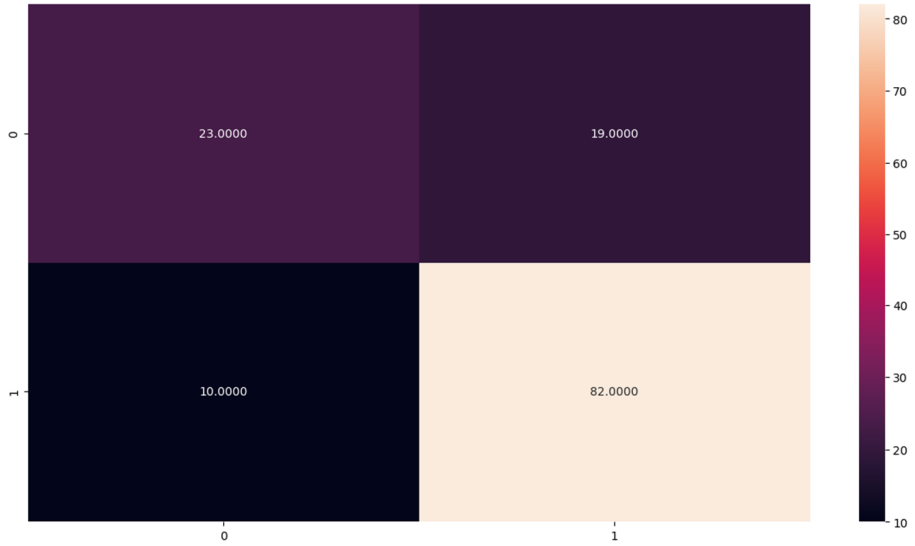

Table 3.2.2–Confusion matrix of test set

Next let us check the classification report of the train and test data of the basic KNN model,

Classification report Train set :

	precision	recall	f1-score	support
0	0.85	0.63	0.72	102
1	0.84	0.95	0.89	208
accuracy			0.84	310
macro avg	0.85	0.79	0.81	310
weighted avg	0.84	0.84	0.83	310

Table 3.2.3 –Classification report of train set

Classification report Test set :

	precision	recall	f1-score	support
0	0.70	0.55	0.61	42
1	0.81	0.89	0.85	92
accuracy			0.78	134
macro avg	0.75	0.72	0.73	134
weighted avg	0.78	0.78	0.78	134

Table 3.2.4 –Classification report of test set

Next let us check the ROC-AUC curve and ROC-AUC score of the train and test data of the basic KNN model,

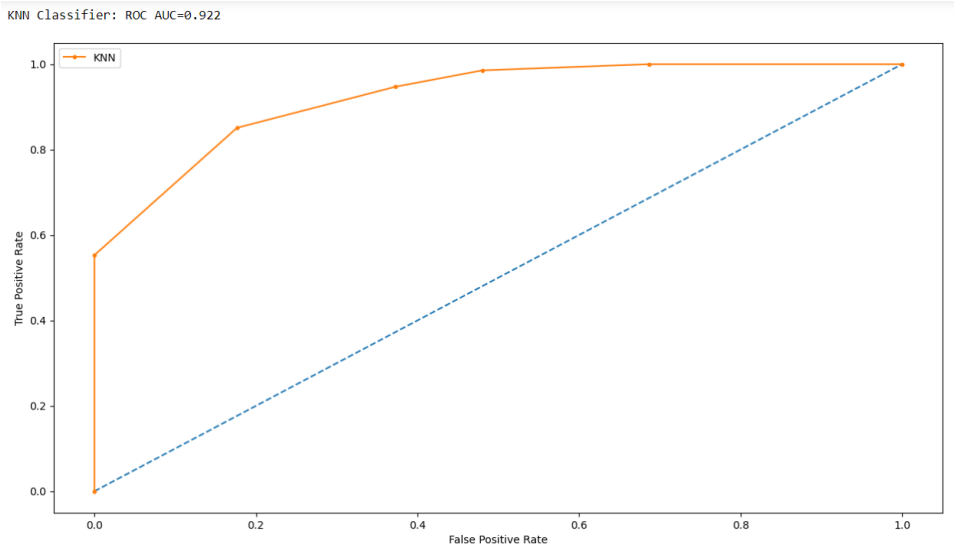

Fig 3.2.1 –ROC-AUC curve of train set.

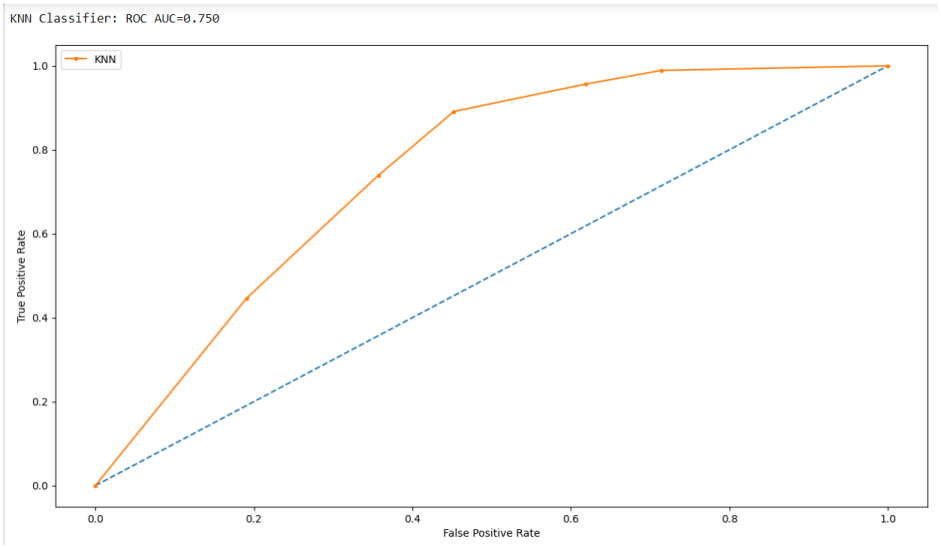

Fig 3.2.2 –ROC-AUC curve of test set.

There is large difference between ROC-AUC score of the test and train data, we need to do model tuning on the data to conform.

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

Bagging

For creating bagging model we have to create random forest model first, using random forest classifier we can create random forest model.

Next we have to import this random forest model to bagging classifier to create the basic bagging model,

The basic parameter using for bagging are,

base_estimator: This parameter specifies the type of base estimator to use. It can be any classifier or regressor that implements the scikit-learn estimator interface.

n_estimators: This parameter specifies the number of base estimators to use in the ensemble. Increasing the number of estimators can improve the performance of the Bagging Classifier, but also increases the training time and memory requirements.

Random state

The model score of train and test data of bagging model is shown below,

Model score of train data : 0.97096

Model score of test data : 0.8059

Next let us check the confusion matrix of the train and test data of the bagging model,

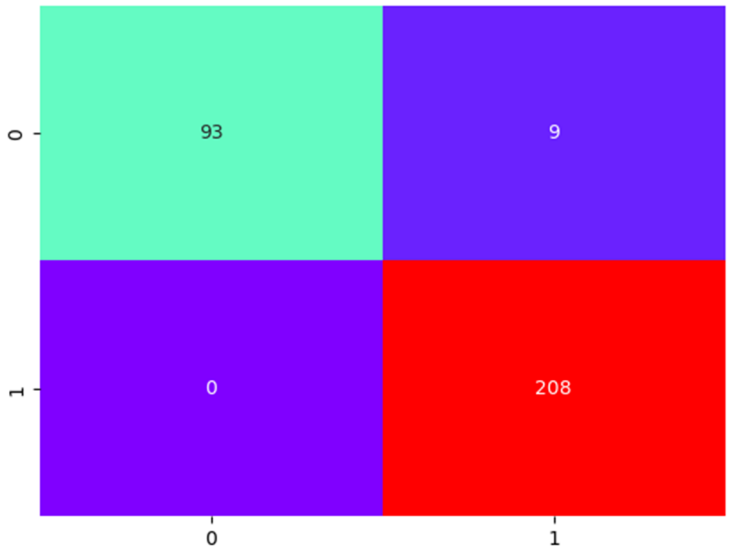

Table 3.3.1 –Confusion matrix of train set

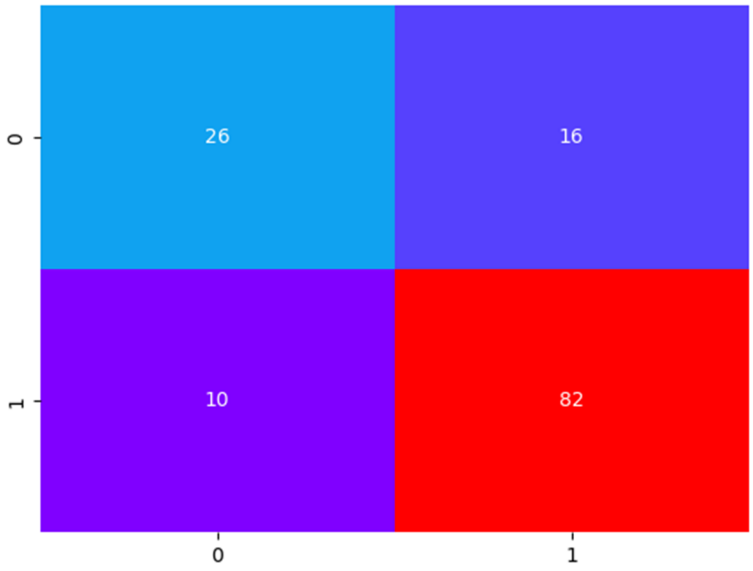

Table 3.3.2 –Confusion matrix of test set

Next let us check the classification report of the train and test data of the basic bagging model,

	precision	recall	f1-score	support
0	1.00	0.91	0.95	102
1	0.96	1.00	0.98	208
accuracy			0.97	310
macro avg	0.98	0.96	0.97	310
weighted avg	0.97	0.97	0.97	310

Table 3.3.3 –Classification report of train data

	precision	recall	f1-score	support
0	0.72	0.62	0.67	42
1	0.84	0.89	0.86	92
accuracy			0.81	134
macro avg	0.78	0.76	0.76	134
weighted avg	0.80	0.81	0.80	134

Table 3.3.4 –Classification report of test data.

Next let us check the ROC-AUC curve and ROC-AUC score of the train and test data of the basic bagging model,

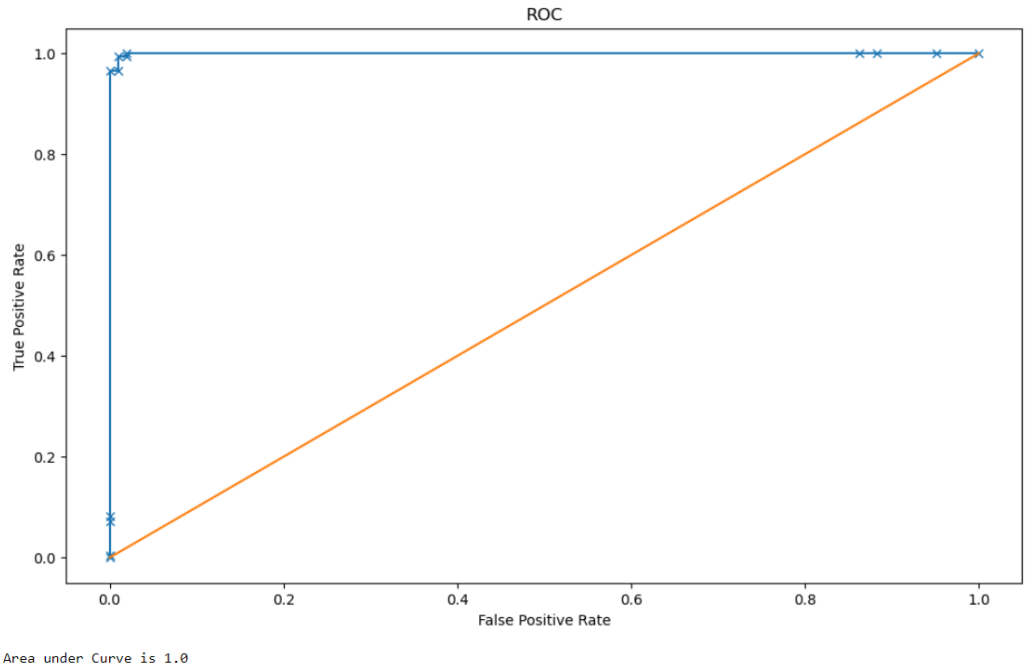

Fig 3.3.1 –ROC-AUC curve of train set.

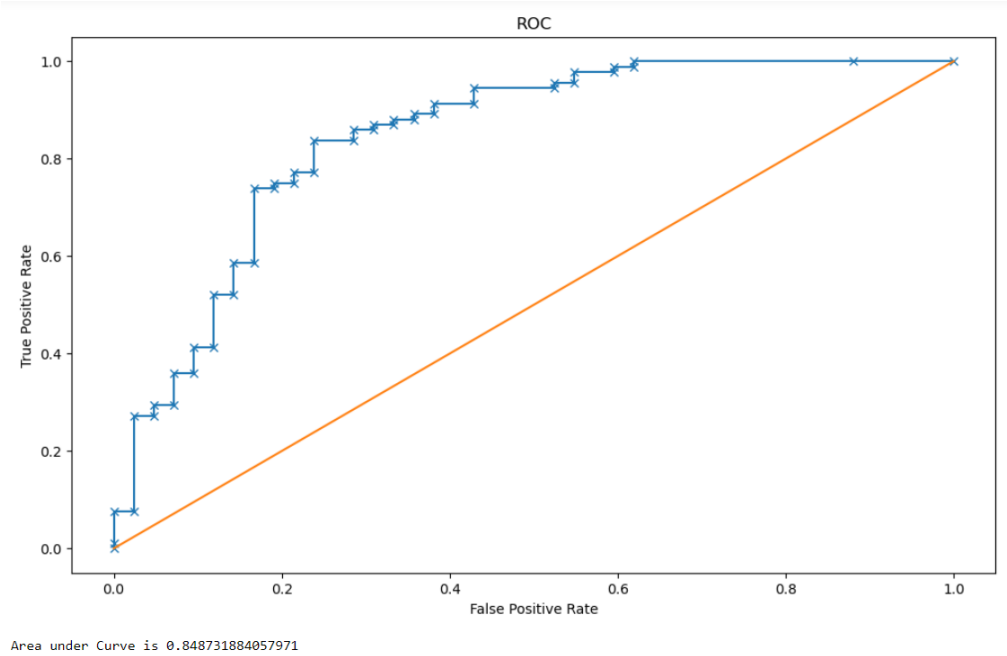

Fig 3.3.2 –ROC-AUC curve of test set.

From the above ROC – AUC score and the accuracy score it is very clear that the model is over fitted model, we have to perform the model tuning technique to check the performance of the bagging model.

Adaboost

For creating adaboost model we have to create random forest model first, using random forest classifier we can create random forest model.

Next we have to import this random forest model to adaboost classifier to create the basic adaboost model,

The basic parameter using for adaboost are,

base_estimator: This parameter specifies the type of base estimator to use. It can be any classifier or regressor that implements the scikit-learn estimator interface. By default, it uses a decision tree with max_depth=1, which is called a "stump".

n_estimators: This parameter specifies the number of base estimators to use in the ensemble. Increasing the number of estimators can improve the performance of the AdaBoost algorithm, but also increases the training time and memory requirements.

The model score of train and test data of adaboost model is shown below,

Model score of train data : 0.825806

Model score of test data : 0.8059

Next let us check the confusion matrix of the train and test data of the adaboost model,

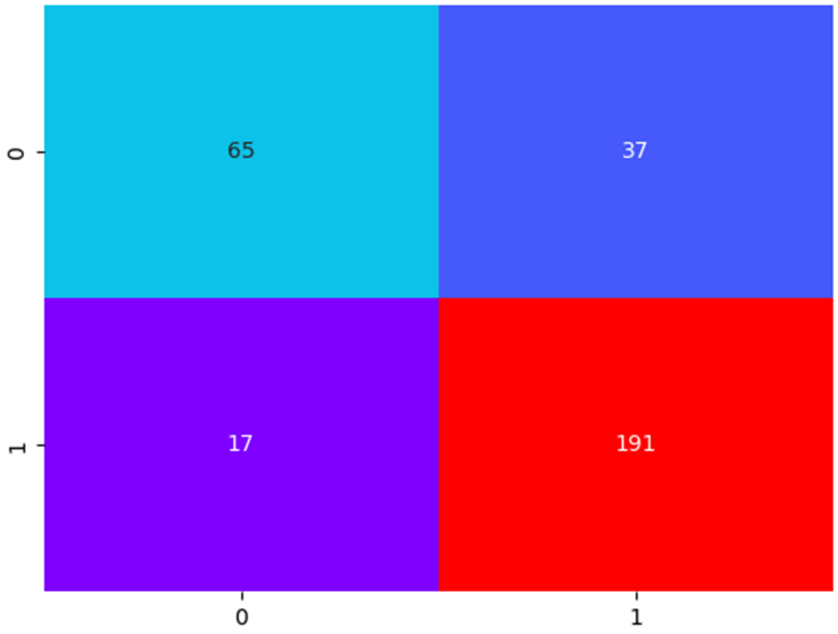

Table 3.3.5 –Confusion matrix of train set

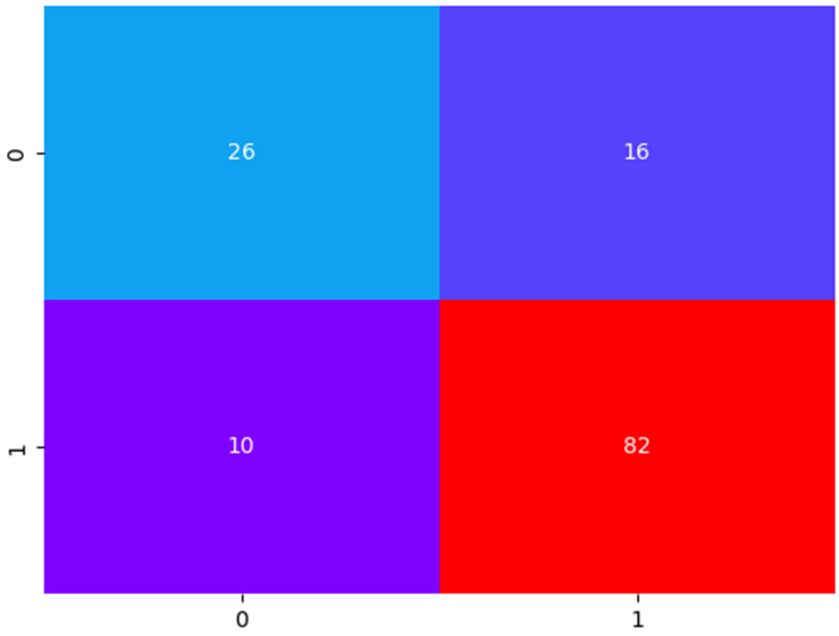

Table 3.3.6 –Confusion matrix of test set

Next let us check the classification report of the train and test data of the adaboost model,

	precision	recall	f1-score	support
0	0.79	0.64	0.71	102
1	0.84	0.92	0.88	208
accuracy			0.83	310
macro avg	0.82	0.78	0.79	310
weighted avg	0.82	0.83	0.82	310

Table 3.3.7 –Classification report of train set

	precision	recall	f1-score	support
0	0.72	0.62	0.67	42
1	0.84	0.89	0.86	92
accuracy			0.81	134
macro avg	0.78	0.76	0.76	134
weighted avg	0.80	0.81	0.80	134

Table 3.3.8 –Classification report of test set

Next let us check the ROC-AUC curve and ROC-AUC score of the train and test data of the basic adaboost model,

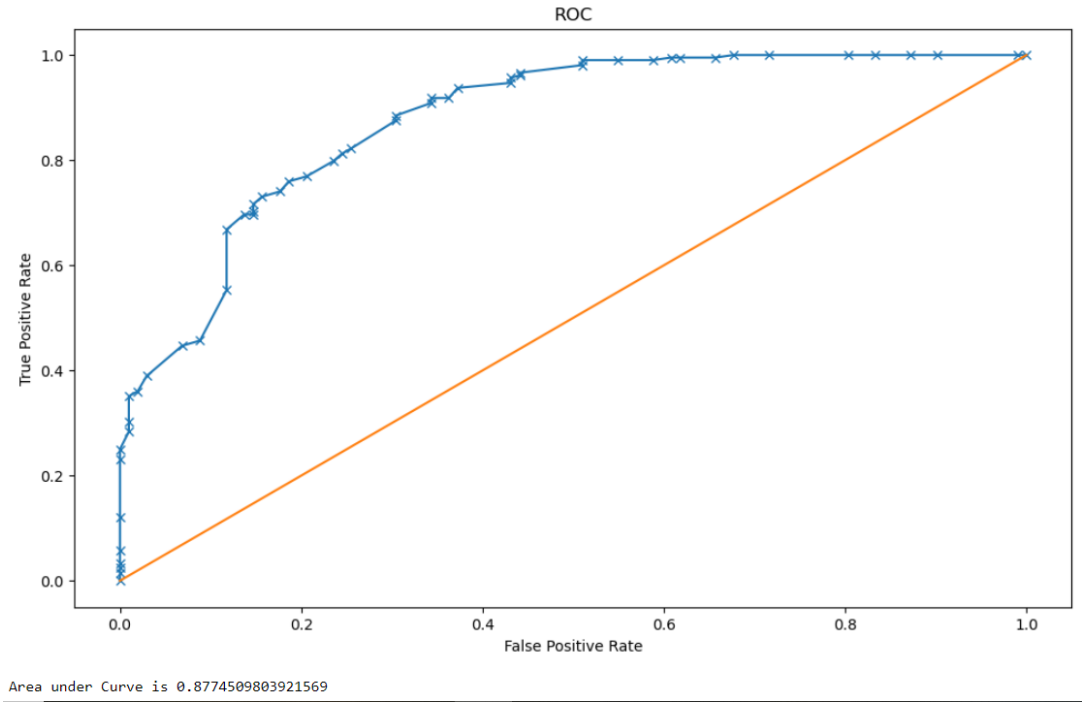

Fig 3.3.3 –ROC-AUC curve of train set.

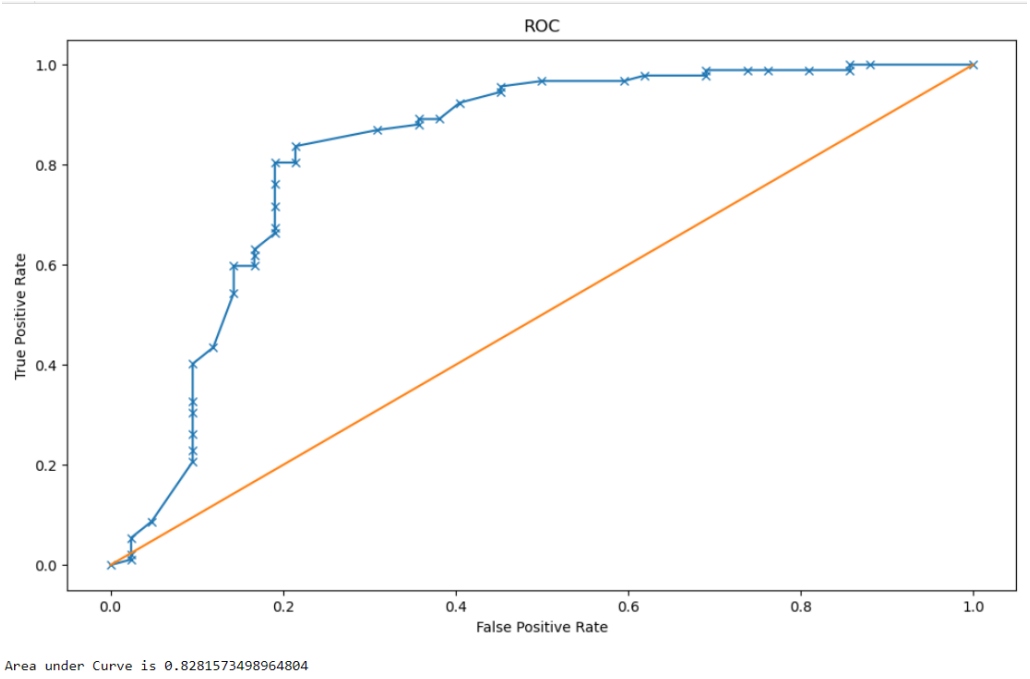

Fig 3.3.4 –ROC-AUC curve of test set.

There is not much difference between the accuracy score and ROC – AUC score of test and train data, so the base model is looks good, need to do model tuning and to check the model still have this consistency.

Model Tuning

Model tuning, also known as hyperparameter tuning, is the process of finding the optimal hyperparameters for a machine learning model in order to achieve the best possible performance on a given dataset. Hyperparameters are the parameters of a model that are not learned during training, but are set before training begins. These can include things like learning rate, regularization strength, number of hidden layers, and more, depending on the type of model being used.

We will use grid search as the model tuning technique here,

Model tuning for logistic regression model.

Below shown are the parameters used for the model tuning of the logistic regression model in grid search.

penalty: The regularization penalty to use. Options include 'l1', 'l2', 'elasticnet', and 'none'.

C: The inverse of regularization strength. A smaller value of C will result in stronger regularization.

solver: The algorithm to use for optimization. Options include 'newton-cg', 'lbfgs', 'liblinear', 'sag', and 'saga'.

max_iter: The maximum number of iterations to run the solver for.

Model tuning for KNN model.

Below shown are the parameters used for the model tuning of the KNN model in grid search.

n_neighbors: The number of nearest neighbors to consider.

weights: The weight function to use in prediction. Options include 'uniform' (all points in the neighborhood are weighted equally) and 'distance' (the weight assigned to each point is proportional to the inverse of its distance from the query point).

p: The power parameter for the Minkowski metric. When p=1, this is equivalent to using the Manhattan distance; when p=2, this is equivalent to using the Euclidean distance.

Model tuning for Bagging model.

Below shown are the parameters used for the model tuning of the bagging model in grid search.

base_estimator: The base estimator to use for each bag. This can be any supervised learning algorithm, such as DecisionTreeClassifier or LogisticRegression.

n_estimators: The number of bags to use in the ensemble.

Model tuning for Adaboost model.

Below shown are the parameters used for the model tuning of the bagging model in grid search

base_estimator: The base estimator to use for each weak learner. This can be any supervised learning algorithm, such as DecisionTreeClassifier or LogisticRegression.

n_estimators: The number of weak learners to use in the ensemble.

learning_rate: The contribution of each weak learner to the final prediction. A smaller learning rate will typically result in a more conservative ensemble.

3.4 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 after Grid search

Let us check the logistic regression model after applying grid search, below shown are the accuracy score and the confusion matrix of the train and test data after applying grid search in logistic regression model.

```
Train Accuracy is : 0.7870967741935484

Test Accuracy is : 0.8208955223880597

Train ROC-AUC score is : 0.8284785067873303

Test ROC-AUC score is : 0.8219461697722568

Confusion matrix for train set :
 [[ 53  49]
 [ 17 191]]

Confusion matrix for test set :
 [[25 17]
 [ 7 85]]
```

Table 3.4.1 –Confusion matrix and accuracy score.

Next let us check the classification report of the logistic regression model after applying grid search.

Classification report Train set :

	precision	recall	f1-score	support
0	0.76	0.52	0.62	102
1	0.80	0.92	0.85	208
accuracy			0.79	310
macro avg	0.78	0.72	0.73	310
weighted avg	0.78	0.79	0.77	310

Table 3.4.2 –Classification report train set.

Classification report Test set :

	precision	recall	f1-score	support
0	0.78	0.60	0.68	42
1	0.83	0.92	0.88	92
accuracy			0.82	134
macro avg	0.81	0.76	0.78	134
weighted avg	0.82	0.82	0.81	134

Table 3.4.3 –Classification report test set.

Next let us check the ROC-AUC of the logistic regression model after applying grid search.

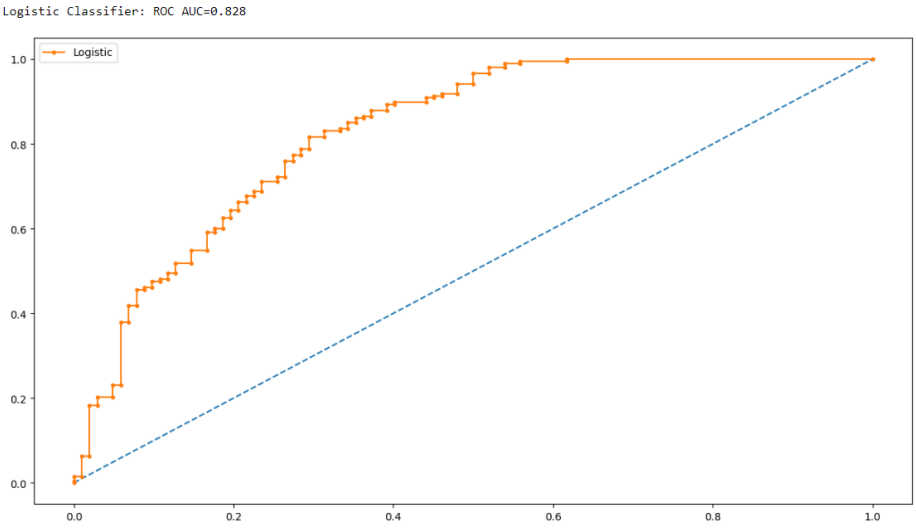

Fig 3.4.1 –ROC-AUC curve train set.

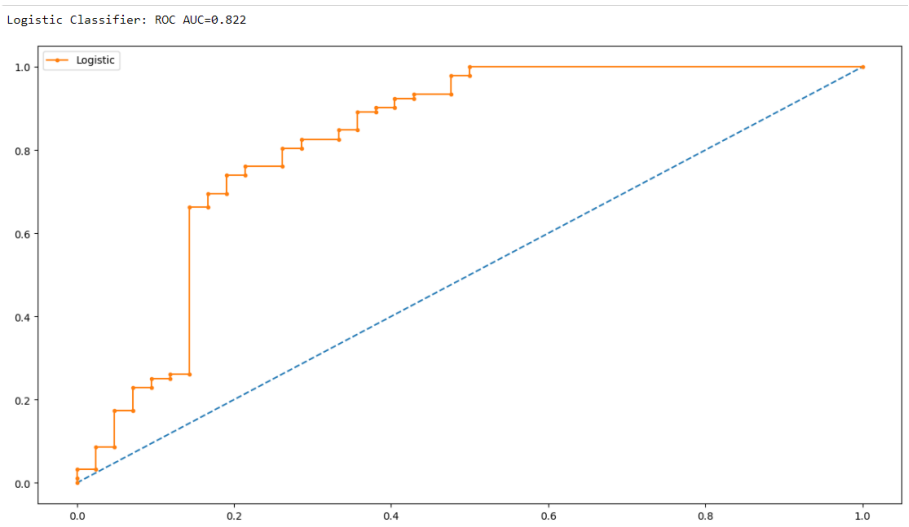

Fig 3.4.2 –ROC-AUC curve test set.

Even after applying grid search the accuracy score and AUC-ROC score seems to be good, so we can consider Logistic regression model as a good model.

KNN Model after Grid search

Let us check the KNN model after applying grid search, below shown are the accuracy score and the confusion matrix of the train and test data after applying grid search in KNN model.

```
Train Accuracy is : 1.0

Test Accuracy is : 0.7611940298507462

Train ROC-AUC score is : 1.0

Test ROC-AUC score is : 0.7458592132505175

Confusion matrix for train set :
 [[102   0]
 [  0 208]]

Confusion matrix for test set :
 [[24 18]
 [14 78]]
```

Table 3.4.4 –Confusion matrix and accuracy score.

Next let us check the classification report of the KNN model after applying grid search.

Classification report Train set :

	precision	recall	f1-score	support
0	1.00	1.00	1.00	102
1	1.00	1.00	1.00	208
accuracy			1.00	310
macro avg	1.00	1.00	1.00	310
weighted avg	1.00	1.00	1.00	310

Table 3.4.5 –Classification report train set.

Classification report Test set :

	precision	recall	f1-score	support
0	0.63	0.57	0.60	42
1	0.81	0.85	0.83	92
accuracy			0.76	134
macro avg	0.72	0.71	0.71	134
weighted avg	0.76	0.76	0.76	134

Table 3.4.6 –Classification report test set.

Next let us check the ROC-AUC of the KNN model after applying grid search.

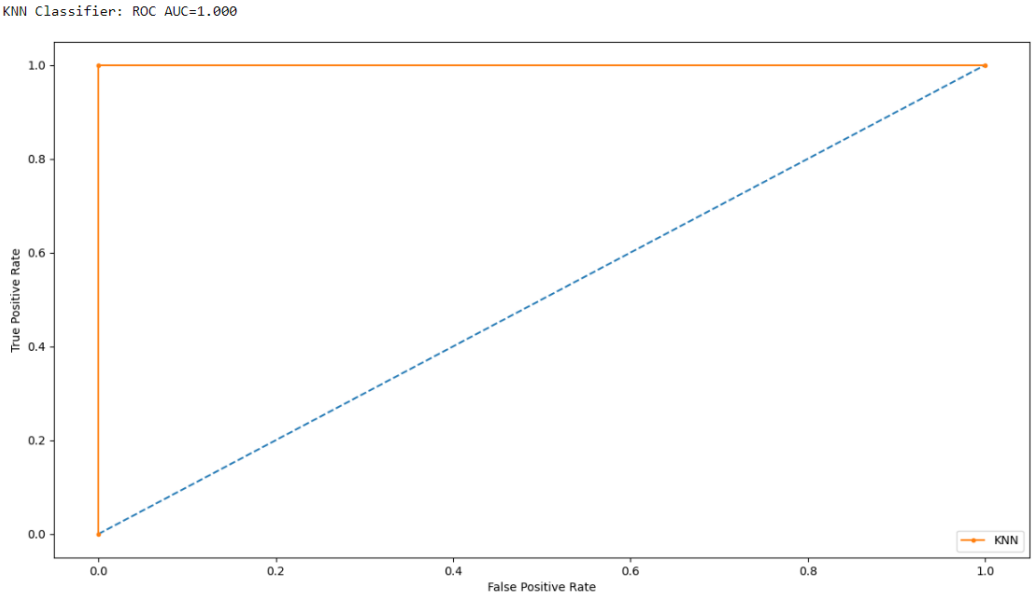

Fig 3.4.3 –ROC-AUC curve train set.

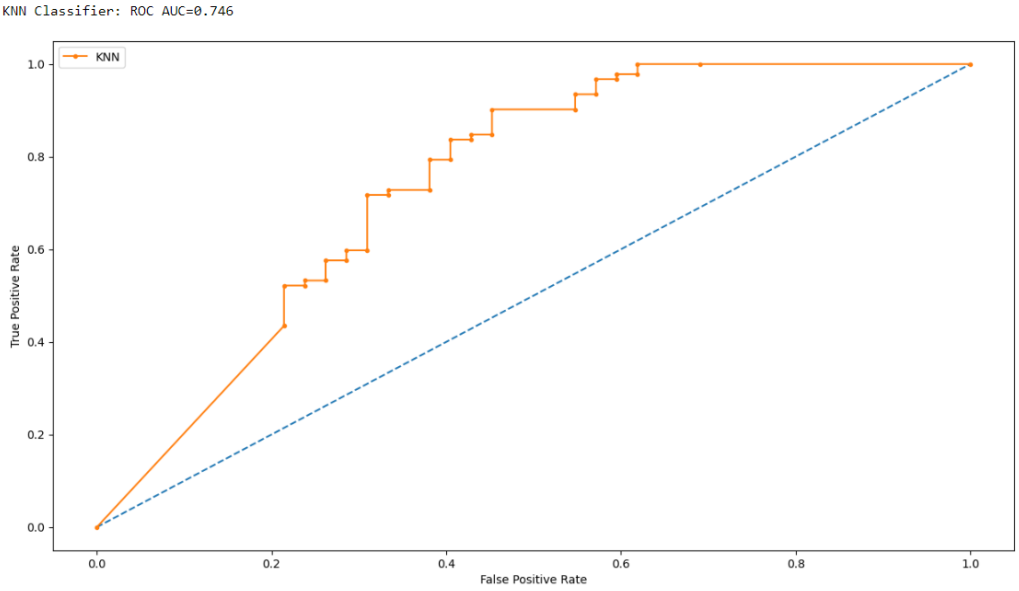

Fig 3.4.4 –ROC-AUC curve test set.

There is a large difference between the accuracy score and ROC-AUC score of the test and train data, and the model is seems to be over fitted, so we cannot use this model.

Bagging Model after Grid search

Let us check the bagging model after applying grid search, below shown are the accuracy score and the confusion matrix of the train and test data after applying grid search in bagging model.

```
Train Accuracy is : 0.9483870967741935

Test Accuracy is : 0.8134328358208955

Train ROC-AUC score is : 0.9972190799396681

Test ROC-AUC score is : 0.8403209109730849

Confusion matrix for train set :
 [[ 86  16]
 [  0 208]]

Confusion matrix for test set :
 [[26 16]
 [ 9 83]]
```

Table 3.4.7 –Confusion matrix and accuracy score.

Next let us check the classification report of the bagging model after applying grid search.

Classification report Train set :

	precision	recall	f1-score	support
0	1.00	0.84	0.91	102
1	0.93	1.00	0.96	208
accuracy			0.95	310
macro avg	0.96	0.92	0.94	310
weighted avg	0.95	0.95	0.95	310

Table 3.4.8 –Classification report train set.

Classification report Test set :

	precision	recall	f1-score	support
0	0.74	0.62	0.68	42
1	0.84	0.90	0.87	92
accuracy			0.81	134
macro avg	0.79	0.76	0.77	134
weighted avg	0.81	0.81	0.81	134

Table 3.4.9 –Classification report test set.

Next let us check the ROC-AUC of the bagging model after applying grid search.

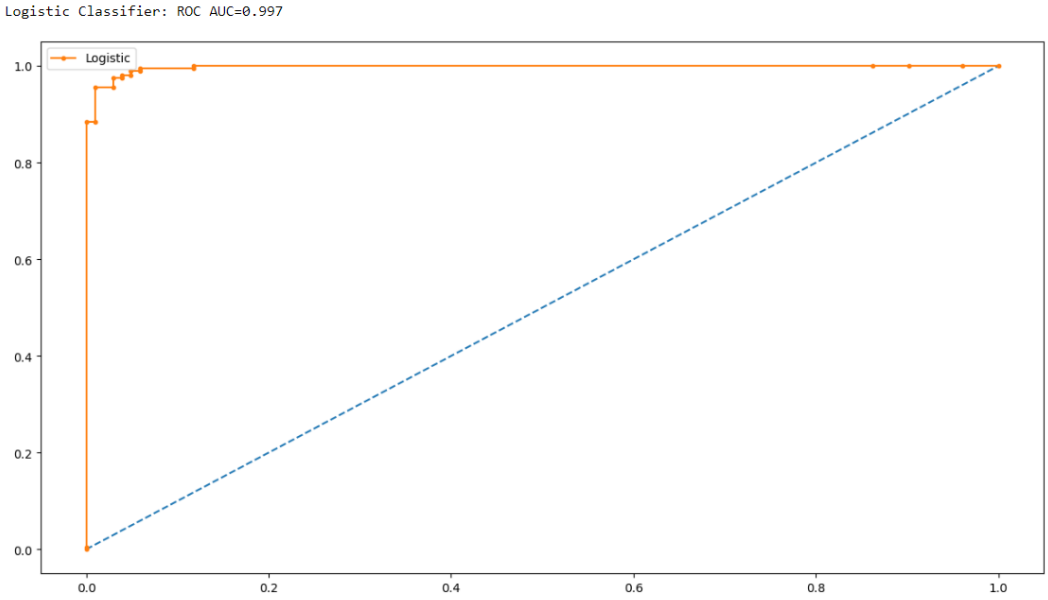

Fig 3.4.5 –ROC-AUC curve train set.

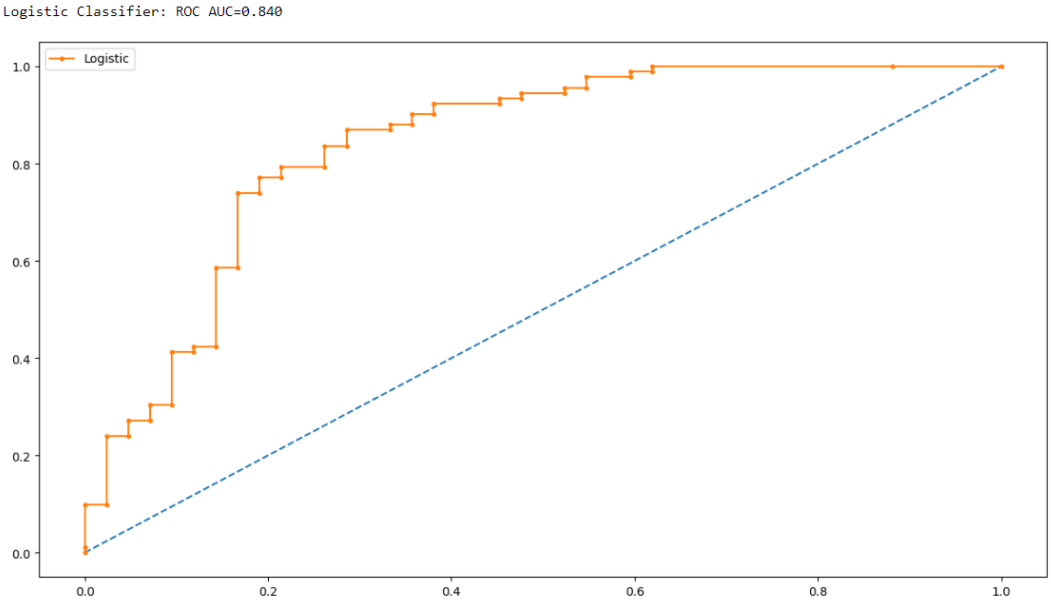

Fig 3.4.6 –ROC-AUC curve test set.

The model is seems to be over fitted.

Adaboost Model after Grid search.

Let us check the adaboost model after applying grid search, below shown are the accuracy score and the confusion matrix of the train and test data after applying grid search in adaboost model.

```
Train Accuracy is : 0.9483870967741935

Test Accuracy is : 0.8134328358208955

Train ROC-AUC score is : 1.0

Test ROC-AUC score is : 0.8330745341614907

Confusion matrix for train set :
 [[ 86  16]
 [  0 208]]

Confusion matrix for test set :
 [[26 16]
 [ 9 83]]
```

Table 3.4.10 –Confusion matrix and accuracy score.

Next let us check the classification report of the adaboost model after applying grid search.

Classification report Train set :

	precision	recall	f1-score	support
0	1.00	0.84	0.91	102
1	0.93	1.00	0.96	208
accuracy			0.95	310
macro avg	0.96	0.92	0.94	310
weighted avg	0.95	0.95	0.95	310

Table 3.4.11 –Classification report train set.

Classification report Test set :

	precision	recall	f1-score	support
0	0.74	0.62	0.68	42
1	0.84	0.90	0.87	92
accuracy			0.81	134
macro avg	0.79	0.76	0.77	134
weighted avg	0.81	0.81	0.81	134

Table 3.4.12 –Classification report test set.

Next let us check the ROC-AUC of the bagging model after applying grid search

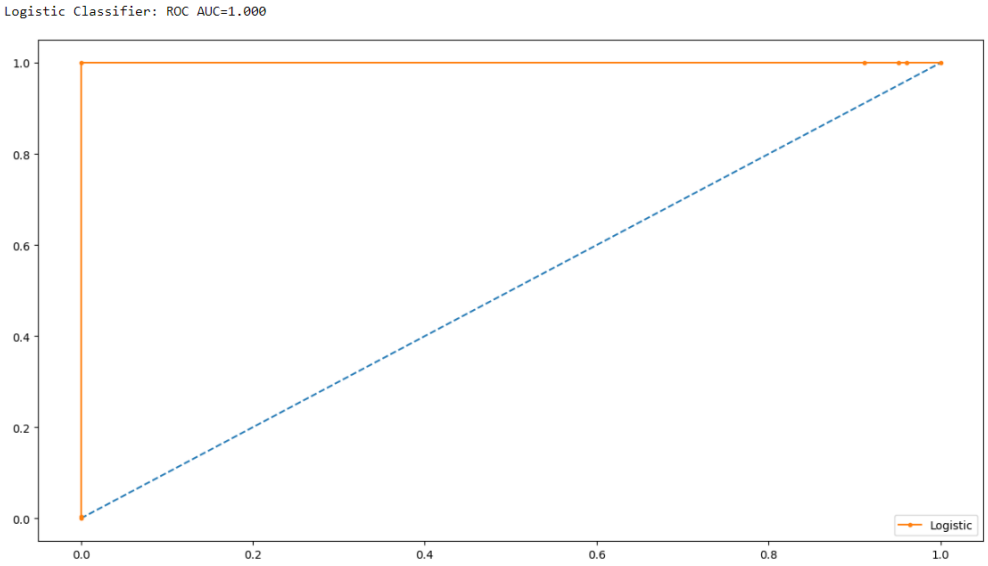

Fig 3.4.7 –ROC-AUC curve train set.

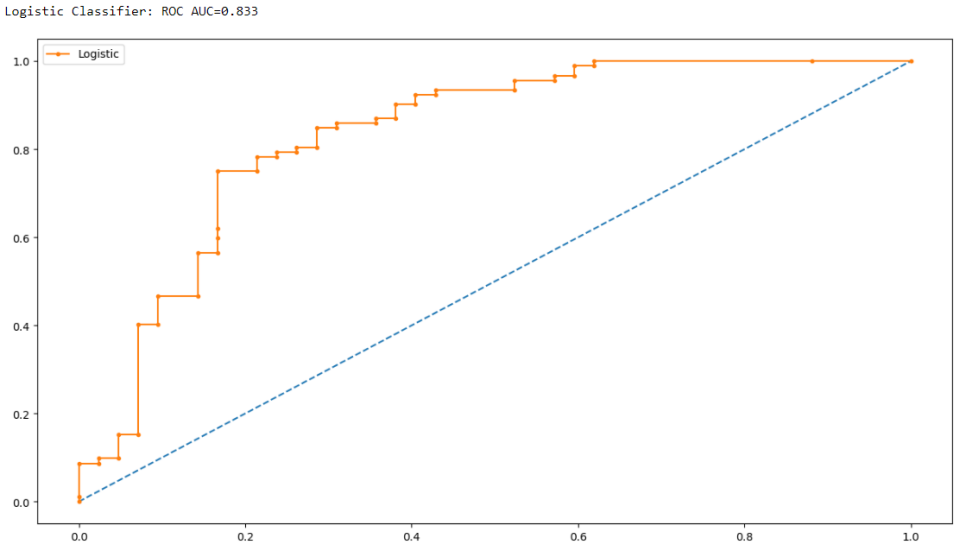

Fig 3.4.8 –ROC-AUC curve test set.

Logistic Regression

	BASE TRAIN	BASE TEST	GRID SEARCH TRAIN	GRID SEARCH TEST
ACCURACY	0.81	0.81	0.79	0.82
AUC	0.84	0.78	0.82	0.82
PRECISION	0.82	0.82	0.82	0.83
RECALL	0.93	0.92	0.92	0.92
F1-SCORE	0.87	0.87	0.87	0.88

Table 3.4.13 –Summary Logistic Regression.

KNN Model

	BASE TRAIN	BASE TEST	GRID SEARCH TRAIN	GRID SEARCH TEST
ACCURACY	0.84	0.78	1.00	0.76
AUC	0.92	0.75	1.00	0.75
PRECISION	0.84	0.81	1.00	0.81
RECALL	0.95	0.89	1.00	0.85
F1-SCORE	0.89	0.85	1.00	0.83

Table 3.4.14 –Summary KNN.

Bagging Model

	BASE TRAIN	BASE TEST	GRID SEARCH TRAIN	GRID SEARCH TEST
ACCURACY	0.97	0.81	0.95	0.81
AUC	1.0	0.84	0.99	0.84
PRECISION	0.96	0.84	0.93	0.84
RECALL	1.0	0.89	1.00	0.9
F1-SCORE	0.98	0.86	0.96	0.87

Table 3.4.15 –Summary Bagging.

Adaboost Model

	BASE TRAIN	BASE TEST	GRID SEARCH TRAIN	GRID SEARCH TEST
ACCURACY	0.83	0.81	0.95	0.81
AUC	0.87	0.82	1.00	0.83
PRECISION	0.84	0.84	0.99	0.83
RECALL	0.92	0.89	1.00	0.91
F1-SCORE	0.88	0.86	0.99	0.87

Table 3.4.16 –Summary Adaboost.

Considering the accuracy score and the F1 score adaboost model seems to be good considering the other 3 models.

PROBLEM 4 – INFERENCE

4.1 Based on these predictions, what are the insights?

1. Since most of the employees are below 30 years, most of them uses public transport, Younger employees may be more motivated by incentives such as discounts, rewards, or recognition programs. The transport company could consider offering these incentives to encourage more employees to use company-provided transportation services.
2. Considering salary, employees who have higher salary prefer mostly private transport system, and employees who have low salary prefer public transport system, so target those who have low salary and provide discounts and offer to them to encourage them to use office transport system.
3. For those employees uses metro and long distance buses, provide transport options form metro stations and bus stations.
4. Those who have high work experience will have high salary and they prefer mostly personal vehicles, and newly joined employees prefer public transport systems, so target these employees.
5. Since most of the employees doesn't have a driving license, they will prefer public transport, so make them use office transport vehicles.
6. Since most of the employees are male and below thirty years give them special offers to attract them to use office transport system.

THE END

CLICK HERE TO GO TO CONTENTS