MACHINE LEARNING PROJECT

AHMED BASHA K

30-10-2022

Project: 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.

The data is given in the File " Transport.csv" As shown below.

	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
...	...	...	...	...	...	...	...	...	...
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

444 rows × 9 columns

1.1) Read the dataset. Do the descriptive statistics and do null value condition check.

Head of the data

	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

Tail of the data

	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

Checking the data info

```
<class 'pandas. core. frame.DataFrame'>
Range Index: 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
```

There are total of 9 variables: float64(2) int64(5) object (2)

Summary of the data

	count	mean	std	min	25%	50%	75%	max
Age	444.0	27.747748	4.416710	18.0	25.0	27.0	30.000	43.0
Engineer	444.0	0.754505	0.430866	0.0	1.0	1.0	1.000	1.0
MBA	444.0	0.252252	0.434795	0.0	0.0	0.0	1.000	1.0
Work Exp	444.0	6.299550	5.112098	0.0	3.0	5.0	8.000	24.0
Salary	444.0	16.238739	10.453851	6.5	9.8	13.6	15.725	57.0
Distance	444.0	11.323198	3.606149	3.2	8.8	11.0	13.425	23.4
license	444.0	0.234234	0.423997	0.0	0.0	0.0	0.000	1.0

shape of the data frame

- Number of rows: 444
- Number of columns: 9

unique counts

```
GENDER:  2
Female    128
Male      316
Name: Gender, dtype: int64
```

```
TRANSPORT:  2
Private Transport    144
Public Transport     300
Name: Transport, dtype: int64
```

Checking for missing value in any column

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

From the above, it is clear that there are no null values.

Checking for duplicate data

Number of duplicate rows = 0

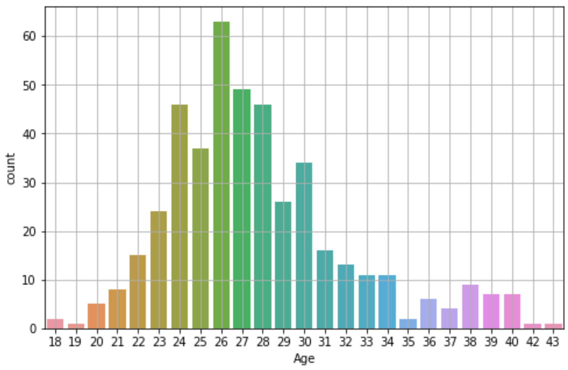
	Age	Gender	Engineer	MBA	Work Exp	Salary	Distance	license	Transport

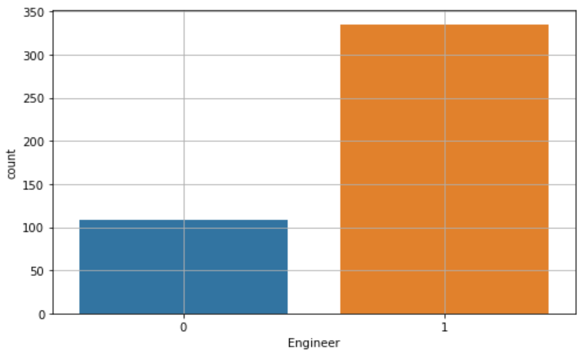
From the above, it is clear that there are no duplicated values.

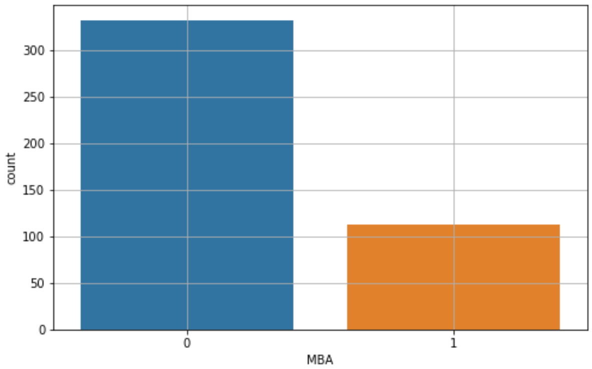
1.2) Perform EDA (Check the null values, Data types, shape, Univariate, bivariate analysis). Also check for outliers (5 pts). Interpret the inferences for each (3 pts)

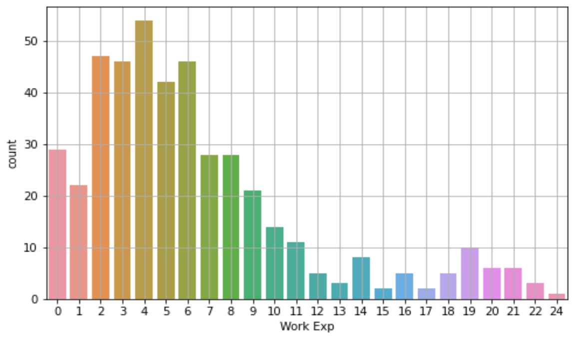
<u>Univariate Analysis</u>

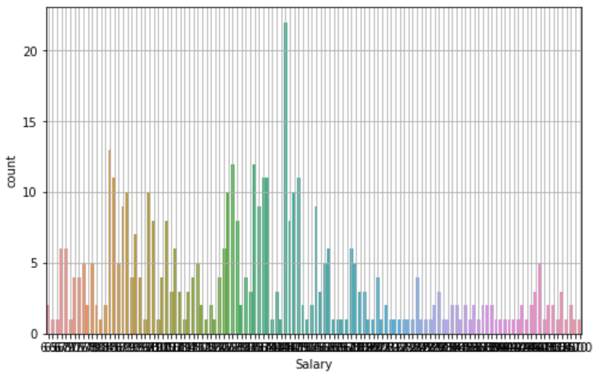
Checking the spread of the data using count plot for the continuous variable.

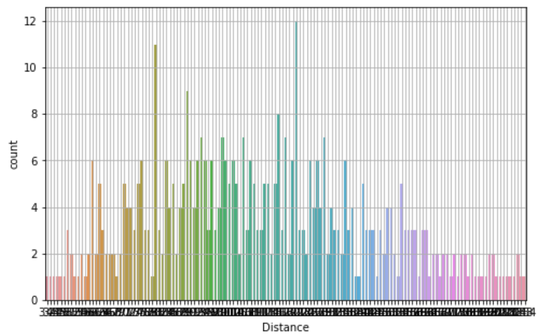


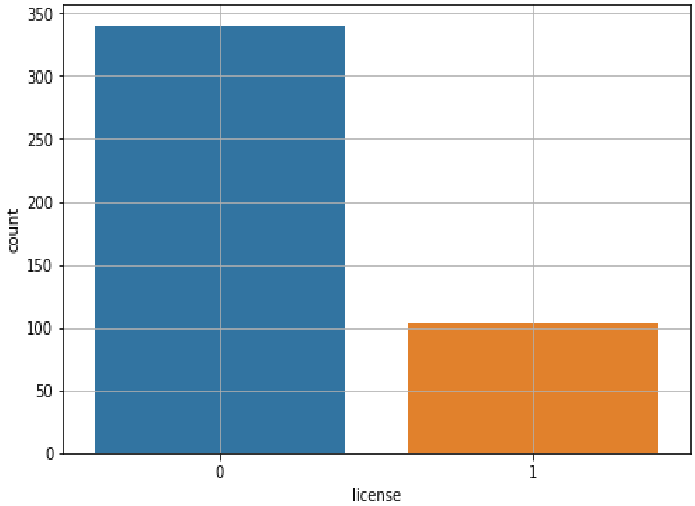




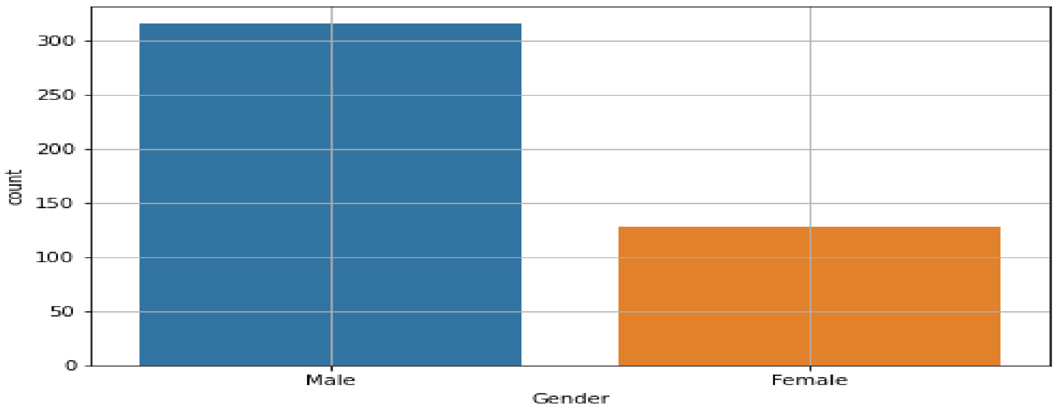


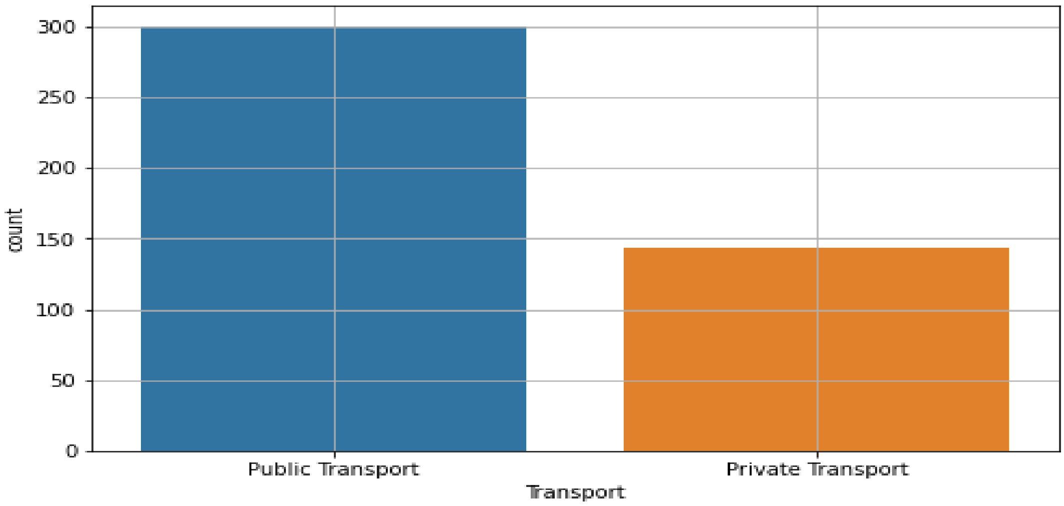




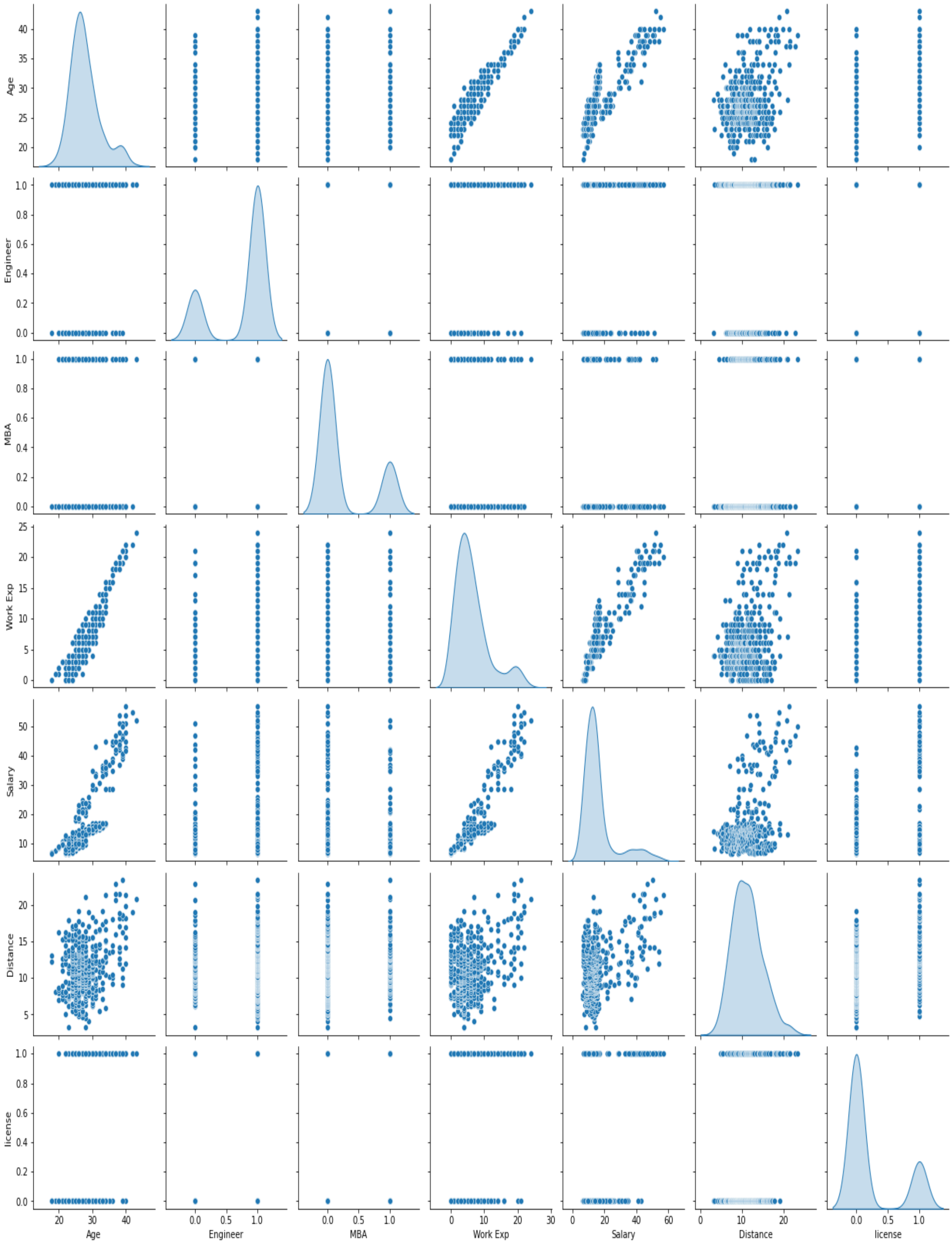


Checking the spread of the data using count plot for the categorical variables.





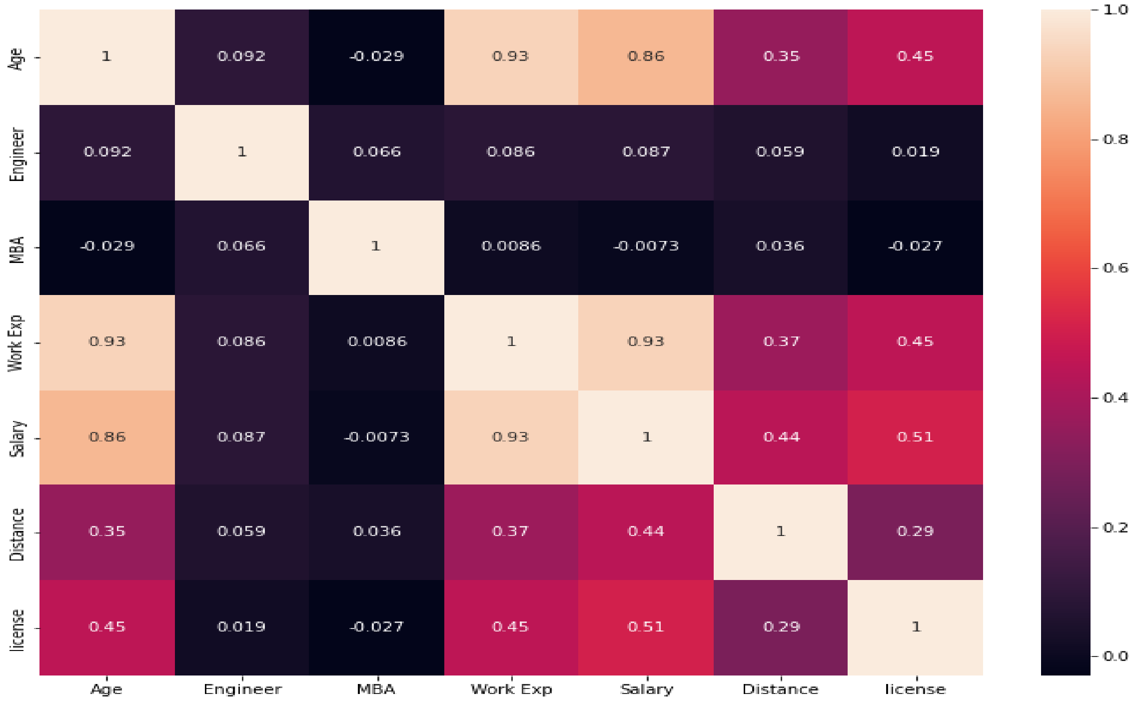
Bivariate Analysis



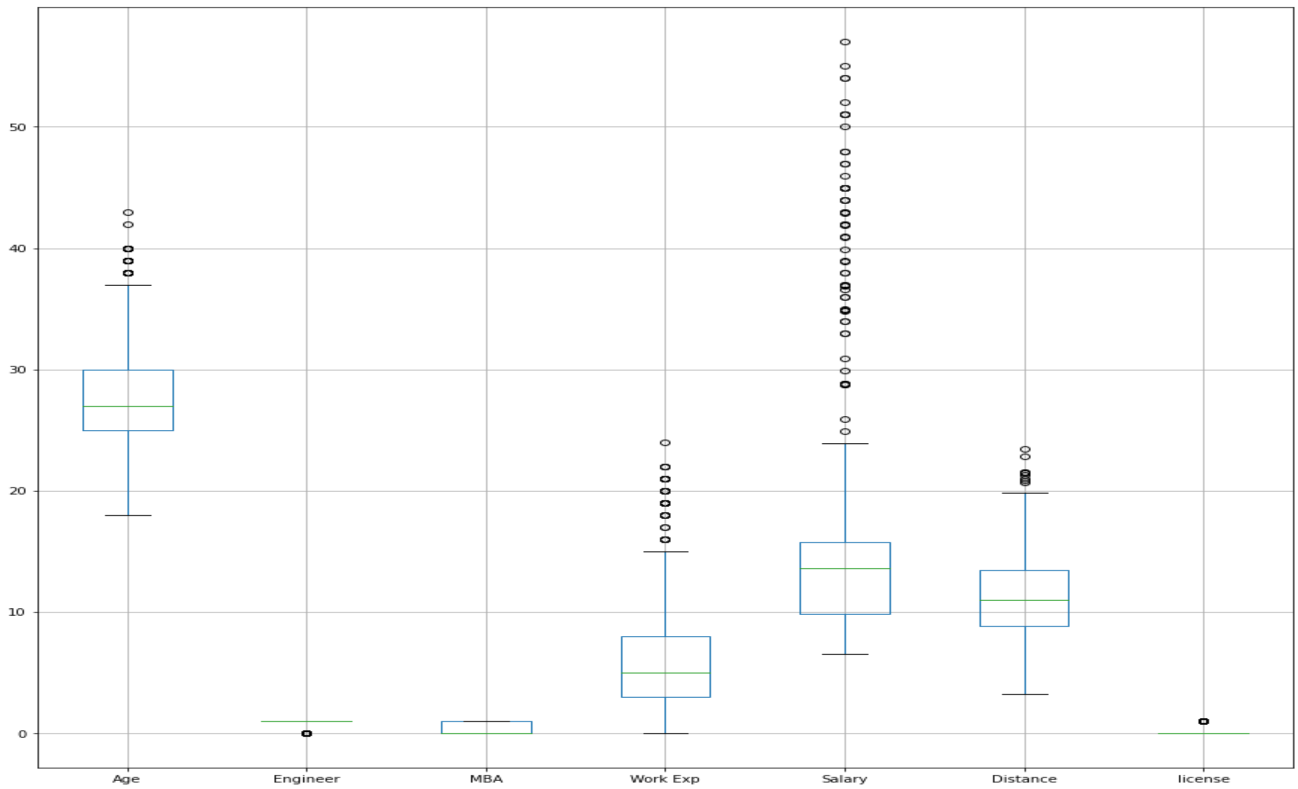
Checking for Correlation

	Age	Engineer	MBA	Work Exp	Salary	Distance	license
Age	1.000000	0.091935	-0.029090	0.932236	0.860673	0.352872	0.452311
Engineer	0.091935	1.000000	0.066218	0.085729	0.086762	0.059316	0.018924
MBA	-0.029090	0.066218	1.000000	0.008582	-0.007270	0.036427	-0.027358
Work Exp	0.932236	0.085729	0.008582	1.000000	0.931974	0.372735	0.452867
Salary	0.860673	0.086762	-0.007270	0.931974	1.000000	0.442359	0.508095
Distance	0.352872	0.059316	0.036427	0.372735	0.442359	1.000000	0.290084
license	0.452311	0.018924	-0.027358	0.452867	0.508095	0.290084	1.000000

Checking on Heat map



Checking for Outliers

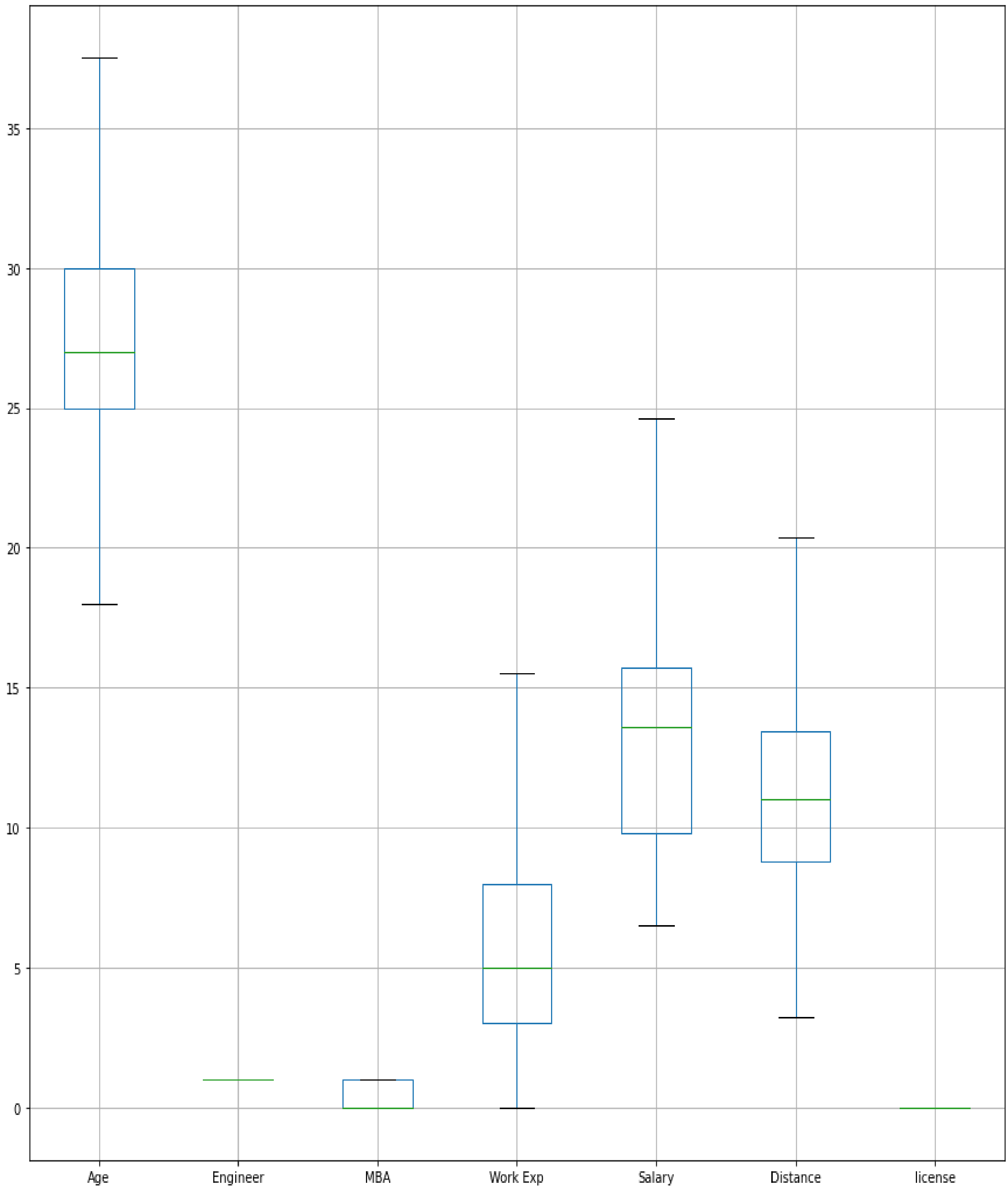


percentage of outliers present in the dataset

	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

Treating the outliers.



1.3) Encode the data (having string values) for Modelling (2 pts). Is Scaling necessary here or not? (2 pts), Data Split: Split the data into train and test (70:30) (2 pts).

Convert all objects to categorical codes

	Age	Gender	Engineer	MBA	Work Exp	Salary	Distance	license	Transport
0	28.0	1	1.0	0.0	4.0	14.3	3.2	0.0	1
1	23.0	0	1.0	0.0	4.0	8.3	3.3	0.0	1
2	29.0	1	1.0	0.0	7.0	13.4	4.1	0.0	1
3	28.0	0	1.0	1.0	5.0	13.4	4.5	0.0	1
4	27.0	1	1.0	0.0	4.0	13.4	4.6	0.0	1
5	26.0	1	1.0	0.0	4.0	12.3	4.8	0.0	1
6	28.0	1	1.0	0.0	5.0	14.4	5.1	0.0	0
7	26.0	0	1.0	0.0	3.0	10.5	5.1	0.0	1
8	22.0	1	1.0	0.0	1.0	7.5	5.1	0.0	1
9	27.0	1	1.0	0.0	4.0	13.5	5.2	0.0	1

Proportion of 1s and 0s

```
1    0.675676
0    0.324324
Name: Transport, dtype: float64
```

Checking the data info

```
<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    float64
 1   Gender     444 non-null    int64
 2   Engineer   444 non-null    float64
 3   MBA        444 non-null    float64
 4   Work Exp   444 non-null    float64
 5   Salary     444 non-null    float64
 6   Distance   444 non-null    float64
 7   license    444 non-null    float64
 8   Transport 444 non-null    int64
dtypes: float64(7), int64(2)
memory usage: 31.3 KB
```

Split data into training and test set

Split X and Y into 70 :30 ratio for training and test data.

Check the dimensions of the training and test data

X train (310, 8)

X test (134, 8)

Y train (310,)

Y test (134,)

Scaling

We need to do scaling before using distance-based models. Standard Scaling, or Min-Max scaling either one of these can used.

We are using Standard Scaler.

Head of training data frame

	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

Head of test data frame

	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

1.4) Apply Logistic Regression (4 pts). Interpret the inferences of both models (2 pts)

<u>Model 1 -</u> Building the model on the Training Data without scaled data**.**

Accuracy Score of Model 1: 0.7774193548387097

<u>Predicting the classes and the probabilities on the Training Data</u>

```
array([1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0,
       1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1,
       1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1,
       1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1,
       0, 1], dtype=int64)
```

<u>Model 2 -</u> Building the model on the Training Data with scaled data.

Accuracy Score of Model 2: 0.6741935483870968

<u>Predicting the classes and the probabilities on the Test Data</u>

```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1], dtype=int64)
```

1.5) Apply KNN Model (4 pts). Interpret the inferences of each model (2 pts)

Model 1

Train Accuracy is: 0.5741935483870968

Test Accuracy is: 0.5522388059701493

Train ROC-AUC score is: 0.4534657320872274

Test ROC-AUC score is: 0.44137596899224807

Confusion matrix for train set:

```
 [[ 18  78]
 [ 54 160]]
```

Confusion matrix for test set:

```
 [[ 9 39]
 [21 65]]
```

Classification report Train set

```
Classification report Train set:
              precision    recall  f1-score   support

           0       0.25      0.19      0.21        96
           1       0.67      0.75      0.71       214

    accuracy                           0.57       310
   macro avg       0.46      0.47      0.46       310
weighted avg       0.54      0.57      0.56       310
```

Classification report Test set

```
Classification report Test set:
              precision    recall  f1-score   support

           0       0.30      0.19      0.23        48
           1       0.62      0.76      0.68        86

    accuracy                           0.55       134
   macro avg       0.46      0.47      0.46       134
weighted avg       0.51      0.55      0.52       134
```

Model 2

Train Accuracy is: 0.7612903225806451

Test Accuracy is: 0.5970149253731343

Train ROC-AUC score is: 0.7773559190031153

Test ROC-AUC score is: 0.4574854651162791

Confusion matrix for train set:

```
[[ 45  51]
 [ 23 191]]
```

Confusion matrix for test set:

```
[[ 8 40]
 [14 72]]
```

Classification report Train set

Classification report Train set:

	precision	recall	f1-score	support
0	0.66	0.47	0.55	96
1	0.79	0.89	0.84	214
accuracy			0.76	310
macro avg	0.73	0.68	0.69	310
weighted avg	0.75	0.76	0.75	310

Classification report Test set

Classification report Test set:

	precision	recall	f1-score	support
0	0.36	0.17	0.23	48
1	0.64	0.84	0.73	86
accuracy			0.60	134
macro avg	0.50	0.50	0.48	134
weighted avg	0.54	0.60	0.55	134

1.6) Bagging (4 pts) and Boosting (4 pts), Model Tuning (4 pts).

Bagging

Model 1

Performance Matrix on train data set

```
0.9967741935483871
[[ 95   1]
 [  0 214]]
              precision    recall  f1-score   support

           0       1.00      0.99      0.99        96
           1       1.00      1.00      1.00       214

    accuracy                           1.00       310
   macro avg       1.00      0.99      1.00       310
weighted avg       1.00      1.00      1.00       310
```

Performance Matrix on test data set

```
0.5373134328358209
[[ 7 41]
 [21 65]]
              precision    recall  f1-score   support

           0       0.25      0.15      0.18        48
           1       0.61      0.76      0.68        86

    accuracy                           0.54       134
   macro avg       0.43      0.45      0.43       134
weighted avg       0.48      0.54      0.50       134
```

Model 2

Performance Matrix on train data set

```
0.9967741935483871
[[ 95   1]
 [  0 214]]
              precision    recall  f1-score   support

           0       1.00      0.99      0.99        96
           1       1.00      1.00      1.00       214

    accuracy                           1.00       310
   macro avg       1.00      0.99      1.00       310
weighted avg       1.00      1.00      1.00       310
```

Performance Matrix on test data set

```
0.5447761194029851
[[ 7 41]
 [20 66]]
              precision    recall  f1-score   support

           0       0.26      0.15      0.19        48
           1       0.62      0.77      0.68        86

    accuracy                           0.54       134
   macro avg       0.44      0.46      0.44       134
weighted avg       0.49      0.54      0.51       134
```

Boosting

Model 1

Model Score with ADA Boosting algorithms is **0.8838709677419355**

```
[[ 66 30]
 [ 6 208]]
              precision    recall  f1-score   support

           0       0.92      0.69      0.79        96
           1       0.87      0.97      0.92       214

    accuracy                           0.88       310
   macro avg       0.90      0.83      0.85       310
weighted avg       0.89      0.88      0.88       310
```

Model 2

Model Score with ADA Boosting algorithms is **0.7580645161290323**

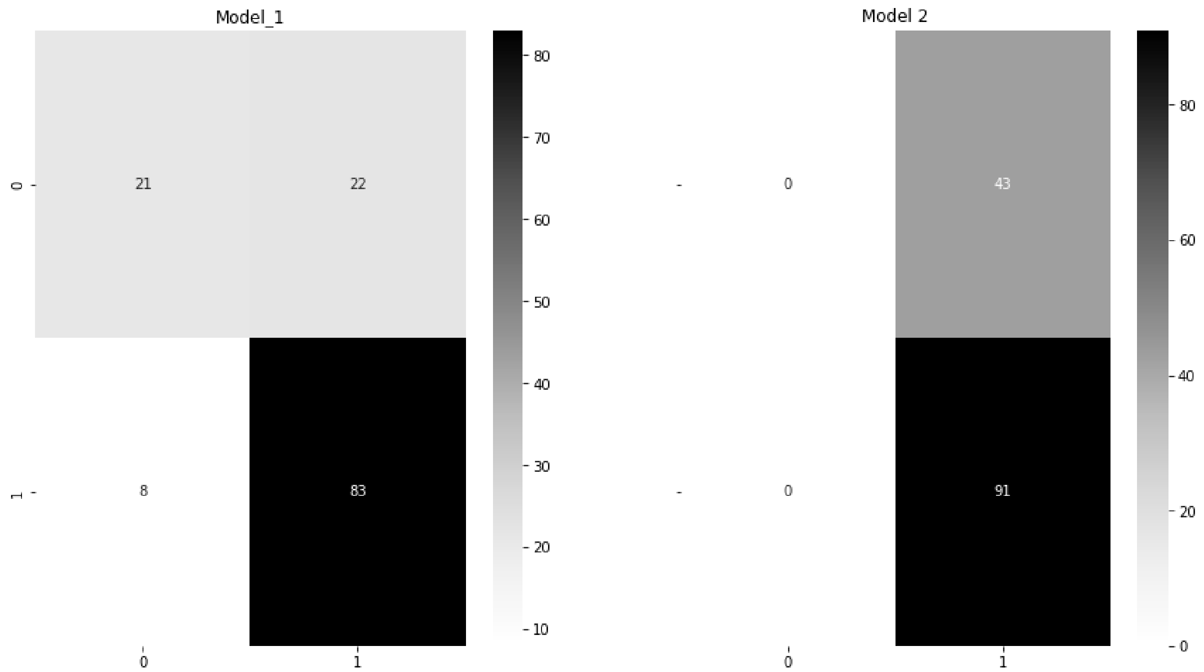
```
[[ 95   1]
 [ 0 214]]
              precision    recall  f1-score   support

           0       1.00      0.99      0.99        96
           1       1.00      1.00      1.00       214

    accuracy                           1.00       310
   macro avg       1.00      0.99      1.00       310
weighted avg       1.00      1.00      1.00       310
```

1.7 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 (5 pts) Final Model - Compare all models on the basis of the performance metrics in a structured tabular manner. Describe on which model is best/optimized (3 pts)

Confusion Matrix



Model 1

True Negative: 21
False Positives: 22
False Negatives: 8
True Positives: 83

Model 2

True Negative: 0
False Positives: 43
False Negatives: 0
True Positives: 91

Model 1

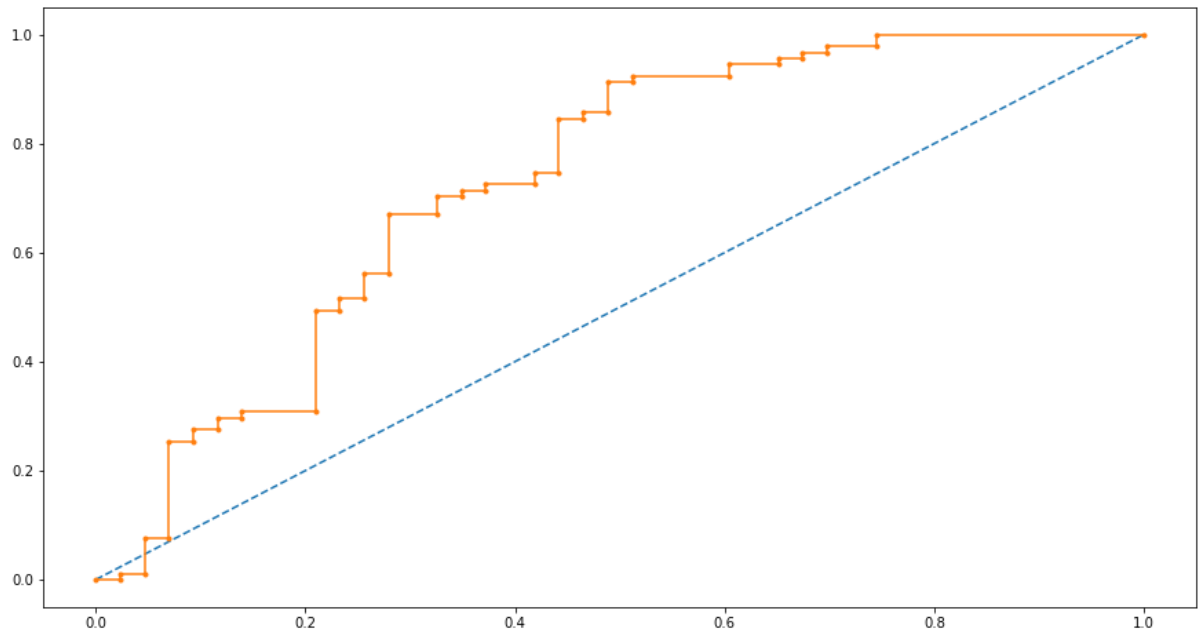
	precision	recall	f1-score	support
0	0.72	0.49	0.58	43
1	0.79	0.91	0.85	91
accuracy			0.78	134
macro avg	0.76	0.70	0.72	134
weighted avg	0.77	0.78	0.76	134

Model 2

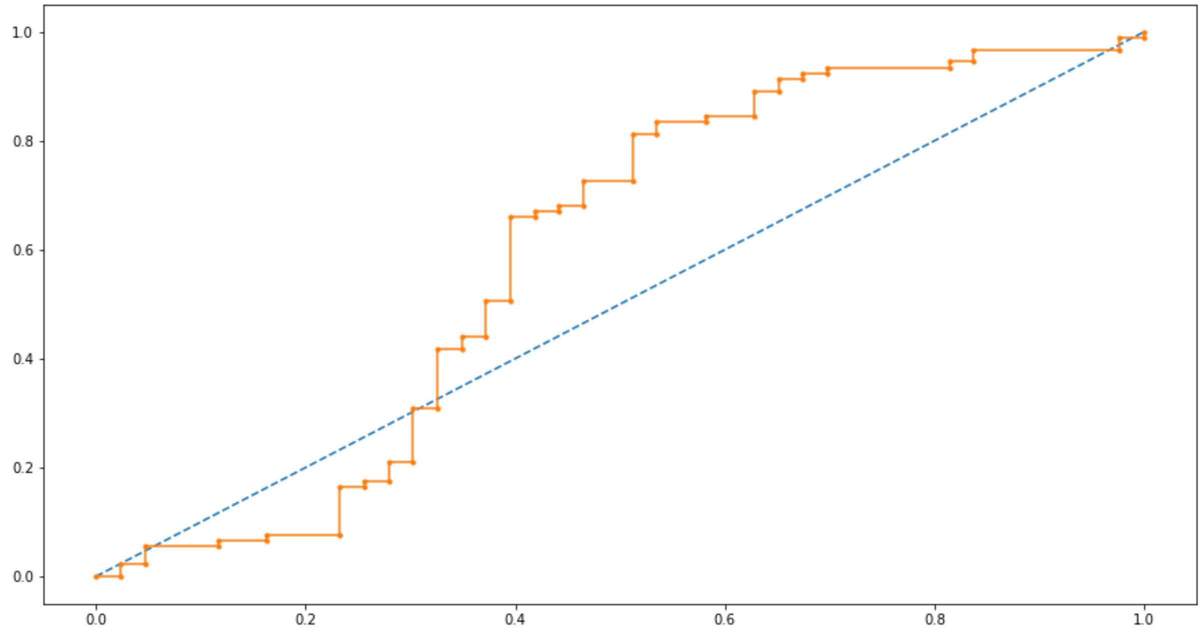
	precision	recall	f1-score	support
0	0.00	0.00	0.00	43
1	0.68	1.00	0.81	91
accuracy			0.68	134
macro avg	0.34	0.50	0.40	134
weighted avg	0.46	0.68	0.55	134

Check the summary statistics of the AUC-ROC curve for the two Models built. This is for the test data.

Model 1 AUC : 0.73115



<u>Model 2 AUC</u>: 0.59392



1.8) Based on your analysis and working on the business problem, detail out appropriate insights and recommendations to help the management solve the business objective.

- After doing EDA on data frame, it shows that:
    - ➢ There are total of 9 variables: float64(2) int64(5) object (2).
    - ➢ shape of the data frame is Number of rows: 444 & Number of columns: 9.
    - ➢ There was no null values & there are no duplicated values in data frame.
    - ➢ We also did descriptive statistics to know the data.
    - ➢ To visualize the data, we did Univariate Analysis & Bivariate Analysis.
    - ➢ We remove the outliers from the data.
- We Convert all objects to categorical codes & splits the data into 70(Training) & 30 (Testing) data.

- we did scale to create a model to compare with original data frame.

- Through Logistic Regression we compare both models to find accurate score.

- And did KNN (to solve both classification and regression problems.) model on both models.

- Did Bagging to reduce variance within a training model.

- Did Boosting to reduce errors in predictive data analysis.

- Did Model Tuning to provides optimized values for hyperparameters, which maximize your model's predictive accuracy.

- And Performance Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model.