

MACHINE LEARNING PROJECT REPORT

ML



June 19, 2022

Brijal Panchal

Contents

[Problem: 2](#_Toc106563663)

[1.1. Read the dataset. Do the descriptive statistics and do null value condition check. 2](#_Toc106563664)

[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) 3](#_Toc106563665)

[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). 7](#_Toc106563666)

[1.4. Apply Logistic Regression (4 pts). Interpret the inferences of both models (2 pts) 8](#_Toc106563667)

[1.5. Apply KNN Model (4 pts). Interpret the inferences of each model (2 pts) 8](#_Toc106563668)

[1.6. Bagging ( 4 pts) and Boosting (4 pts), Model Tuning (4 pts) 9](#_Toc106563669)

[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) 11](#_Toc106563670)

[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 14](#_Toc106563671)

# Problem:

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 Discerption

|  |  |
| --- | --- |
| **Variable Name** | **Description** |
| Age | Age of an employee |
| Gender | Gender of an employee (male or female) |
| Engineer | 0 for non-engineer, 1 for engineer |
| MBA | 0 for non-MBA, 1 for MBA |
| Work Exp | Work experience of an employee |
| Salary | Salary of an employee |
| Distance | Distance travelled by an employee |
| license | 0 for not having licence, 1 for having licence |
| Transport | Mode of transport, Public Transport and Private Transport |

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

Sample of the dataset:



* There are 444 samples available in the dataset having column Age, Gender, Engineer, MBA, Work Exp, Salary, Distance, license and transport.
* There are no duplicate value present in the dataset
* There are no null value present in the dataset

## 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)

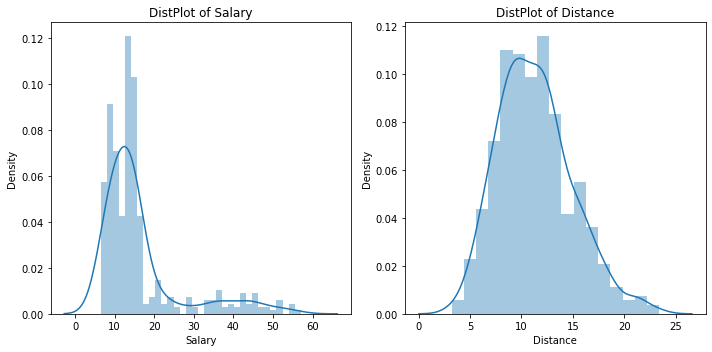
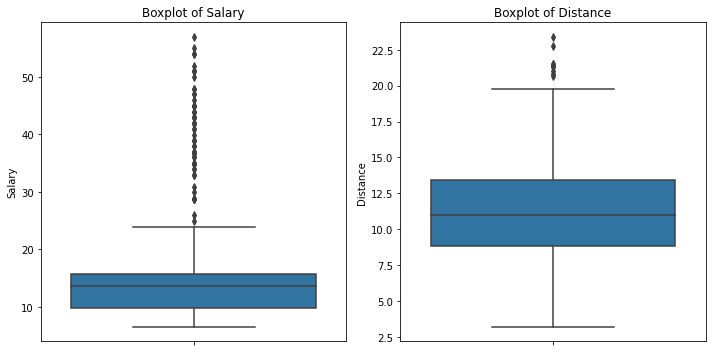
Summary of description



* Minimum age is 18 year while maximum is 43 year. Average age is 27 year.
* Minimum work experience is 0 year while maximum is 24 year with average of 6 year.
* Average salary is 16.23 with range of 6.5 to 57.
* Average distance is 11.32 with range of 3.2 to 23.4.
* Median Price is significantly higher than the mean, the Standard deviation is quite high Interpretation Univariate Analysis The purpose of Univariate Analysis is to find out which variables have clear separation for the target variable in terms of skewness

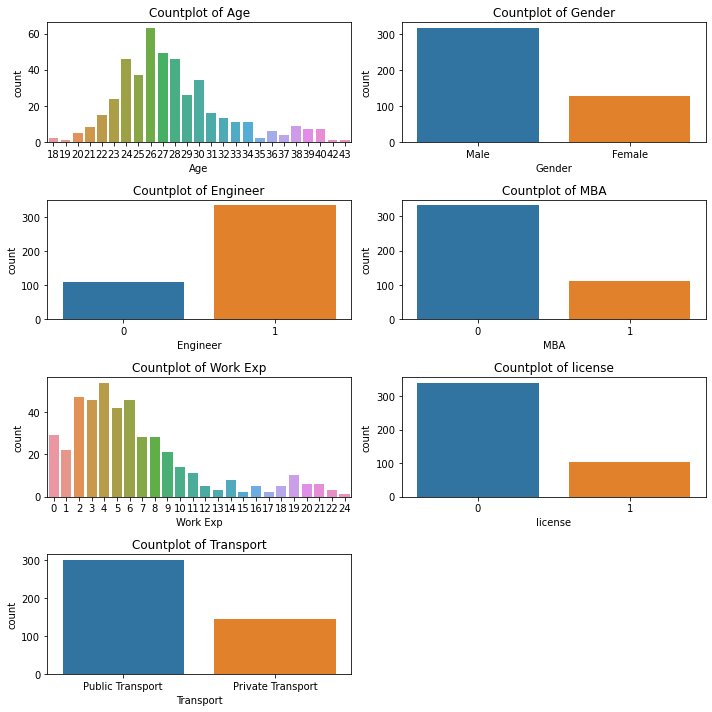
Univariate Analysis:

* Histogram and Box-plot of continuous variables



From above graph, we see that both continuous variables have outliers in their data in which Salary and Distance have 13.29% and 2.03% outliers present in respectively. Salary is right skewed data while Distance is normally distributed samples.

* Count plot for categorical variables



Maximum number of employees are of age 26 years while early age and older are having less numbers

More male employee are there as compare to woman

Most of the employees are engineers

Very low number of employees are having MBA and licenses

Very few number of employees are having higher experience

Most of the employee prefers public transport as compare to private transport

Multivariate analysis:

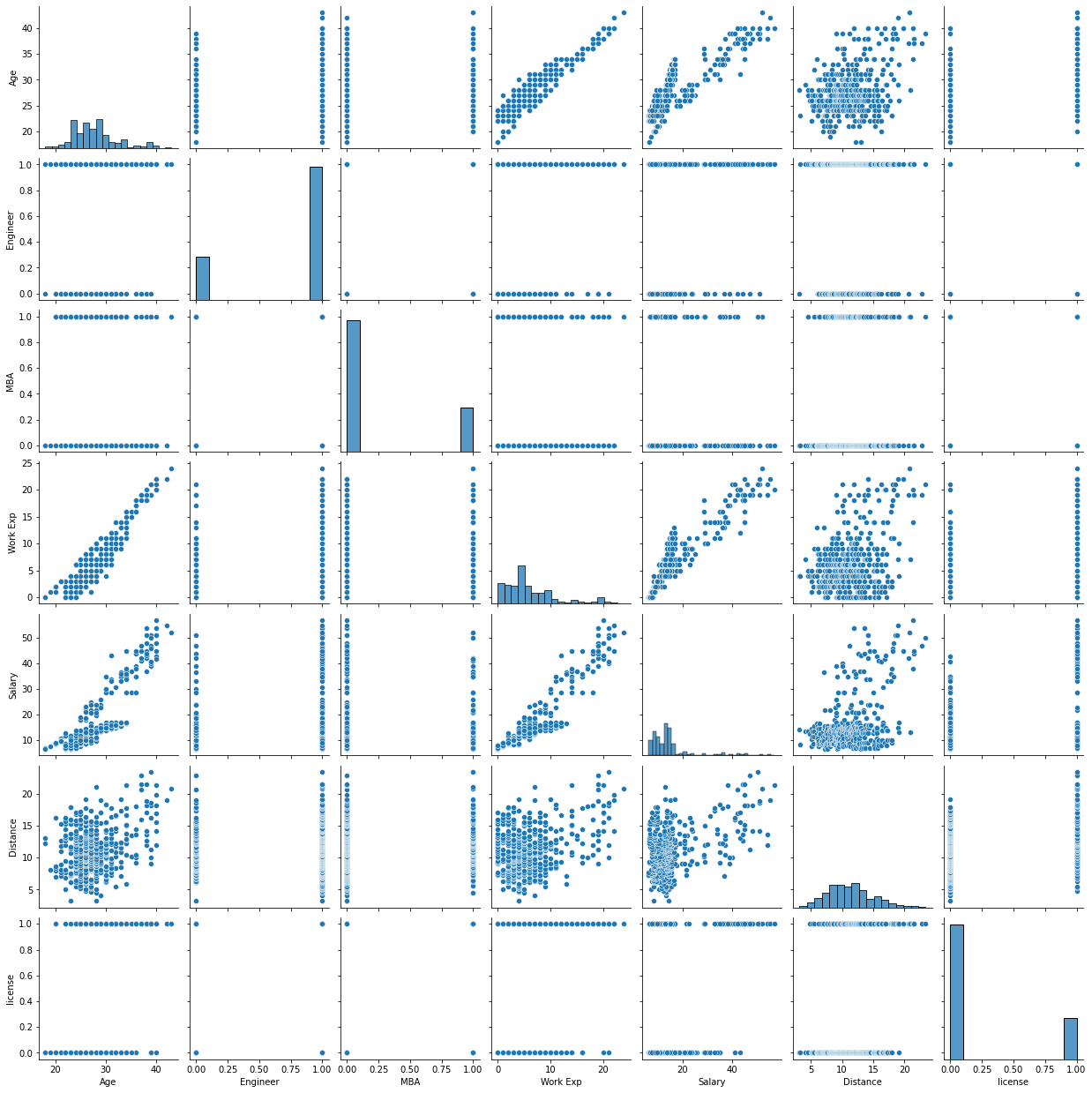
* Heat map



There are mostly positive correlations between variables, and very few negative correlations.

Overall, the magnitude of correlations between the variables are very less except Work Exp, Salary and Distance.

* Pair plot



From pair plot, we can see that there is positive relation present Work Exp, Salary and Distance predictors

## 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).

Gender and Transport are in object data type. Converting this data type into integer by applying label encoding.

After applying label encoding in Gender category 0 represents as male and 1 represents as female as well as in Transport category 0 represents as Public Transport and 1 represents as Private Transport

In the data, Transport is dependent variable whereas rest of the column is independent variable

After splitting the data into (70:30) we can see 310 samples in training while 134 samples in testing

We need to do scaling before using distance-based models (ex. Logistic regression, KNN) as unit of columns are different and it may influence weight to another column for model building. Standard Scaling or Min-Max scaling one of these can used. Here I am applying standard scaling.

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

Applying Logistic Regression on train samples with default parameters with random state as 100, C as 1.0, solver as lbfgs and penalty as l2

* + Accuracy of default model after fit the train data for building model

Train Accuracy: 80.32%

Test Accuracy: 81.34%

Tuning the model using GridSearchCV using bellow

* + tuning parameter is

{ 'penalty' : ['l1','l2','elasticnet'],

'C': np.logspace(-3,3,7),

'solver': ['newton-cg', 'lbfgs', 'liblinear'] }

* + Best parameter found after applying tuning to the model is

{'C': 0.1,

'penalty': 'l2',

'solver': 'newton-cg'}

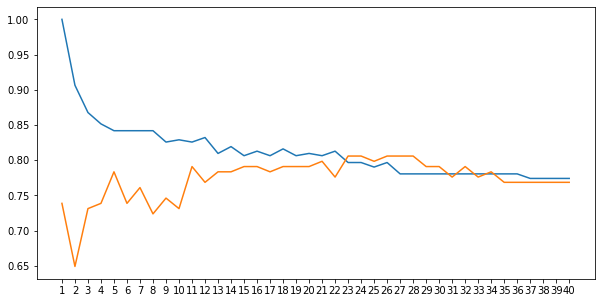
* + Accuracy of Tuned model

Train Accuracy: 82.25%

Test Accuracy: 82.83%

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

Applying different nearest neighbour count value between ranges 1 to 41 we found bellow accuracy table between train and test data and found best K value as 26 having less error



Applying K value as 26 into KNeighborsClassifier model with default other parameter as leaf\_size as 30 and p as 2.

* + Accuracy of default model after fit the train data for building model

Train Accuracy: 79.67%

Test Accuracy: 80.59%

Tuning the model using GridSearchCV using bellow

* + tuning parameter is

{'n\_neighbors': [7,15,17,19,21,31],

'leaf\_size': [10, 30, 40,50],

'p': [1,2,3,4]}

* + Best parameter found after applying tuning to the model is

{'leaf\_size': 10,

'n\_neighbors': 15,

'p': 3}

* + Accuracy of Tuned model

Train Accuracy: 81.93%

Test Accuracy: 79.10%

## Bagging ( 4 pts) and Boosting (4 pts), Model Tuning (4 pts)

* Boosting Classifier:

Applying GradientBoostingClassifier on train samples with default parameters with random state as 0, learning\_rate as 0.1, max\_depth as 3, min\_samples\_split as 2 and n\_estimators as 1

* + Accuracy of default model after fit the train data for building model which shows overfitting model

Train Accuracy: 96.77%

Test Accuracy: 76.86%

Tuning the model using GridSearchCV using bellow

* + tuning parameter is

{ 'learning\_rate': [0.05,0.075, 0.095],

'max\_depth':[6,7,8],

'min\_samples\_split': [10,11,12],

'n\_estimators':[30,50,100],

'random\_state': [100]}

* + Best parameter found after applying tuning to the model is

{'learning\_rate': 0.05,

'max\_depth': 6,

'min\_samples\_split': 10,

'n\_estimators': 30,

'random\_state': 100}

* + Accuracy of Tuned model which is over fit model

Train Accuracy: 93.54%

Test Accuracy: 84.32%

* Bagging Classifier:

Applying BaggingClassifieron train samples with default parameters with random state as 100, base\_estimator as none, max\_features as 1.0, max\_samples as 1.0 and n\_estimators as 10

* + Accuracy of default model after fit the train data for building model which is over fitting model

Train Accuracy: 98.38%

Test Accuracy: 79.10%

Tuning the model using GridSearchCV using bellow

* + tuning parameter is

{'base\_estimator':[LogisticRegression(),DecisionTreeClassifier()],

'max\_samples':[0.3,0.5,0.6,0.7,0.8],

'max\_features':[0.3,0.4,0.5,0.6],

'n\_estimators' :[30,40,50,100,150],

'random\_state': [100]}

* + Best parameter found after applying tuning to the model is

{ 'base\_estimator': DecisionTreeClassifier(),

'max\_features': 0.5,

'max\_samples': 0.6,

'n\_estimators': 50,

'random\_state': 100}

* + Accuracy of Tuned model which is over fit model

Train Accuracy: 96.12%

Test Accuracy: 78.35%

## 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)

* Train and test accuracy of all model as well as test AUC score:



* Confusion matrix off all model for train samples

Logistic Regression Model

precision recall f1-score support

0 0.76 0.60 0.67 42

1 0.83 0.91 0.87 92

accuracy 0.81 134

macro avg 0.79 0.75 0.77 134

weighted avg 0.81 0.81 0.81 134

Tuned Logistic Regression Model

precision recall f1-score support

0 0.81 0.60 0.68 42

1 0.83 0.93 0.88 92

accuracy 0.83 134

macro avg 0.82 0.77 0.78 134

weighted avg 0.83 0.83 0.82 134

KNN Model

precision recall f1-score support

0 0.94 0.40 0.57 42

1 0.78 0.99 0.87 92

accuracy 0.81 134

macro avg 0.86 0.70 0.72 134

weighted avg 0.83 0.81 0.78 134

Tuned KNN Model

precision recall f1-score support

0 0.75 0.50 0.60 42

1 0.80 0.92 0.86 92

accuracy 0.79 134

macro avg 0.78 0.71 0.73 134

weighted avg 0.79 0.79 0.78 134

Bagging Model

precision recall f1-score support

0 0.65 0.71 0.68 42

1 0.86 0.83 0.84 92

accuracy 0.79 134

macro avg 0.76 0.77 0.76 134

weighted avg 0.80 0.79 0.79 134

Tuned Bagging Model

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

Boosting Model

precision recall f1-score support

0 0.64 0.60 0.62 42

1 0.82 0.85 0.83 92

accuracy 0.77 134

macro avg 0.73 0.72 0.73 134

weighted avg 0.76 0.77 0.77 134

Tuned Boosting Model

precision recall f1-score support

0 0.80 0.67 0.73 42

1 0.86 0.92 0.89 92

accuracy 0.84 134

macro avg 0.83 0.80 0.81 134

weighted avg 0.84 0.84 0.84 134

* Confusion matrix off all model for test samples

Logistic Regression Model

precision recall f1-score support

0 0.77 0.57 0.66 102

1 0.81 0.92 0.86 208

accuracy 0.80 310

macro avg 0.79 0.74 0.76 310

weighted avg 0.80 0.80 0.79 310

Tuned Logistic Regression Model

precision recall f1-score support

0 0.86 0.55 0.67 102

1 0.81 0.96 0.88 208

accuracy 0.82 310

macro avg 0.84 0.75 0.77 310

weighted avg 0.83 0.82 0.81 310

KNN Model

precision recall f1-score support

0 0.91 0.42 0.58 102

1 0.78 0.98 0.87 208

accuracy 0.80 310

macro avg 0.85 0.70 0.72 310

weighted avg 0.82 0.80 0.77 310

Tuned KNN Model

precision recall f1-score support

0 0.88 0.52 0.65 102

1 0.80 0.97 0.88 208

accuracy 0.82 310

macro avg 0.84 0.74 0.77 310

weighted avg 0.83 0.82 0.80 310

Bagging Model

precision recall f1-score support

0 0.98 0.97 0.98 102

1 0.99 0.99 0.99 208

accuracy 0.98 310

macro avg 0.98 0.98 0.98 310

weighted avg 0.98 0.98 0.98 310

Tuned Bagging Model

precision recall f1-score support

0 1.00 0.88 0.94 102

1 0.95 1.00 0.97 208

accuracy 0.96 310

macro avg 0.97 0.94 0.95 310

weighted avg 0.96 0.96 0.96 310

Boosting Model

precision recall f1-score support

0 0.99 0.91 0.95 102

1 0.96 1.00 0.98 208

accuracy 0.97 310

macro avg 0.97 0.95 0.96 310

weighted avg 0.97 0.97 0.97 310

Tuned Boosting Model

precision recall f1-score support

0 1.00 0.80 0.89 102

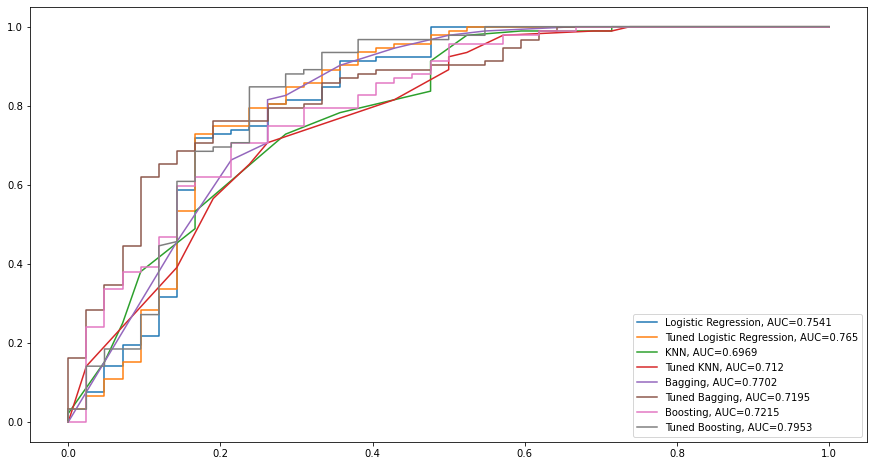
1 0.91 1.00 0.95 208

accuracy 0.94 310

macro avg 0.96 0.90 0.92 310

weighted avg 0.94 0.94 0.93 310

* ROC curve for all models with AUC score



By seeing accuracy score, ROC\_AUC score and ROC curve we can find that tuned boosting model would be the best model to move forward.

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

Very less samples are available for training the model and testing the model, which are not enough.

No description of data available for the given datasheet.

It is an under sampling model because around 144 samples are available for Public transport while 300 samples are available for Private transport.

There are many outliers present in the data of Salary and Distance column which need to be clarify is the data are provide are valid or not for those rows.

After going through different model, the highest accuracy we can achieve through tuned boosting model of about 94% training accuracy and 84% testing accuracy and 0.7953 ROC\_AUC score.

However, there are many errors present in the data by looking precession and recall values, which can be reduced by gathering more number of samples.