Project – Machine Learning

Exploratory Data Analysis

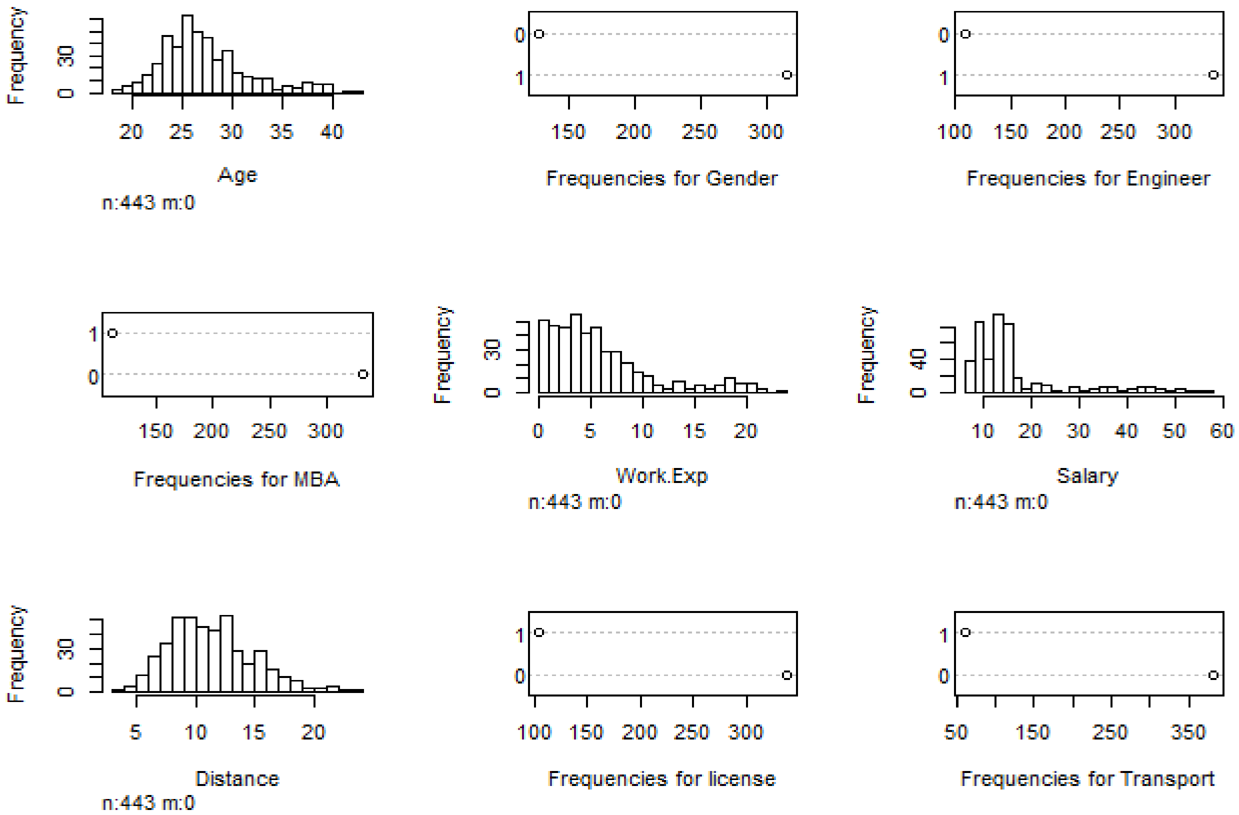
EDA was performed on the dataset after it being imported in R environment. We discovered the following in the Dataset.

- We have 443 observations and 9 variables (8 Predictors + 1 Target)
- MBA has 1 row as NA
- The data set provided is highly imbalanced.
- High correlation is observed among the predictor variables.

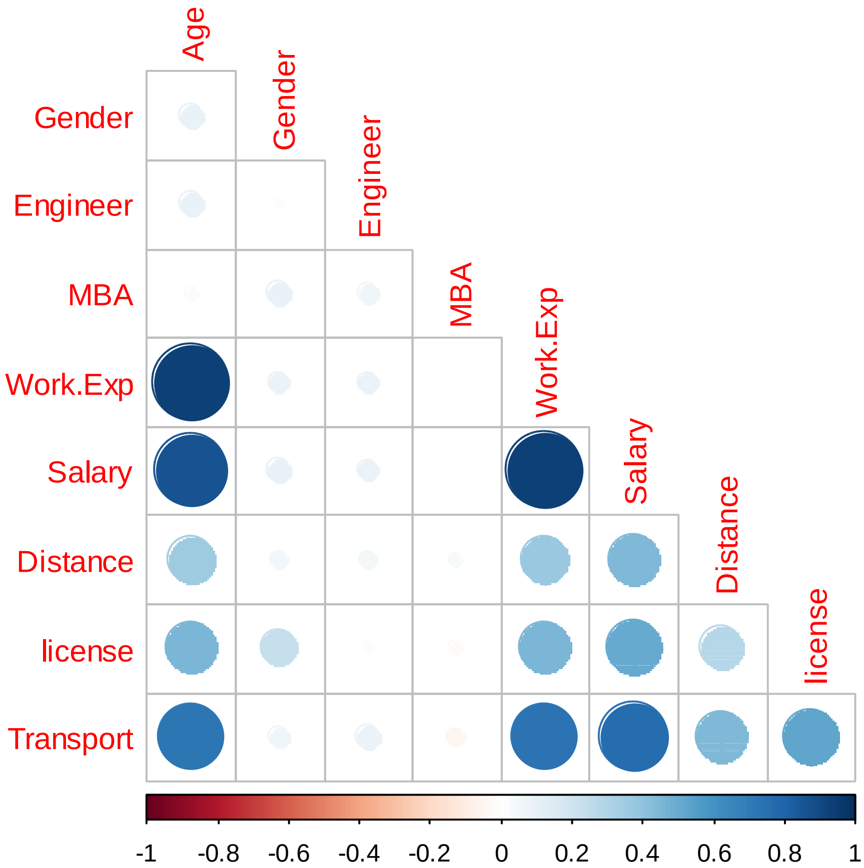
The following remediation were taken due to the above finding:

- The 1 NA rows in MBA is removed.
- The Target variable (Transport) which has 3 labels – Public Transport, 2 Wheeler & Car was converted to a binary variable – 1 if Transport is Car & 0 otherwise
- For Gender variable, we have assumed, if Gender is Male then 1 & 0 otherwise
- Engineer, MBA & License were converted to factor variables from numeric.

After all the operations were performed above, We have the frequency plot of the variables.



We have also found out the correlation between the Variables of the dataset.



Insights from the above plot:

- We can see correlation between Age, distance & Salary and also between Distance and License.
- The above correlation are quite natural given Salary increases with Age and so does Work Experience.
- Also, if the employee lives far away from the office, he/she is more likely to get a license for himself/ herself.

Exploratory Data Analysis:

Target Variable – Transport

Transport is a factor Variable with 443 rows and 2 distinct labels.

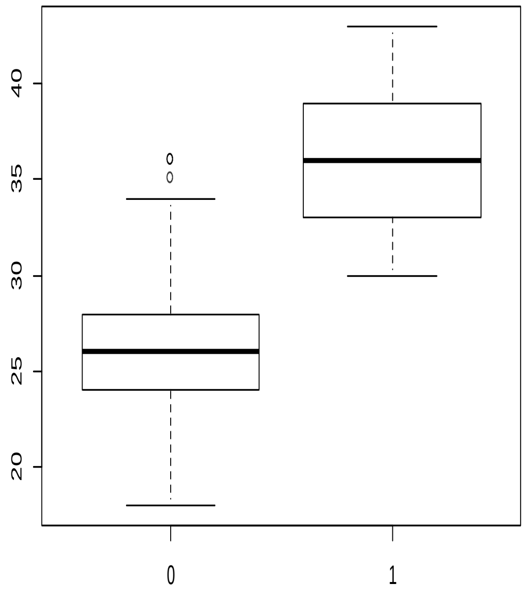
```
Transport
       n  missing distinct
     443        0        2

Value         0    1
Frequency   382   61
Proportion 0.86 0.14
```

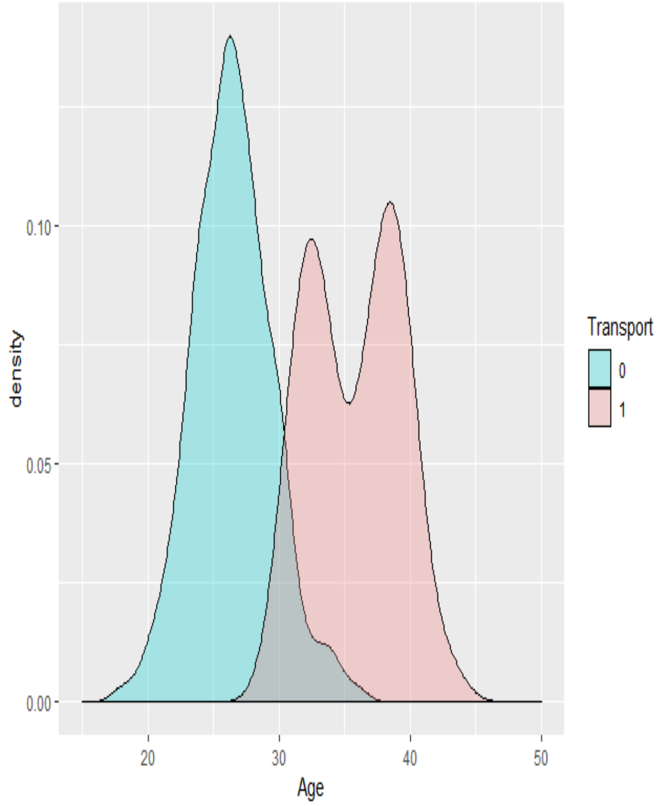
As said, the is highly imbalanced with label 1 accounting only for 14% of the entire dataset.

Response Variables – Continuous

AGE

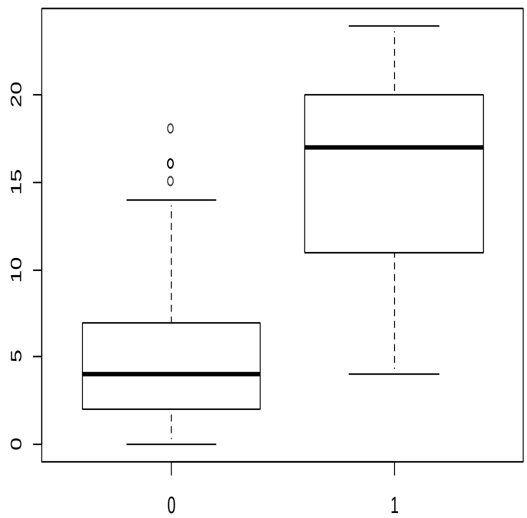
Age is ranging between 18 and 43 years with a mean of 28 and median of 27

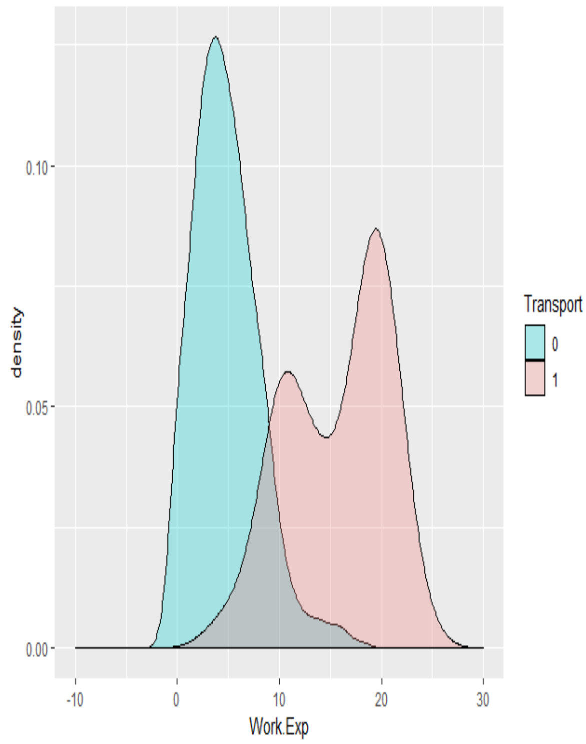




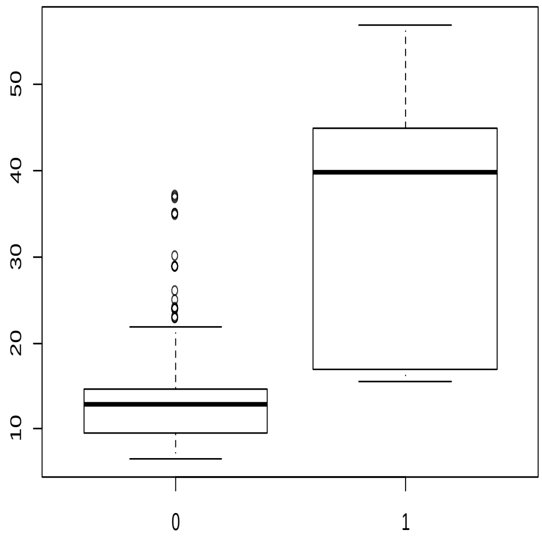
Work Experience

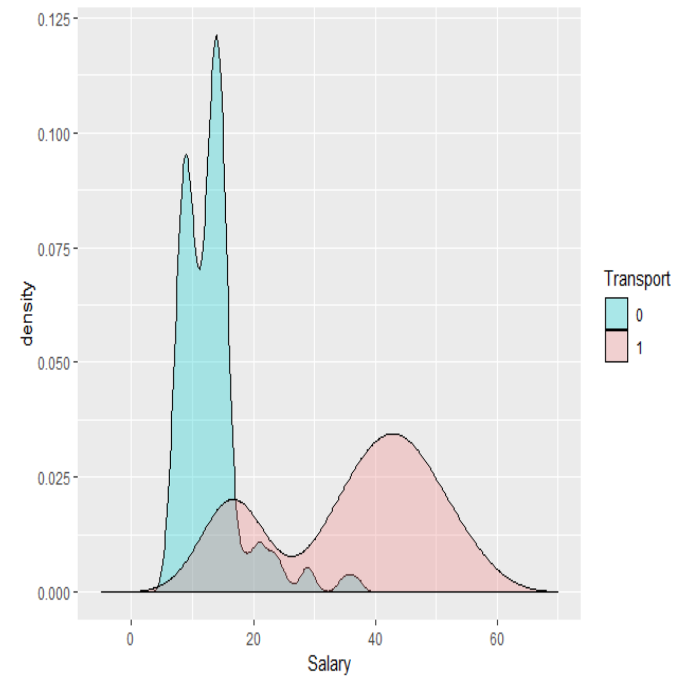
Work Experience ranges between 0 and 24 years with mean of 6.3 years.



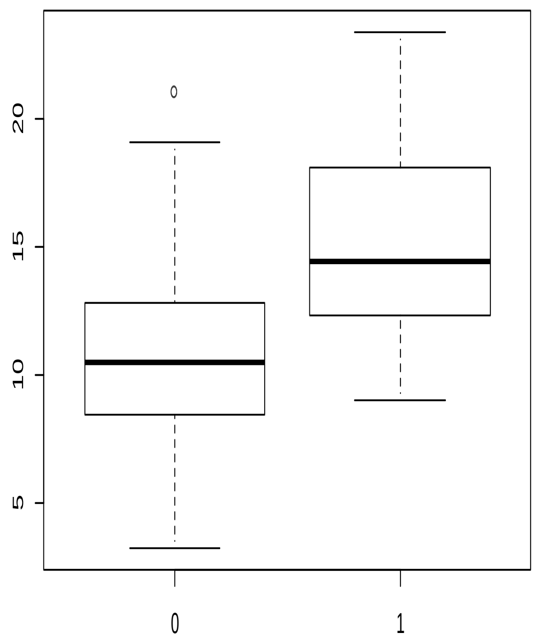


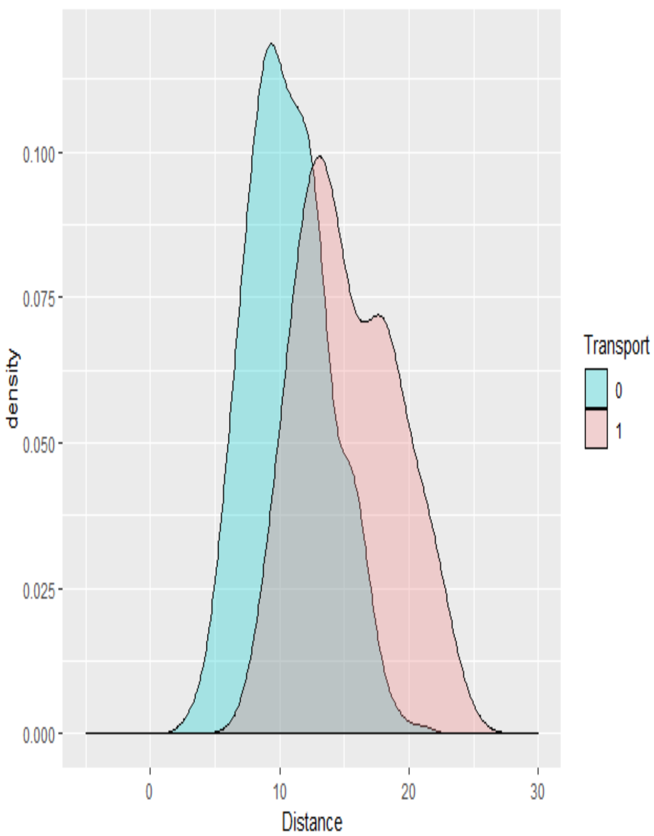
Salary
Salary ranges between 6 and 57 with a mean of 16 and median 14.





Distance
Distance ranges between 3.2 and 23.4 KM with a mean of 11.3KM and median 11 KM.



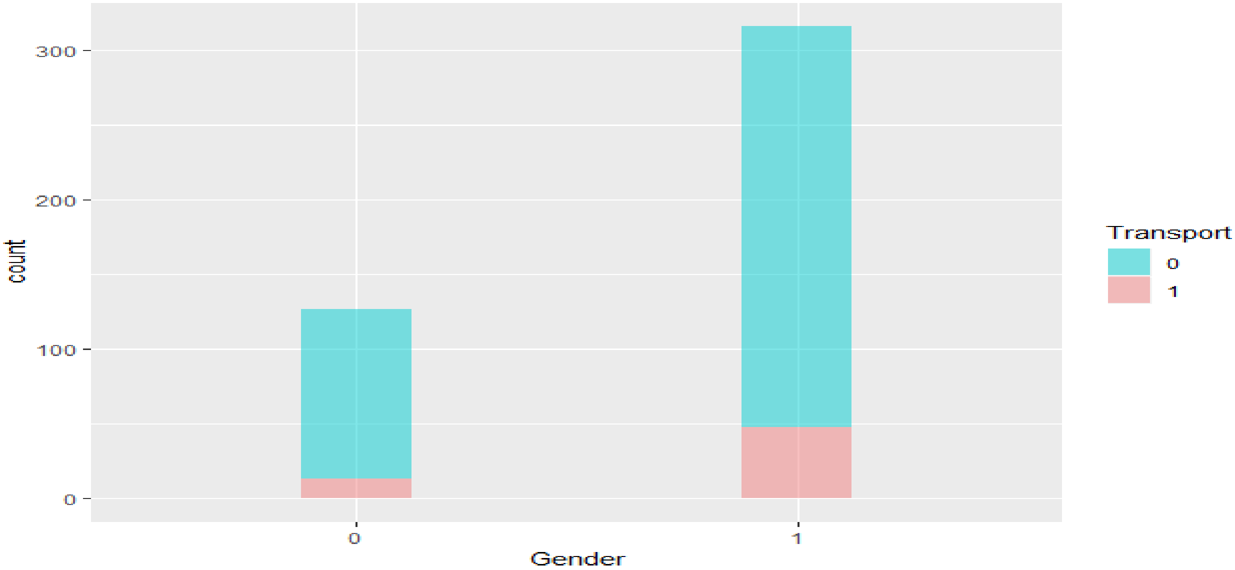


Response Variable – Categorical

<u>Gender</u>

The Gender variable is a dichotomous which determines whether the Employee is a Male or a Female. Frequency of 0 and 1 is shown below:

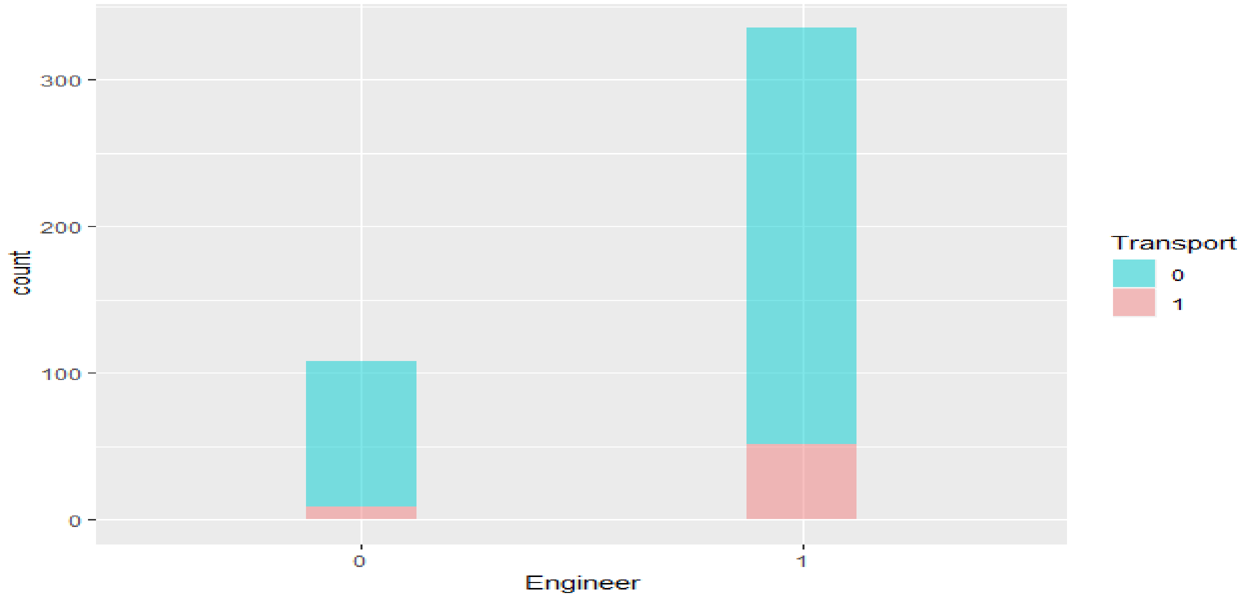
0 : 127
1 : 316



<u>Engineer</u>

It is a dichotomous variable showing whether the Employee is an Engineer or not. Frequency is shown below for 0 & 1:

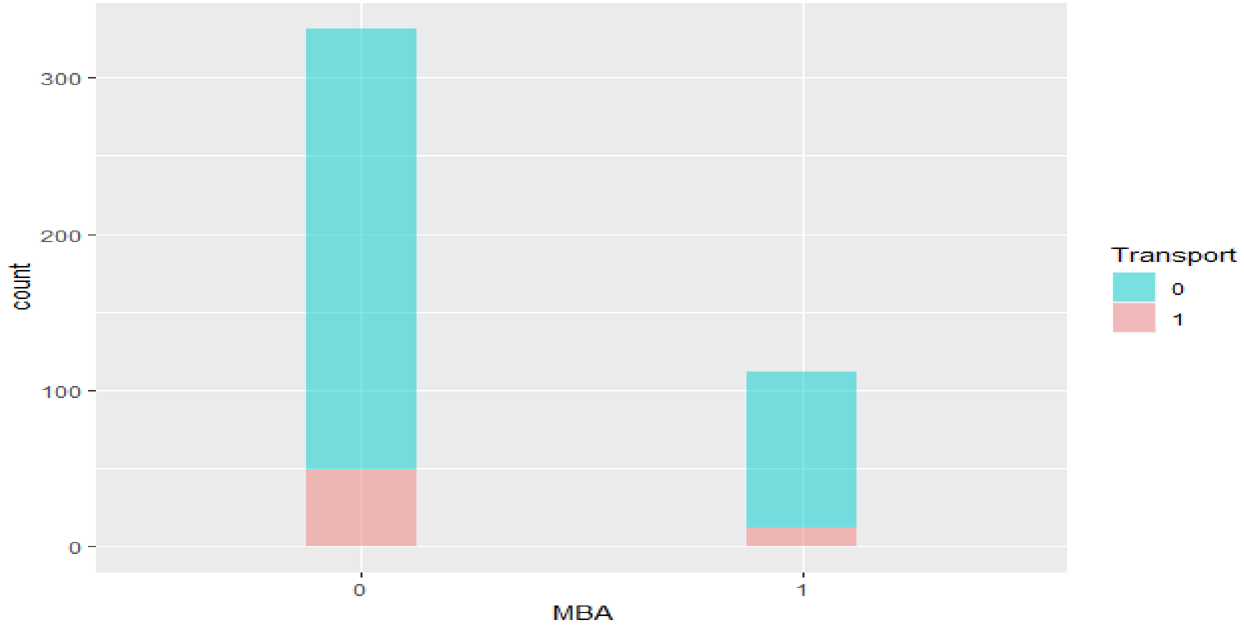
0 : 108
1 : 335



MBA

MBA variable shows whether an Employee has a MBA degree or not.

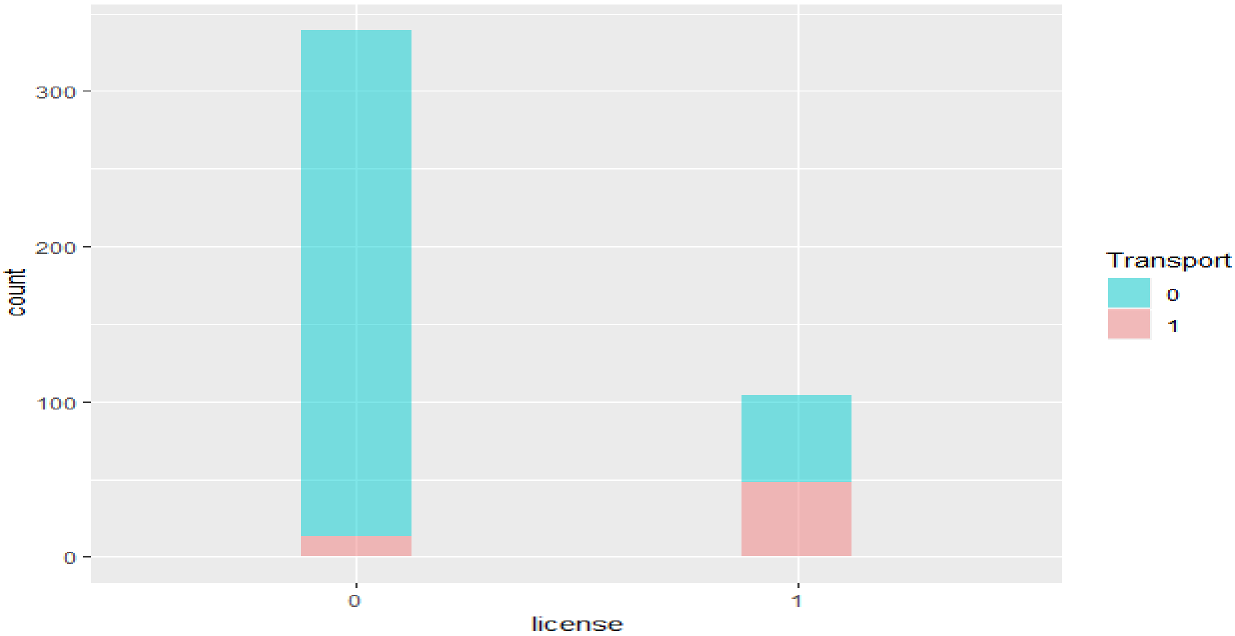
0 : 331
1 : 112



License

License variable shows whether an employee holds a Driving License or not.
Frequency is shown below for 0 & 1:

0 : 339
1 : 104



Data Slicing

The data was sliced into train & test in 75:25 ratio. Since we have an unbalanced dataset, we will use SMOTE in the train dataset. On the other hand, we shall leave the test dataset as it is and shall use it for determining the Confusion Matrix of the models we are going to train forward.
The proportion table for the test &train dataset are as below:

```
> prop.table(table(data$Transport))

   0    1 
0.86 0.14 
> prop.table(table(train$Transport))

   0    1 
0.86 0.14 
> prop.table(table(test$Transport))

   0    1 
0.86 0.14 
```

DATA preparation using SMOTE

Since the dataset supplied is highly imbalanced, we have used SMOTE to synthetically oversample the minority class. As a result, minority class has increased from 14% to 34%.

```
library(DMwR)
balanced.data.train <- SMOTE(Transport ~., train, perc.over = 4700, k = 5, perc.under = 200)
prop.table(table(balanced.data.train$Transport))
```

The proportion table for the Transport variable after SMOTE operation:

```
> prop.table(table(balanced.data.train$Transport))

   0    1 
0.66 0.34 
```

Logistic Regression

Logistic regression is performed at first using Target at the response and all other variables as predictor.

```
logit_model1 = glm(Transport ~ ., data = balanced.data.train, family = binomial(link="logit"))
summary(logit_model1)
vif(logit_model1)
```

Except Engineer & MBA, all other variables were found to be significant.
AIC: 1169

```
> vif(logit_model1)
     Age   Gender Engineer      MBA Work.Exp   Salary Distance  license
     6.9      1.1      1.0      1.1     10.6      3.3      1.9      1.1
>
```

We can see VIF of Work.Exp is more than 10. We are going to treat these in the next logistic regression model we're going to train.

The next logistic regression we are going to train will have predictors Work.Exp , Engineer & MBA removed.

```
logit_model2 = glm(Transport ~ Age+Gender+Salary+Distance+license, data = balanced.data.train, family = binomial(link="logit"))
summary(logit_model2)
vif(logit_model2)
```

AIC : 1453
All predictor variables were found to be significant.

```
> vif(logit_model2)
     Age   Gender   Salary Distance  license
     1.5      1.0      1.4      1.2      1.0
```

VIF of the predictor variables taken into consideration also looks good in consideration to the previous model. Though there's a increase in AIC, but we have to do a trade-off between AIC and VIF. The above model was found to be showing the best performance in logistic regression.

Model Performance Evaluation

- Model Significance Test

Model significance is tested using log likelihood test.

```
lrtest(logit_model2)
```

Likelihood ratio test

Model 1: Transport ~ Age + Gender + Salary + Distance + license
Model 2: Transport ~ 1
  #Df LogLik Df Chisq Pr(>Chisq)

```
1   6   -721
2   1  -3906    -5   6371 <0.0000000000000002 ***
---
Signif. Codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

H0: All betas are zero
H1: At least 1 beta is nonzero

From the log likelihood, we can see that, intercept only model -3906 variance was unknown to us. When we take the full model, -721 variance was unknown to us. So we can say that, 1-(-721 /-3906)= 81.45% of the uncertainty inherent in the intercept only model is calibrated by the full model. Chisq likelihood ratio is significant. Also the p value suggests that we can accept the Alternate Hypothesis that at least one of the beta is not zero.
The model is significant.

- Robustness of the Model.

Since we have just concluded that the model is significant, we are not going to determine the robustness of the model.

```
pR2(logit_model2)
     llh      llhNull      G2     McFadden     r2ML   r2CU
 -720.71 -3906.12  6370.84       0.82           0.65     0.90
```

The McFadden’s pseudo-R Squared test suggests that at least 82% variance of the data is captured by our Model, which suggests it’s a robust model.

- Odds Explanatory power

```
> # Odds Ratio
> exp(coef(logit_model2))
          (Intercept)                       Age                     Gender1
0.00000000000000000082 3.41059002074046624386 0.42792602733687556960
              Salary                  Distance                    license1
0.97358603484723271748 1.36434648636499278318 3.07486516403297382993
> # Probability
> exp(coef(logit_model2))/(1+exp(coef(logit_model2)))
          (Intercept)                         Age                   Gender1
0.00000000000000000082 0.77327296454724303576 0.29968361045632752049
              Salary                            Distance              license1
0.49330812929196371508 0.57705014651324393338 0.75459310682803537595
```

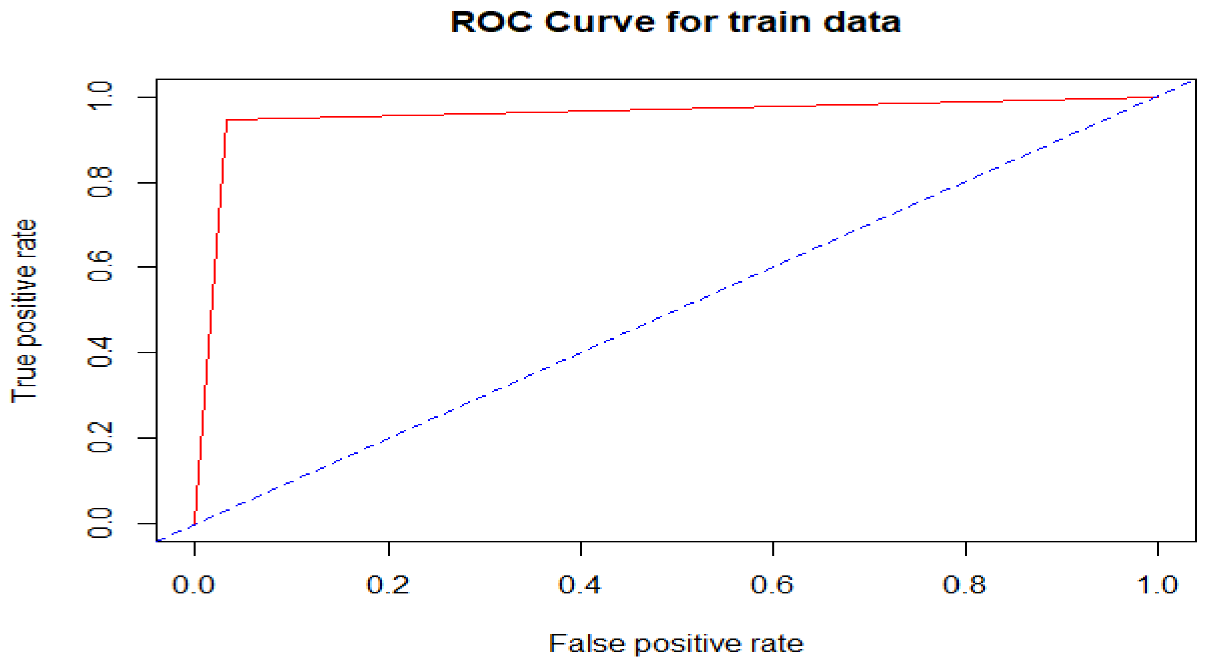
- Performance metric – In sample & 0ut- of- sample

```
Confusion Matrix and Statistics


      0    1
 0 3907  135
 1  112 1952

               Accuracy : 0.96
                 95% CI : (0.954, 0.964)
    No Information Rate : 0.658
    P-Value [Acc > NIR] : <0.0000000000000002

                  Kappa : 0.91

 Mcnemar's Test P-Value : 0.162

            Sensitivity : 0.972
            Specificity : 0.935
         Pos Pred Value : 0.967
         Neg Pred Value : 0.946
             Prevalence : 0.658
         Detection Rate : 0.640
   Detection Prevalence : 0.662
      Balanced Accuracy : 0.954

       'Positive' Class : 0
```

```
Confusion Matrix and Statistics


     0   1
 0 109   6
 1   1  17

               Accuracy : 0.947
                 95% CI : (0.895, 0.979)
    No Information Rate : 0.827
    P-Value [Acc > NIR] : 0.0000313

                  Kappa : 0.799

 Mcnemar's Test P-Value : 0.131

            Sensitivity : 0.991
            Specificity : 0.739
         Pos Pred Value : 0.948
         Neg Pred Value : 0.944
             Prevalence : 0.827
         Detection Rate : 0.820
   Detection Prevalence : 0.865
      Balanced Accuracy : 0.865

       'Positive' Class : 0
```

- ROC Plot

Finally, let's draw the Receiver Operating Characteristic (ROC) plot. It is a plot of the True Positive Rate against the False Positive Rate for the different possible cut-points of a diagnostic test.
Accuracy is measured by the area under the ROC curve. An area of 1 represents a perfect test; an area of 0.5 represents a worthless test.



- Area under curve

AUC of a classifier is equal to the probability that the classifier will rank a randomly chosen positive data point higher than a randomly chosen negative data point. Higher the probability better is the classifier.

```
> balanced.train.auc
 0.96
```

AUC of 0.96 shows that our model shows excellently good results.

- Gini coefficient

Gini coefficient is a ratio of two areas: the area between the ROC curve and the random model line.
It can also be simplified as: (2 * AUC – 1)

```
> balanced.train.gini
 0.91
```

- Kolmogorov–Smirnov test

This performance measure is defined as maximum difference between TPR and FPR. Higher KS stat value indicates better model.

```
> balanced.train.ks
 0.91
```

K Nearest Neighbour Model

The k-nearest neighbors (KNN) algorithm is a simple, easy-to-implement instance based supervised machine learning algorithm that can be used to solve both classification and regression problems.

For implementing KNN, we normalize our continuous variables in our dataset.

```
### Model Building - KNN
#Use KNN Classifier
#normalize the test & train data
norm=function(x){(x-min(x))/(max(x)-min(x))}
norm.balanced.data=as.data.frame(lapply(balanced.data.train[,c(1,5,6,7)],norm))
norm.balanced.data=cbind(balanced.data.train[,c(2,3,4,8,9)],norm.balanced.data)
test.knn=as.data.frame(lapply(test[,c(1,5,6,7)],norm))
test.knn=cbind(test[,c(2,3,4,8,9)],test.knn)
str(norm.balanced.data)
#KNN Algorithm
library(class)
knn.pred = knn(norm.balanced.data[,-c(5)], test.knn[,-c(5)], norm.balanced.data[,5], k =5)
confusionMatrix( table(test.knn$Transport, knn.pred))
```

Model Performance:

- Out of sample – confusion Matrix

```
Confusion Matrix and Statistics

   knn.pred
      0   1
  0 108   7
  1   0  18

               Accuracy : 0.947
                 95% CI : (0.895, 0.979)
    No Information Rate : 0.812
    P-Value [Acc > NIR] : 0.00000539

                  Kappa : 0.807

 Mcnemar's Test P-Value : 0.0233

            Sensitivity : 1.000
            Specificity : 0.720
         Pos Pred Value : 0.939
         Neg Pred Value : 1.000
             Prevalence : 0.812
         Detection Rate : 0.812
   Detection Prevalence : 0.865
      Balanced Accuracy : 0.860

       'Positive' Class : 0
```

Naive Bayes

Naive Bayes is a classification technique with the assumption that the predictors are independent of one another. It assumes that the presence of a particular feature in the given dataset has nothing to do with the presence of other features in the data set.

In the dataset, we have observed multicollinearity among the predictors. In that case, we can not typically apply Naive Bayes here. But we have applied Naive Bayes algorithm, only with a few predictors like Age, Gender & License.

```
### Naive Bayes
library(e1071)
NB = naiveBayes(Transport ~ Age+Gender+license, data = balanced.data.train[,-10])
predNB = predict(NB, test, type = "class")
confusionMatrix(table(test[,9], predNB))
```

The confusion Matrix for out-of-sample is shown below.

```
Confusion Matrix and Statistics

   predNB
      0   1
  0 110   5
  1   0  18

               Accuracy : 0.962
                 95% CI : (0.914, 0.988)
    No Information Rate : 0.827
    P-Value [Acc > NIR] : 0.00000168

                  Kappa : 0.856

 Mcnemar's Test P-Value : 0.0736

            Sensitivity : 1.000
            Specificity : 0.783
         Pos Pred Value : 0.957
         Neg Pred Value : 1.000
             Prevalence : 0.827
         Detection Rate : 0.827
   Detection Prevalence : 0.865
      Balanced Accuracy : 0.891

       'Positive' Class : 0
```

Model Comparison:

In this section, we are going to compare the accuracy, sensitivity ans specificity of the 3 models done above. In that way, we are going to determine which model performed best.

Model	Accuracy	Sensitivity	Specificity
Logistic Regression	0.947	0.991	0.739
K Nearest Neighbour	0.947	1.000	0.720
Naive Bayes	0.962	1.000	0.783

Sensitivity is the ability of the model to correctly identify the positives where as specificity is the ability of the model to correctly identify the negatives. Based Accuracy, sensitivity ans sensitivity, we got excellent results for Naive Bayes using only a few predictor variables compared to Logistic Regression and KNN. So, Naive Bayes has performed the best.

<u>Ensemble Methods:</u>

Ensemble Learning is a learning method in which instead of trying to learn one super accurate model, we create a number of weak models and then use the predictions given by those weak models to obtain a high accuracy model.
Two ensemble learning methods are Bagging and Boosting.

<u>Bagging</u>

Bagging consists of creating many copies of the training data with each copy slightly different from one another. We then apply our weak learners to each copy to create weak models and combine them.

We have applied bagging with minsplit = 4 and maxdepth = 5 and have used the unbalanced dataset.

```
data.bagging <- bagging(Transport ~.,data=train,control=rpart.control(maxdepth=5, minsplit=4))
test$pred.class <- predict(data.bagging, test)
confusionMatrix(table(test$Transport,test$pred.class))
```

The Confusion Matrix obtained is shown below:

```
Confusion Matrix and Statistics


      0   1
  0 113   2
  1   1  17

               Accuracy : 0.977
                 95% CI : (0.935, 0.995)
    No Information Rate : 0.857
    P-Value [Acc > NIR] : 0.00000255

                  Kappa : 0.906

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.991
            Specificity : 0.895
         Pos Pred Value : 0.983
         Neg Pred Value : 0.944
             Prevalence : 0.857
         Detection Rate : 0.850
   Detection Prevalence : 0.865
      Balanced Accuracy : 0.943

       'Positive' Class : 0
```

Bagging has provided excellent results even with the unbalanced data set.

Boosting

Boosting consists of using the original training data and iteratively create multiple weak learning models. Each new model would be different from the previous ones in the sense that the weak learner, by building each new model tries to fix the errors which the previous models make. The final ensemble model is a combination of those multiple weak models built iteratively.

We have used XGBOOST because of the better performance it provides w.r.t the other boosting techniques.

We have used unbalanced dataset for boosting algorithm.

```
xgb.fit <- xgboost(
  data = features.train,
  label = target.train,
  eta = 0.1,
  max_depth = 7,
  nrounds = 2,
  nfold = 5,
  objective = "binary:logistic",  # for regression models
  verbose = 1,               # silent,
  early_stopping_rounds = 10 )# stop if no improvement for 10 consecutive trees
test$xgb.pred.class=ifelse(test$xgb.pred.class>=0.5,1,0)
confusionMatrix(table(test$Transport,test$xgb.pred.class))
```

The confusion matrix is shown below:

```
Confusion Matrix and Statistics


      0   1
  0 110   5
  1   1  17

               Accuracy : 0.955
                 95% CI : (0.904, 0.983)
    No Information Rate : 0.835
    P-Value [Acc > NIR] : 0.0000193

                  Kappa : 0.824

 Mcnemar's Test P-Value : 0.221

            Sensitivity : 0.991
            Specificity : 0.773
         Pos Pred Value : 0.957
         Neg Pred Value : 0.944
             Prevalence : 0.835
         Detection Rate : 0.827
   Detection Prevalence : 0.865
      Balanced Accuracy : 0.882

       'Positive' Class : 0
```

Boosting has also provided excellent results with the unbalanced dataset.

Actionable Insights

We were here to understand whether an Employee will use CAR for transport based on certain predictor variables that were supplied.

- Gender

Only 15% females and 10% males are supposed to use Transport as CAR.

- License

Employees with license show a mixed preference to using CAR for transport where as Employees with no license show a 96% preference to using CAR as transport.

- Age

Employees with Age near or around 50 years show a greater preference to use CAR as transport.