CARS CASE STUDY

Classification Models

Smitayan Nandy

Table of Contents

1. Problem Statement/Objective

The objective of this exercise is to determine which employees will opt for Car as mode of transport for office commute. The dataset provided has attributes Age, Gender, Highest qualification, Work experience, Salary, Distance, license and mode of transport. Using these attributes we have to build a model which can predict effectively which employees are more likely to use car.

From the problem statement/objective it is clear that we need to build a classification model. Transport will be the dependent variable and we will use multiple models and compare the results to determine which algorithm is most effective

2. Data

Dataset provided consists of 418 observations and 9 variables.

Description of data attributes is as below:

Age	Employee age in years
Gender	Male/Female
Engineer	1 - Engineer; 0 – Not Engineer
MBA	1 – MBA; 0 – Not MBA
Work.Exp	Work experience in years
Salary	Employee salary
Distance	Distance between home and office
License	1 – Has license; 0 – Doesn't have license
Transport	2wheeler; Car; Public Transport

3. Exploratory Data Analysis

We will start with trying to understand the data attributes, their spread and how they are related to each other. Since our objective is to find out mode of transport used by an employee, hence Transport will be the dependent variable.

a. Attributes

```
[1] "Age"        "Gender"     "Engineer"   "MBA"        "Work.Exp"   "Salary"
[7] "Distance"   "license"    "Transport"
```

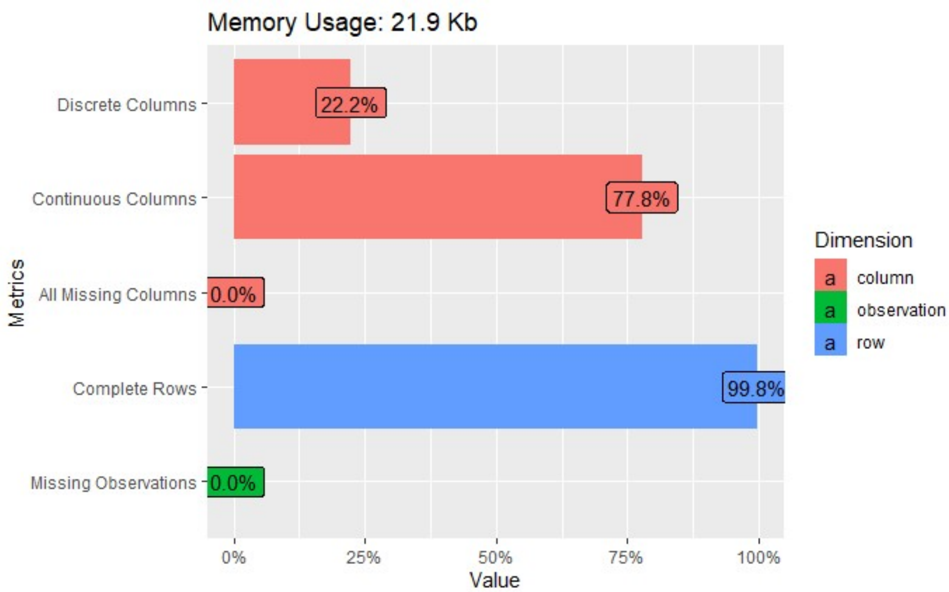
b. Data types

```
'data.frame':	418 obs. of  9 variables:
 $ Age      : int  28 24 27 25 25 21 23 23 24 28 ...
 $ Gender   : Factor w/ 2 levels "Female","Male": 2 2 1 2 1 2 2 2 2 2 ...
 $ Engineer : int  1 1 1 0 0 0 1 0 1 1 ...
 $ MBA      : int  0 0 0 0 0 0 1 0 0 0 ...
 $ Work.Exp : int  5 6 9 1 3 3 3 0 4 6 ...
 $ Salary   : num  14.4 10.6 15.5 7.6 9.6 9.5 11.7 6.5 8.5 13.7 ...
 $ Distance : num  5.1 6.1 6.1 6.3 6.7 7.1 7.2 7.3 7.5 7.5 ...
 $ license  : int  0 0 0 0 0 0 0 0 0 1 ...
 $ Transport: Factor w/ 3 levels "2wheeler","Car",..: 1 1 1 1 1 1 1 1 1 1 ...
```

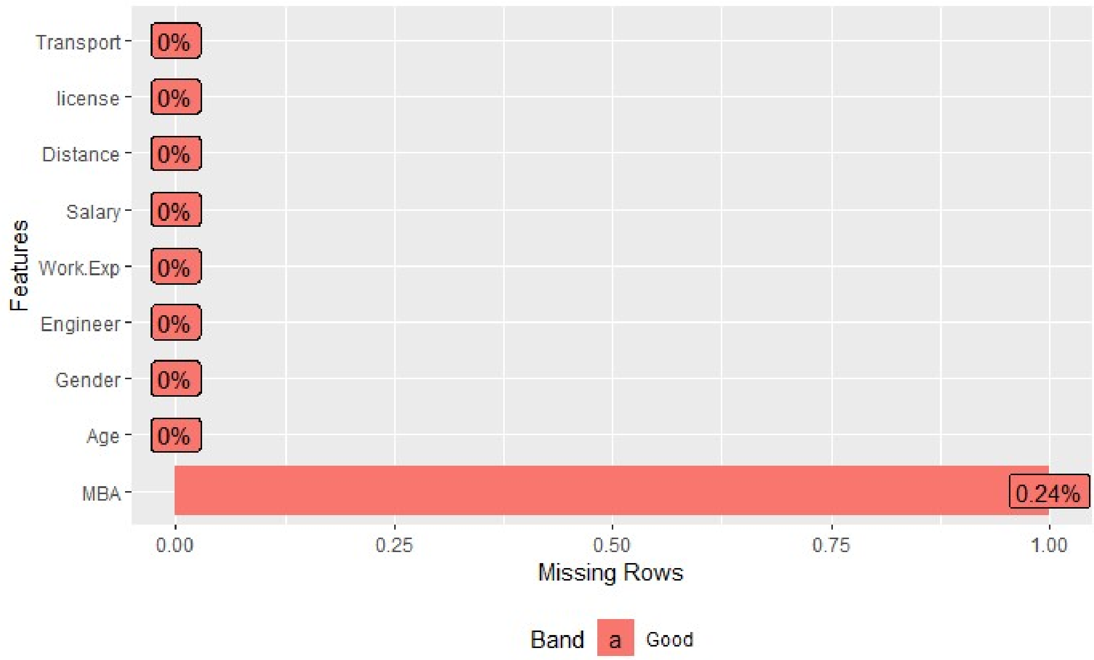
Age, Engineer, MBA, Work Exp, Salary, Distance, licence are numerical fields. Gender and Transport are factor variables. Engineer, MBA and license seem to be good candidate for factor variables.

c. Summary

```
      Age              Gender        Engineer           MBA             Work.Exp
 Min.   :18.00   Female:121   Min.   :0.0000   Min.   :0.0000   Min.   : 0.000
 1st Qu.:25.00   Male  :297   1st Qu.:0.2500   1st Qu.:0.0000   1st Qu.: 3.000
 Median :27.00                Median :1.0000   Median :0.0000   Median : 5.000
 Mean   :27.33                Mean   :0.7488   Mean   :0.2614   Mean   : 5.873
 3rd Qu.:29.00                3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.: 8.000
 Max.   :43.00                Max.   :1.0000   Max.   :1.0000   Max.   :24.000
                                               NA's   :1
     Salary          Distance        license                  Transport
 Min.   : 6.500   Min.   : 3.20   Min.   :0.0000   2Wheeler        : 83
 1st Qu.: 9.625   1st Qu.: 8.60   1st Qu.:0.0000   Car             : 35
 Median :13.000   Median :10.90   Median :0.0000   Public Transport:300
 Mean   :15.418   Mean   :11.29   Mean   :0.2033
 3rd Qu.:14.900   3rd Qu.:13.57   3rd Qu.:0.0000
 Max.   :57.000   Max.   :23.40   Max.   :1.0000
```



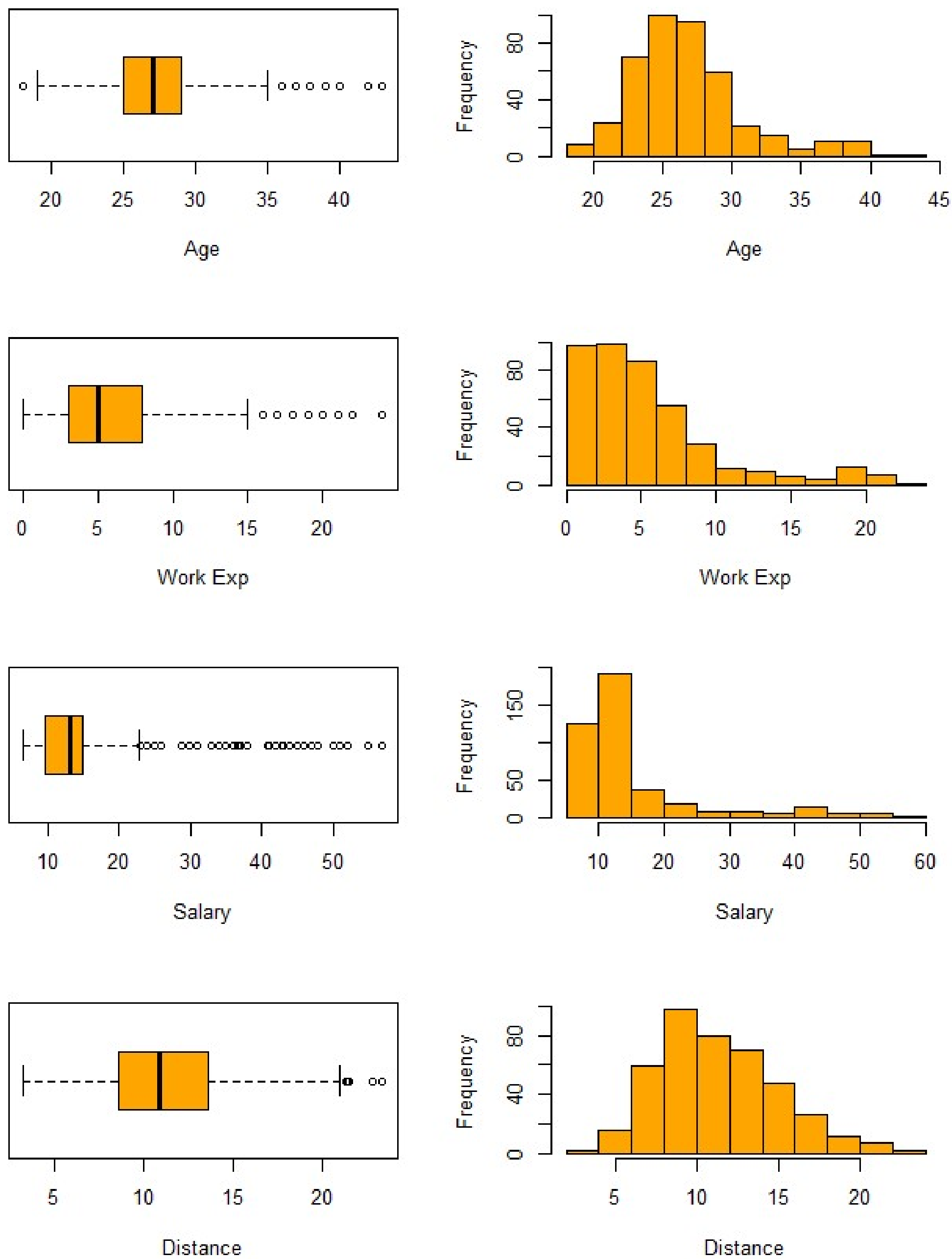
d. Null Check



MBA attribute has 1 Null value.

Engineer, MBA and license were converted to factor variables. Value of 1 means ‘Yes’ and 0 means ‘No’.

e. Boxplots and Histograms



Findings:

Age and Distance have pretty much uniform distribution, but have some outliers. Work Exp and Salary have highly right skewed distribution and have outliers.

Bar plots of Factor variables

Majority of the employees use Public Transport: 300, next highest group is of 2wheelers with 83 employees, and 35 employees use Car. Employees who use car constitute 8.39 % of the dataset.

f. Chi-square tests between factor variables

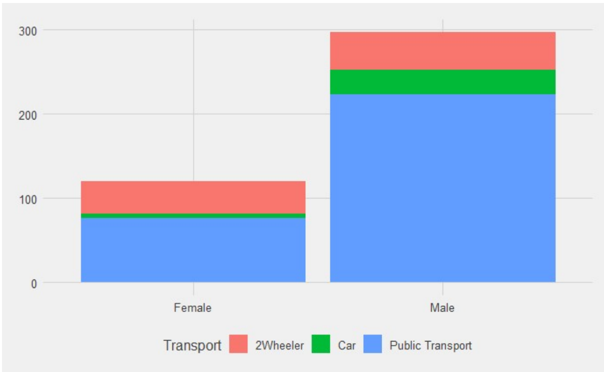
Since Personal Loan is our dependent variable, we will assess the dependency of other factor variables on Personal Loan.

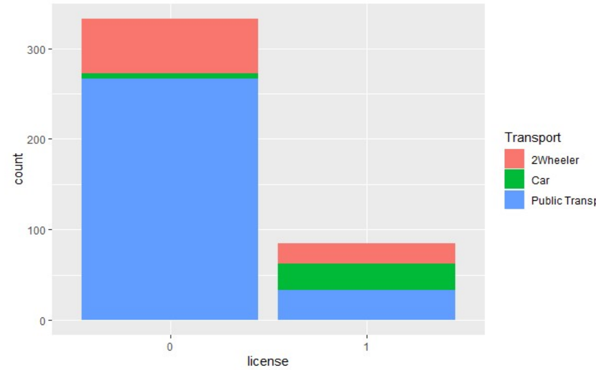
Ho = Variables are independent

Ha = Variables are not independent

chisq.test(cars.study$Gender,cars.study$Transport)$p.value	0.0003958196
chisq.test(cars.study$Engineer,cars.study$Transport)$p.value	0.2866151
chisq.test(cars.study$MBA,cars.study$Transport)$p.value	0.409505
chisq.test(cars.study$license,cars.study$Transport)$p.value	4.271117e-23

Above table shows that Transport has dependency on Gender and license.





From the above two plots and chi-square tests we see that:

- Male Employees have higher probability of opting Car.
- Employees who have license have higher probability of opting Car.

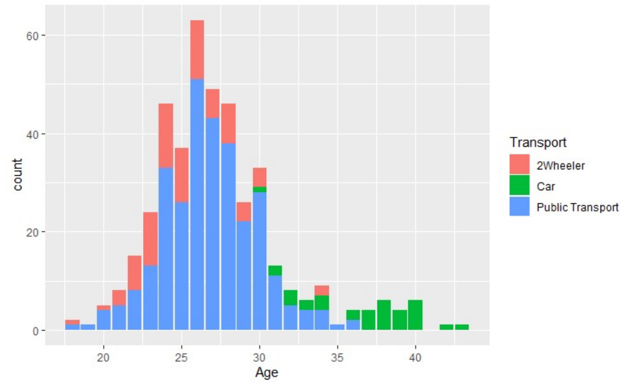
g. Correlation between continuous variables

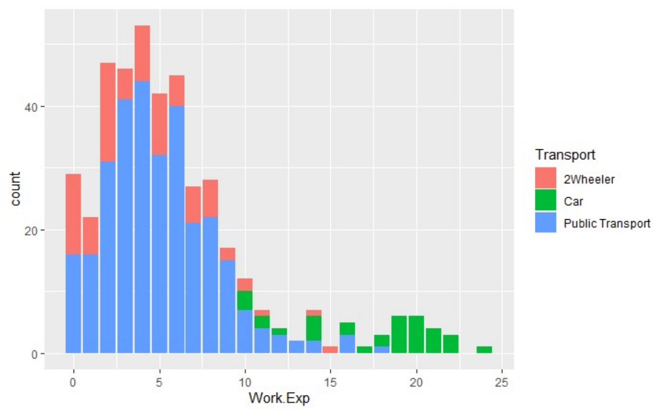
Summary from correlation plot:

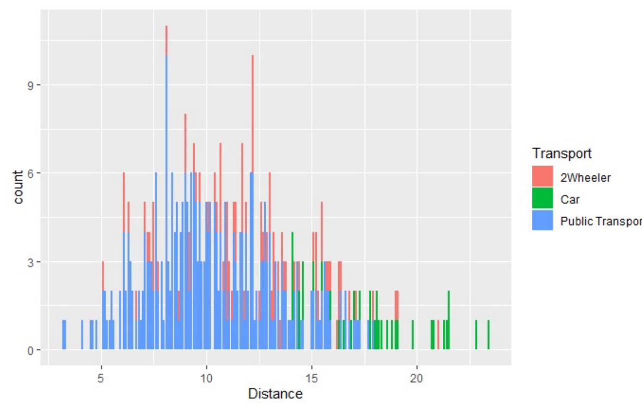
The correlation plot shows very strong correlation between Age Work Experience and Salary.

Weak positive correlation exists between Age and Distance; Work Ex and Distance; Salary and Distance.

h. Plots between continuous variables and Transport







Findings:

From the plots above we see that employees with higher work age, work exp and salary have the ability to own a car and hence are more likely to use car to commute to office.

Employees who stay far away from office also are more likely to commute by car, in order to reduce the commute time.

4. Data Preparation for classification models

Imbalanced Data: The dataset provided has only 35 observations with mode of Transport as Car. This constitutes ~8.4% of the data provided, hence it looks to be an imbalanced dataset, but before applying any oversampling or under sampling of the data we would first try see the accuracy of the models with the current dataset.

Null Treatment: From EDA we have seen that MBA is not a significant attribute in predicting who will use Car, hence we will retain the observation with Null value in MBA attribute.

Outlier Treatment: All attributes like Age, Salary and Work.Exp have outliers, but from EDA we see that these attributes might have significant impact on classification, hence we will keep the values as is.

Multiple output class: Attribute Transport has multiple classes, but before applying any classification model on the data we will reduce it to two classes. Our objective is to predict which employees will opt for Car, so our two classes can be 'Car' and 'Others'. This way we convert this task to a binary classification problem.

```
cars.study$Transport = as.character(cars.study$Transport)

cars.study$Transport = ifelse(cars.study$Transport !='Car','Others','Car')

cars.study$Transport = as.factor(cars.study$Transport)

table(cars.study$Transport)
```

```
   Car Others 
    35    383 
```

Finally we will split the data into training and test with a ratio of 70-30. The same training and test split will be used with Logistic Regression, Naïve Bayes, KNN, Bagging and Extreme Gradient Boosting, so that we can compare which model performs the best.

For Extreme Gradient Boosting the input data has to be numerical, so all the factor variables will be converted to numerical fields and one hot encoding will be applied wherever necessary.

5. Logistic Regression

We will start with Logistic Regression model to predict which employees will opt for Car.

Initially we will include all variables in the model and then asses which are significant. Based on the significance level of each attribute and the findings from EDA we will fine tune the model.

a. Model building

```
cars.logistic = glm(Transport~.,data = cars.training,family = 'binomial')

summary(cars.logistic)
```

```
Call:
glm(formula = Transport ~ ., family = "binomial", data = cars.training)

Deviance Residuals: 
     Min        1Q    Median        3Q       Max  
-2.11902   0.00012   0.00108   0.00854   1.53910  
```

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  75.3575    44.2195   1.704   0.0883 .
Age          -2.0188     1.4301  -1.412   0.1581
GenderMale    1.2982     1.7540   0.740   0.4592
Engineer     -0.4323     1.7672  -0.245   0.8068
MBA           1.8562     2.1357   0.869   0.3848
Work.Exp      0.8418     1.0654   0.790   0.4294
Salary       -0.1456     0.2038  -0.715   0.4748
Distance     -1.0086     0.4477  -2.253   0.0243 *
license      -2.8730     2.6916  -1.067   0.2858
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 175.196  on 290  degrees of freedom
Residual deviance:  16.317  on 282  degrees of freedom
  (1 observation deleted due to missingness)
AIC: 34.317

Number of Fisher Scoring iterations: 11
```

vif(cars.logistic)

```
       Age GenderMale   Engineer        MBA   Work.Exp     Salary   Distance
 24.392514   1.893063   1.212887   2.709972  29.764629   9.211644   4.030199
   license
  4.490734
```

The initial model shows most of the attributes as insignificant, only the intercept and distance are bit significant.

Variation Inflation Factor shows that there is high correlation between Age, Work.Exp and Salary.

Based on the findings of EDA and the VIF stats we will try to include only significant attributes and remove some of the collinear attributes and build a new model.

b. Model Tuning

We will update the model and include the attributes Age, Gender, Distance and license only.

cars.logistic = glm(Transport~Age+Gender+Distance+license,data = cars.training,family = 'binomial')

summary(cars.logistic)

```
Call:
glm(formula = Transport ~ Age + Gender + Distance + license,
    family = "binomial", data = cars.training)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.17122   0.00044   0.00277   0.01182   1.33292

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  51.5286    18.7068   2.755  0.00588 **
Age          -1.1272     0.4356  -2.588  0.00966 **
GenderMale    0.5148     1.3267   0.388  0.69802
Distance     -0.9042     0.3364  -2.688  0.00718 **
```

```
license      -1.2989     1.3244  -0.981  0.32673
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 175.38  on 291  degrees of freedom
Residual deviance:  18.29  on 287  degrees of freedom
AIC: 28.29

Number of Fisher Scoring iterations: 11
```

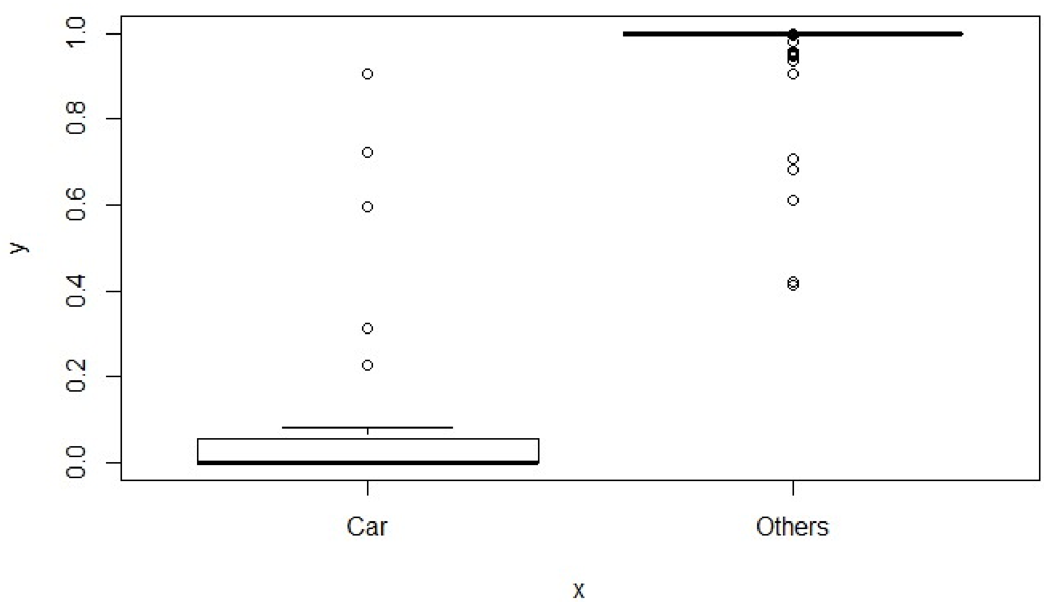
vif(cars.logistic)

```
       Age  GenderMale    Distance    license
  3.252862    1.179232    3.031925   1.204285
```

The revised model shows much better significance values for the attributes. Although Gender is showing as insignificant, but I will stick to it based on the findings of the chi-square test. VIF results shows that there is no collinearity between any of the independent variables used in the model. AIC value has also decreased to 28.29 which proves that the new model is better than the previous one.

c. Determining decision boundary

plot(cars.training$Transport,cars.logistic$fitted.values)



Based on the plot above, we can say that probability value of less than 0.92 can be considered as ‘Car’.

predicted.transport = ifelse(cars.logistic$fitted.values<0.92,'Car','Others')

table(cars.training$Transport,predicted.transport)

```
        predicted.transport
```

```
        Car Others
  Car     26      0
  Others   7    259
```

accuracy = sum(diag(table(cars.training$Transport,predicted.transport)))/nrow(cars.training)

accuracy
```
[1] 0.9760274
```

Results on the training data shows that we have a perfect TPR of 1 and an overall accuracy of 97.6%.

d. Model Performance

Confusion matrix on training data:

	Predicted Car	Predicted Others
Actual Car	26	0
Actual Others	7	259

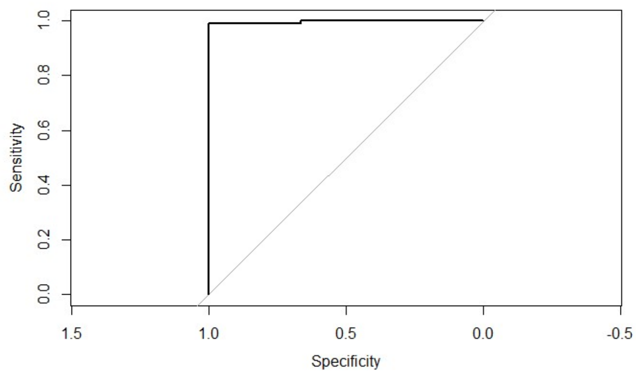
Confusion matrix on test data:

	Predicted Car	Predicted Others
Actual Car	8	1
Actual Others	1	116

Stats derived from confusion matrix

	Training Data	Test Data
TPR(Sensitivity)	1	.88
TNR(Specificity)	.973	.991
Accuracy	.976	.984

ROC Curve and other stats derived from Test data:



AUC(Area under the ROC Curve): 0.9972

6. Naïve Bayes

We will create a Naïve Bayes model on the same data set and compare it with Logistic Regression model to establish which model performed better in predicting customers who will opt for Car.

a. Data Preparation

We will use the same training and test data used in Logistic regression.

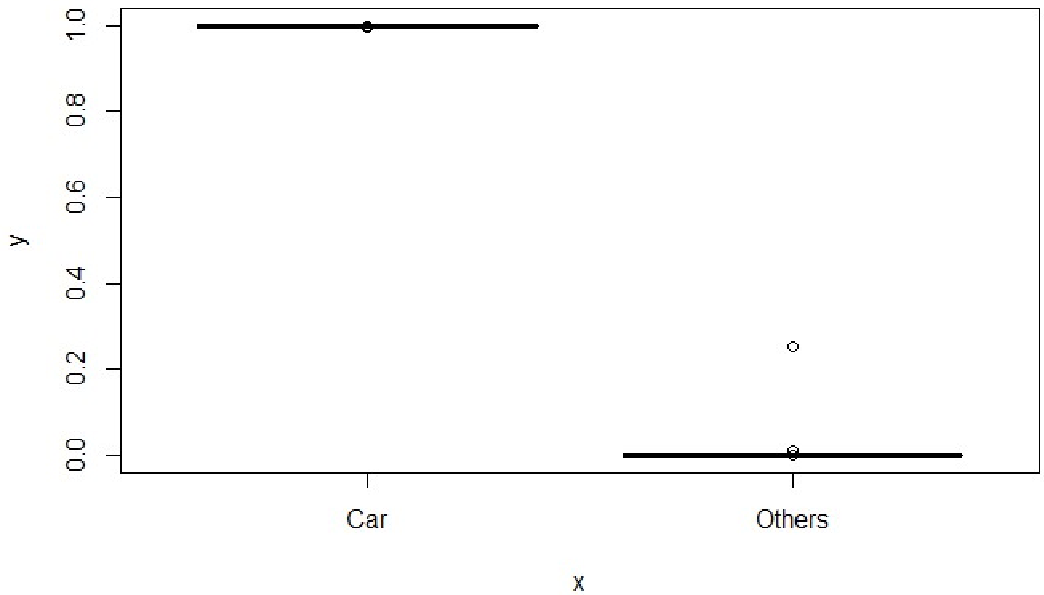
b. Model Building

Unlike logistic regression, we will use all important variables deduced from EDS in our model and use the same on test data for prediction.

```
cars.nb = naiveBayes(Transport~Age+Gender+Work.Exp+Salary+Distance+license,data=cars.training,laplace = T)

predicted.probs = predict(cars.nb,newdata = cars.test,type = 'raw')
```

c. Determining Decision boundary

```
plot(cars.test$Transport,predicted.probs[,1])
```



We see that there is a distinct decision boundary here, ideally anything above 0.5 can be considered as 'Car'. I will still try to keep it as 0.92, the same used in Logistic Regression and see the result.

predicted.transport = ifelse(predicted.probs[,1]>.92,'Car','Others')

table(cars.test$Transport,predicted.transport)

```
        predicted.transport
         Car Others
  Car      9      0
  Others   0    117
```

d. Model Performance

Confusion matrix on test data:

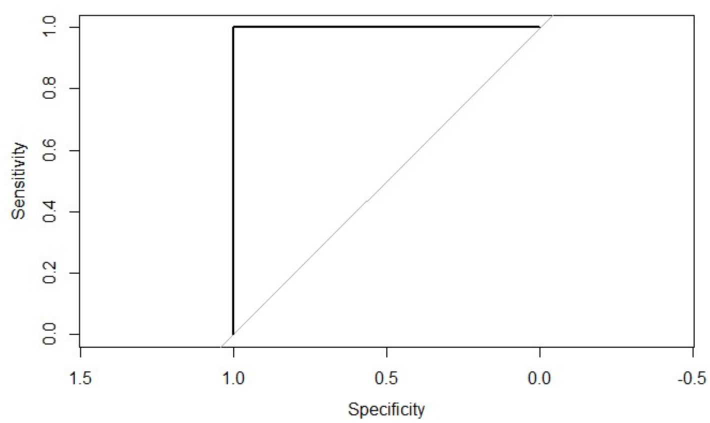
	Predicted Car	Predicted Others
Actual Car	9	0
Actual Others	0	117

Stats derived from confusion matrix

	Test Data
TPR(Sensitivity)	1

TNR(Specificity)	1
Accuracy	1

ROC Curve and other stats derived from Test data:



AUC(Area under the ROC Curve): 1

7. KNN

KNN is a distance based algorithm, so we will be using all numerical fields in the data set.

a. Data Preparation

We will use the same training and test data used in Logistic regression but only the attributes – Age, Work.Exp, Salary and Distance. Since the algorithm works on measures of distance, we will be scaling the data.

b. Model Building

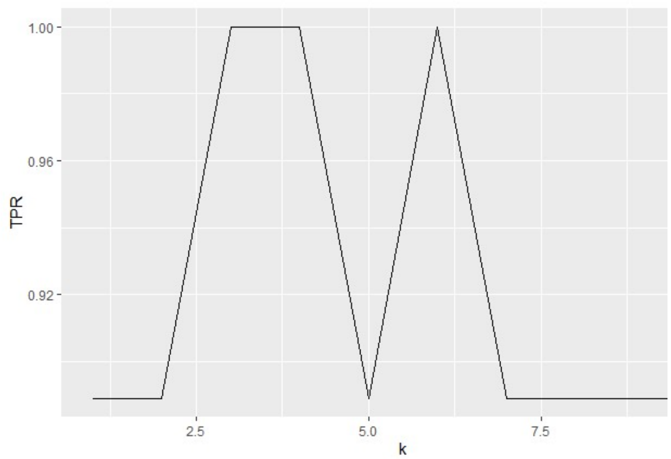
In KNN model building we will first need to identify the optimum K value. Since we don't have any automated way to find that out, we will have to try various values of K and determine the optimum value.

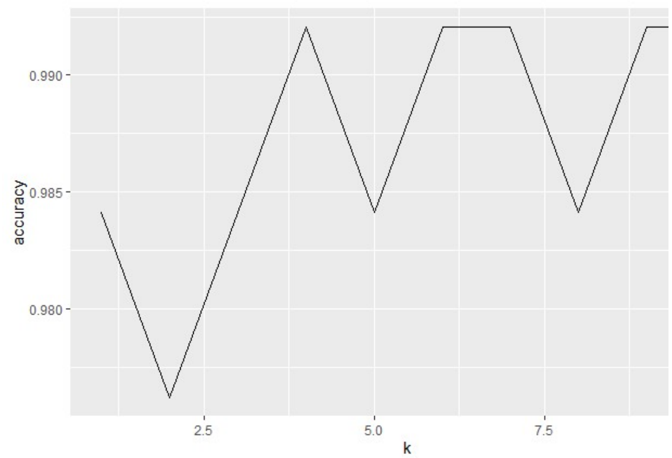
We will try select a K value for which we have the highest TPR and highest accuracy.

```
set.seed(10)

TPR = c()

accuracy = c()

for (i in 1:10){

  cars.knn = knn(scale(cars.training[,c(1,5,6,7)]),scale(cars.test[,c(1,5,6,7)]),cars.training[,c(9)],k=i)
```

```
  conf.matrix = table(cars.test$Transport,cars.knn)

  TPR[i] = diag(conf.matrix)[1]/sum(conf.matrix[1,])

  accuracy[i] = sum(diag(conf.matrix))/nrow(cars.test)

}
```

Plot between K and TPR; and Plot between K and accuracy





From the plot and the values derived we can use K=6 to get an optimum TPR and Accuracy.

c. Model Performance

Confusion matrix on test data:

	Predicted Car	Predicted Others
Actual Car	9	0
Actual Others	0	117

Stats derived from confusion matrix

	Test Data
TPR(Sensitivity)	1
TNR(Specificity)	1
Accuracy	1

8. Bagging

Bagging is a tree based algorithm, so we will use all the attributes for model building.

a. Data Preparation

We will use the same training and test data used in Logistic regression.

b. Model Building

The control parameters used in Bagging is similar to Decision Trees, we will use complexity parameter of 0 to let the tree grow completely. Terminal nodes will have atleast 5 observations. By default 25 splits of the data will be created and decision trees created, the final output will be based on an average of all the trees.

```
cars.bagging = bagging(Transport ~.,data = cars.training,control = rpart.control(minbucket = 5,cp=0,xval = 10),na.action=na.rpart)

varImp(cars.bagging)
```

	Overall <dbl>
Age	35.27835714
Distance	28.34621662
Engineer	0.00745758
Gender	0.36577802
License	10.52281186
MBA	0.39299695
Salary	39.59553236

The above output shows that Salary,Age,Distance and license are the most important variables in determining whether an employee will opt for Car. This is very much in line with our findings in EDA.

So we see a scope of tuning the model, hence we update the model using just the important attributes.

```
set.seed(1)

cars.bagging = bagging(Transport ~ Age+Distance+license+Salary+Work.Exp,data = cars.training,control = rpart.control(minbucket = 5, cp = 0, xval = 10))
```

c. Model Performance

Confusion matrix on training data:

	Predicted Car	Predicted Others
Actual Car	26	0
Actual Others	0	266

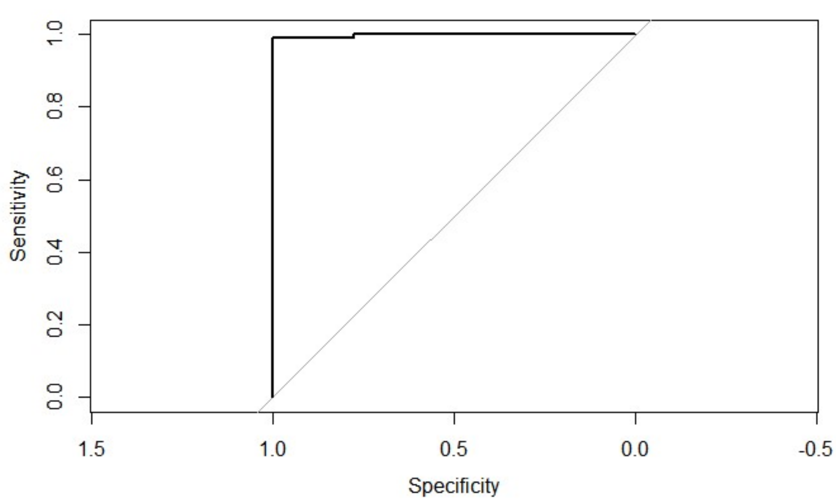
Confusion matrix on test data:

	Predicted Car	Predicted Others
Actual Car	6	3
Actual Others	0	117

Stats derived from confusion matrix

	Training Data	Test Data
TPR(Sensitivity)	1	.667
TNR(Specificity)	1	1
Accuracy	1	.976

ROC Curve and other stats derived from Test data:



Area under the curve: 0.9981

9. Boosting

The results using bagging wasn't great on test data, probably there was overfitting on training data. We will try use boosting to see if we get better performance than bagging.

a. Data Preparation

Boosting requires data to be in numeric form, hence we will have to convert all non numeric factor attributes into numeric form. Both training and test data have to be applied the same treatment. One hot encoding is done to convert factor attribute 'Gender' into numeric attributes 'Male' and 'Female'; rest of the factor attributes which have binary values are simply type casted to integer.

Dependent attribute Transport has been converted to a binary value attribute with the mapping 'Car'=1 and 'Others'=0.

```
cars.training.gender = one_hot(as.data.table(cars.training$Gender))

names(cars.training.gender) = c('Female','Male')

cars.training.1h = cbind(cars.training[-c(2)],cars.training.gender)

cars.training.1h$Transport = ifelse(cars.training.1h$Transport=='Car','1','0')

cars.training.1h$Engineer = as.integer(cars.training.1h$Engineer)

cars.training.1h$MBA = as.integer(cars.training.1h$MBA)

cars.training.1h$license = as.integer(cars.training.1h$license)

cars.training.1h$Transport = as.integer(cars.training.1h$Transport)

head(cars.training.1h)
```

	Age <int>	Engineer <int>	MBA <int>	Work.Exp <int>	Salary <dbl>	Distance <dbl>	license <int>	Transport <int>	Female <int>
416	27	1	1	4	13.9	17.3	1	0	1
179	29	1	1	7	14.6	7.7	1	0	1
14	24	2	1	6	12.7	8.7	1	0	0
195	27	1	2	4	13.6	8.2	1	0	1
307	29	2	1	5	14.9	11.2	1	0	0

```
str(cars.training.1h)
'data.frame':   291 obs. of  10 variables:
 $ Age      : int  27 29 24 27 29 39 33 28 25 23 ...
```

```
 $ Engineer : int  1 1 2 1 2 2 2 2 2 1 ...
 $ MBA      : int  1 1 1 2 1 2 2 1 1 1 ...
 $ Work.Exp : int  4 7 6 4 5 21 14 5 1 3 ...
 $ Salary   : num  13.9 14.6 12.7 13.6 14.9 50 34.9 14.6 8.6 9.9 ...
 $ Distance : num  17.3 7.7 8.7 8.2 11.2 23.4 10.9 9 9.4 17.9 ...
 $ license  : int  1 1 1 1 1 2 1 1 1 1 ...
 $ Transport: int  0 0 0 0 0 1 0 0 0 0 ...
 $ Female   : int  1 1 0 1 0 0 0 1 1 0 ...
 $ Male     : int  0 0 1 0 1 1 1 0 0 1 ...
```

b. Model Building

We will start by creating a model with the following parameters:

eta=0.3 (selecting the default learning rate)

max_depth=5 (since it is not a very complicated data set, we are going with a value lower than the default 6)

min_child_weight=5 (going with the same value as used in bagging)

nrounds = 100 (since its not a huge dataset)

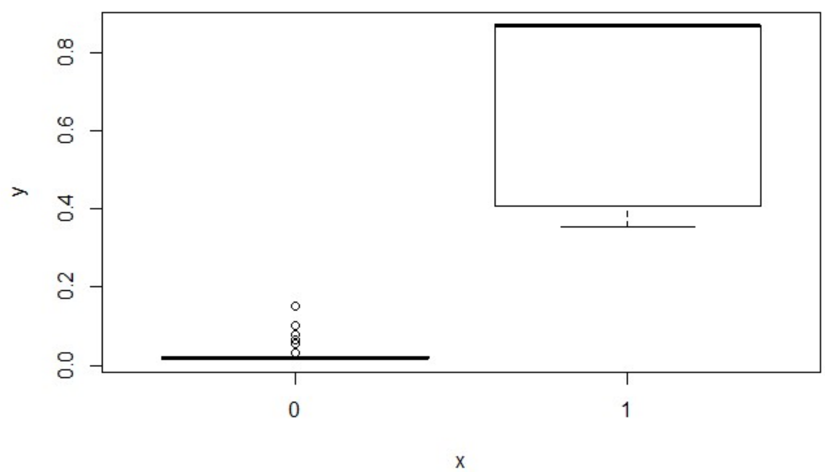
objective = "binary:logistic" (as we are doing binary classification between Cars and Others)

```
cars.xgb.fit = xgboost(

  data = as.matrix(cars.training.1h[,-c(8)]),

  label = as.matrix(cars.training.1h[,c(8)]),

  eta = 0.3,#this is like shrinkage in the previous algorithm

  max_depth = 5,#Larger the depth, more complex the model; higher chances of overfitting. There is no standard
value for max_depth. Larger data sets require deep trees to learn the rules from data.

  min_child_weight = 5,#it blocks the potential feature interactions to prevent overfitting

  nrounds = 100,#controls the maximum number of iterations. For classification, it is similar to the number of
trees to grow.

  nfold = 5,

  objective = "binary:logistic",  # for regression models

  verbose = 0,               # silent,

  early_stopping_rounds = 10 # stop if no improvement for 10 consecutive trees

)
```

predicted.probs = predict(cars.xgb.fit,as.matrix(cars.test.1h[,-c(8)]))

c. Determining Decision boundary

plot(as.factor(cars.test.1h$Transport),predicted.probs)



From the above plot we can say that any probability value greater than 0.3 can be classified as ‘1’ or ‘Car’.

predicted.transport = ifelse(predicted.probs>0.3,'1','0')

d. Model Performance

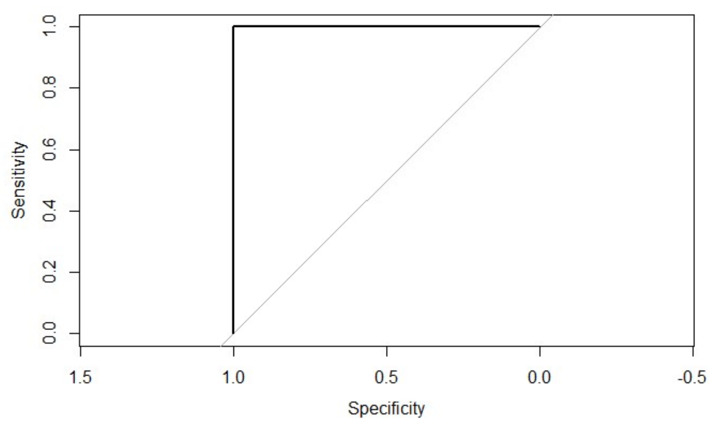
Confusion matrix on test data:

	Predicted Car	Predicted Others
Actual Car	9	0
Actual Others	0	117

Stats derived from confusion matrix

	Test Data
TPR(Sensitivity)	1
TNR(Specificity)	1
Accuracy	1

ROC Curve and other stats derived from Test data:



AUC(Area under the ROC Curve): 1

10. SMOTE

We started off with a slightly imbalanced dataset which had around 8% of the minority class observations. Although we got perfect results with Naïve Bayes,KNN and Boosting, we got decent TPR and accuracy with Logistic Regression and average TPR with Bagging. We will try balance the training data using SMOTE and try logistic regression and bagging again to check whether there is any improvement in results.

a. Data Preparation using SMOTE for Logistic Regression

cars.training.smote = SMOTE(Transport~Age+Gender+Distance+license,data = cars.training,perc.over = 200,k=5,perc.under = 500)

table(cars.training.smote$Transport)

```
   Car Others
    78    260
```

We increase the minority class by 200% and we have now around 30% of observations as Car in the training data.

b. Model building

We will use the same parameters as used earlier to build the logistic regression model.

cars.logistic.smote = glm(Transport~Age+Gender+Distance+license,data = cars.training.smote,family = 'binomial')

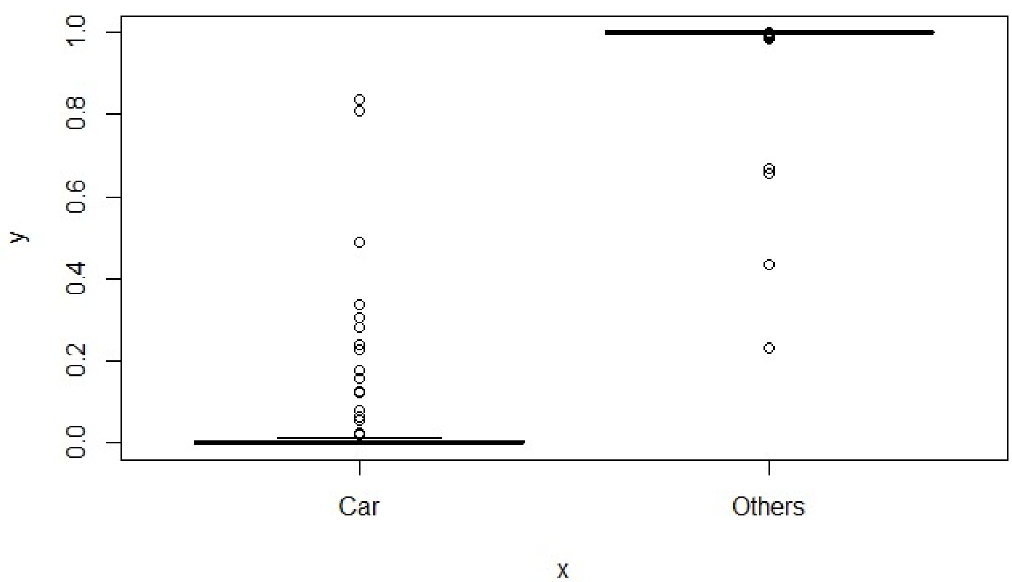
summary(cars.logistic.smote)

```
Call:
glm(formula = Transport ~ Age + Gender + Distance + license,
    family = "binomial", data = cars.training.smote)

Deviance Residuals:
     Min         1Q     Median         3Q        Max
-1.90690    0.00004    0.00126    0.01148    1.71020
```

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)   
(Intercept)    54.9142    16.9123   3.247  0.00117 **
Age            -1.2288     0.4151  -2.960  0.00307 **
GenderMale      0.2488     1.0936   0.227  0.82005   
Distance       -0.9556     0.3433  -2.784  0.00538 **
license1       -2.7925     1.3328  -2.095  0.03615 * 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 365.18  on 337  degrees of freedom
Residual deviance:  27.15  on 333  degrees of freedom
AIC: 37.15

Number of Fisher Scoring iterations: 10
```

The model summary shows that all variables are significant except for Gender, however will keep it in the model as it was a significant variable as part of EDA.

c. Determining decision boundary



Above plot shows that probability value of less than .90 can be considered as ‘Car’.

predicted.transport = ifelse(cars.logistic.smote$fitted.values<.90,'Car','Others')

table(cars.training.smote$Transport,predicted.transport)

```
        predicted.transport
         Car Others
  Car     78      0
  Others   8    252
```

We see a TPR of 1 on training data and accuracy of 97.63%.

d. Model Performance

Confusion matrix on training data:

	Predicted Car	Predicted Others
Actual Car	78	0
Actual Others	8	252

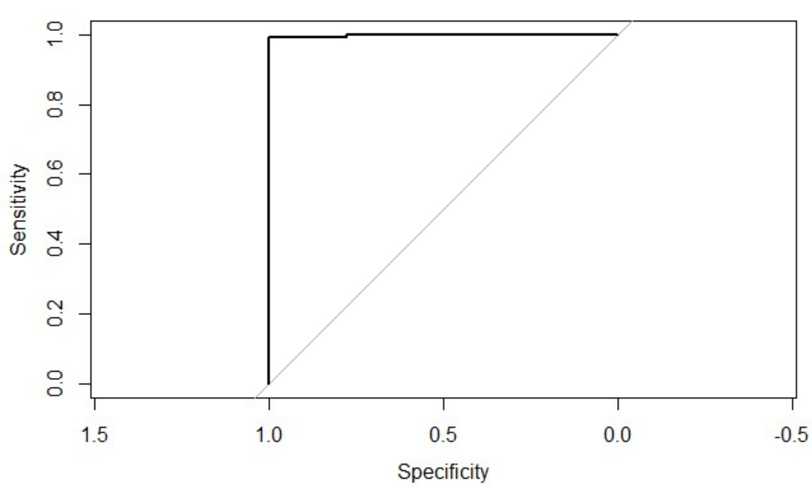
Confusion matrix on test data:

	Predicted Car	Predicted Others
Actual Car	9	0
Actual Others	3	114

Stats derived from confusion matrix

	Training Data	Test Data
TPR(Sensitivity)	1	1
TNR(Specificity)	.969	.974
Accuracy	.976	.976

ROC Curve and other stats derived from Test data:



AUC(Area under the ROC Curve):
0.9981

e. Data Preparation using SMOTE for Bagging

cars.training.smote = SMOTE(Transport ~ .,data = cars.training,perc.over = 200,k=5,perc.under = 600)

table(cars.training.smote$Transport)

```
   Car Others
    78    312
```

We increase the number of observations in the training data and now the minority class consists of 20% of the total.

f. Model building

We will use the same variables used for model building in Bagging earlier.

set.seed(2)

cars.bagging.smote = bagging(Transport ~ Age+Distance+license+Salary+Work.Exp,data = cars.training.smote,control = rpart.control(minbucket = 5, cp = 0, xval = 10,na.action=na.rpart))

g. Model Performance

Confusion matrix on training data:

	Predicted Car	Predicted Others
Actual Car	78	0
Actual Others	0	312

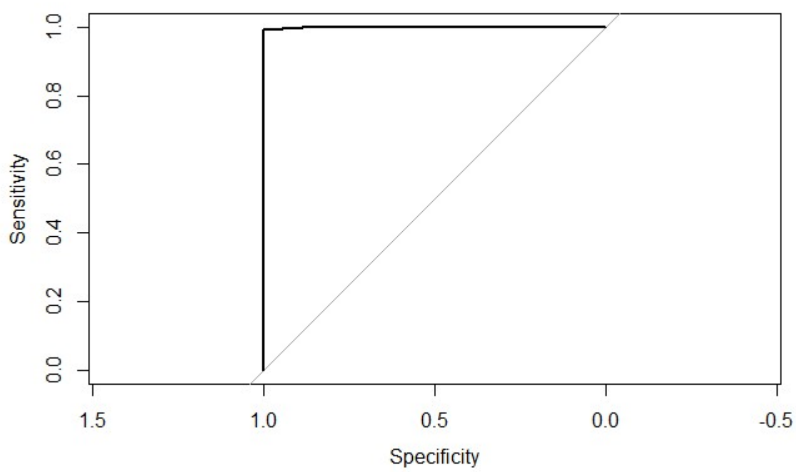
Confusion matrix on test data:

	Predicted Car	Predicted Others
Actual Car	9	0
Actual Others	1	116

Stats derived from confusion matrix

	Training Data	Test Data
TPR(Sensitivity)	1	1
TNR(Specificity)	1	.991
Accuracy	1	.992

ROC Curve and other stats derived from Test data:



AUC(Area under the ROC Curve): 0.9995

11. Insights and Conclusion

All the models have done a good job with respect to identifying which employees are more likely to opt for Car.

From the results its evident that, people who most likely own a car prefer to commute by Car. This group primarily have Employees who are:

1. Having Age > 30 years
2. Earning salary > 30
3. Having Work.Exp >= 10 years
4. Stay at a Distance >= 14 kms
5. Have driving license

So Age, Salary, Work Ex, Distance and license are the key attributes from our study.

We have used a lot of classification techniques on a fairly simple, clean and small data set. The dataset had some attributes which have a strong impact on the mode of Transport. Hence the classification wasn't a difficult task even with small number of observations who use 'Car'.

We got perfect results from Naïve Bayes,KNN and Extreme Gradient Boosting.

With the application of SMOTE we could marginally increase the accuracy of Logistic Regression and Bagging, at least we achieved a perfect TPR after applying SMOTE.

Model comparison stats:

Algorithm	TPR	Accuracy	AUC
Naïve Bayes	1	1	1
KNN	1	1	NA
Gradient Boosting	1	1	1
Logistic Regression	1	.976	.998
Bagging	1	.992	.999

Recommendation:

It would be good to continue this study with a bigger data set with more employees and attributes, to make the model more robust.