Machine Learning

Predicting preferred mode of commute

[1 Introduction 2](#_Toc52131937)

[2 Business Perspective 2](#_Toc52131938)

[3 Exploratory Data Analysis 3](#_Toc52131939)

[3.1 Basic Data Summary and preparation 3](#_Toc52131940)

[3.2 Outliers check 5](#_Toc52131941)

[3.3 Multicollinearity Check 6](#_Toc52131942)

[3.4 Variables impact on “Target” - hypothesis validations 7](#_Toc52131943)

[3.5 Splitting the data in training and testing dataset 9](#_Toc52131944)

[4 Building Logistic Regression 10](#_Toc52131945)

[4.1 Logistic Regression Model - – Iteration one (All variable) 10](#_Toc52131946)

[4.2 Multicollinearity – Iteration one 11](#_Toc52131947)

[4.3 Logistic Regression Model - – Iteration two (only relevant variables) 12](#_Toc52131948)

[4.4 Multicollinearity effect - Iteration two (only relevant variables) 14](#_Toc52131949)

[4.5 Assigning probabilities and class to the training set 14](#_Toc52131950)

[4.6 Model Performance Measures – Logistics Training 14](#_Toc52131951)

[4.7 Model Performance Measures – Logistics Testing 16](#_Toc52131952)

[4.8 Comparing model performance training and testing set. 18](#_Toc52131953)

[5 KNN Model 18](#_Toc52131954)

[5.1 Building the model 18](#_Toc52131955)

[5.2 Model Performance Measures 20](#_Toc52131956)

[6 Naive Bayes Model 21](#_Toc52131957)

[6.1 Building the Model 21](#_Toc52131958)

[6.2 Model Performance Measures 22](#_Toc52131959)

[7 Boosting 22](#_Toc52131960)

[7.1 Building the model 23](#_Toc52131961)

[7.2 Model Performance Measures 23](#_Toc52131962)

[8 Bagging 24](#_Toc52131963)

[8.1 Building the model 24](#_Toc52131964)

[8.2 Model Performance Measures 25](#_Toc52131965)

[9 Comparison of all models an Variable Importance 26](#_Toc52131966)

[10 Actionable insights and Recommendations 28](#_Toc52131967)

# Introduction

A sample dataset of 444 employees is given with information about their mode of transport as well as their personal and professional details like age, salary, work exp. The assignment requires to predict whether or not an employee will use Car as a mode of transport.

The R codes in the document are given in a grey box.

The R output in the document is given in a white box.

# Business Perspective

The business problem we are trying to solve here is predicting an employee will use Car as a mode of transport and interpret the factors that influence the employee’s decision to use of cars as a preferred mode of transport.

This prediction can be used by the company itself to plan for parking spaces or to design Car Lease schemes. This a class prediction problem hence different models such as logistics regression, KNN can be applied to build the model.

# Exploratory Data Analysis

## Basic Data Summary and preparation

We begin with understanding the data and summary statistics.

Following code is executed to load the dataset.

#loading file

getwd()

car = read.csv('Cars.csv',header = T)

**View**(car)

To understand the data and its structure, the following command is executed in R which returns the summary descriptive statistics and structure of each variable in the dataset.

#understand structure

str(car)

dim(car)

The dataset consists of 444 observations and 9 variables. the following variables were available for analysis (Data dictionary)

|  |  |
| --- | --- |
| Age | Age of the employee, read as an integer variable by R |
| Gender | Gender of the employee, read as a factor variable by R, Contains 2 levels  1 = Male, 2 = Female |
| Engineer | No of employees who are engineers, read as a integer variable by R, but is a factor variable containing 2 levels  0 = Non - engineer, 1 = Engineer |
| MBA | No of employees who are management graduates, read as a integer variable by R, but is a factor variable containing 2 levels  0 = Non - engineer, 1 = Engineer |
| Work.Exp | Work experience of the employee, read as a integer variable by R |
| Salary | Salary drawn by the employee, read as a numeric variable by R |
| Distance | Distance commuted by the by the employee, read as a numeric variable by R |
| License | No of employees who have a license, read as a integer variable by R, but is a factor variable containing 2 levels  0 = Non - engineer, 1 = Engineer |
| Transport | Different modes of transport preferred by the employee to reach office , read as a factor variable by R, containing 3 levels ;  1 = 2 wheeler; 2 = Car; 3 = Public transport |

Variables like Engineer, MBA and license need to be converted into factors for further analysis, following codes were written for changing the structure of the data.

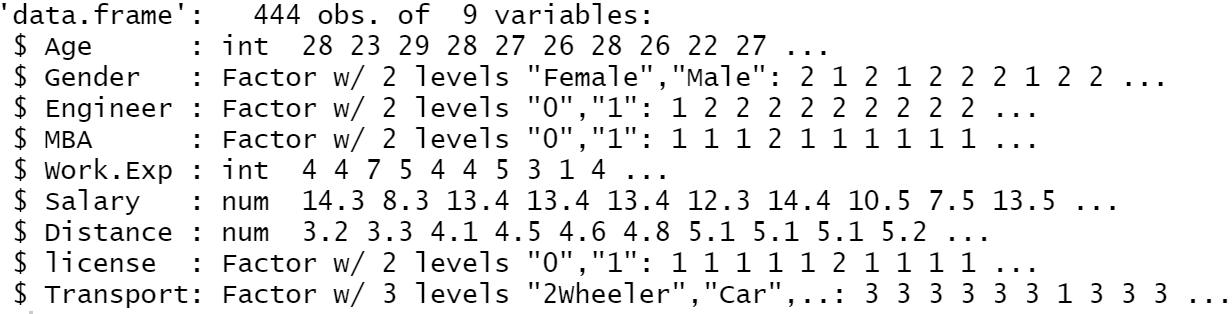
## Factor conversion - Engineer,MBA and license

car$Engineer = as.factor(car$Engineer)

car$MBA = as.factor(car$MBA)

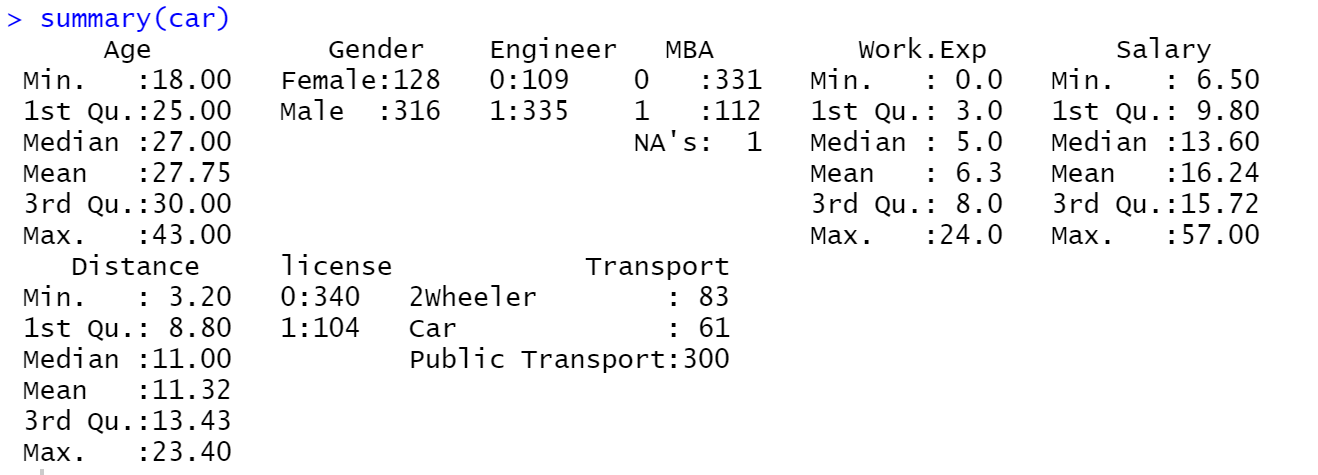
car$license = as.factor(car$license)

Data structure post converting variables can be seen in the below



Statistical summary of the data set is available in the below:

summary(car)



The variable “MBA” has one missing value there are no missing values in other variables.

This NA would be rectified using KNN imputation

#impute

library(DMwR)

car = knnImputation(car)

Major observation of the summary statistics include:

* The data contains information of 444 employee of which male are 72% & female are 28%
* 75% of the employee have an engineering degree
* Minimum age is 18 & maximum age is 43, median age of the work force is 27 years.
* 76% of the population doesn’t have a license.
* It can be seen that maximum employees travel between 5 & 15 miles. 75% of the population travels less than 13.43 miles for reaching workplace. The maximum distance commuted by the employees is around
* Outliers are observed in the variable “salary” which need to be to be addressed

## Outliers check

From the observations of the data summary it was suspected that the data had a few abnormal values that affected the normal distribution of data. Hence, the data was checked for outliers and the Following box blots confirm the presence. The plot below tells us that the 4 numeric variables Age, Work experience, salary & distance have outliers.

|  |  |
| --- | --- |
|  |  |
| Age | Work Experience |
|  |  |
| Salary | Distance |

The outliers are not treated as of now are not treated

## Multicollinearity Check

The correlation amongst variables can be checked only for numeric variables, hence we remove these and plot a correlation plot

These variables are Age, Work.Experinece, Salary and Distance

Plotting the correlation:

#correlation plot

library(GGally)

ggcorr(car)



It can be observed from the correlation matrix that Age, Work Experience and Salary are correlated, which is quite obvious as from a general understanding as well.

There may have a Multicollinearity effect on logistic regression model.

## Variables impact on “Target” - hypothesis validations

For further analysis, we require to check how each variable impact whether an employee will use Car as a mode of transport. We will first convert the target variable into binary for easier use in models.

#as we have to predict if the employee uses car or not we plan to create a variable

#Car use of 2 levels 0 = not using car & 1 = using car

car$Target = ifelse(car$Transport =='Car',1,0)

car$Target<-as.factor(car$Target)

If Target = 1, then employee uses car

It Target = 0 employee uses other modes of transport

Lets first write our Hypothesis and then validate it by performing t-test for numeric variables and Chi-square test for categorical variables.

Hypothesis for “Age” variable

* Null Hypothesis: Ages does not have an impact on Target. True difference in means is equal to zero
* Alternate Hypothesis: Age impacts on Target. True difference in means is not equal to zero

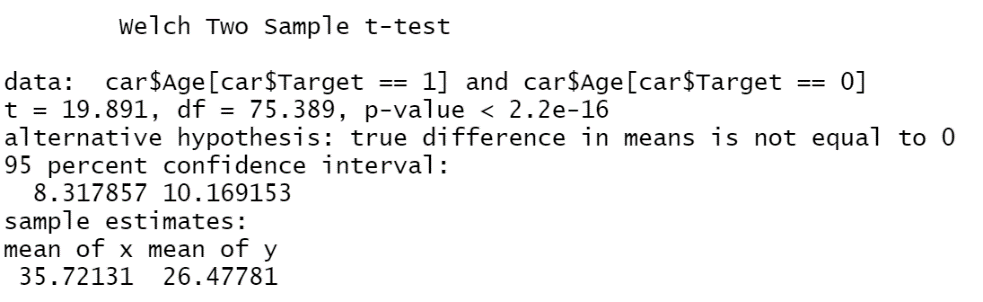
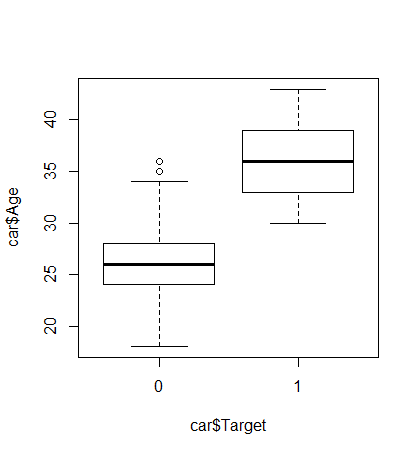
Similarly, we can write hypothesis for all variables. For all variables in which Null Hypothesis cannot be rejected, we can possible drop those for model creation.

Age being a numeric variable lets perform a t-test and plot a boxplot to understand if it individually impacts Attrition.

#age

boxplot(car$Age~car$Target)

t.test(car$Age[car$Target==1],car$Age[car$Target==0])arget)

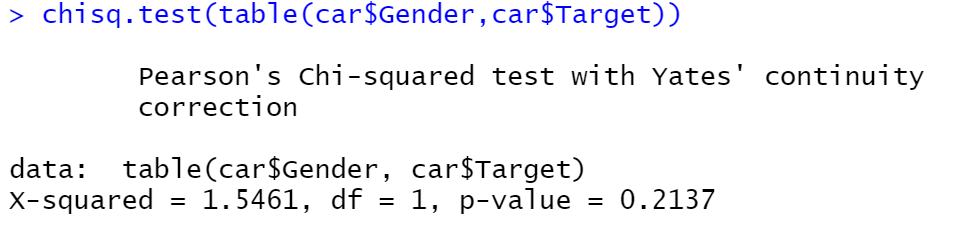


From the t-test we can see that p-value is < 2.2e-16 is significantly less than alpha of 5%, which mean that null hypothesis can be rejected and we can conclude that “Age” has an impact on “Target”. This can also from the boxplot above that mean age of employees travelling in car is more than those using other modes. We will continue to retain “Age” variable in out model.

Let see on example of Chi-Square test for a categorical variable.

#Gender - ChiSquare

chisq.test(table(car$Gender,car$Target))



From the Chi-Square test we can see that p-value is 0.2137 is more than alpha of 5%, which mean that null hypothesis cannot be rejected and we can conclude that “Gender” does has an impact on Car usage. We Can remove this from the

Following above steps for all variable, the conclusion is given in the following table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variables** | **Test** | **P-Value** | **Inference** |
| Age | t-test | < 2.2e-16 | Impacts Target |
| Gender | *Chi-Square test* | 0.2137 | Does not impact Target |
| Engineer | Chi-Square test | 0.07947 | Does not impact Target |
| MBA | Chi-Square test | 0.3539 | Does not impact Target |
| Work.Exp | t-test | < 2.2e-16 | Impacts Target |
| Salary | t-test | < 2.2e-16 | Impacts Target |
| Distance | t-test | 9.812e-14 | Impacts Target |
| license | Chi-Square test | < 2.2e-16 | Impacts Target |

## Splitting the data in training and testing dataset

The original data set will be split into training (Development) and testing data set (Holdout) as suggested in the 70:30 ratio. The testing data will help to test how well the model performs on a new data set.

The splitting of data will be done using random sample as shown in the R code below.

# training and Testing splitting data set

**set**.seed(123)

totalrows <- nrow(car)

training.sample = round(((totalrows)\*70/100),0)

s <- sample(totalrows, **size** = training.sample)

car\_train <- car[s,]

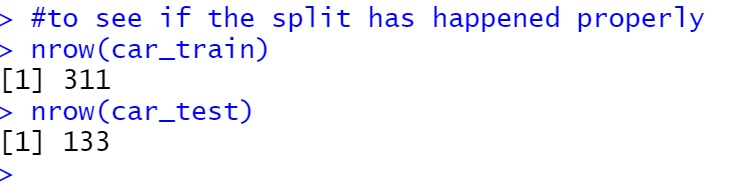
car\_test <- car[-s,]

Let see the total number of rows in each data set to see if the split has happened properly.

#to see if the split has happened properly

nrow(car\_train)

nrow(car\_test)

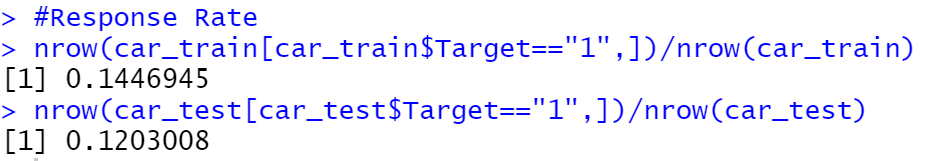


Let us also see target response rate for training and testing sample without prior to modelling.

#Response Rate

nrow(EmpAtt.train[EmpAtt.train$Attrition=="1",])/nrow(EmpAtt.train)

nrow(EmpAtt.test[EmpAtt.test$Attrition=="1",])/nrow(EmpAtt.test)



The target response rate for both the data sets is approximately around 12-14%. This does not classify as unbalance data set, so we can proceed to build model without oversampling or under sampling the data.

# Building Logistic Regression

Logistic regression is applied to predict the categorical variables, here, Car Usage.

It will help us in understanding which are the variables that impact the target and how.

## Logistic Regression Model - – Iteration one (All variable)

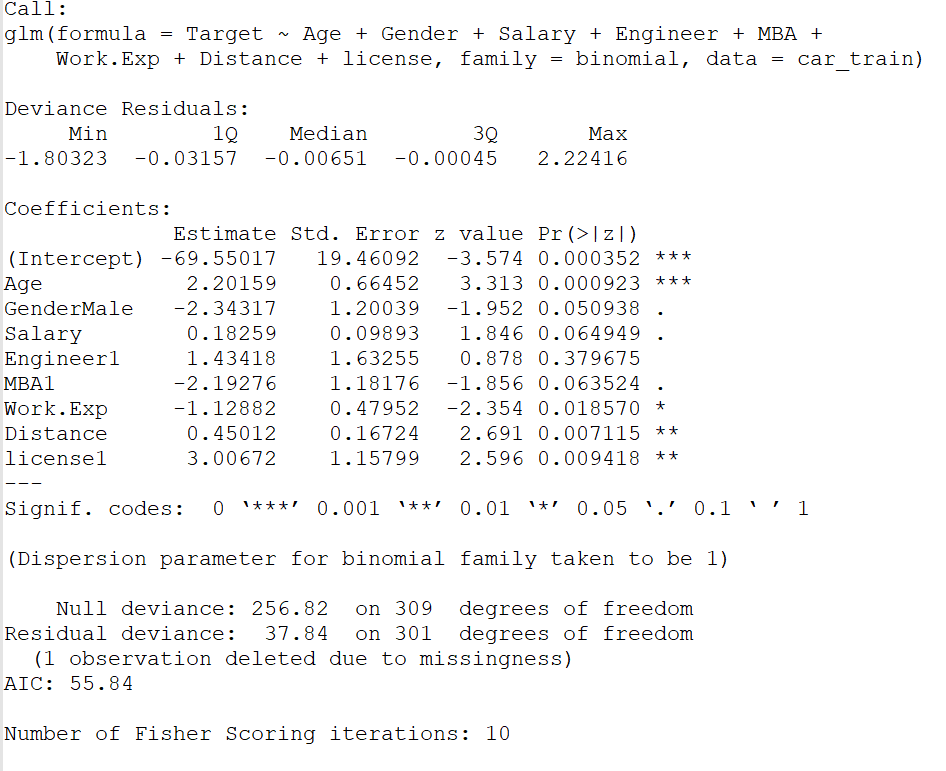
Let us perform logistic regression using all the independent variables on training set.

#logistic regression model

car\_glm\_model1 = glm(Target ~ Age+Gender+Salary+Engineer+MBA+Work.Exp+Distance+license, data=car\_train, family = binomial)

summary(car\_glm\_model1)

The summary of model.1 that we have built is as follows:



Following interpretations can be made from the above summary:

* The variable “Engineer” is not significant for predicting the Car usage as its p-value is much greater than the alpha of 5%.
* This was also indicated by the Chi-square test

Lets remove this variable

## Multicollinearity – Iteration one

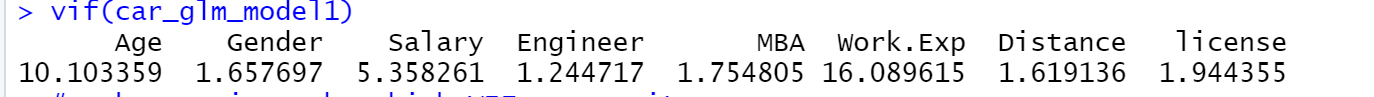
From the correlation plot, we know that few variables have medium -to high correlation. For a regression model multicollinearity, independent variables makes the model unstable.

Let’s check “Variation Inflation Factor “ (VIF) of each dependent variable to see how large its standard error of the estimated coefficient is with respect to other dependent variable.

#Multicollinearity

library(car)

vif(car\_glm\_model1)



Work Experience has a relatively high VIF compared to other variables.

## Logistic Regression Model - – Iteration two (only relevant variables)

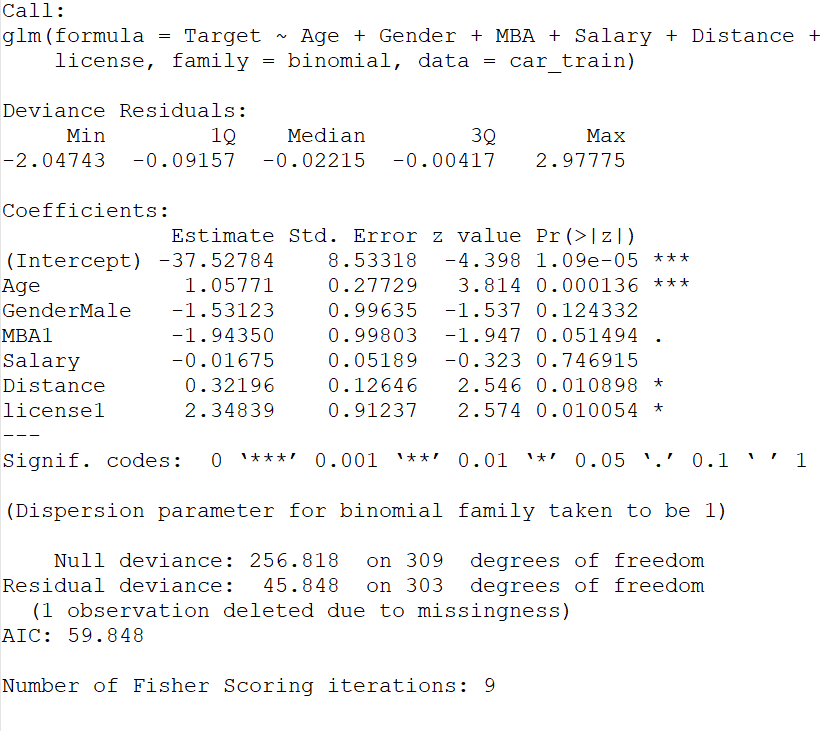
From the first logistic regression model and VIF output we have observed that some, some are insignificant to predict Target, there is multicollinearity effect which may render the model unstable.

Lets’ remove this Work Experience and Engineer variables and build the mode again and interpret the results

#model removing Engineer

car\_glm\_model2 = glm(Target ~ Age+Gender+MBA+Salary+Distance+license, data=car\_train, family = binomial)

summary(car\_glm\_model2)



Following interpretations can be made from the above summary:

* Variables MBA and Salary will be retained as they have P-value only slightly higher than 0.05
* Rest all variables are important in this logistic regression model as their p-value is close less than than the alpha of 5%.

The prediction equation can be formed as:

Logit (Churn) =

e -37.52784 + 1.05771 (x1) -1.53123 (x2) -1.94350 (x3) -0.01675 (x4 ) +0.32196 (x5) + 2.34839 (x6))

1+ e -37.52784 + 1.05771 (x1) -1.53123 (x2) -1.94350 (x3) -0.01675 (x4 ) +0.32196 (x5) + 2.34839 (x6))

Where

-37.52784 = Coeffecint of Intercept

X1 = Age

X2 = Gender if value is “Male”

X3 = MBA if value is “1”

X4 = Salary

X5 = Distance

X6 = license if value is “1”

Let’s interpret the equation

* + If the Age of employee increases by one unit, the odds for Target (opting for Car) increases by 1.05771times (i.e 51% probability) keeping other variables at the same level.
  + If the employee is Male the odds for Target (opting for Car) decreases by 1.53123 times (i.e 60% probability) keeping other variables at the same level.
  + Likewise, we can interpret the equation for other variables.
  + We observe that longer Distance and having a license increases the probability of using Car.
  + Whereas having an MBA, or higher Salary influences target negatively.

The null deviance has not reduced much from the previous model.

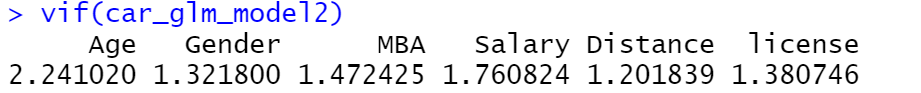
## Multicollinearity effect - Iteration two (only relevant variables)

Let’s check multicollinearity of this model by calculating “Variation Inflation Factor “ (VIF) for model 2

#Multicollinearity

library(car)

vif(car\_glm\_model2)



We observe that VIF for all variables is small, hence we can conclude that model 2 does not have multicollinearity effect.

## Assigning probabilities and class to the training set

Based on the logistic regression model worked initiation 2 let us calculate probabilities and class for the training set. For probalities greater than 0.5 is taken as class 1 meaning they will use Car

# assigning probablities and class

car\_train$fittedvalue <-car\_glm\_model2$fitted.values

car\_train$prediction <-ifelse(car\_train$fittedvalue >0.5, 1, 0)

## Model Performance Measures – Logistics Training

The logistic regression model built by retaining only relevant variables seems to be a good start, we would calculate the model performance measures for that model first on training set then on the testing set.

Confusion matrix and other performance measures such gini, KS AUC as will be used as the model performance measure. We will load the required libraries and run for training and test data.

#confusion matrix

library(caret)

library(e1071)

confusionMatrix( as.factor(car\_train$prediction),as.factor(car\_train$Target))

# Other Model Performance Measures

library(ROCR)

library(ineq)

pred <- ROCR::prediction(car\_train$fittedvalue, car\_train$Target)

perf <- performance(pred, "tpr", "fpr")

KS <- **max**(attr(perf, 'y.values')[[1]]-attr(perf, 'x.values')[[1]])

auc <- performance(pred,"auc");

auc <- as.numeric(auc@y.values)

gini = ineq(car\_train$fittedvalue, type="Gini")

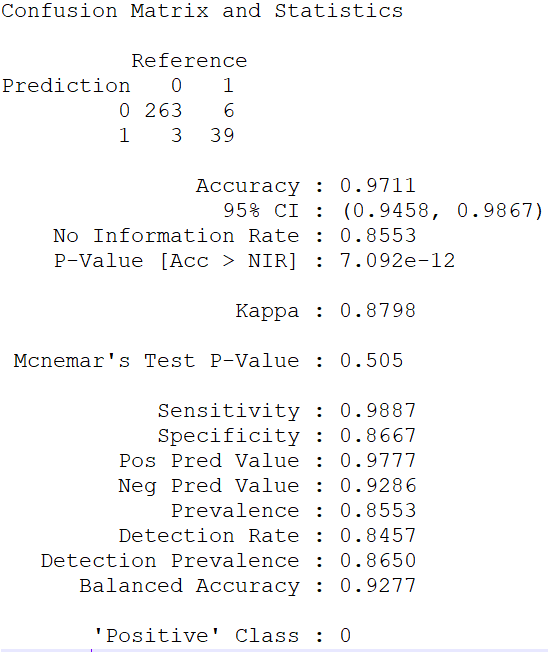
with( car\_train, table(Target, as.factor(prediction) ))

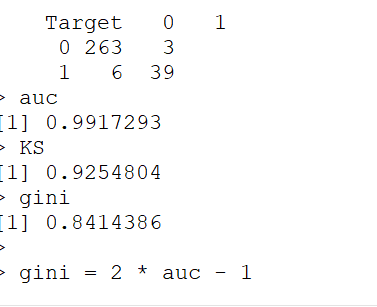
auc

KS

gini

gini = 2 \* auc - 1





The accuracy of the trained dataset is 97% which seems to be good. The sensitivity and specificity is 99% and 88% respectively. Accuracy of the model is the predictive power of the model, the higher the number better is the predictive power.

AUC stands for Area under the curve. AUC gives the rate of successful classification by the logistic model, which is 99 %. Higher the better.

K-S is a measure of the degree of separation between the positive and negative distributions, here it is 92%. The higher the value the better the model is at separating the positive from negative cases. KS value of more than 40% is considered to be a good model.

Gini coefficient of 84% indicated there is inequality in the data.

All the model performance measure indicates that the model is a fair model. The next step is to see how they fare in the testing model.

## Model Performance Measures – Logistics Testing

To predict the scores of the testing data set, we would use the predict function on testing data set and assign class with the same threshold as taken for the training set.

#testing

car\_test$fittedvalue <-predict(car\_glm\_model2, newdata=car\_test, type = "response")

## Assgining 0 / 1 class based on certain threshold

car\_test$prediction <- ifelse(car\_test$fittedvalue >0.5, 1, 0)

Model performance measure for testing set

#Confusion Matrix testing

confusionMatrix( as.factor(car\_test$prediction),as.factor(car\_test$Target))

# Other Model Performance Measures - testing

pred <- ROCR::prediction(car\_test$fittedvalue, car\_test$Target)

perf <- performance(pred, "tpr", "fpr")

KS <- **max**(attr(perf, 'y.values')[[1]]-attr(perf, 'x.values')[[1]])

auc <- performance(pred,"auc");

auc <- as.numeric(auc@y.values)

gini = ineq(car\_test$fittedvalue, type="Gini")

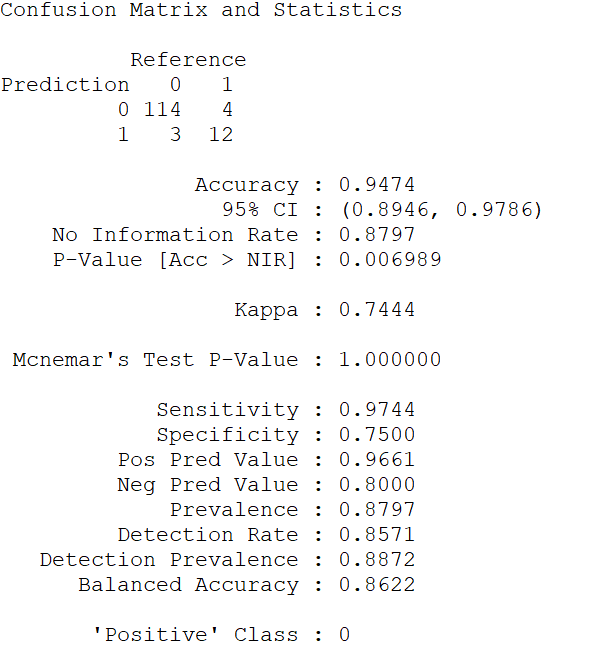
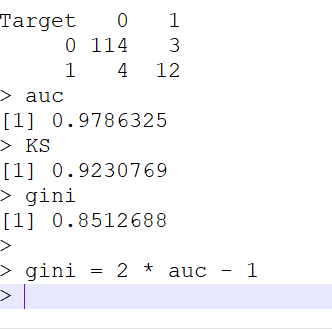
with( car\_test, table(Target, as.factor(prediction) ))

auc

KS

gini

gini = 2 \* auc - 1

The accuracy of the testing dataset is 94%, The area under the Curve (AUC) is 97% KS is 92% and gini coefficient is 0.85 seems to be good for this model on the training set and comparable to training model.

## Comparing model performance training and testing set.

The following table gives a comparative matrix on model performance measure of training and testing data set.

|  |  |  |
| --- | --- | --- |
| Performance Metric | Training Data | Testing Data |
| Accuracy Classification Score | 97% | 94% |
| Area Under the Receiver Operating Characteristic Curve | 99% | 97% |
| KS | 92% | 92% |
| Gini Coefficient | 82% | 85% |

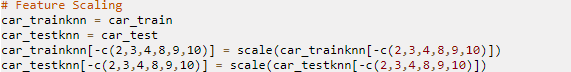
The model performance measure of testing data set as compared to the training data set is similar, which seem to suggest it is s good model and has not overfitted.

# KNN Model

KNN is a non-parametric model which classifies the target variable based on the features of its nearest neighbouring records.

## Building the model

In our dataset, we have many numeric variables which are in different value ranges. Hence, we need to scale these variables else the output of the model might turn out to be biased.



Also, let us assign levels to the factor variable Gender, i.e.,

0 → Male

1 → Female

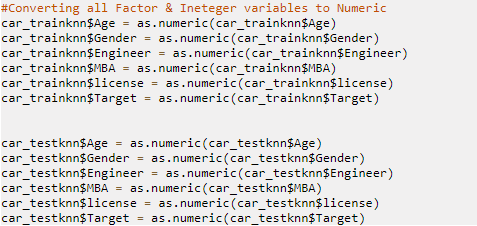


We have derived a new variable called Target from the Transport variable for our prediction, hence we can discard the Transport variable.

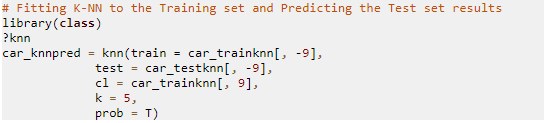


KNN algorithm requires its predictors to be numeric, as it deals with the distances.

Converting the factor variables to numeric,



Now the training dataset is ready to be run on the KNN algorithm and predict the Target variable in the test dataset using the same.



While building the model, the parameter k, no. of neighbours considered is taken as 5.

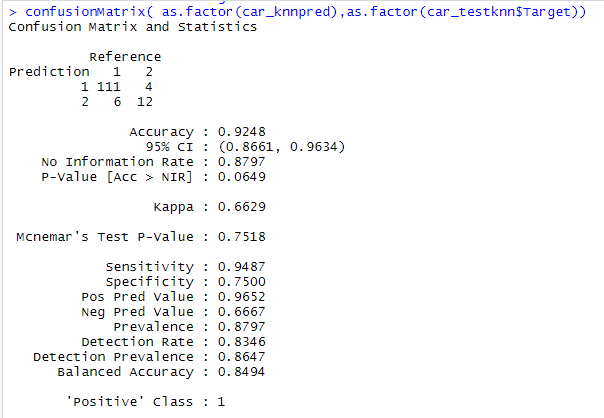


Out of total 133 records, 115 records have been classified as people who will not take cars, whereas 18 people would take car as a mode of transport.

## Model Performance Measures

To calculate the performance of the KNN model, let us compute confusion matrix and check for metrics such as accuracy, sensitivity, specificity.





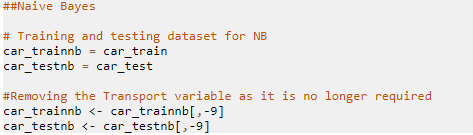
The above model shows an accuracy of 92% which is fairly good. Here, the sensitivity (True positive rate) is quite high (95%) but the specificity is decent (75%). The model is good at predicting the people who will not take car as a mode of transport.

# Naive Bayes Model

Naive Bayes algorithm has its basis in Bayes’ theorem and it calculates the probability of the target variable based on the prior knowledge of the features related to the target variable.

## Building the Model

Preparing the Training and Test dataset for Naive Bayes and removing the Transport variable as it is not to be used in the model



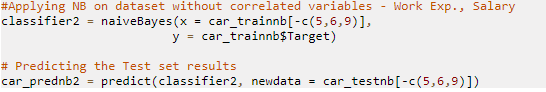
Naive Bayes algorithm cannot be applied to this dataset directly as it makes a very strong assumption that one predictor variable is not impacted by the value of another predictor variable, that is the variables are independent of each other.

Also, Naive Bayes model is more suited for the dataset with categorical variables.

Here, we observe there is strong correlation between Age, Work Excp. and Salary.

If we apply Naive Bayes in this dataset, these variables might inflate their effect on the target variable.

Hence, let us apply Naive Bayes while keeping only one of these correlated variables and predict the test set:

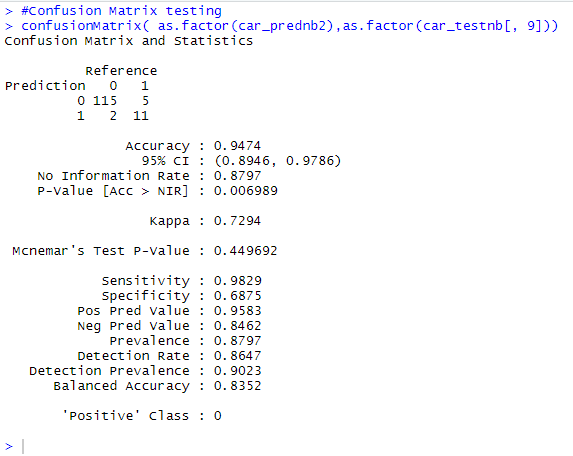


The model has predicted 120 people to be using modes of transport other than car, and 13 people to be using cars.



## Model Performance Measures





The predictive power of the model (accuracy) is 95% which has increased by 3% from the KNN model. The sensitivity of this model is very high (98%) but the specificity is not that high (69%).

# Boosting

Boosting algorithms aim to improve the prediction power of the machine learning models by training a number of models sequentially while compensating on the error in each model.

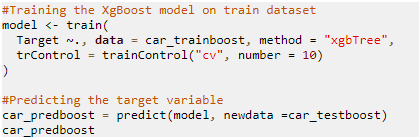
The gradient boosting method calculates the difference between the prediction and the actual value and adjusts the weights of the predictor variables accordingly so as to minimize the errors.

## Building the model

the transport variable is not required in the model, hence removing it:



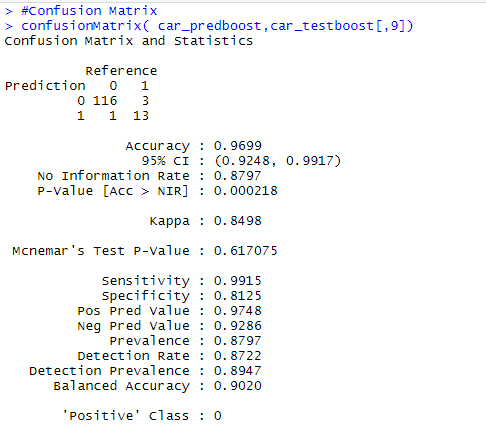
Let us build the model and perform prediction on the test dataset:



Here, the number of boosting iterations have been specified as 10.

## Model Performance Measures





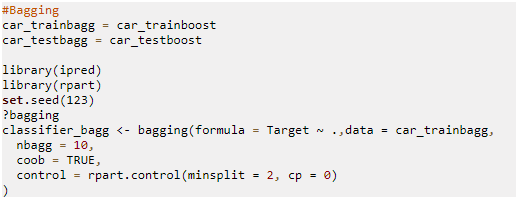
It can be observed the predictive model has improved to 97% and the Sensitivity (99%) and Specificity(81%)which is increased.

# Bagging

Bagging is another ensemble method which builds multiple models independently and parallely, and then combines the results in some deterministic averaging method.

## Building the model

Let us build the model using Bagging function in R:



Here, the number of bootstrap replications is set as 10, which means 10 models will be built independently. The control parameter sets the number of minimum splits for classification as 2.

Let us predict the test dataset value based on this model:

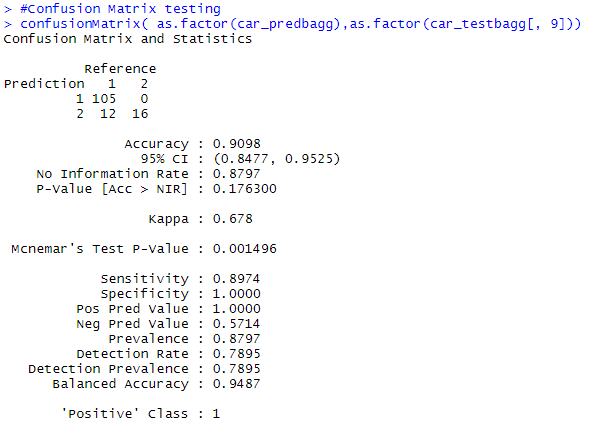




It can be observed that the prediction for the number of people preferring car s a mode of transport has significantly increased in this model (28).

## Model Performance Measures





The accuracy of the model is 90% which is fairly good but the important measure here is Specificity that is 100% for this model. It correctly predicts all the people that would take car as a mode of transport in the test set, while impacting sensitivity a bit.

# Comparison of all models an Variable Importance

We have applied 5 different approaches on the same dataset to predict the mode of transport employees might take to commute to work -

1. Logistic Regression
2. KNN algorithm
3. Naive Bayes algorithm
4. Boosting
5. Bagging

Let us compare the performance of each model:

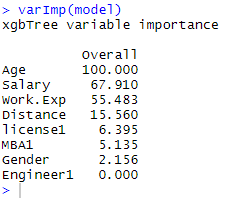
|  |  |  |  |
| --- | --- | --- | --- |
|  | Accuracy | Sensitivity | Specificity |
| Logistic Regression | 95% | 97% | 75% |
| KNN | 92% | 95% | 75% |
| Naive Bayes | 95% | 98% | 69% |
| Boosting | 97% | 99% | 81% |
| Bagging | 91% | 90% | 100% |

It can be observed that the predictive power for all the models remains between 90% - 97%, but the specificity varies largely.

The specificity for Logistic regression, KNN & Naive Bayes is relatively low.

The models after applying boosting and bagging have high specificity, i.e., predict the people taking car to work correctly, but bagging affects the sensitivity and thus the accuracy also. We conclude that the accuracy of the model with boosting is much greater than the Logistic Regression model.

Also, the importance given to variables used in Boosting model are as follows:



The variables Age(32%), Salary(13%), Distance(40%) largely impact the mode of transport an employee would take.

# Actionable insights and Recommendations

Boosting model seems to be a better model as compared to others. This model can be used to identify employees who will probably opt for Car. Outside of the model using the variable Importance we can also identify targeted employees. We understand that Age, Distance and Salary significantly impacts weather and employee will choose Car or other modes of car. In order to filter the probable employees following criteria can be chosen.

• Employees above the age of 30

• Employees with Salary above 20

• Employees commuting more than 12 Km to office

The number of such identified employees can be used as negotiating point for tie-up with Car dealers for rolling out attractive corporate plans.

This also can be used for restructuring salaries such that Car incentives or petrol usage allowance are factored in to give employees better tax-free payout.