**PROBLEM STATEMENT:**

**The attached data has records of 444 employees in a firm. The variables are described below**

**Age: Age of the employee**

**Gender: Gender of the employee**

**Engineer: Whether the employee is an Engineer**

**MBA: Whether the employee is a MBA**

**Work Exp: work experience in completed years**

**Salary: Salary in Rs Lakhs**

**Distance (in Km): The distance between the employee’s residence and office**

**License: Whether the employee has a license**

**Transport: Main mode of transport taken by the employee**

**Build a model that best explains the employee’s decision to use cars as the main means of transport?**

**What would your predictions regarding their choice of transport for the given two employees?**

**INTRODUCTION:**

The problem statement shows that the response variable has three classes.The following algorithms holds good for multi-class classification problems.

1. Linear Discriminant Analysis
2. Support Vector Machine
3. CART
4. K-Nearest Neighbour
5. Naïve Bayes

Here, we have used the four models to find out the best one for the given problem with high accuracy.

**DATA EXPLORATION:**

The data is initially read and the analysis of the variables is done.

car = read.csv("Cars.csv")

str(car)

'data.frame': 444 obs. of 9 variables:

$ Age : int 28 23 29 28 27 26 28 26 22 27 ...

$ Gender : Factor w/ 2 levels "Female","Male": 2 1 2 1 2 2 2 1 2 2 ...

$ Engineer : int 0 1 1 1 1 1 1 1 1 1 ...

$ MBA : int 0 0 0 1 0 0 0 0 0 0 ...

$ Work.Exp : int 4 4 7 5 4 4 5 3 1 4 ...

$ Salary : num 14.3 8.3 13.4 13.4 13.4 12.3 14.4 10.5 7.5 13.5 ...

$ Distance : num 3.2 3.3 4.1 4.5 4.6 4.8 5.1 5.1 5.1 5.2 ...

$ license : int 0 0 0 0 0 1 0 0 0 0 ...

$ Transport: Factor w/ 3 levels "2Wheeler","Car",..: 3 3 3 3 3 3 1 3 3 3 ...

Then, the categorical variables are converted to factor variables.

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

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

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

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

**Imputing Missing Values:**

Then, the variables are checked for missing values.

summary(car)

Age Gender Engineer MBA Work.Exp Salary

Min. :18.00 0:316 0:109 0 :331 Min. : 0.0 Min. : 6.50

1st Qu.:25.00 1:128 1:335 1 :112 1st Qu.: 3.0 1st Qu.: 9.80

Median :27.00 NA's: 1 Median : 5.0 Median :13.60

Mean :27.75 Mean : 6.3 Mean :16.24

3rd Qu.:30.00 3rd Qu.: 8.0 3rd Qu.:15.72

Max. :43.00 Max. :24.0 Max. :57.00

Distance license Transport

Min. : 3.20 0:340 2Wheeler : 83

1st Qu.: 8.80 1:104 Car : 61

Median :11.00 Public Transport:300

Mean :11.32

3rd Qu.:13.43

Max. :23.40

It is evident that the variable MBA has a missing value. Since it is a categorical variable, the missing value can be replaced with mode.

Mode <- function (x, na.rm) {

xtab <- table(x)

xmode <- names(which(xtab == max(xtab)))

if (length(xmode) > 1)

xmode <- ">1 mode"

return(xmode)

}

for (var in 1:ncol(car)) {

if (class(car[,var])=="numeric") {

car[is.na(car[,var]),var] <- mean(car[,var], na.rm = TRUE)

} else if (class(car[,var]) %in% c("character", "factor")) {

car[is.na(car[,var]),var] <- Mode(car[,var], na.rm = TRUE)

}

}

table(car$MBA)

0 1

332 112

**Splitting into training and test data:**

The data is split into training and the test and train data.

set.seed(1234)

split<-sample.split(car$Transport, SplitRatio = 0.70)

train<-subset(car, split == TRUE)

test<-subset(car, split == FALSE)

**LINEAR DISCRIMINANT ANALYSIS:**

When we use the linear discriminant analysis to develop a model for the given data, we get the following output.

library(MASS)

lda.model = lda (Transport~., data=train)

lda.model

Prior probabilities of groups:

2Wheeler Car Public Transport

0.1864952 0.1382637 0.6752412

Group means:

Age Gender1 Engineer1 MBA1 Work.Exp Salary Distance

2Wheeler 25.39655 0.4137931 0.7241379 0.2413793 4.37931 13.26379 12.16897

Car 35.20930 0.2093023 0.8372093 0.1395349 14.97674 34.81628 15.16744

Public Transport 26.64286 0.2809524 0.7238095 0.2904762 4.80000 13.03667 10.19095

license1

2Wheeler 0.2931034

Car 0.7906977

Public Transport 0.1047619

Coefficients of linear discriminants:

LD1 LD2

Age -0.07531569 -0.39108439

Gender1 -0.19367363 0.79865948

Engineer1 -0.12946752 0.01049790

MBA1 0.41433002 -0.32637143

Work.Exp -0.01598952 0.20073936

Salary -0.08007046 -0.02249602

Distance -0.08425613 0.16746531

license1 -1.01833559 1.59742227

Proportion of trace:

LD1 LD2

0.8918 0.1082

* Since the target variable has three classes, LDA produces two discriminant functions(LD1 and LD2). The first discriminant function captures the maximum variance. i.e.**89.18%**
* Here, in this model, we are car as the trying to explain the employee’s decision to use car as the main means of transport.
* So, from the output we could clearly see that the variables which influence the employees to choose car as the means of transport are **Engineer, Age, Salary, Work Experience and license to some extent.**

**Model Accuracy:**

The accuracy for the training data is **79.42%.**

predmodel.car.lda = predict(lda.model, data=train)

table(Predicted=predmodel.car.lda$class, train$Transport)

Predicted 2Wheeler Car Public Transport

2Wheeler 18 1 10

Car 3 31 2

Public Transport 37 11 198

lda<-table(Predicted=predmodel.car.lda$class, train$Transport)

accuracy.lda<-sum(diag(lda))/sum(lda)

accuracy.lda

[1] 0.7942122

The accuracy for the testing data is **75.93%.**

test.lda<-predict(lda.model,newdata=test)

table(Predicted=test.lda$class, test$Transport)

Predicted 2Wheeler Car Public Transport

2Wheeler 6 0 8

Car 0 14 1

Public Transport 19 4 81

lda.test<-table(Predicted=test.lda$class, test$Transport)

accuracy.lda.test<-sum(diag(lda.test))/sum(lda.test)

accuracy.lda.test

[1] 0.7593985

**Prediction for the new data values:**

When we try to predict the mode of transport for the given data values using LDA model, we get **public** **transport** as the predicted mode of transport.

newtest<-read.csv("car\_test.csv",header=TRUE)

newtest$Gender=ifelse(newtest$Gender=="Male",0,1)

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

newtest$Gender=as.factor(newtest$Gender)

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

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

new1= predict(lda.model, newdata=newtest)

new1$class

Public Transport Public Transport

**But, considering the accuracy for the car alone as the means of transport it is 72.09% for the training data and 77.77% for the test data.**

**CART:**

Now, we try to build CART model for the given dataset.

library(rpart)

library(rpart.plot)

r.ctrl<-rpart.control(minsplit=50,minbucket=10,cp=0,xval=10)

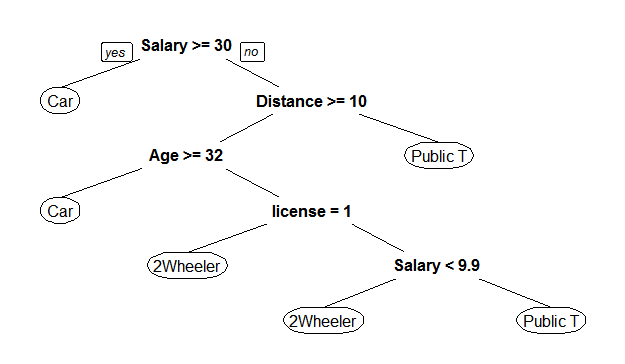
m2<-rpart(formula=Transport ~ .,data=train,method="class",control=r.ctrl)

library(rattle)

library(RColorBrewer)

fancyRpartPlot(m2)

prp(m2)



Based on the CART model, the most important variables that influence the mode of transport are **Salary, Distance, Age and Distance.**

**Model Accuracy:**

The accuracy for the training data is **82.63%.**

train$predict.class<-predict(m2,train,type="class")

train$predict.score<-predict(m2,train,type="prob")

View(train)

ptree<-table(train$Transport,train$predict.class)

accuracy.tree<-sum(diag(ptree))/sum(ptree)

accuracy.tree

[1] 0.8263666

The accuracy for the testing data is **79.69%.**

test$predict.class<-predict(m2,test,type="class")

ptree1<-table(test$Transport,test$predict.class)

accuracy.tree1<-sum(diag(ptree1))/sum(ptree1)

accuracy.tree1

[1] 0.7969925

When we try to predict the mode of transport for the given data values using CART, we get **public** **transport** as the predicted mode of transport.

newtree= predict(m2, newdata=newtest)

newtest$predict.class<-predict(m2,newdata=newtest,type="class")

newtest$predict.score<-predict(m2,newdata=newtest,type="prob")

View(newtest)

newtest$predict.class

[1] Public Transport Public Transport

**But, considering the accuracy for the car alone as the means of transport it is 93.02% for the training data and 94.44% for the test data.**

**K-NEAREST NEIGHBOR:**

**Normalization of Variables:**

For K-Nearest Neighbor and the Support Vector Machine, the data has to be **normalized** before building the model.

normalize<-function(x){

+ +return((x-min(x))/(max(x)-min(x)))}

train<-transform(train, Work.Exp=ave(train$Work.Exp,FUN = normalize))

train<-transform(train, Salary=ave(train$Salary,FUN = normalize))

train<-transform(train, Distance=ave(train$Distance,FUN = normalize))

train<-transform(train, Age=ave(train$Age,FUN = normalize))

test<-transform(test, Work.Exp=ave(test$Work.Exp,FUN = normalize))

test<-transform(test, Salary=ave(test$Salary,FUN = normalize))

test<-transform(test, Distance=ave(test$Distance,FUN = normalize))

test<-transform(test, Age=ave(test$Age,FUN = normalize))

str(train)

'data.frame': 311 obs. of 11 variables:

$ Age : num 0.455 0.227 0.5 0.455 0.364 ...

$ Gender : Factor w/ 2 levels "0","1": 1 2 1 2 1 1 2 1 2 1 ...

$ Engineer : Factor w/ 2 levels "0","1": 1 2 2 2 2 2 2 2 2 2 ...

$ MBA : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 1 1 1 1 ...

$ Work.Exp : num 0.182 0.182 0.318 0.227 0.182 ...

$ Salary : num 0.1545 0.0356 0.1366 0.1366 0.1149 ...

$ Distance : num 0 0.00495 0.04455 0.06436 0.07921 ...

$ license : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 1 1 1 ...

$ Transport : Factor w/ 3 levels "2Wheeler","Car",..: 3 3 3 3 3 1 3 3 3 3

train$Male<-ifelse(train$Gender=="0",1,0)

train$Eng<-ifelse(train$Engineer=="1",1,0)

train$lic<-ifelse(train$lic=="1",1,0)

train$Masters<-ifelse(train$MBA=="1",1,0)

test$Male<-ifelse(test$Gender=="0",1,0)

test$Eng<-ifelse(test$Engineer=="1",1,0)

test$lic<-ifelse(test$lic=="1",1,0)

test$Masters<-ifelse(test$MBA=="1",1,0)

After Normalization, we have developed a model without the categorical variables as normalization doesn’t have a great effect on them.

train.2fact<-train[,c(1,5,6,7,9)]

val.2fact<-test[,c(1,5,6,7,9)]

library(MASS)

library(class)

y\_pred<-knn(train=train.2fact[,-5],test=val.2fact[-5], cl=train.2fact[,5],k=23)

cm.knn<-table(val.2fact[,5],y\_pred)

cm.knn

y\_pred

2Wheeler Car Public Transport

2Wheeler 4 0 21

Car 0 14 4

Public Transport 5 1 84

accuracy.cm.knn<-sum(diag(cm.knn))/sum(cm.knn)

accuracy.cm.knn

[1] 0.7669173

The accuracy of the model with only the normalized numerical variables is **76.69%.**

Next, we develop the full model with all the variables.

##full model

train.count<-train[,c(1,5,6,7,9,10,11,12,13)]

test.count<-test[,c(1,5,6,7,9,10,11,12,13)]

View(test.count)

y\_pred1<-knn(train=train.count[,-5],test=test.count[-5], cl=train.count[,5],k=23)

tab.knn.3<-table(test.count[,5],y\_pred1)

tab.knn.3

y\_pred1

2Wheeler Car Public Transport

2Wheeler 1 0 24

Car 0 13 5

Public Transport 0 0 90

accuracy.knn.3<-sum(diag(tab.knn.3))/sum(tab.knn.3)

accuracy.knn.3

[1] 0.7819549

The accuracy of the full model with all the variables is **78.19%.**

y\_p<-knn(train=test.count[,-5],test=newtest1, cl=test.count[,5],k=3)

y\_p

[1] Car Car

When we try to predict the mode of transport for the given data values using KNN, we get **car** as the predicted mode of transport unlike the other two models.

**But, considering the accuracy for the car alone as the means of transport it is 77.77% for the model with only the numerical variables and 72.22% for the model with all the variables.**

**SUPPORT VECTOR MACHINE:**

For support vector machine, we have to normalize the variables before we build up the model.

svm1 <- svm(Transport~., data=train.2fact,

method="C-classification",probability=TRUE, cost=10)

summary(svm1)

Call:

svm(formula = Transport ~ ., data = train.2fact, method = "C-classification",

probability = TRUE, cost = 10)

Parameters:

SVM-Type: C-classification

SVM-Kernel: radial

cost: 10

gamma: 0.125

Number of Support Vectors: 153

( 80 51 22 )

Number of Classes: 3

Levels:

2Wheeler Car Public Transport

**Model Accuracy:**

The model accuracy considering all the variables is **76.69%.**

prediction <- predict(svm1, val.2fact)

xtab <- table(val.2fact$Transport, prediction)

xtab

prediction

2Wheeler Car Public Transport

2Wheeler 11 0 14

Car 1 14 3

Public Transport 12 1 77

accuracy.svm<-sum(diag(xtab))/sum(xtab)

accuracy.svm

[1] 0.7669173

newtest1<-newtest[,c(1,5,6,7,9,10,11,12)]

newsvm= predict(svm1, newdata=newtest1)

newsvm

1 2

Public Transport Public Transport

When we try to predict the mode of transport for the given data values using SVM, we get **public transport** as the predicted mode of transport.

**But, considering the accuracy for the car alone as the means of transport it is 77.77%.**

**NAIVE BAYES ALGORITHM:**

Naïve Bayes has little importance in the multiclass classification problem.

library(caret)

> library(e1071)

> NBclassifier=naiveBayes(Transport~., data=train)

> print(NBclassifier)

Naive Bayes Classifier for Discrete Predictors

Call:

naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:

Y

2Wheeler Car Public Transport

0.1864952 0.1382637 0.6752412

Conditional probabilities:

Age

Y [,1] [,2]

2Wheeler 25.39655 3.195060

Car 35.20930 3.240798

Public Transport 26.64286 3.005748

Gender

Y 0 1

2Wheeler 0.5862069 0.4137931

Car 0.7906977 0.2093023

Public Transport 0.7190476 0.2809524

Engineer

Y 0 1

2Wheeler 0.2758621 0.7241379

Car 0.1627907 0.8372093

Public Transport 0.2761905 0.7238095

MBA

Y 0 1

2Wheeler 0.7586207 0.2413793

Car 0.8604651 0.1395349

Public Transport 0.7081340 0.2918660

Work.Exp

Y [,1] [,2]

2Wheeler 4.37931 3.626547

Car 14.97674 4.847623

Public Transport 4.80000 3.105482

Salary

Y [,1] [,2]

2Wheeler 13.26379 6.815815

Car 34.81628 12.720735

Public Transport 13.03667 4.778795

Distance

Y [,1] [,2]

2Wheeler 12.16897 3.450309

Car 15.16744 3.851389

Public Transport 10.19095 3.014203

license

Y 0 1

2Wheeler 0.7068966 0.2931034

Car 0.2093023 0.7906977

Public Transport 0.8952381 0.1047619

The Y values are the means and the standard deviations of the predictors within each class.

**Model Accuracy:**

The model accuracy for the training data is **78.77%.**

trainPred=predict(NBclassifier, newdata = train, type = "class")

NaiveBayes=table(train$Transport,trainPred)

NaiveBayes

trainPred

2Wheeler Car Public Transport

2Wheeler 13 4 41

Car 0 35 8

Public Transport 5 8 197

accuracy.NaiveBayes<-sum(diag(NaiveBayes))/sum(NaiveBayes)

accuracy.NaiveBayes

[1] 0.7877814

The model accuracy for the testing data is **78.19%.**

trainPred1=predict(NBclassifier, newdata = test, type = "class")

NaiveBayes1=table(test$Transport,trainPred1)

NaiveBayes1

trainPred1

2Wheeler Car Public Transport

2Wheeler 4 0 21

Car 0 15 3

Public Transport 3 2 85

accuracy.NaiveBayes1<-sum(diag(NaiveBayes1))/sum(NaiveBayes1)

accuracy.NaiveBayes1

0.7819549

Pred1=predict(NBclassifier, newdata = newtest, type = "class")

Pred1

[1] Public Transport Public Transport

When we try to predict the mode of transport for the given data values using Naïve Bayes, we get **public transport** as the predicted mode of transport.

**But, considering the accuracy for the car alone as the means of transport it is 81.39%.**

**COMPARISION OF MODELS:**

When we compare the accuracy of the models,

|  |  |
| --- | --- |
| **Model** | **Overall Accuracy** |
| Linear Discriminant Analysis | 79.42% |
| CART | 89.63% |
| K-Nearest Neighbor | 78.19% |
| Support Vector Machine | 76.69% |
| Naïve Bayes | 78.77% |

When we try to obtain the accuracy of the models in prediction of car as a mode of transport, the models provide following accuracy:

|  |  |
| --- | --- |
| **Model** | **Accuracy in Prediction of Car as a means of Transport** |
| Linear Discriminant Analysis | 72.09% |
| CART | 93.03% |
| K-Nearest Neighbor | 77.77% |
| Support Vector Machine | 77.77% |
| Naïve Bayes | 81.39% |

From the accuracy values, **CART** could be the best model for this dataset**.**

In predicting the mode of transport for the two given records,

|  |  |  |
| --- | --- | --- |
| **Model** | **Prediction of mode of transport-Record1** | **Prediction of mode of transport-Record2** |
| Linear Discriminant Analysis | Public Transport | Public Transport |
| CART | Public Transport | Public Transport |
| K-Nearest Neighbor | Car | Car |
| Support Vector Machine | Public Transport | Public Transport |
| Naïve Bayes | Public Transport | Public Transport |

So, we could conclude that **Public transport** could be the preferred mode for the given data values.