Travel Insurance Claim Prediction Analysis

Bamidele Tella

9/5/2021

Overview

Travel insurance is a type of insurance that covers the costs and losses associated with traveling. It is useful protection for those traveling domestically or abroad.

Many companies selling tickets or travel packages, give consumers the option to purchase travel insurance, also known as travelers insurance. Some travel policies cover damage to personal property, rented equipment, such as rental cars, or even the cost of paying a ransom.

Problem Statement

As a data scientist in an insurance company in the USA. The company has collected the data of earlier travel insurance buyers. In this season of vacation, the company wants to know which person will claim their travel insurance and who will not. The company has chosen you to apply your Machine Learning knowledge and provide them with a model that achieves this vision.

Objective

You are responsible for building a machine learning model for the insurance company to predict if the insurance buyer will claim their travel insurance or not.

Evaluation Criteria Submissions are evaluated using F1 Score.

Package Importation

First, we import the required packages, that would aid this analysis

```
library(caret)
library(rpart)
library(randomForest)
library(ggcorrplot)
library(ggplot2)
library(GGally)
library(data.table)
library(MLmetrics)
```

Loading Dataset

First,I load the data set from my local directory.

```
train <- read.csv("~/R Studio/Dataset/Churn Data/train.csv")
test <- read.csv("~/R Studio/Dataset/Churn Data/test.csv")
str(train)
```

```
## 'data.frame':    48260 obs. of  11 variables:
##  $ Agency              : chr  "CWT" "EPX" "EPX" "C2B" ...
##  $ Agency.Type         : chr  "Travel Agency" "Travel Agency" "Travel Agency" "Airlines" ...
##  $ Distribution.Channel: chr  "Online" "Online" "Online" "Online" ...
##  $ Product.Name        : chr  "Rental Vehicle Excess Insurance" "Cancellation Plan" "2 way Comprehens
##  $ Duration            : int  61 93 22 14 90 36 13 4 95 30 ...
##  $ Destination         : chr  "UNITED KINGDOM" "NEW ZEALAND" "UNITED STATES" "SINGAPORE" ...
##  $ Net.Sales           : num  19.8 63 22 54.5 10 47 25 27 20 10 ...
##  $ Commision..in.value.: num  11.9 0 0 13.6 0 ...
##  $ Gender              : chr  "" "" "" "M" ...
##  $ Age                 : int  29 36 25 24 23 36 36 35 36 36 ...
##  $ Claim               : int  0 0 0 0 0 0 0 0 0 0 ...
```

Cleaning the Data

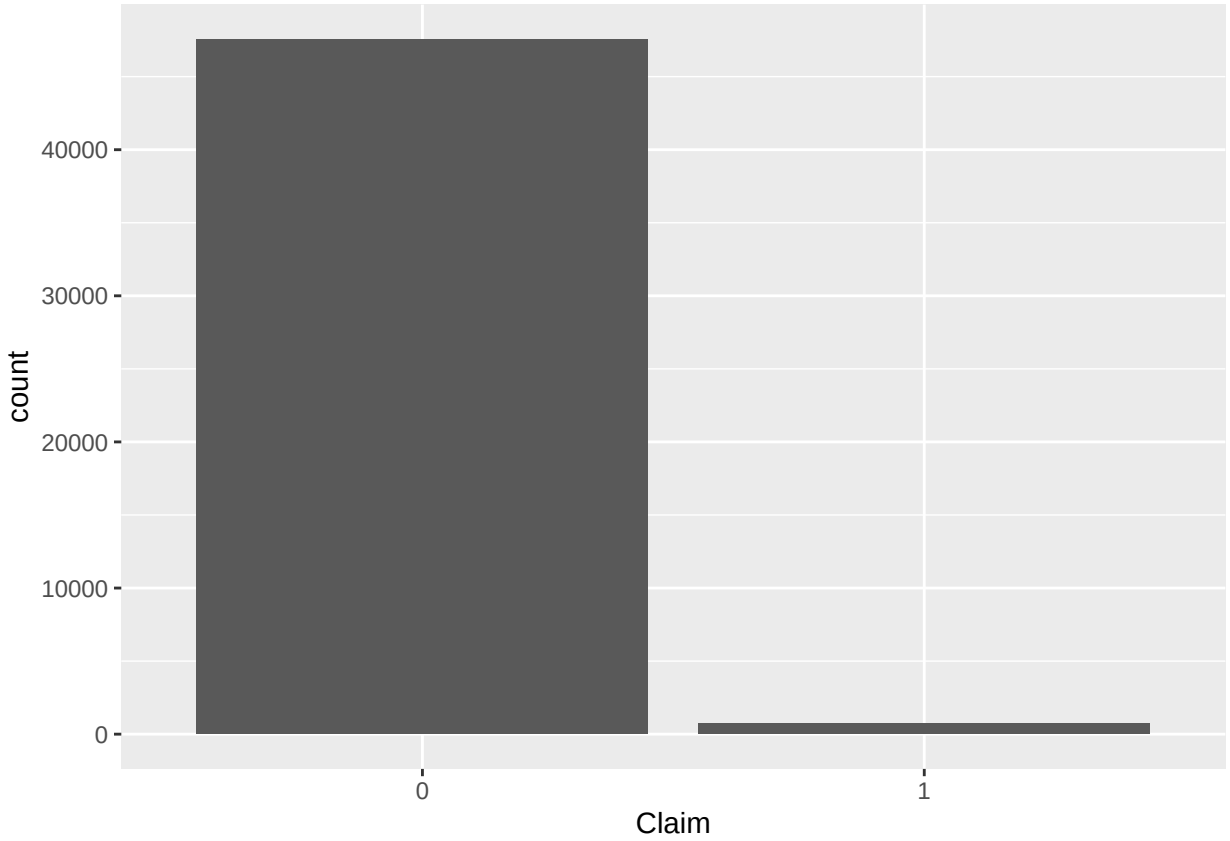
Handling Missing values

For the gender, I name the missing values 'Unspecified'. I make some reassignment during analysis to serve as a restore point. Then I perform some Exploratory Data Analysis to view what we are working with.

```
train$Gender[train$Gender==""] <- "Unspecified"
test$Gender[test$Gender==""] <- "Unspecified"

trainData2 <- train
testData2 <- test
print(table(trainData2$Claim))
```

```
##
##     0     1
## 47552   708
```

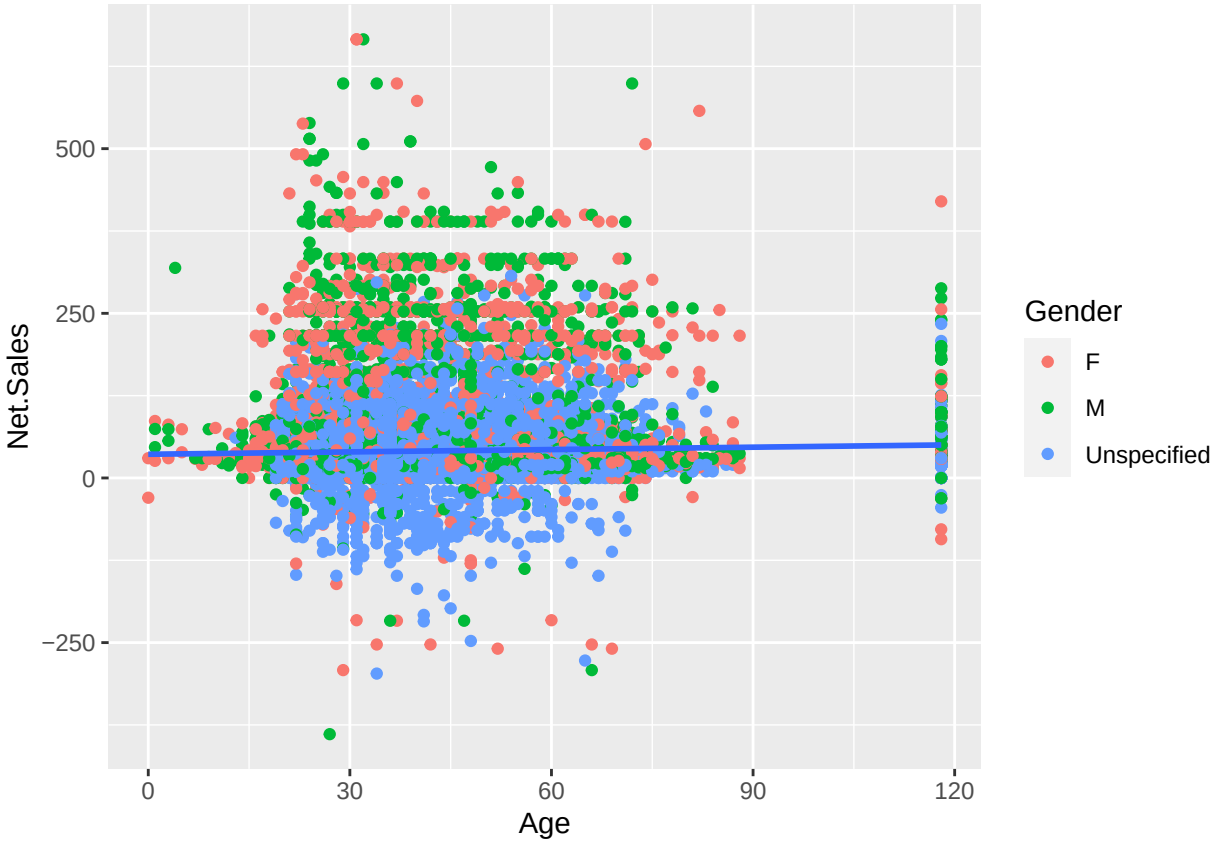
```
trainData2$Claim <- as.factor(trainData2$Claim)
trainData2$Gender <- as.factor(trainData2$Gender)

ggplot1 <- ggplot(trainData2, aes(Claim)) + geom_histogram(stat="count")
ggplot1
```



From the above plot, we can see that our target column is imbalanced. Now I take a look at our Data with the different individual ages.

Next I view the Net Sales according to individual age range.

```
ggplot2 = ggplot(trainData2,aes(Age, Net.Sales)) + geom_point(aes(col=Gender))
ggplot2 = ggplot2 + geom_smooth(method = lm)
ggplot2
```



We can see that the people within the age range of about 25 to about 80 are most likely going to purchase a travel insurance. Most importantly, we see that there is a wild range of outliers in the data which needs to be amended.

Feature Selection

I decided to remove the Agency Type, Distribution Channel, Product Name and Destination as I did not see how it affected a customer claiming an insurance.

```
trainData2 <- trainData2[, -c(2,3,4,6)]
testData2 <- testData2[, -c(2,3,4,6)]
head(trainData2)
```

```
##   Agency Duration Net.Sales Commision..in.value.      Gender Age Claim
## 1    CWT       61      19.8                11.88 Unspecified  29     0
## 2    EPX       93      63.0                 0.00 Unspecified  36     0
## 3    EPX       22      22.0                 0.00 Unspecified  25     0
## 4    C2B       14      54.5                13.63           M  24     0
## 5    EPX       90      10.0                 0.00 Unspecified  23     0
## 6    EPX       36      47.0                 0.00 Unspecified  36     0
```

Next I perform some feature engineering like converting categorical data to numeric values that can be interpreted by R.

```
unique_agency <- unique(trainData2$Agency)
unique_gender <- unique(trainData2$Gender)
label_matrix <- matrix(0,nrow = nrow(trainData2),ncol = length(c(unique_agency,unique_gender)))
colnames(label_matrix) <- c(unique_agency,as.character(unique_gender))
label_matrix <- as.data.frame(label_matrix)

train_labels <- cbind(trainData2,label_matrix)

for (i in 1:nrow(trainData2)) {
  for (j in colnames(train_labels)){
    if(train_labels[i,1]==j){
      train_labels[i,j] <- 1
    }
  }
}

for (i3 in 1:nrow(trainData2)){
  for (j3 in colnames(train_labels)){
    if(train_labels[i3, 5]==j3){
      train_labels[i3,j3] <- 1
    }
  }
}

head(train_labels)
```

```
##   Agency Duration Net.Sales Commision..in.value.      Gender Age Claim CWT EPX
## 1    CWT       61      19.8                11.88 Unspecified  29     0   1   0
## 2    EPX       93      63.0                 0.00 Unspecified  36     0   0   1
## 3    EPX       22      22.0                 0.00 Unspecified  25     0   0   1
## 4    C2B       14      54.5                13.63           M  24     0   0   0
## 5    EPX       90      10.0                 0.00 Unspecified  23     0   0   1
## 6    EPX       36      47.0                 0.00 Unspecified  36     0   0   1
##   C2B JZI TST ART RAB SSI JWT CCR LWC KML TTW CSR ADM CBH Unspecified M F
## 1   0   0   0   0   0   0   0   0   0   0   0   0   0   0           1 0 0
## 2   0   0   0   0   0   0   0   0   0   0   0   0   0   0           1 0 0
## 3   0   0   0   0   0   0   0   0   0   0   0   0   0   0           1 0 0
## 4   1   0   0   0   0   0   0   0   0   0   0   0   0   0           0 1 0
## 5   0   0   0   0   0   0   0   0   0   0   0   0   0   0           1 0 0
## 6   0   0   0   0   0   0   0   0   0   0   0   0   0   0           1 0 0
```

```
train_labels2 <- train_labels
train_labels2 <- train_labels2[,-c(1,5)]  # The relabeled columns were removed (Agency Type, Gender).
```

Handling Outliers

Next, I replaced outliers with values in between the first and third quadrant in the Duration and Net Sales columns. The values were gotten from the statistical summary of the original data.

```
for(i4 in 1:nrow(train_labels2)){
  if(train_labels2[i4,1]<9.00){
    train_labels2[i4,1] <- 9.00
```

```
  }else if(train_labels2[i4,1]>53.00){
    train_labels2[i4,1] <- 53.00
  }else{

  }
}

for(i4 in 1:nrow(train_labels2)){
  if(train_labels2[i4,4]<18){
    train_labels2[i4,4] <- 18
  }else if(train_labels2[i4,4]>85){
    train_labels2[i4,4] <- 85
  }else{

  }
}

for(i5 in 1:nrow(train_labels2)){
  if(train_labels2[i5,2]<18.00){
    train_labels2[i5,2] <- 18.00
  }else if(train_labels2[i5,2]>48.00){
    train_labels2[i5,2] <- 48.00
  }else{
  }
}
```

Then I convert the Target column to category(Factor),Age column to Numeric, Agencies and Genders columns into Categorical Columns indicating 1 for 'Yes',and 0 for 'No'.

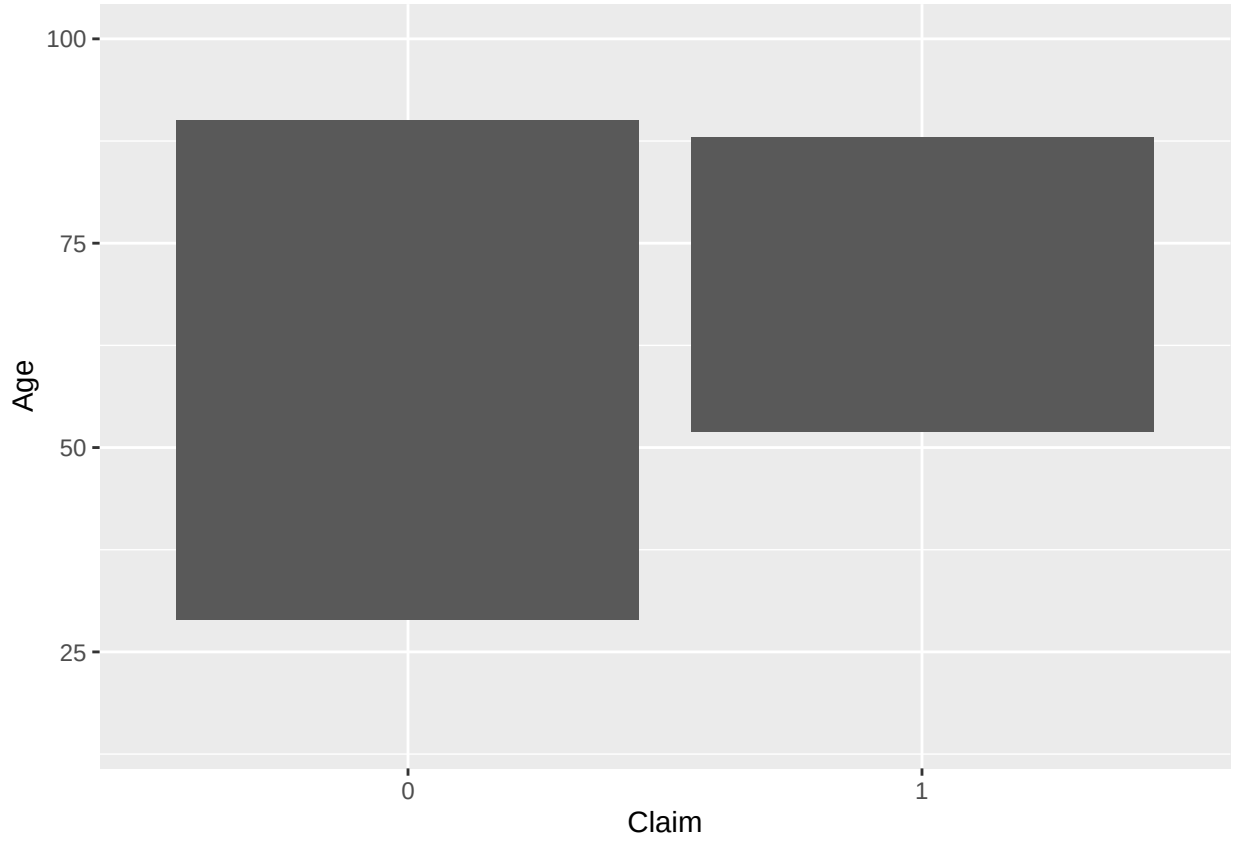
```
train_labels2$Age <- as.numeric(train_labels2$Age)
colnames(train_labels2) <- make.names(colnames(train_labels2),unique = T)

train_labels3 <- train_labels2

for(i6 in c(6:ncol(train_labels3))){
  train_labels3[,i6] <- as.factor(train_labels3[,i6])
}
summary(train_labels3)
```

```
##     Duration       Net.Sales     Commision..in.value.      Age        Claim
##  Min.   : 9.00   Min.   :18.00   Min.   :  0.000      Min.   :18.00   0:47552
##  1st Qu.: 9.00   1st Qu.:18.00   1st Qu.:  0.000      1st Qu.:35.00   1:  708
##  Median :22.00   Median :27.00   Median :  0.000      Median :36.00
##  Mean   :28.08   Mean   :30.62   Mean   :  9.812      Mean   :39.42
##  3rd Qu.:53.00   3rd Qu.:48.00   3rd Qu.: 11.630      3rd Qu.:43.00
##  Max.   :53.00   Max.   :48.00   Max.   :262.760      Max.   :85.00
##  CWT       EPX       C2B       JZI       TST       ART       RAB
##  0:41688   0:21548   0:41980   0:43409   0:47871   0:48012   0:47683
##  1: 6572   1:26712   1: 6280   1: 4851   1:  389   1:  248   1:  577
##
##
##
##
```

```
##  SSI       JWT       CCR       LWC       KML       TTW       CSR
##  0:47453   0:47680   0:48105   0:47728   0:47967   0:48188   0:48194
##  1:  807   1:  580   1:  155   1:  532   1:  293   1:   72   1:   66
##
##
##
##
##  ADM       CBH       Unspecified M         F
##  0:48204   0:48190   0:13899     0:41123   0:41498
##  1:   56   1:   70   1:34361     1: 7137   1: 6762
##
##
##
##
```

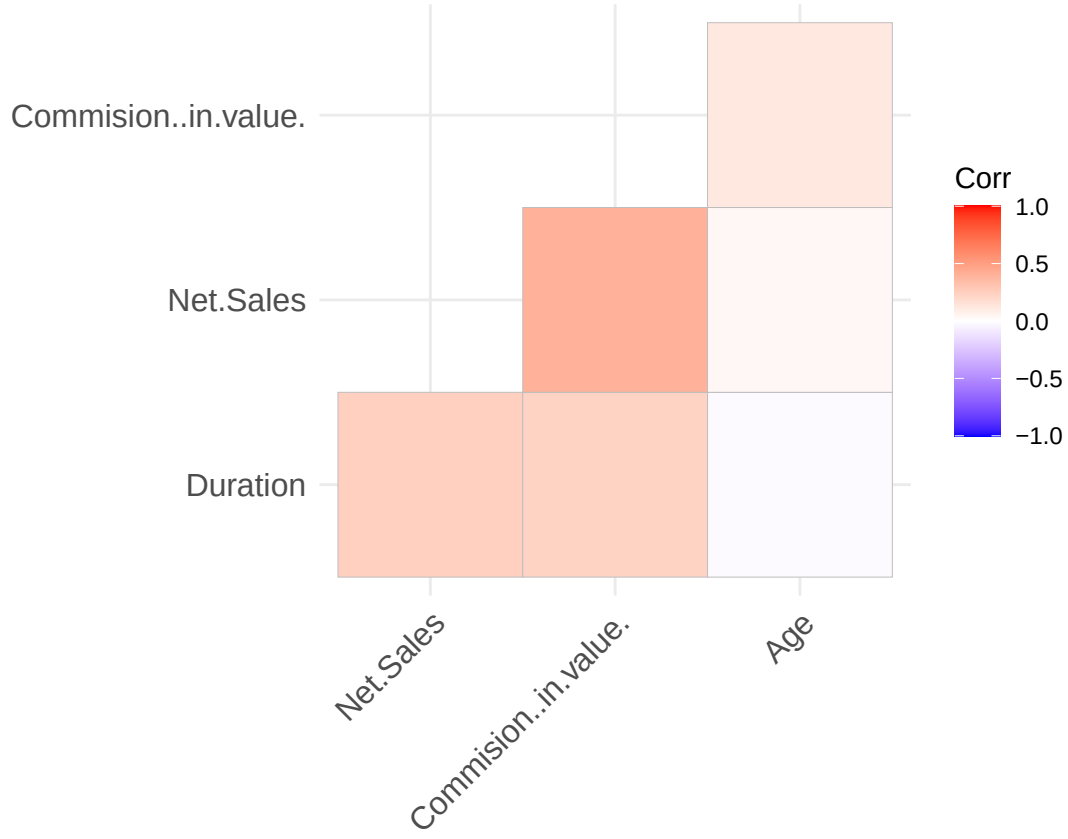
Now we take a look at what age range are more likely to claim the insurance and how the target column imbalance affects the different Gender of people that claimed their travel insurance.

```
ggplot3 <-  ggplot(train_labels3, aes(Claim,Age), fill= Claim)+ geom_bar(stat='identity') + ylim(c(15,
ggplot3
```



Next, I take a look at the correlation between Duration of the insurance purchased by the different individuals, Net Sales of Insurance, Commission in value of each individual, and the different Ages of the different individuals.

```
corr = cor(train_labels3[c(1,2,3,4)])
ggcorrplot(corr, method = "square", type="lower")
```



Modelling and Evaluation

Next, I develop multiple models and decide the best using the Confusion Matrix and F1_Score.

Data Split into Training and Evaluation Set

```
set.seed(12)
intrain <- createDataPartition(train_labels3$Claim, p=0.6,list = F)
trainer <- train_labels3[intrain,]
val <- train_labels3[-intrain,]
```

Decision Tree

```
fit.rpart <- train(Claim~.,data=trainer,method="rpart")
pred.rpart <- predict(fit.rpart, newdata=val)
confusionMatrix(pred.rpart,val$Claim)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 19020   283
##          1     0     0
##
##                Accuracy : 0.9853
##                  95% CI : (0.9835, 0.987)
##     No Information Rate : 0.9853
##     P-Value [Acc > NIR] : 0.5158
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.0000
##             Specificity : 0.0000
##          Pos Pred Value : 0.9853
##          Neg Pred Value :    NaN
##              Prevalence : 0.9853
##          Detection Rate : 0.9853
##    Detection Prevalence : 1.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : 0
##
```

```
F1_Score(y_true = val$Claim, y_pred = pred.rpart)
```

```
## [1] 0.9926154
```

Random Forest

```
fit.rf2 <- randomForest(Claim ~ ., data=trainer, proximity = F)
rfpred3 <- predict(fit.rf2, newdata=val, type = "response")
confusionMatrix(rfpred3,val$Claim)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 19020   283
##          1     0     0
##
##                Accuracy : 0.9853
##                  95% CI : (0.9835, 0.987)
##     No Information Rate : 0.9853
##     P-Value [Acc > NIR] : 0.5158
##
##                   Kappa : 0
```

```
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.0000
##             Specificity : 0.0000
##          Pos Pred Value : 0.9853
##          Neg Pred Value :    NaN
##              Prevalence : 0.9853
##          Detection Rate : 0.9853
##    Detection Prevalence : 1.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : 0
##
```

```
F1_Score(y_true = val$Claim,y_pred = rfpred3)
```

```
## [1] 0.9926154
```

Generalised Linear Model

```
gfit3 <- glm(Claim ~ ., data = trainer, family = "binomial"(link = 'logit'))
gpred2 <- predict(gfit3, newdata = val, type = "response")
table(val$Claim, gpred2 >= 0.5)
```

```
##
##     FALSE
##   0 19020
##   1   283
```

Latent Diriclet Allocation Model

```
lda.fit <- train(Claim~.,method="lda",data=trainer)
pred.lda <- predict(lda.fit,newdata=val)
confusionMatrix(pred.lda,val$Claim )
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 18779   250
##          1   241    33
##
##                Accuracy : 0.9746
##                  95% CI : (0.9722, 0.9767)
##     No Information Rate : 0.9853
##     P-Value [Acc > NIR] : 1.0000
##
```

```
##                   Kappa : 0.1056
##
##  Mcnemar's Test P-Value : 0.7181
##
##             Sensitivity : 0.9873
##             Specificity : 0.1166
##          Pos Pred Value : 0.9869
##          Neg Pred Value : 0.1204
##              Prevalence : 0.9853
##          Detection Rate : 0.9729
##    Detection Prevalence : 0.9858
##       Balanced Accuracy : 0.5520
##
##        'Positive' Class : 0
##
```

```
F1_Score(y_true = val$Claim, y_pred=pred.lda)
```

```
## [1] 0.9870956
```

Prediction of Test Data

The Latent Dirichlet Allocation (LDA) model among all models has a better sensitivity to the data set's imbalance and its also more specific. Hence, we use this model for prediction of the test data. To attain reasonable results, we must clean up the test data and perform feature engineering as we did for the train data. Then we predict with our resulting test Data.

Feature Selection

```
test_agency <- unique(testData2$Agency)
test_gender <- unique(testData2$Gender)
test_matrix <- matrix(0, nrow = nrow(testData2), ncol = length(c(test_agency,test_gender)))
colnames(test_matrix) <- c(test_agency,test_gender)
test_matrix <- as.data.frame(test_matrix)
test_encode <- cbind(testData2,test_matrix)

for (u in 1:nrow(testData2)) {
  for (v in colnames(test_encode)){
    if(test_encode[u,1]==v){
      test_encode[u,v] <- 1
    }
  }
}
for (u3 in 1:nrow(testData2)) {
  for (v3 in colnames(test_encode)){
    if(test_encode[u3,5]==v3){
      test_encode[u3,v3] <- 1
    }
  }
}
```

```
test_encode2 <- test_encode
test_encode2 <- test_encode2[,-c(1,5)]
summary(test_encode2)
```

```
##     Duration        Net.Sales       Commision..in.value.      Age
##  Min.   :  -1.0   Min.   :-357.50   Min.   :  0.000      Min.   :  1.00
##  1st Qu.:   9.0   1st Qu.:  18.00   1st Qu.:  0.000      1st Qu.: 35.00
##  Median :  22.0   Median :  26.00   Median :  0.000      Median : 36.00
##  Mean   :  48.6   Mean   :  40.62   Mean   :  9.791      Mean   : 40.08
##  3rd Qu.:  53.0   3rd Qu.:  48.00   3rd Qu.: 10.640      3rd Qu.: 44.00
##  Max.   :4784.0   Max.   : 810.00   Max.   :283.500      Max.   :118.00
##       EPX              CWT              C2B              TST
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.000000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000000
##  Median :1.0000   Median :0.0000   Median :0.0000   Median :0.000000
##  Mean   :0.5575   Mean   :0.1338   Mean   :0.1323   Mean   :0.008969
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.000000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.000000
##       JZI               RAB               KML               JWT
##  Min.   :0.00000   Min.   :0.000000   Min.   :0.000000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:0.000000   1st Qu.:0.00000
##  Median :0.00000   Median :0.000000   Median :0.000000   Median :0.00000
##  Mean   :0.09872   Mean   :0.009853   Mean   :0.006316   Mean   :0.01105
##  3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:0.000000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.000000   Max.   :1.000000   Max.   :1.00000
##       CBH                ART               LWC               TTW
##  Min.   :0.000000   Min.   :0.000000   Min.   :0.00000   Min.   :0.000000
##  1st Qu.:0.000000   1st Qu.:0.000000   1st Qu.:0.00000   1st Qu.:0.000000
##  Median :0.000000   Median :0.000000   Median :0.00000   Median :0.000000
##  Mean   :0.002084   Mean   :0.005558   Mean   :0.01049   Mean   :0.001642
##  3rd Qu.:0.000000   3rd Qu.:0.000000   3rd Qu.:0.00000   3rd Qu.:0.000000
##  Max.   :1.000000   Max.   :1.000000   Max.   :1.00000   Max.   :1.000000
##       CCR               SSI               CSR               ADM
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.000000   Min.   :0.000000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:0.000000
##  Median :0.00000   Median :0.00000   Median :0.000000   Median :0.000000
##  Mean   :0.00259   Mean   :0.01617   Mean   :0.001326   Mean   :0.001642
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:0.000000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.000000   Max.   :1.000000
##   Unspecified            M                F
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.00
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00
##  Median :1.0000   Median :0.0000   Median :0.00
##  Mean   :0.7129   Mean   :0.1472   Mean   :0.14
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.00
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.00
```

Handling Outliers

```
for(u4 in 1:nrow(test_encode2)){
  if(test_encode2[u4,1]<9.00){
```

```
    test_encode2[u4,1] <- 9.00
  }else if(test_encode2[u4,1]>53.00){
    test_encode2[u4,1] <- 53.00
  }else{

  }
}

for(i4 in 1:nrow(test_encode2)){
  if(test_encode2[i4,4]<18){
    test_encode2[i4,4] <- 18
  }else if(test_encode2[i4,4]>85){
    test_encode2[i4,4] <- 85
  }else{

  }
}

for(u5 in 1:nrow(test_encode2)){
  if(test_encode2[u5,2]<18.00){
    test_encode2[u5,2] <- 18.00
  }else if(test_encode2[u5,2]>48.00){
    test_encode2[u5,2] <- 48.00
  }else{

  }
}
```

Classification and Prediction

```
test_encode2$Age <- as.numeric(test_encode2$Age)
colnames(test_encode2) <- make.names(colnames(test_encode2),unique = T)
test_encode3 <- test_encode2
for(u6 in c(5:ncol(test_encode3))) {
  test_encode3[,u6] <- as.factor(test_encode3[,u6])
}
```

Final Prediction

```
finalpred2 <- predict(lda.fit,newdata=test_encode3)
finalpred2 <- as.data.frame(finalpred2,stringsAsFactors=F )
colnames(finalpred2) <- "prediction"
table(finalpred2$prediction)
```

```
##
##     0     1
## 15615   217
```

```
write.csv(finalpred2,"./Final_Prediction2.csv",row.names = F)
```