Experimental Design

Venukanan Subenthiran

2022-11-19

Experimental Design

Splitting the Data into Training and Test Sets

```
#install.packages("caTools")
library(caTools)



set.seed(123)

TrainTestData <- sample.split(Y = fraudTotal.db$is_fraud, SplitRatio = 0.7)
FD_Train <- fraudTotal.db[TrainTestData,]
FD_Test <- fraudTotal.db[!TrainTestData,]

#converting dob to numeric
FD_Train$dob <- as.numeric(FD_Train$dob)
FD_Test$dob <- as.numeric(FD_Test$dob)


#convertig trans_date_trans_time to numeric
FD_Train$trans_date_trans_time <- as.numeric(as.POSIXct(FD_Train$trans_date_trans_time))
FD_Test$trans_date_trans_time <- as.numeric(as.POSIXct(FD_Test$trans_date_trans_time))



#If there is a need to convert back then need to figure out the way to do that.
###FD_Train$trans_date_trans_time <- strptime(FD_Train$trans_date_trans_time, format = "%Y%m%d %
H:%M:%s")
###FD_Train_trial2 <- as.numeric(FD_Train$trans_date_trans_time)

###FD_Train$trans_date_trans_time <- as.POSIXct(format(FD_Train$trans_date_trans_time))
###as.Date.POSIXct(FD_Train$trans_date_trans_time, origin = "1970-01-01 00:00")



#This will convert dob back to date.
###FD_Train$dob <- as.Date(FD_Train$dob, origin = "1970-01-01")
```

Treatment for Imbalance Data using ROSE

```
#install.packages("ROSE")
library(ROSE)
```

```
## Loaded ROSE 0.0-4
```

```
table(FD_Train$is_fraud)
```

```
##
##       0       1
## 1289920    6756
```

```
FD_ROSE <- ROSE(formula = is_fraud~.,data = FD_Train, seed = 345)

table(FD_ROSE$data$is_fraud)
```

```
##
##      0      1
## 647554 649122
```

```
FD_Train_ROSE <- FD_ROSE$data

Reduced_FD_Train_ROSE <- subset(FD_Train_ROSE, select = c(2, 5, 6, 10, 11, 13, 14, 18, 20, 22, 2
3))
```

Cross Validation???

Modeling

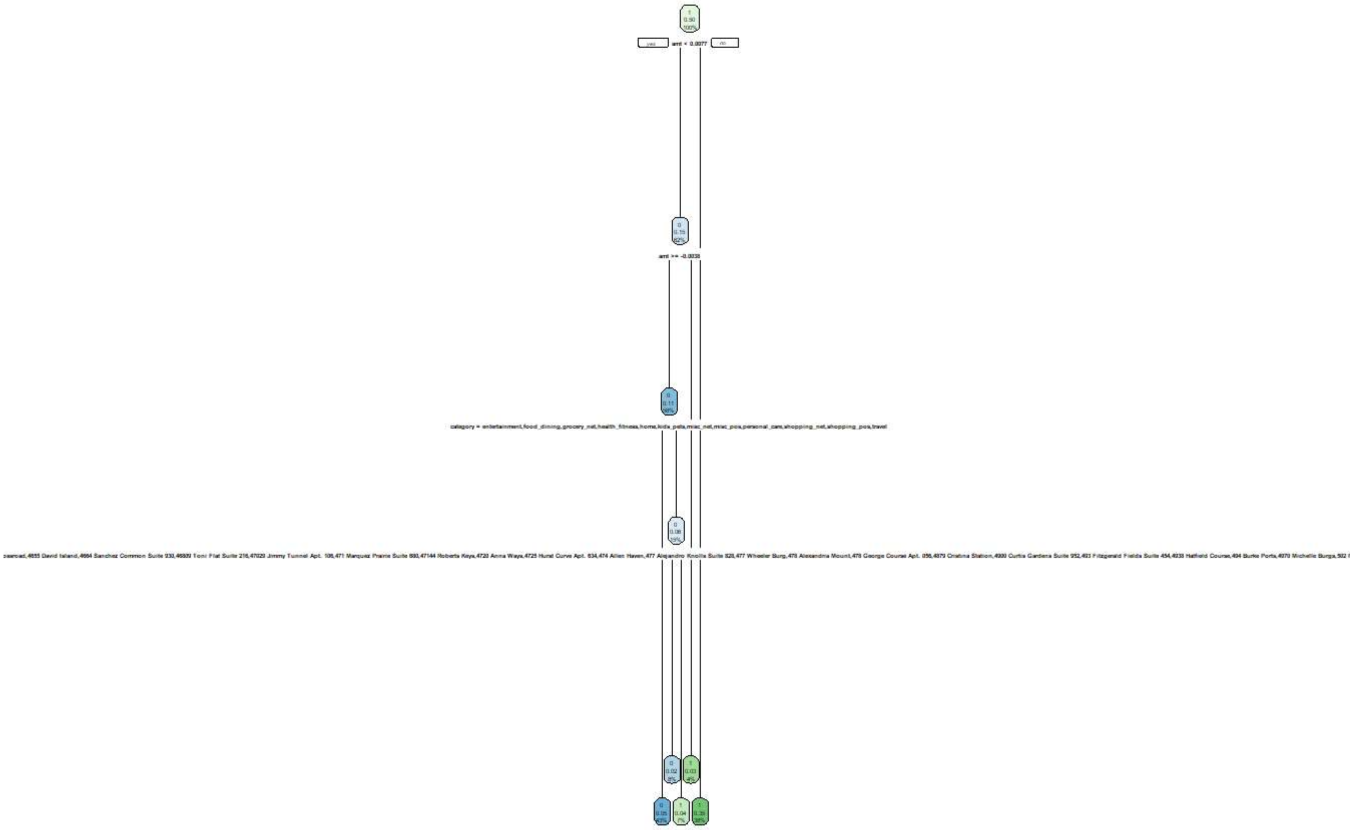
Classification

Decision Tree

```
#install.packages("rpart.plot")
library(rpart.plot)
```

```
## Loading required package: rpart
```

```
#Train with model
fit_tree <- rpart(is_fraud~., data = Reduced_FD_Train_ROSE, method = "class")
rpart.plot(fit_tree, extra = 110)
```



```
# Prediction
predict_tree <- predict(fit_tree, FD_Test, type = "class")
```

Testing merging of categorical variables

Random Forest

K-Nearest Neighbours

Regression

Logistic Regression biraviate??

Evaluation

Confusion Matrix

Decision Tree

```
confusionMatrix_DS <- table(FD_Test$is_fraud, predict_tree)
```

Random Forest

K-Nearest Neighbours

Logistic Regression

Accuracy, Recall, and Precision

Accuracy, Recall, and Precision for Decision Tree

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
confusionMatrix_DS
```

```
##    predict_tree
##          0      1
##   0 504267  48556
##   1    612   2283
```

```
FD_Test$is_fraud <- as.factor(FD_Test$is_fraud)

#Accuracy
accuracy.meas(FD_Test$is_fraud, predict_tree)
```

```
##
## Call:
## accuracy.meas(response = FD_Test$is_fraud, predicted = predict_tree)
##
## Examples are labelled as positive when predicted is greater than 0.5
##
## precision: 0.005
## recall: 1.000
## F: 0.005
```

```
# Recall
recall(FD_Test$is_fraud, predict_tree)
```

```
## [1] 0.9987878
```

```
# Precision
precision(FD_Test$is_fraud, predict_tree)
```

```
## [1] 0.9121672
```

```
print(confusionMatrix(data = FD_Test$is_fraud, reference = predict_tree))
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction      0      1
##           0 504267  48556
##           1    612   2283
##
##                Accuracy : 0.9115
##                  95% CI : (0.9108, 0.9123)
##     No Information Rate : 0.9085
##     P-Value [Acc > NIR] : 2.835e-15
##
##                   Kappa : 0.0759
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.99879
##             Specificity : 0.04491
##          Pos Pred Value : 0.91217
##          Neg Pred Value : 0.78860
##              Prevalence : 0.90852
##          Detection Rate : 0.90742
##    Detection Prevalence : 0.99479
##       Balanced Accuracy : 0.52185
##
##        'Positive' Class : 0
##
```