Challenge: Catch the Fraudster

Lindsey Erickson

June 26, 2017

Introduction

Every retail chain faces a potential fraud instances where people order a product and then return it after some days claiming either the product doesn't work or doesn't provide desired utility. However, each such transaction has some precursors that may point towards a potential fraud instances.

The goal of this project is to create a predictive model that will identify the fraud propensity for a retail company.

Comments

This project will be submitted to the coursera data science challenge provided by Sunil Kappal. This report is not meant to be used for actual fraud detection.

Import Data

My first step is to load the necessary libraries and import the data from the website. Once the data is loaded, I will need to coerce it to a data frame in order to prepare it for exploratory analysis.

```
# load the necessary packages
library(RCurl)
library(gsheet)
library(ggplot2)
library("PerformanceAnalytics")
library(rpart)
library(rattle)

# pull the data from the website and assign it to tempData
url <-
'https://docs.google.com/spreadsheets/d/1TufF3QBHK8RsC06V0arvF3PwN3gfz5kg5eV6
BjRxEjc/edit#gid=581816440'
tempData <- gsheet2tbl(url)

# coerce tempData to a data frame and assign it to fraudData
fraudData <- as.data.frame(tempData)
```

Clean Data

I now have a data frame, and want to look at some basic properties. If necessary, I will update any variables.

```
# check the number of rows and columns (observations and variables)
dim(fraudData)
## [1] 4349   13

# check to see if there are any missing values
sum(is.na(fraudData))
## [1] 0

# check the structure of the data frame
str(fraudData)
## 'data.frame':    4349 obs. of  13 variables:
##  $ #                 : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Fraud Instance    : int  0 0 0 1 0 0 1 0 1 1 ...
##  $ Damaged Item      : int  1 0 1 0 1 1 0 0 1 0 ...
##  $ Item Not Avaiable: int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Item Not In Stock: int  1 1 0 1 0 1 0 1 0 0 ...
##  $ Product Care Plan: int  0 1 1 0 1 0 1 0 0 1 ...
##  $ Claim Amount      : chr  "$89" "$290" "$67" "$350" ...
##  $ Registered Online: int  1 0 0 0 0 1 1 1 1 0 ...
##  $ Age Group         : int  29 33 39 49 37 25 55 34 49 42 ...
##  $ Marital Status    : chr  "In-Relationship" "Married" "Married" "In-
Relationship" ...
##  $ Owns a Vehicle    : int  1 1 1 1 1 1 1 1 1 0 ...
##  $ Accomodation Type: chr  "Owns a house" "Staying with Family" "Staying
with Family" "Rented" ...
##  $ Height (cms)      : int  155 178 156 187 184 157 173 169 185 159 ...
##  - attr(*, "spec")=List of 2
##   ..$ cols   :List of 13
##   .. ..$ #                 : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ Fraud Instance    : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ Damaged Item      : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ Item Not Avaiable: list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ Item Not In Stock: list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ Product Care Plan: list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ Claim Amount      : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ Registered Online: list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ Age Group         : list()
```

```
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ Marital Status   : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ Owns a Vehicle   : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ Accomodation Type: list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ Height (cms)     : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   ..$ default: list()
##   .. ..- attr(*, "class")= chr  "collector_guess" "collector"
##   ..- attr(*, "class")= chr "col_spec"
```

The data frame contains 4,349 observations and 13 variables, and does not contain any missing variables. Looking at the structure of the data, I notice there are some changes that need to be made to some variables:

1. Some variables are classified as integers when they should be factors. For example, the *Fraud Instance* variable is classified as an integer, as it contains 0's and 1's; however, the 0's stand for "not fraud" and the 1's stand for "fraud". This means that the variable is not necessarily an integer, but a factor with two levels. I need to change all the necessary variables to factors.

2. I noticed a variable called # that numbers each observation in order from 1 to 4,349. This variable is not providing any useful information; therefore, I will drop the variable.
3. The dollar sign in the *Claim Amount* variable is causing the variable to be classified as a character. I will drop the dollar sign and coerce the variable to an integer.

```
# changing the integer variables to factors
fraudData$`Fraud Instance` <- as.factor(fraudData$`Fraud Instance`)
fraudData$`Damaged Item` <- as.factor(fraudData$`Damaged Item`)
fraudData$`Item Not Avaiable` <- as.factor(fraudData$`Item Not Avaiable`)
fraudData$`Item Not In Stock` <- as.factor(fraudData$`Item Not In Stock`)
fraudData$`Product Care Plan` <- as.factor(fraudData$`Product Care Plan`)
fraudData$`Registered Online` <- as.factor(fraudData$`Registered Online`)
fraudData$`Marital Status` <- as.factor(fraudData$`Marital Status`)
fraudData$`Owns a Vehicle` <- as.factor(fraudData$`Owns a Vehicle`)
fraudData$`Accomodation Type` <- as.factor(fraudData$`Accomodation Type`)

# dropping the observation number variable, as it adds nothing to the dataset
and dropping Height variable, as I have no idea what it is and cannot find
any documentation on this variable.
fraudData <- subset(fraudData, select = -`#`)
fraudData <- subset(fraudData, select = -`Height (cms)`)
```

```
# drop the dollar sign and change the variable to an integer
fraudData$`Claim Amount` <- gsub("[$]","",fraudData$`Claim Amount`)
fraudData$`Claim Amount` <- as.integer(fraudData$`Claim Amount`)

# check the structure of the revised data frame
str(fraudData)
## 'data.frame':    4349 obs. of  11 variables:
##  $ Fraud Instance   : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 2 1 2 2 ...
##  $ Damaged Item     : Factor w/ 2 levels "0","1": 2 1 2 1 2 2 1 1 2 1 ...
##  $ Item Not Avaiable: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
##  $ Item Not In Stock: Factor w/ 2 levels "0","1": 2 2 1 2 1 2 1 2 1 1 ...
##  $ Product Care Plan: Factor w/ 2 levels "0","1": 1 2 2 1 2 1 2 1 1 2 ...
##  $ Claim Amount     : int  89 290 67 350 297 52 294 329 122 246 ...
##  $ Registered Online: Factor w/ 2 levels "0","1": 2 1 1 1 1 2 2 2 2 1 ...
##  $ Age Group        : int  29 33 39 49 37 25 55 34 49 42 ...
##  $ Marital Status   : Factor w/ 3 levels "In-Relationship",..: 1 2 2 1 1 1
1 2 2 1 ...
##  $ Owns a Vehicle   : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 1 ...
##  $ Accomodation Type: Factor w/ 3 levels "Owns a house",..: 1 3 3 2 3 1 1
2 2 1 ...
```

This revised data frame looks much better! I now have 11 variables to work with. My next step is to do some exploratory analysis on the data.

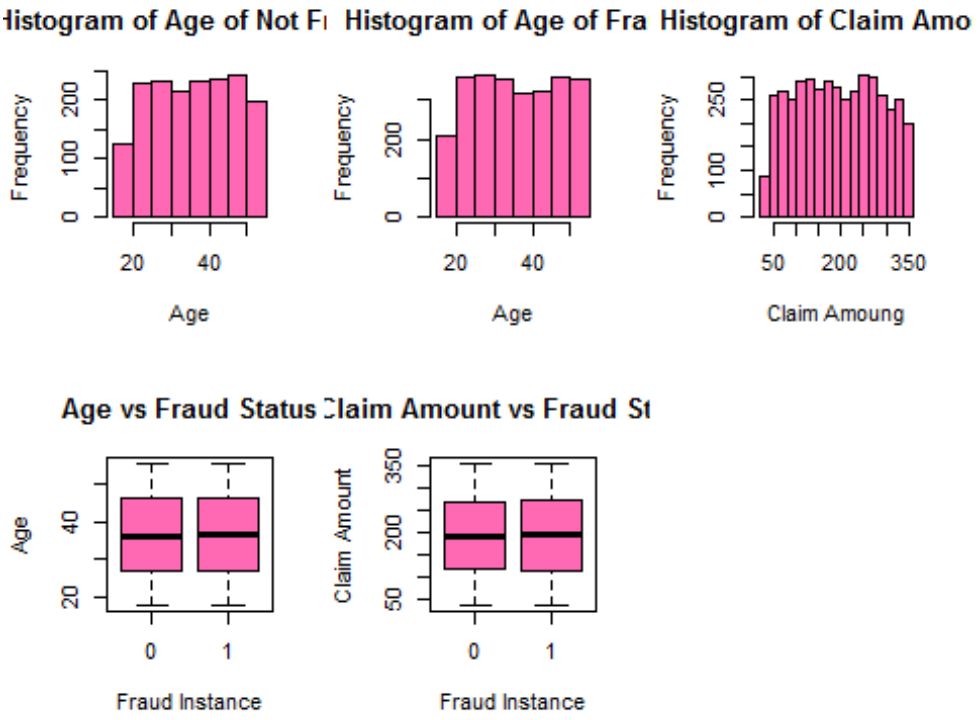
Exploratory Analysis

Here, I want to summerize the main characteristics of the data. I will start by looking at some summary statistics of the data. Looking at the dataset, I want to see the averages between fraud and not fraud among certain variables (e.g. age, claim amount, etc.). Secondly, I want to see if there are certain groups that seem to be more prone to fraud than other groups (e.g. age groups, marital status, etc.).

```
# summary statistics
summary(fraudData)
##  Fraud Instance Damaged Item Item Not Avaiable Item Not In Stock
##  0:2643         0:1439       0:3741            0:2178
##  1:1706         1:2910       1: 608            1:2171
##
##
##
##
##  Product Care Plan  Claim Amount   Registered Online   Age Group
##  0:3046            Min.   : 35.0   0:2212            Min.   :18.0
##  1:1303            1st Qu.:116.0   1:2137            1st Qu.:27.0
##                    Median :192.0                     Median :36.0
##                    Mean   :193.8                     Mean   :36.4
##                    3rd Qu.:270.0                     3rd Qu.:46.0
##                    Max.   :355.0                     Max.   :55.0
##          Marital Status Owns a Vehicle           Accomodation Type
```

```
##  In-Relationship:1408   0:2185         Owns a house       :1441
##  Married        :1503   1:2164         Rented             :1409
##  Unmarried,     :1438                  Staying with Family:1499
##
##
##

# split the dataset into two separate datasets: one which includes fraud and
one which includes not fraud
actualFraud <- fraudData[fraudData$`Fraud Instance` == 1,]
notFraud <- fraudData[fraudData$`Fraud Instance` == 0,]

# find the average age between fraud/not fraud
mean(actualFraud$`Age Group`) #mean is 36.45
## [1] 36.45135
mean(notFraud$`Age Group`)   #mean is 36.4
## [1] 36.37117
#it seems the average age is the same between fraud and not fraud

# find the average claim amount between fraud/not fraud
mean(actualFraud$`Claim Amount`) #mean is $195
## [1] 194.9601
mean(notFraud$`Claim Amount`) #mean is $193
## [1] 192.9822
#the average claim amount is fairly equal

# compile a table showing relationship status vs fraud
table(fraudData$`Marital Status`, fraudData$`Fraud Instance`)
##
##                     0   1
##   In-Relationship 865 543
##   Married         919 584
##   Unmarried,      859 579

# compile a table showing accomodation type vs fraud
table(fraudData$`Marital Status`, fraudData$`Fraud Instance`)
##
##                     0   1
##   In-Relationship 865 543
##   Married         919 584
##   Unmarried,      859 579

# compile a table showing vehicle ownership vs fraud
table(fraudData$`Owns a Vehicle`, fraudData$`Fraud Instance`)
##
##        0    1
##   0 1343  842
##   1 1300  864
```

```
par(mfrow=c(2,3))
hist(actualFraud$`Age Group`, col = "hot pink", xlab = 'Age', ylab =
'Frequency', main = 'Histogram of Age of Not Fraud')
hist(notFraud$`Age Group`, col = "hot pink", xlab = 'Age', ylab =
'Frequency', main = 'Histogram of Age of Fraud')
hist(fraudData$`Claim Amount`, col = "hot pink", xlab = 'Claim Amoung', ylab
= 'Frequency', main = 'Histogram of Claim Amount')
plot(fraudData$`Age Group` ~ fraudData$`Fraud Instance`, col = "hot pink",
xlab = 'Fraud Instance', ylab = 'Age', main = 'Age vs Fraud Status')
plot(fraudData$`Claim Amount` ~ fraudData$`Fraud Instance`, col = "hot pink",
xlab = 'Fraud Instance', ylab = 'Claim Amount', main = 'Claim Amount vs Fraud
Status')
```



Based on the above analysis, I found a lot of interesting information in the data. Note that statistical tests were not conducted; therefore, the below findings have not been tested for statistical significance. I'll breakdown the findings below:

1. Of the 4,349 observations in the dataset, 39% were considered fraud.
2. The minimum claim amount was $35, while the highest amount was $355
3. The youngest age was 18 years old, while the oldest was 55 years old.
4. The average age of individuals' claims that were considered fraud was about 34 years old, as was the average age of individuals who's claims were not considered fraud (34 y/o). Therefore, there does not seem to be a difference between fraud/not fraud and age.

5. The average claim amount of a fraud transaction was $195, while the average claim amount of a non-fraud transaction was $193. Again, there does not seem to be a difference between fraud/not fraud and claim amount.
6. Looking at the table that categorizes marital status and fraud instance, it shows that there there doesn't seem to be any differences between fraud and not fraud and the three marital statuses: *In-Relationship*, *Married*, and *Unmarried*.
7. Looking at the table that categorizes vehicle ownership between fraud and not fraud, it shows that there doesn't seem to be a difference between fraud and not fraud and whether or not one owns a vehicle.
8. The histograms that show the average age of non-fraud cases, average age of fraud cases, and histogram of the claim amount all seem to have somewhat of a uniform distribution.
9. The boxplot of age vs fraud status shows that the distribution of age is the same for fraud cases and non-fraud cases.
10. The boxplot of claim amount vs fraud status shows that the distribution of claim amount is the same for fraud cases and non-fraud cases.

Model Building

Next, I want to build a model that will detect whether or not a transaction is fraud or not fraud. I want to build a decision tree, as there are only two predictive outcomes: fraud and not fraud. Secondly, a decision tree will be easy for the end-user to interpret. Lastly, the decision tree will implicitly perform feature selection. To build this model, I first want to train and test the data using cross-validation in a for-loop. With each iteration the for-loop will resample a train and test set and use that to predict and train the tree. I will then see how accurate the trained model is.

```
#train and test data using cross-validation
set.seed(199)
n <- nrow(fraudData)
shuffled <- fraudData[sample(n),]
accs <- rep(0,6)

for(i in 1:6) {
  # These indices indicate the interval of the test set

  indices <- (((i-1) * round((1/6)*nrow(shuffled))) + 1):((i*round((1/6) *
nrow(shuffled))))

  # Exclude the from the train set
  train <- shuffled[-indices,]

  # Include them in the test set
  test <- shuffled[indices,]

  # A model is learned using each training set
  tree <- rpart(`Fraud Instance` ~ ., train, method = "class")
```

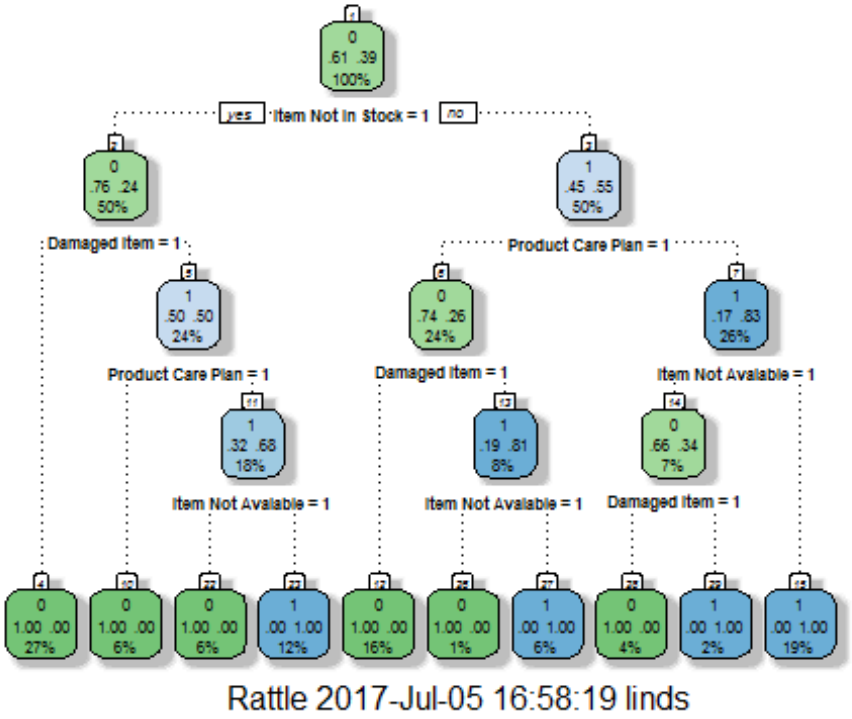
```
  # Make a prediction on the test set using tree
  pred <- predict(tree, test, type = "class")

  # Assign the confusion matrix to conf
  conf <- table(test$`Fraud Instance`, pred)

  # Assign the accuracy of this model to the ith index in accs
  accs[i] <- sum(diag(conf))/sum(conf)

}

mean(accs)
## [1] 1
conf
##    pred
##       0   1
##   0 446   0
##   1   0 278
```

Looking at the average accuracy of the trained model, we can see that we have a 100% accuracy! Looking at the confusion matrix, you can see that in fact, it is 100% accurate. I'm very happy with this model and will call it my final model.

Final Model: Decision Tree

Here's the final model that will detect whether or not a claim is considered fraud:

```
# plot the tree
fancyRpartPlot(tree)
```



Conclusion

This project walked through the steps on how to analyze data and build a model that will successfully predict whether or not a transaction is considered fraud or not fraud. After cleaning the data, I was able to create a decision tree that was 100% accurate in predicting fraud.

Notice that the top nodes of the tree: *Item Not In Stock*, *Damaged Item*, and *Product Care Plan* are the most important variables in the dataset, as they are in the top of the tree. By looking at the decision tree, it seems that one should be suspecious if one returns an item that is in stock, if there is not a product care plan, and if the item is not available. If the item is in stock and the item is damaged, then you can feel confident that the transaction is not fraud.

One should feel confident in implementing this decision tree to predict whether or not a transaction is fraud.