Credit Data Analysis

Abstract

In this data exploration, the goal is to create an effective model in the binary classification of credit card transactions as fraudulent or not. To achieve this, logistic regression and random forest models are employed for the use of classification. Since the data set that is used consisted of 284,807 total observational data point transactions, a subset of 10,000 data points were first used to determine which of the two models would be the most effective. The obtained accuracy of the two models were 99.9% and 99.95% accuracy for logistic regression and random forest models respectively. Looking at other metrics of validation, the AUC-ROC curve was utilized to determine which method would be better at distinguishing between the two binary classes. Logistic regression was found to have a lower AUC by .08% which is not much. In addition, a confusion matrix was used to determine that the random forest model had one less false positive than the model from logistic regression. Therefore, a random forest model was trained over the entire data set which obtained a model with 99.99% accuracy.

Introduction

Credit card fraudulent activity is a rampant issue in today's society where we have switched from carrying around large amounts of currency in order to simply use a card that carries funds electronically. When someone other than yourself utilizes your credit card for transactions, this is defined as credit card fraudulence as they've used funds charged to your name to purchase products and services for themselves. In order to combat this issue, it is important for credit card companies to recognize fraudulent transactions so that their clients will not be charged for these fraudulent transactions. As such, statistical learning methods can be made in order to predict whether a credit card transaction is fraudulent or not. ***

Methods

In order to solve this issue the Logistic regression and random forest method will be employed to create a means to classify credit card transactions as either fraudulent (1) or not fraudulent (0). A sub-set of 10,000 points from the 284,807 total observed transactions will be used to initially train the models and create a comparison between the two models.

Data

The data set that is observed is collected in September 2013 by European cardholders and was accrued over a two day period. There are a total of 284,807 observed transactions in which 492 of them accounted for fraudulent transactions.

```
##   fraud genuine 
##     492  284315
```

This small portion of fraudulent transactions of the total amount of observations indicate a highly unbalanced data set. In addition, Principal Component Analysis (PCA) transformation is performed on all features (V1-V28) in order to maintain anonymity of the card holders. The two features of each observed data set is "Time" and "Amount" where time is the number of seconds elapsed between the current transaction and the first transaction and amount is the amount cost of the transaction. Finally "Class" will act as a dummy variable that marks 0 for non fraudulent charges while 1 indicates a fraudulent charge.

To preprocess this data we should first check to see if the data set contains any NA values in any of the observations. This particular data set contains no null values for data so no further changes have to be done to make the data set workable.

```
##   Time     V1     V2     V3     V4     V5     V6     V7     V8     V9    V
10 
##      0      0      0      0      0      0      0      0      0      0     
0 
##    V11    V12    V13    V14    V15    V16    V17    V18    V19    V20    V
21 
##      0      0      0      0      0      0      0      0      0      0     
0 
##    V22    V23    V24    V25    V26    V27    V28 Amount  Class 
##      0      0      0      0      0      0      0      0      0
```

In order to deal with the imbalance in the data set under-sampling and over-sampling can be used to balance the class distribution for classification. This will eliminate the skewness of the distribution and allow for more accurate classification models.

```
## genuine   fraud 
##    5001    4999
```

The subset of the overall data set is split into train and test sets where 80% of the entire set will be used to train the logistic regression model, while the other 20% is used to test the data set.

```
set.seed(42)
ind_Training=sample(nrow(Credit_Data_TT),round(nrow(Credit_Data_TT)*0.80),rep
lace = FALSE)
Credit_Data_Training=Credit_Data_TT[ind_Training,];
Credit_Data_Test=Credit_Data_TT[-ind_Training,];
```

Modeling

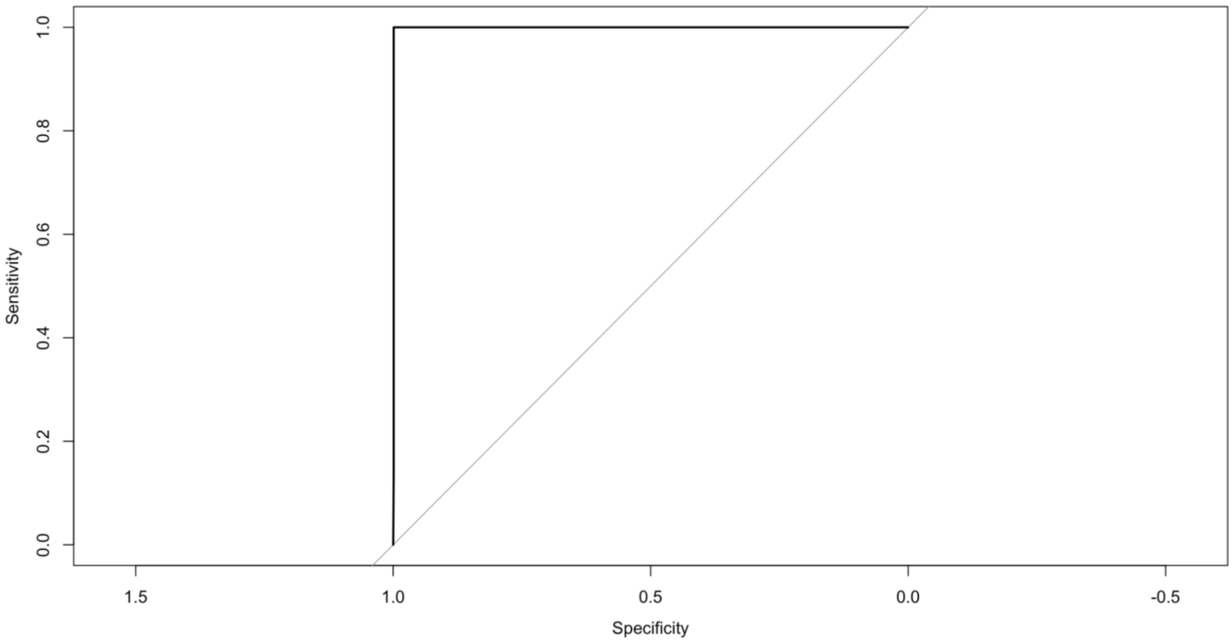
In order to create a comparison for logistic regression and random forest classification the subset of the credit card data is used to fit an overall model for both types of learning methods. From there the assessment of accuracy is obtained to see which method acquired a higher. In addition, AUC-ROC curves will be observed in order to determine which model is better capable of distinguishing between fraudulent (1) and non-fraudulent (0) transactions.

```
# logistic regression fitted model
set.seed(42)
glm_model <- train(Class ~ ., data = Credit_Data_Training, preProcess=
c("center", "scale"),method = "glm")
glm_model


# confusion matrix for logistic regression
pred=predict(glm_model, newdata = Credit_Data_Test)
CM= confusionMatrix(data=pred, Credit_Data_Test$Class)
CM

# fit random forest model
set.seed(42)
rf_model <- randomForest(Class ~ ., data = Credit_Data_Training, ntree
= 100)
rf_model

#confusion matrix for random forest model
rf_predict <- predict(rf_model, Credit_Data_Test)
rf_cm<- confusionMatrix(data = rf_predict, Credit_Data_Test$Class)
rf_cm
```

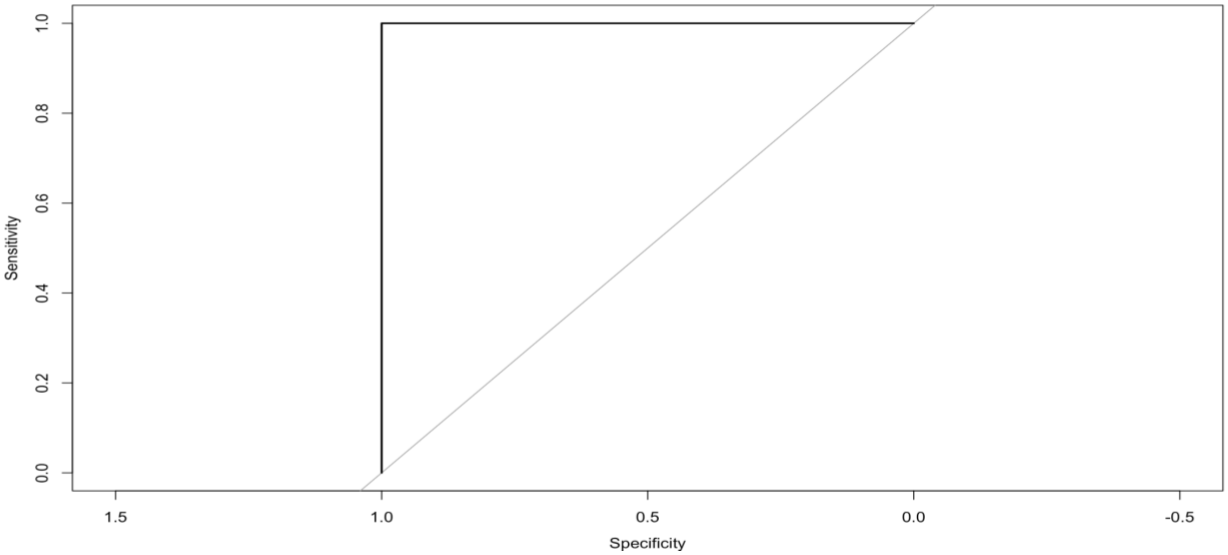
Results

From the acquired models the accuracy that is obtained from the two fitted models are both high and relatively similar 99.9% and 99.95% for logistic regression and random forest model respectively. In order to further stress test these models the AUC-ROC curves are used.

```
roc(Credit_Data_Test$Class,
    predict(glm_model, newdata = Credit_Data_Test, type = "prob")[,2],
    plot=T)
```



```
roc(Credit_Data_Test$Class,
    predict(rf_model, newdata = Credit_Data_Test, type = "prob")[,2],
    plot=T)
```



The area under the curve for logistic regression and random forest models were found to be 99.91% and 99.99% which are both large. This indicates that both models do a good job at creating distinction between the fraudulent and non fraudulent transactions. Upon observation of the confusion matrix, it is observed that classification of the test data was relatively similar in terms of correctness, however, random forest was able to further prevent a single false negative. For this scenario, false positives are better than false negatives as it is better to have transaction not go through as opposed to having undetected fraudulent activity.

Discussion

From the obtained data, the random forest model was found to be slightly more effective than logistic regression. Still, the accuracy of both the models and their obtained value of AUC from ROC curves prove that these two models are effective in producing a solution of fraudulent prediction. Since the data set was very large at 284,807 total transnational observations, taking the subset of this data set to determine the best type of model for the problem was helpful in reducing the amount of time required to train the final model.

The final model was selected to be a random forest model. This model was trained over the entire data set and achieved a large accuracy of 99.99%. There were no false negatives and three false positives detected. Therefore achieving a highly effective model.

Appendix

The code below is used to build and analyze the random forest model over the entire dataset.

~~~
Credit_Data=ovun.sample(Class~.,data = cc, method = "both",
                                 p = 0.5,seed = 42)$data
summary(Credit_Data$Class)

set.seed(42)
ind_Training_full=sample(nrow(Credit_Data),round(nrow(Credit_Data)*0.8
0),replace = FALSE)
Credit_Data_Training_full=Credit_Data[ind_Training_full,];
Credit_Data_Test_full=Credit_Data[-ind_Training_full,];

set.seed(42)
rf_model <- randomForest(Class ~ ., data = Credit_Data_Training_full,
ntree = 100)
rf_model

rf_predict <- predict(rf_model, Credit_Data_Test)
rf_cm<- confusionMatrix(data = rf_predict, Credit_Data_Test$Class)
rf_cm
```
~~~