Term Project: Identifying Fraudulent Transactions Using Logistic Regression

Ayan Ashkenov

Department of Computer Science, Metropolitan College, Boston University

CS555: Foundations of Machine Learning

Dr. Ming Zhang

May 7th, 2023

1. Research Objective

Fraud detection is defined as "a set of activities undertaken to prevent money or property from being obtained through false pretenses"[1]. It is an integral part in many industries across the board, like e-commerce, online banking, and more. In many ways, fraud detection is an arms race against the ever-evolving fraudsters, a constant battle to prevent new malevolent techniques and methods. According to some research, the fraud detection and prevention market is valued around 63.5 billion U.S. dollars, an increase of 49.13 billion since 2016[2].

The general objective of this paper is to analyze the differences between fraudulent and normal online transactions. The ultimate objective, however, is to determine whether fraud could be identified by the logistic regression algorithm. The term "fraud detection" invites the idea that only the most sophisticated algorithms can manage such a task. There is a lot of truth in this notion, as many large institutions rely on very advanced artificial intelligence systems to distinguish between fraud and normal behavior. However, such high-end methods are often expensive, in both development and upkeep, and unnecessary for smaller organizations. This research attempts to test whether a simple classification algorithm, namely logistic regression, can do a satisfactory job in correctly classifying fraudulent online behavior.

[1] Gillis, S. Alexander. "Fraud detection". Tech Target, September 2021.
[2] Statista Research Department. "Size of the fraud detection and prevention (FDP) market worldwide from 2016 to 2023 (in billions of U.S. dollars)". Statista, January 2023.

2. Data Description

The dataset was acquired from Kaggle, with 5 downloads and 251 views as of May 3rd, 2023[3]. The provider of this dataset is Sudharshan Jayakumar and it has a total of 14 columns and 176,650 records. This dataset presents online transactions with binary fraud classification (isFraud = 1 or 0). No other information was provided by the author.

Out of the 14 variables, only 3 were used:

- type (type of the online transaction)
- amount (amount of the online transaction; currency unknown)
- isFraud (classification of the transaction; fraudulent or not)

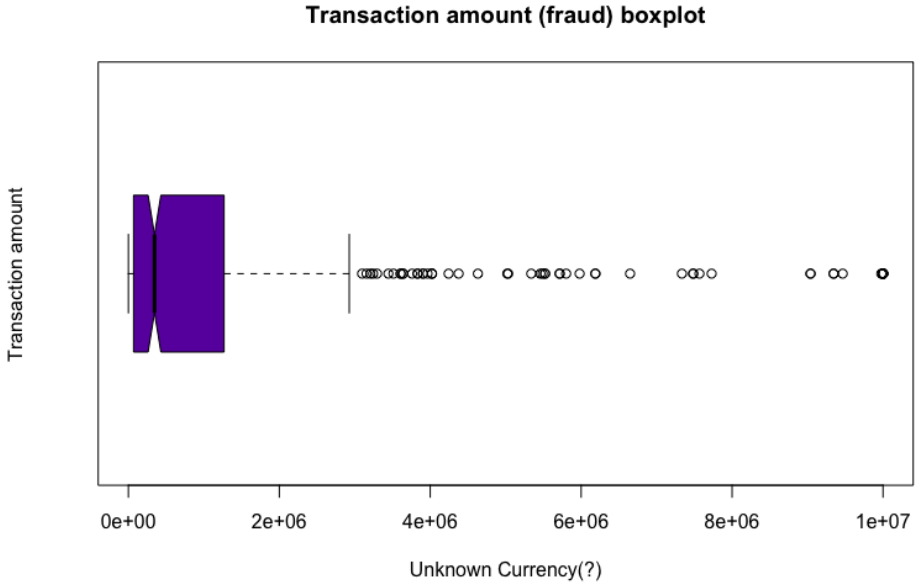
Following exploratory data analysis, other columns did not showcase any value for the research goals. Many columns, like isFraudPrevented, had a 100% correlation with the classification variable (isFraud). Others had the exact same value for all rows, providing no information about the nature of the transactions.

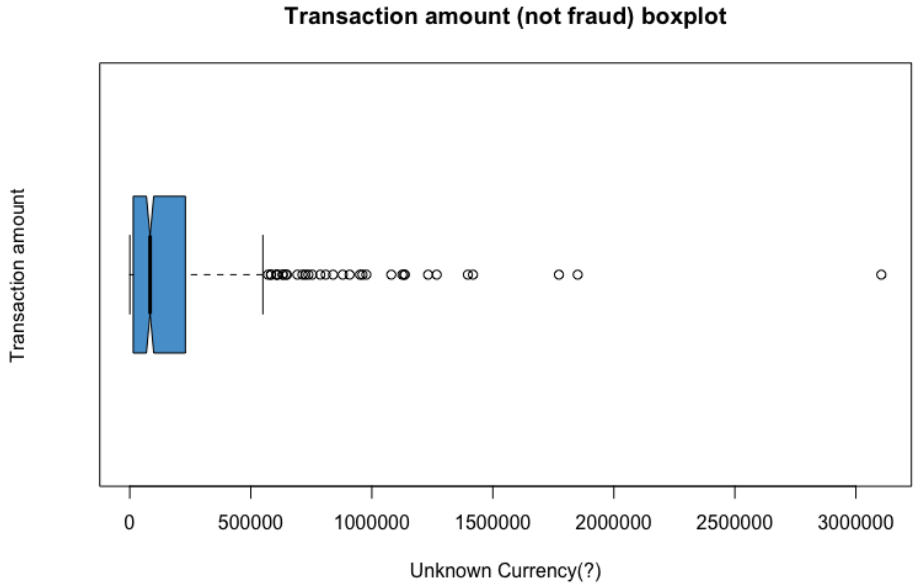
As required by the term project guidelines, the dataset was reduced to 1,000 rows. Originally, the dataset contained 1,142 fraudulent transactions and 175,508 regular transactions. This ratio makes sense, because fraudulent behavior is a rare occurrence when compared to everyday, normal transaction traffic. Fraud by definition is not common and can be considered an anomaly.

To avoid oversampling or undersampling the data, it was decided to take an equal number of transactions from each group, equaling 500 fraudulent and 500 normal transactions. They were selected randomly utilizing simple-random-sampling methodology. This approach will also benefit the final logistic regression model, by providing it with an equal number of cases on both sides.

[3] https://www.kaggle.com/datasets/sudharshanjayakumar/dataset

The two diagrams below present a box plot of fraudulent and regular transaction amounts, respectively. While the plots do show conventional outliers, it was chosen to keep them for the analysis, because it was theorized that higher amounts corresponded with more likelihood that a transaction is fraudulent.





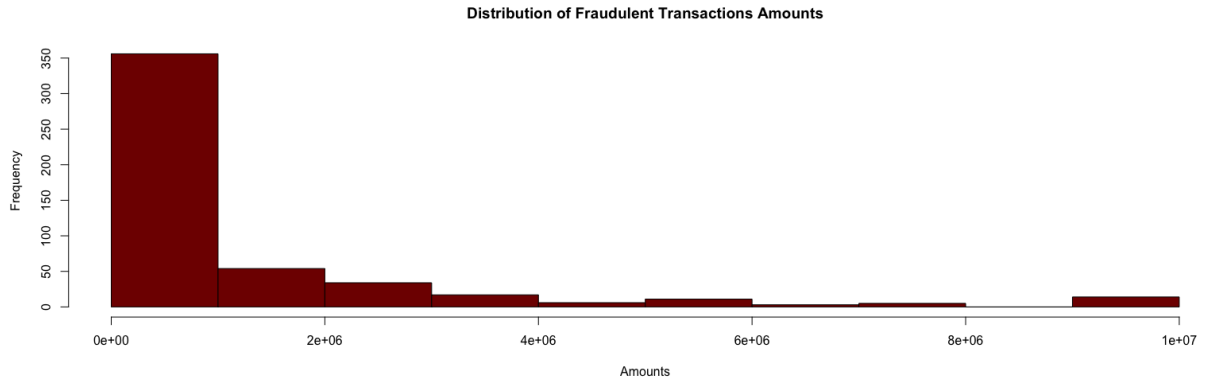
The largest fraudulent transaction amount is 10,000,000 units, with the minimum at 119. The maximum for the regular transactions in the sample is 3,105,173 units, with the minimum at 265.8. The averages for the two groups are 1,177,240 and 181,216.9 units, respectively. The differences between them are substantial at first glance, but it is interesting that the sample of fraudulent transactions has a smaller minimum amount than the regular sample. It must also be noted that it does not make sense to run a sample mean t-test on the created sample. This is because the sample used in this research has a balanced ratio of fraudulent to normal transactions, and is not similar to the original population ratio. As observed earlier, fraudulent transactions are, on average, much larger than regular ones, guaranteeing that one sample mean test would not produce meaningful results.

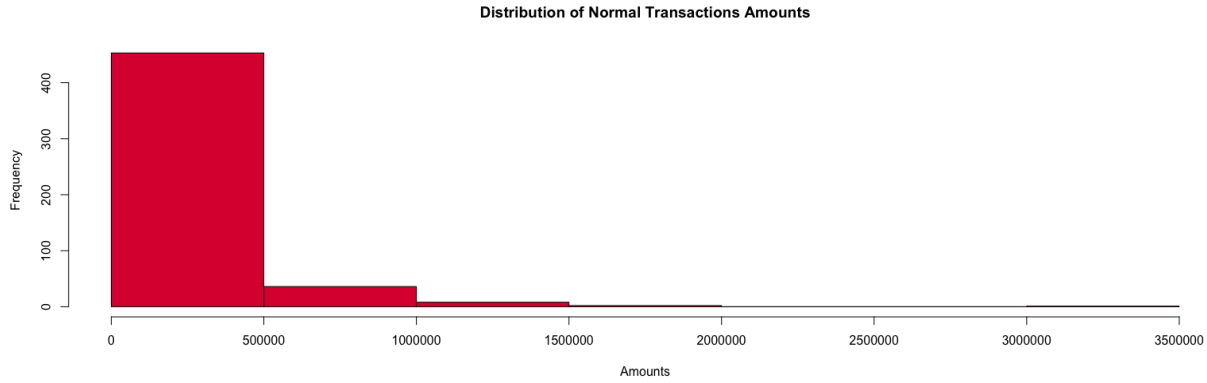
Lastly, preliminary analysis of the dataset showed that all fraudulent transactions are either a cash-out operation or a transfer of money from one account to another. The same cannot be said about normal transactions, with many cash-in operations and payment receival types, among others.

3. Statistical Methods Introduction

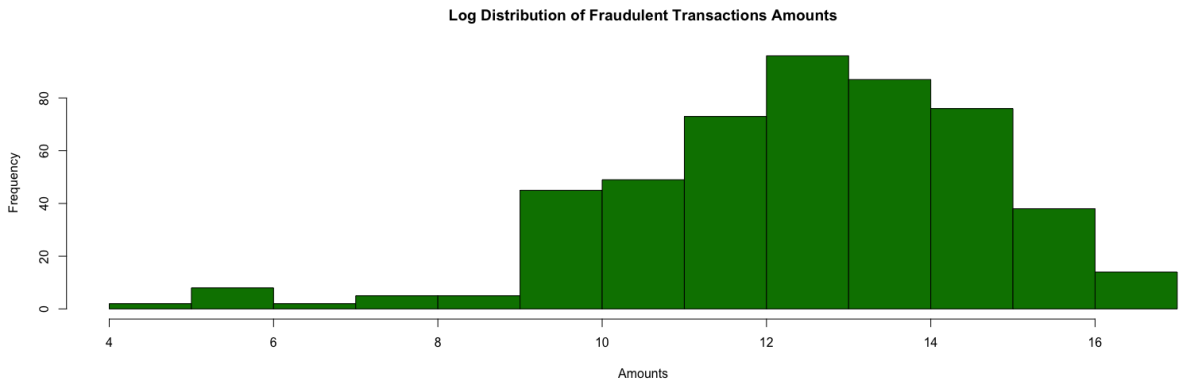
Initially, two sample mean tests were planned for this research. However, upon further investigation of the data, neither the t-test of means nor the ANOVA test could be used.

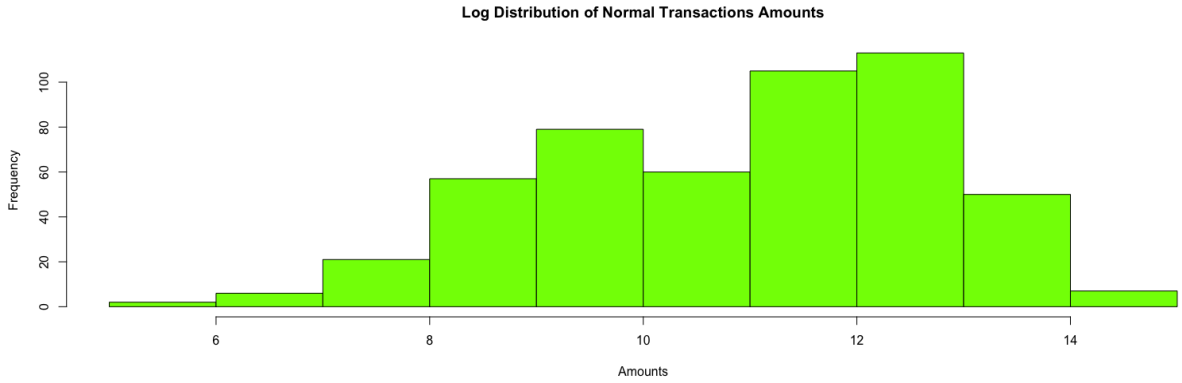
T-test of means is used when the sample size is small, generally below 30 observations, or when the population variance is unknown. However, it assumes that the experiment data is normally distributed, or at least, approximately so. The histogram of fraudulent and normal transaction amounts can be observed below:



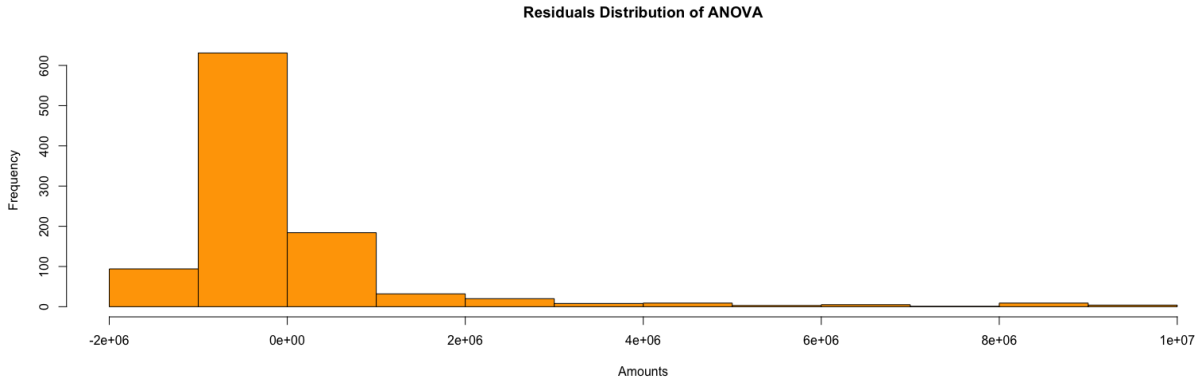


As is obvious from the histograms, both groups have a heavy right skew. Removing outliers will not improve the situation, since the right skew is very pronounced even without the extreme values. To further confirm this, a Shapiro-Wilk test was carried out on both groups, showing that neither are normally distributed (p-values less than 2.2e-16 for both groups). Finally, a logarithmic transformation was applied on the data, to make sure that the normal distribution requirement cannot be met. The histograms below show the transformation and while both groups do look closer to being normally distributed, the Shapiro-Wilk tests carried out on the transformed values did not support this (2.117e-8 and 2.597e-11 p-values, respectively).



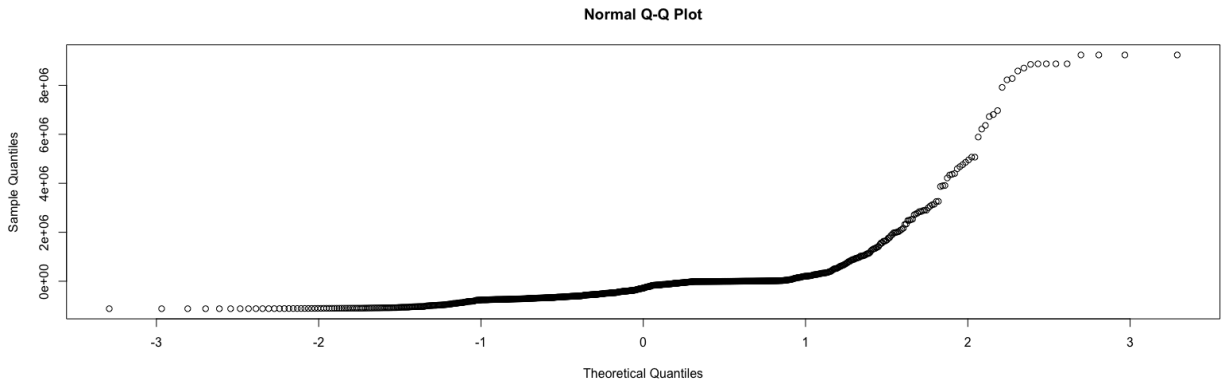


The ANOVA test was also not applicable, because it did not meet the normally distributed requirement and the homogeneity of variance test. The histogram of the residuals can be found below:



Even though there is a heavy right skew, the data looks approximately normally distributed. Upon further analysis, the Shapiro-Wilk test did not confirm this, resulting in a

p-value of less than 2.2e-16. In addition, the Q-Q plot displayed a snake-like shape, far away from a straight line, which indicates normal distribution. Lastly, the Levene test indicated that the group variances are not exactly equal, presenting a p-value of 4.94e-14.



While comparing means across fraudulent and normal transactions, along with between transaction types, was not possible, the research could finally enter its final stage. Two logistic regression models were used, one that only used the amount of a transaction (Simple Logistic Regression) and the other that also used the type of a transaction (Multiple Logistic Regression).

4. Research Result

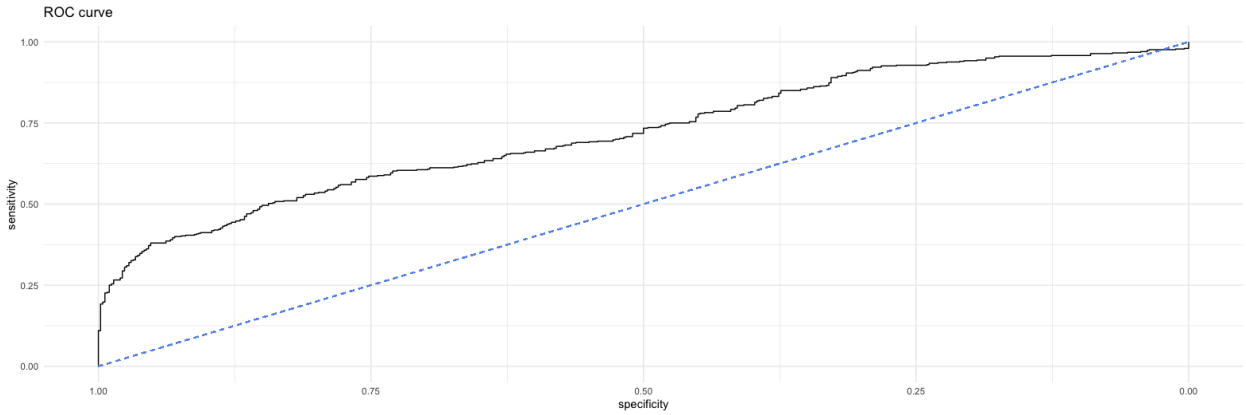
It was hypothesized that the type of the transaction and its amount would combine into a strong fraud detection model. This theory did not prove correct. The type of a transaction was not significant in the multiple logistic regression model:

```
Call:
glm(formula = srs_df_filtered$isFraud ~ srs_df_filtered$amount +
    srs_df_filtered$type, family = binomial)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.69082  -0.00008   0.00699   0.75889   1.17915

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)
(Intercept)                -1.971e+01  9.931e+02  -0.020    0.984
srs_df_filtered$amount      8.167e-07  1.619e-07   5.045 4.53e-07 ***
srs_df_filtered$typeCASH_OUT  1.971e+01  9.931e+02   0.020    0.984
srs_df_filtered$typeDEBIT     1.438e-01  7.669e+03   0.000    1.000
srs_df_filtered$typePAYMENT   1.380e-01  1.299e+03   0.000    1.000
srs_df_filtered$typeTRANSFER  2.077e+01  9.931e+02   0.021    0.983
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1386.3  on 999  degrees of freedom
Residual deviance:  776.0  on 994  degrees of freedom
AIC: 788

Number of Fisher Scoring iterations: 18
```

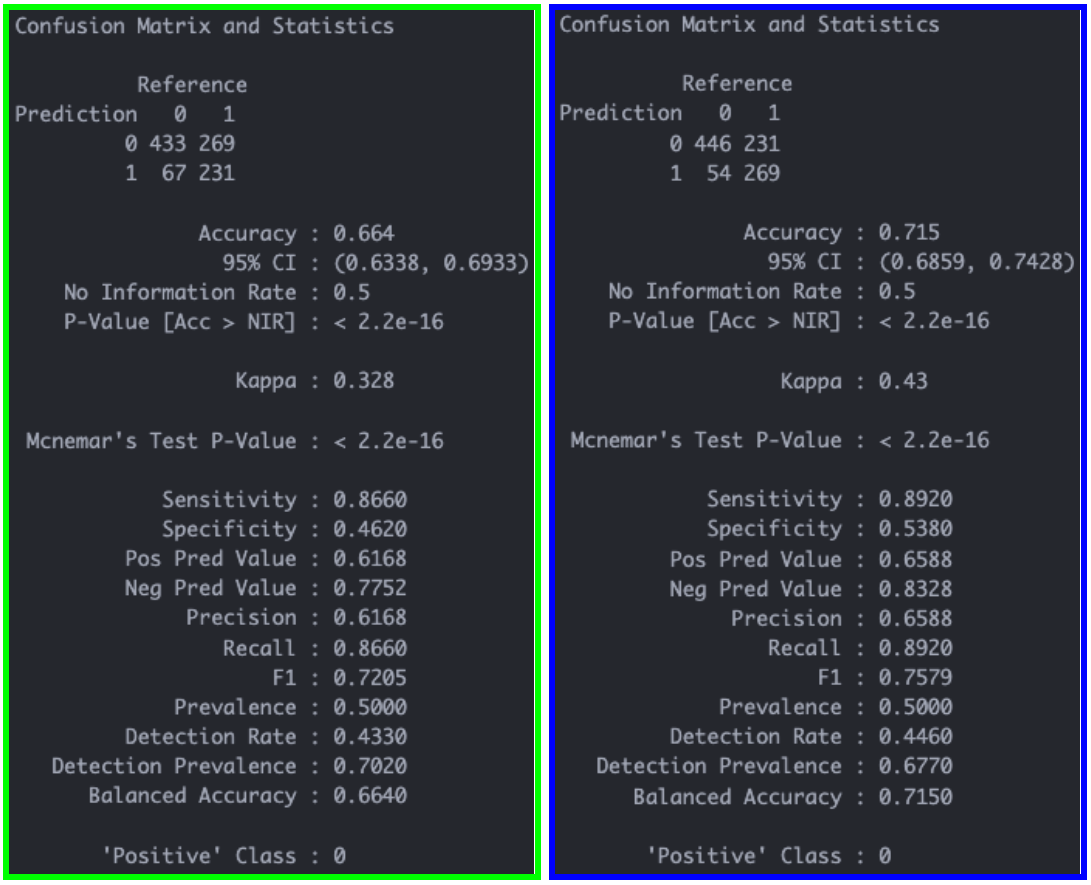
As is evident from the result, only the coefficient associated with the amount of a transaction was significant at low alpha levels (the universal alpha for this research was 0.05). The p-value for types DEBIT and PAYMENT was 1, which makes sense since fraudulent transactions are never of these types. In stark contrast, the simple logistic regression model that only used the amount variable, was significant for the transaction coefficient and the x-intercept:

```
Call:
glm(formula = srs_df_filtered$isFraud ~ srs_df_filtered$amount,
    family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.1192  -0.9644  -0.4544   1.2004   1.4720

Coefficients:
                         Estimate Std. Error z value Pr(>|z|)
(Intercept)            -6.705e-01  8.554e-02  -7.839 4.56e-15 ***
srs_df_filtered$amount  1.780e-06  2.002e-07   8.891  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1386.3  on 999  degrees of freedom
Residual deviance: 1177.7  on 998  degrees of freedom
AIC: 1181.7

Number of Fisher Scoring iterations: 6
```

The area under the curve for this simple logistic regression model is 0.7205, which is an acceptable value for general modeling:



In order to determine if a simple logistic regression model performs on par with a more advanced algorithm, the same exact data was used with a boosted classification tree model. The side-to-side outputs, along with the confusion matrices, can be found below on the next page. The boosted trees model outperformed the logistic regression model across all relevant metrics, such as Sensitivity and Specificity, along with the F-1 and Balanced Accuracy scores. However, while the boosted trees model was clearly the better choice, the gap between the two models was relatively small. The difference in the F-1 score was only 0.0374 and the difference between Balanced Accuracy was 0.051.



```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 433 269
         1  67 231

               Accuracy : 0.664
                 95% CI : (0.6338, 0.6933)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.328

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.8660
            Specificity : 0.4620
         Pos Pred Value : 0.6168
         Neg Pred Value : 0.7752
              Precision : 0.6168
                 Recall : 0.8660
                     F1 : 0.7205
             Prevalence : 0.5000
         Detection Rate : 0.4330
   Detection Prevalence : 0.7020
      Balanced Accuracy : 0.6640

       'Positive' Class : 0
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 446 231
         1  54 269

               Accuracy : 0.715
                 95% CI : (0.6859, 0.7428)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.43

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.8920
            Specificity : 0.5380
         Pos Pred Value : 0.6588
         Neg Pred Value : 0.8328
              Precision : 0.6588
                 Recall : 0.8920
                     F1 : 0.7579
             Prevalence : 0.5000
         Detection Rate : 0.4460
   Detection Prevalence : 0.6770
      Balanced Accuracy : 0.7150

       'Positive' Class : 0
```

Logistic regression model (green) vs. Ada Boosted Trees model (blue)

5. Conclusion

This paper attempted to use a simple logistic regression model to detect fraudulent online behavior. The data used in this experiment was extremely limited, since real-world applications likely use big, high-dimensional datasets. Nonetheless, it was enough to compare two very different machine learning models and evaluate whether logistic regression can be used.

Logistic regression, as an algorithm, is very simple in comparison to other classification systems. With no parameters to tune, logistic regression is a general model that shines in very specific, limited environments. In comparison, boosted classification trees have 3 parameters that

can be adjusted to improve the model performance and is a much better option for higher dimensional data.

A significant improvement in this research would see the two algorithms tested on larger, more complex data. In an ideal world, this experiment would be run on real-life, anonymized online transaction data. The research could also be extended to more algorithms, with a heavier emphasis on parameter tuning. With this in mind, the results of this research are inconclusive and should only be seen as an introduction to the fraud detection dilemma. While preliminary evidence suggests that logistic regression models will not outperform more complex algorithms, more evidence is needed to confirm this notion.

References

Gillis, S. Alexander. “Fraud detection”. Tech Target, September 2021. https://www.techtarget.com/searchsecurity/definition/fraud-detection

Jayakumar, Sudharshan. “Online_Fraud_with_MFA_enabled”. Kaggle, March, 2023. https://www.kaggle.com/datasets/sudharshanjayakumar/dataset

Statista Research Department. “Size of the fraud detection and prevention (FDP) market worldwide from 2016 to 2023 (in billions of U.S. dollars)”. Statista, January 2023. https://www.statista.com/statistics/786778/worldwide-fraud-detection-and-prevention-market-size/

R code:

```
setwd("/users/ayan/Desktop/BU/Spring 2023/CS555_project")

df <- read.csv("project_data.csv")

fraud_subset <- subset(df, isFraud == 1)
not_fraud_subset <- subset(df, isFraud == 0)

library(dplyr)
set.seed(17)
srs_fraud <- sample_n(fraud_subset, 500)
srs_not_fraud <- sample_n(not_fraud_subset, 500)

srs_df <- rbind(srs_fraud, srs_not_fraud)

relevant_cols <- c("type", "amount", "isFraud")
srs_df_filtered <- srs_df[relevant_cols]

srs_df_filtered$type <- as.factor(srs_df_filtered$type)
srs_df_filtered$isFraud <- as.factor(srs_df_filtered$isFraud)

write.csv(srs_df_filtered, "Ashkenov_Final_Project_data.csv", row.names=FALSE)

table(srs_fraud$type)
table(srs_not_fraud$type)

# boxplot
boxplot(srs_fraud$amount,
        main = "Transaction amount (fraud) boxplot",
        xlab = "Unknown Currency(?)",
        ylab = "Transaction amount",
        col = "#6A0DAD",
        border = "black",
        horizontal = TRUE,
        notch = TRUE
)
boxplot(srs_not_fraud$amount,
        main = "Transaction amount (not fraud) boxplot",
        xlab = "Unknown Currency(?)",
        ylab = "Transaction amount",
        col = "#56A0D3",
        border = "black",
        horizontal = TRUE,
        notch = TRUE
)
```

```
### T-test
hist(srs_fraud$amount, main="Distribution of Fraudulent Transactions Amounts", xlab =
"Amounts", col="#800000")
shapiro.test(srs_fraud$amount)
hist(srs_not_fraud$amount, main="Distribution of Normal Transactions Amounts", xlab =
"Amounts", col="#DC143C")
shapiro.test(srs_not_fraud$amount)

log_fraud_amount <- log(srs_fraud$amount)
hist(log_fraud_amount, main="Log Distribution of Fraudulent Transactions Amounts", xlab =
"Amounts", col="#008000")
mean(log_fraud_amount)
shapiro.test(log_fraud_amount)

log_not_fraud_amount <- log(srs_not_fraud$amount)
hist(log_not_fraud_amount, main="Log Distribution of Normal Transactions Amounts", xlab =
"Amounts", col="#7FFF00")
mean(log_not_fraud_amount)
shapiro.test(log_not_fraud_amount)

### ANOVA (types column)
# normal distribution test
fit <- aov(amount ~ type, data = srs_df_filtered)
resid <- residuals(fit)
hist(resid, main="Residuals Distribution of ANOVA", xlab = "Amounts", col="#FFA500")
qqnorm(resid)
shapiro.test(resid)

# homogeneity of variance test
leveneTest(amount ~ type, data = srs_df_filtered)

### Logistic Regression
log_reg_model <- glm(srs_df_filtered$isFraud ~ srs_df_filtered$amount, family = binomial)
summary(log_reg_model)

log_reg_model <- glm(srs_df_filtered$isFraud ~ srs_df_filtered$amount + srs_df_filtered$type,
family = binomial)
summary(log_reg_model)

predictions <- predict(log_reg_model, type = "response")
library(pROC)
roc(srs_df_filtered$isFraud, predictions)

srs_df_filtered$predictions <- predictions
srs_df_filtered$predictions <- ifelse(predictions>=0.5, 1,0)
```

```
srs_df_filtered$predictions <- as.factor(srs_df_filtered$predictions)

confusionMatrix(srs_df_filtered$predictions, srs_df_filtered$isFraud, mode = "everything")

library(pROC)
library(ggplot2)

ggroc(roc(srs_df_filtered$isFraud, predictions)) +
  theme_minimal() +
  ggtitle("ROC curve") +
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1), color="#6495ED", linetype="dashed")

library(caret)
ctrl <- trainControl(method = "repeatedcv",
                     number = 10,
                     repeats = 5,
                     verboseIter = FALSE)
ada_grid <- expand.grid(iter = 5, maxdepth = 1:5, nu = seq(0.1, 0.5, by=0.1))
ada_model <- train(isFraud ~ amount,
                   data = srs_df_filtered,
                   method = "ada",
                   preProcess = c("scale", "center"),
                   trControl = ctrl,
                   tuneGrid = ada_grid
                   )
summary(ada_model)
ada_predictions <- predict(ada_model)
confusionMatrix(ada_predictions, srs_df_filtered$isFraud, mode = "everything")
```