Anukriti Sharma  
Joseph Burke  
Pranav Naphad  
Sukanya Wadawadagi

Credit Card Fraud DETECTION  
final project

**BANA 7038**

**Outline**

1. Introduction
   1. Dataset
2. Problem Statement
3. General outlook
4. Data exploration
   1. Exploratory Data Analysis
5. Data preparation
6. Logistic Regression Modeling
7. Data Visualization
8. Results
9. Conclusion
10. Appendix
11. **Introduction**

Credit cards are convenient ways to make purchases and allow consumers to make purchases anywhere.  They are not perfect and come with risk to both the consumer and the Financial Institutions that issue them. There are credit card scams, card skimming devices, and Malware/Virus compromises everywhere.  Credit card fraud is growing at a record pace, which makes it critical for Financial Institutions to be able to recognize fraudulent credit card transactions.  When you these fraud events happen, they cost companies millions of dollars and wreak havoc on the lives of consumers.  While all may seem dire, ecommerce ventures and businesses that take online payments can take heart when it comes to credit card fraud detection and protecting data.

Unlike the other machine learning problems, in credit card fraud detection the target class distribution is not equally distributed. It is popularly known as the class imbalance problem or unbalanced data issue. This makes this problem even more challenging to solve. So, through this project we have attempted to answer this problem using three approaches and tried to find out which approach suites the problem better (Random Forest, Logistic Regression, K nearest neighbor)

* 1. **Dataset**

Coming to dataset, we have selected the most popular credit card dataset which was downloaded from Kaggle.com platform. It contains two-day transactions made on 09/2013 by European cardholders. The dataset contains 492 frauds out of 284,807 transactions. Thus, it is highly unbalanced, with the positive (frauds) accounting for only 0.17%. The only features which have not been transformed are ‘Time’ and ‘Amount’. ‘Time’ is the seconds elapsed between each transaction and the first. ‘Amount’ is the transaction amount. ‘Class’ is the response variable with 1 as fraud and 0 otherwise.  Scaling was used to mask 28 of the variables as the underlying data entails confidential and proprietary information.  In the Credit Card Fraud industry, these variables would typically be customer profile data (e.g. typical location, limits, average number transactions), Transactional data (e.g. type of transaction, Point of Sale type, card type), and additional variables created by the analyst (e.g. number of transactions in past 30 mins, card is compromised).  Not knowing the specific variables somewhat hinders our ability to build new variables like an analyst would in the field.  Based on Kaggle.com, this data was cleaned and ready to be modeled.

1. **Problem Statement**

The credit card fraud classification problem is used to identify fraud transactions or fraudulent activities before they become a major problem to credit card companies. To ensure that, we have concentrated more on the fraud which goes undetected than frauds which are not even present but classified incorrectly. As we have realized the undetected fraud if continues over a long period might weaken the system that we cannot recover.

Hence, the major approach is to minimize the cost incurred through false negatives and false positives.

1. **General Outlook**

We have built a logistic regression model using R.

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is (binary).  Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more predictor variables.

Advantages : Easy, fast and a simple classification method. Can be used for multiclass classifications also.

Disadvantages : For it to work properly, a good selection of features is required. Outliers tamper the accuracy of Logistic Regression model as well.

1. **Data Exploration**

Exploratory Data Analysis is a thorough examination meant to uncover the underlying structure of a data set and is important for a company because it exposes trends, patterns, and relationships that are not readily apparent. We performed EDA on our dataset in R language. It helped identify errors, understand patterns within the data, detect outliers among the variables.

These are the following steps we went through to complete EDA.

* Install and load the libraries by using: library(DataExplorer)

str(all\_data)

Text, letter

Description automatically generated

* **We checked the dimension of the input dataset and the time of variables.**

plot\_str(all\_data)

Diagram

Description automatically generated with medium confidence

With that, we saw we have some Continuous variables and some Categorical variables. We can see an overview of each variable with its datatype.

* **Check for Missing values before diving deep in the analysis.**

plot\_missing(all\_data)

Graphical user interface, application

Description automatically generated

In our dataset, there are no missing values.

plot\_density(all\_data)

Graphical user interface, diagram

Description automatically generated

* **Bivariate Analysis**

plot\_correlation(all\_data, type = 'continuous','Review.Date')

Calendar

Description automatically generated

plot\_bar(all\_data)

A picture containing chart

Description automatically generated

We then used the function create\_report() that gives a sharable rendered markdown in html.

The Data Profiling Report can be accessed by double clicking on this html link:

create\_report(all\_data)

Graphical user interface, application

Description automatically generated

1. **Data Preparation**

Preprocessing is the process of cleaning and transforming the dataset. In this cleaning step, we have applied different methods to clean the raw data to feed more meaningful data for the modeling phase. This method includes-

Remove duplicates or irrelevant samples like dropping the unused columns such as date, name, dob, street, latitude, longitude etc. Update missing values with the most relevant values, for example if the customer or user does not fill all information in the forms. Blank values are treated as null or non-values. Although our dataset doesn’t have any null values.

As per below observation, we don’t have any duplicated values in our dataset.

duplicated(all\_data)

A picture containing text

Description automatically generated

We have dropped the unused columns such as date, name, dob, street, latitude, longitude etc.

df = subset(all\_data, select = -c(lat,long,street,city\_pop,first,last,trans\_date\_trans\_time))

A close-up of a document

Description automatically generated with medium confidence

Thus, our dataset is now ready to be built a model upon.

1. **Logistic Regression Modeling**

[Logistic regression](https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/logistic-regression/) is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary).  Like all regression analyses, the logistic regression is a predictive analysis.  Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval, or ratio-level independent variables.

In our case, the target variable is\_fraud is a binary one which is the reason we have carried out logistic regression modeling.

* Loading the required packages-

install.packages("caTools")

install.packages("ROCR")

library(caTools)

library(ROCR)

# Training model

logreg <- glm(formula = is\_fraud ~ amt,

              family=binomial,

              data=traindata)

summary(logreg)

Text, letter

Description automatically generated

* Predict probabilities of default using the logistic model

prob.hat <- predict(logreg, testdata , type='response')

summary(prob.hat)

A picture containing text

Description automatically generated

* Add a dummy variable, default\_hat\_model

testdata <- testdata %>%

  mutate(default\_hat\_model = as.numeric(prob.hat > 0.25))

* Evaluating model accuracy using confusion matrix:

table(testdata[['is\_fraud']],

      testdata[['default\_hat\_model']],

      dnn=c("Actual", "Predicted"))

Text

Description automatically generated

* Capture Rate

100\*mean(testdata[['is\_fraud']] == testdata[['default\_hat\_model']])



* ROC-AUC Curve

predict\_reg <- ifelse(predict\_reg >0.5, 1, 0)

table(testdata$is\_fraud, predict\_reg)

missing\_classerr <- mean(predict\_reg != testdata$is\_fraud)  
print(paste('Accuracy =', 1 - missing\_classerr))

ROCPred <- prediction(predict\_reg, testdata$is\_fraud)  
ROCPer <- performance(ROCPred, measure = "tpr",  
x.measure = "fpr")

auc <- performance(ROCPred, measure = "auc")  
auc <- auc@y.values[[1]]  
auc

Text, letter

Description automatically generated

* Plotting Curve

plot(ROCPer)  
plot(ROCPer, colorize = TRUE,  
print.cutoffs.at = seq(0.1, by = 0.1),  
main = "ROC CURVE")  
abline(a = 0, b = 1)

Chart, line chart, scatter chart

Description automatically generated

auc <- round(auc, 4)  
legend(.6, .4, auc, title = "AUC", cex = 1)

Chart, scatter chart

Description automatically generated

1. **Data Visualization**

*“The simple graph has brought more information to the data analyst’s mind than any other device.”* — John Tukey

Data visualization is perhaps the fastest and most useful way to summarize and learn more about your data. You'll start by exploring the numeric variables individually.

Histograms provide a bar chart of a numeric variable split into bins with the height showing the number of instances that fall into each bin. They are useful to get an indication of the distribution of an attribute.

1. Percentage of Fraud

common\_theme <- theme(plot.title = element\_text(hjust = 0.5, face = "bold"))

ggplot(data = testdata, aes(x = factor(is\_fraud),

                      y = prop.table(stat(count)), fill = factor(is\_fraud),

                      label = scales::percent(prop.table(stat(count))))) +

  geom\_bar(position = "dodge") +

  geom\_text(stat = 'count',

            position = position\_dodge(.9),

            vjust = -0.5,

            size = 3) +

  scale\_x\_discrete(labels = c("no fraud", "fraud"))+

  scale\_y\_continuous(labels = scales::percent)+

  labs(x = 'Fraud', y = 'Percentage') +

  ggtitle("Distribution of Fraud") +

  common\_theme

Chart, bar chart

Description automatically generated

Clearly the dataset is very imbalanced with 99.8% of cases being non-fraudulent transactions. A simple measure like accuracy is not appropriate here as even a classifier which labels all transactions as non-fraudulent will have over 99% accuracy. An appropriate measure of model performance here would be AUC (Area Under the Precision-Recall Curve) which we would be covering later in the project.

1. Distribution of Time Transaction by Fraud

#Distribution of variable 'Time' by class

testdata %>%

ggplot(aes(x = unix\_time, fill = factor(is\_fraud))) + geom\_histogram(bins = 100)+

labs(x = 'Time in seconds since first transaction', y = 'No. of transactions') +

ggtitle('Distribution of time of transaction by Fraud') +

facet\_grid(is\_fraud ~ ., scales = 'free\_y') + common\_theme

Chart

Description automatically generated

1. State-wise Fraud data

fig <- plot\_ly(Category\_data, labels = ~ state, values = ~Fraud, type = 'pie')

fig <- fig %>% layout(title = 'State wise fraud',

                      xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),

                      yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))

fig

Chart, pie chart

Description automatically generated

1. Category-wise Fraud data

FraudCategory <- sqldf(

  'select category,

  count(is\_fraud) as Fraud

  from all\_data

  where is\_fraud = 1

  group by state'

)

ggplot(FraudCategory, aes(x=Fraud, y=category)) +   geom\_bar(stat = "identity" ,fill = "#FF6666")

Chart, bar chart

Description automatically generated

1. **Results**

Biggest Challenge in our business case is the tradeoff between the cost incurred by False Negatives and False positives.

The evaluation metric to use depends heavily on the task at hand. Earlier, accuracy was the only measure we used, which is really a vague option for an imbalanced dataset.

Although PR curve is considered better for a highly imbalanced dataset, we went ahead to evaluate the model using ROC AUC as reducing both False negatives and false positives is important for our case.

ROC stands for receiver operating characteristic and the graph is plotted against TPR and FPR. As TPR increases FPR also increases.

Area Under Curve(AUC) is one of the most widely used metrics for evaluation. The greater the value, the better is the performance of our model.

ROC- AUC Curve

Chart, scatter chart

Description automatically generated

The accuracy of the model comes out to be 99.54%

