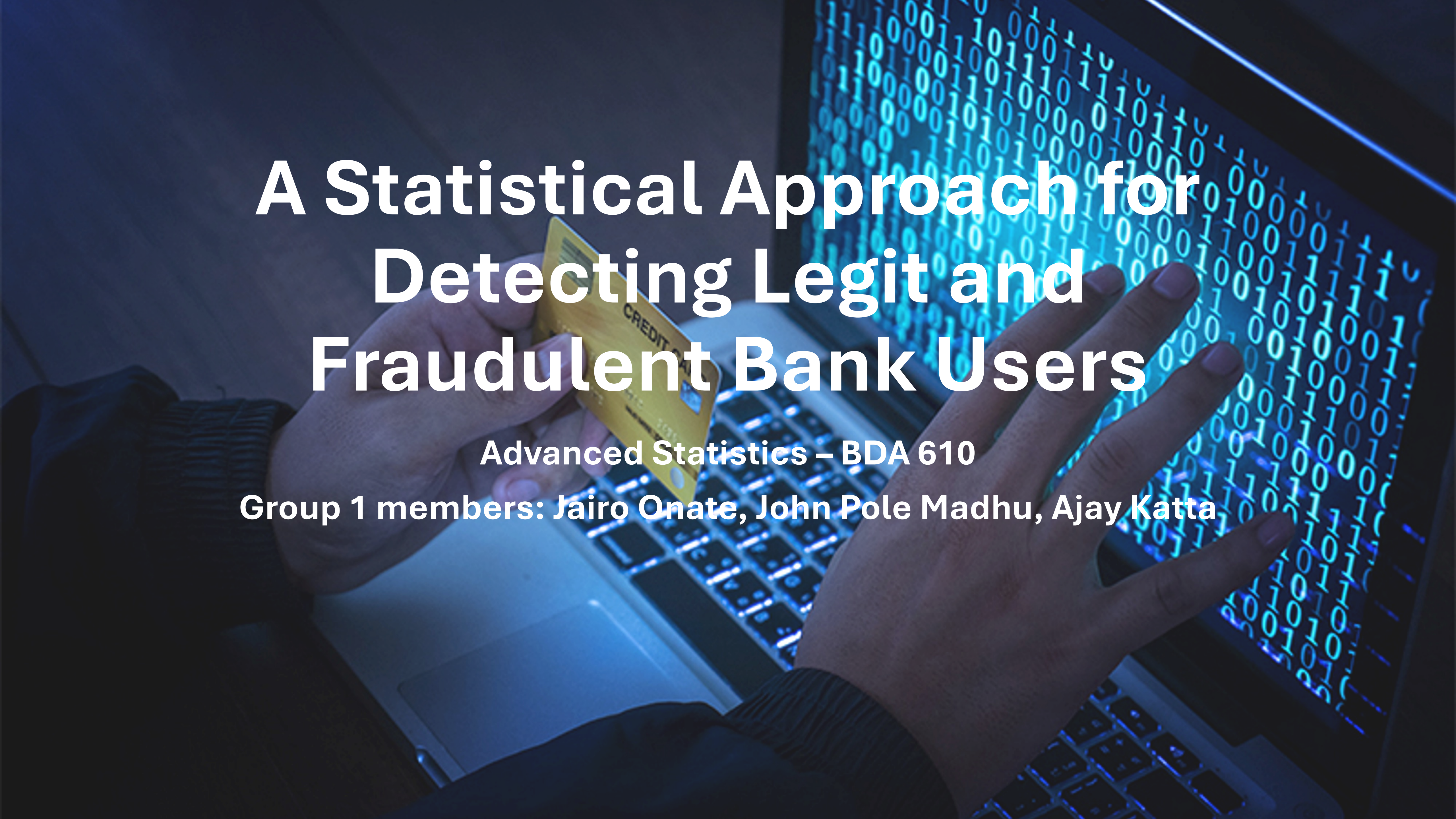


A Statistical Approach for Detecting Legit and Fraudulent Bank Users

Advanced Statistics – BDA 610

Group 1 members: Jairo Onate, John Pole Madhu, Ajay Katta

CONTENTS

INTRODUCTION TO BANK FRAUD

Background

- By the 1980s, advances in technology made computers accessible, leading to online banking services. Wells Fargo launched the first online platform in 1995. However, this evolution also opened doors to new types of fraud.

Problem

- In 2023, consumer losses from fraud in the U.S. totaled $10 billion (FTC). The most common fraud types were investment scams ($4.6B) and imposter scams ($2.7B).

Objective

- How can classification algorithms distinguish between legit and fraudulent bank users?

Data Source

- The dataset, titled ***Bank Account Fraud Dataset (NeurIPS 2022)***, is part of the **NeurIPS 2022 competition** and is focused on the detection of fraudulent activities associated with bank accounts

Key Factors

- Analyzing customer characteristics like annual income, email/legal name similarity, age, transfer amounts, and address history to classify a user's application as legit or fraudulent.

MODELING: DATA MANIPULATION

Dimensionality and Reduction

- Records with variables containing −1 and negative values specified by the authors as missing values were excluded from the data selection.
- Dimensionality reduction from 1,000,000 to 124,260 records with 32 variables.

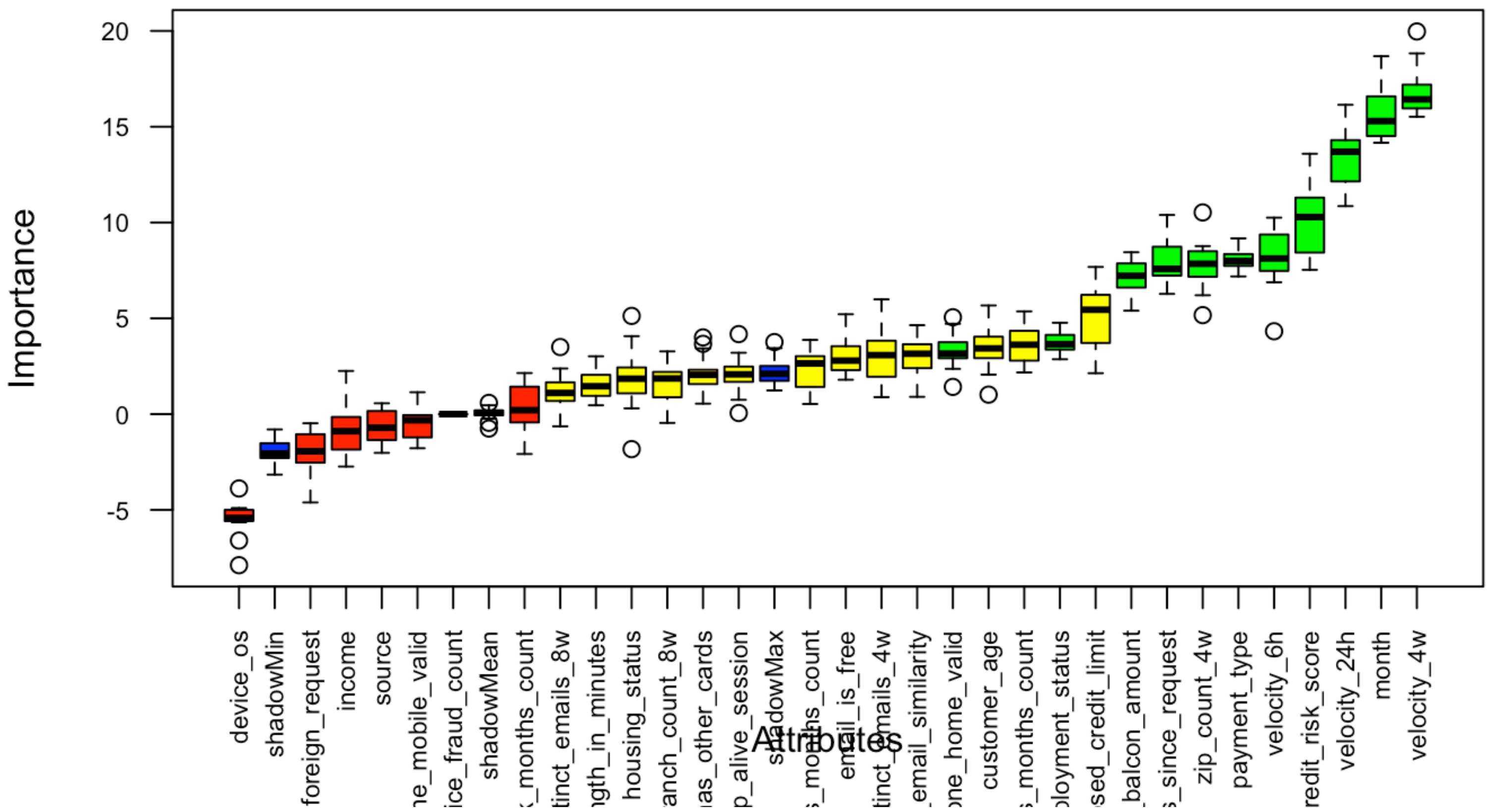
Challenges:

1. Feature Selection:

Aim: Selecting the best variables to build the model.

Tool: Boruta package which uses Random Forest to classify the importance of the attributes.

MODELING: FEATURE SELECTION



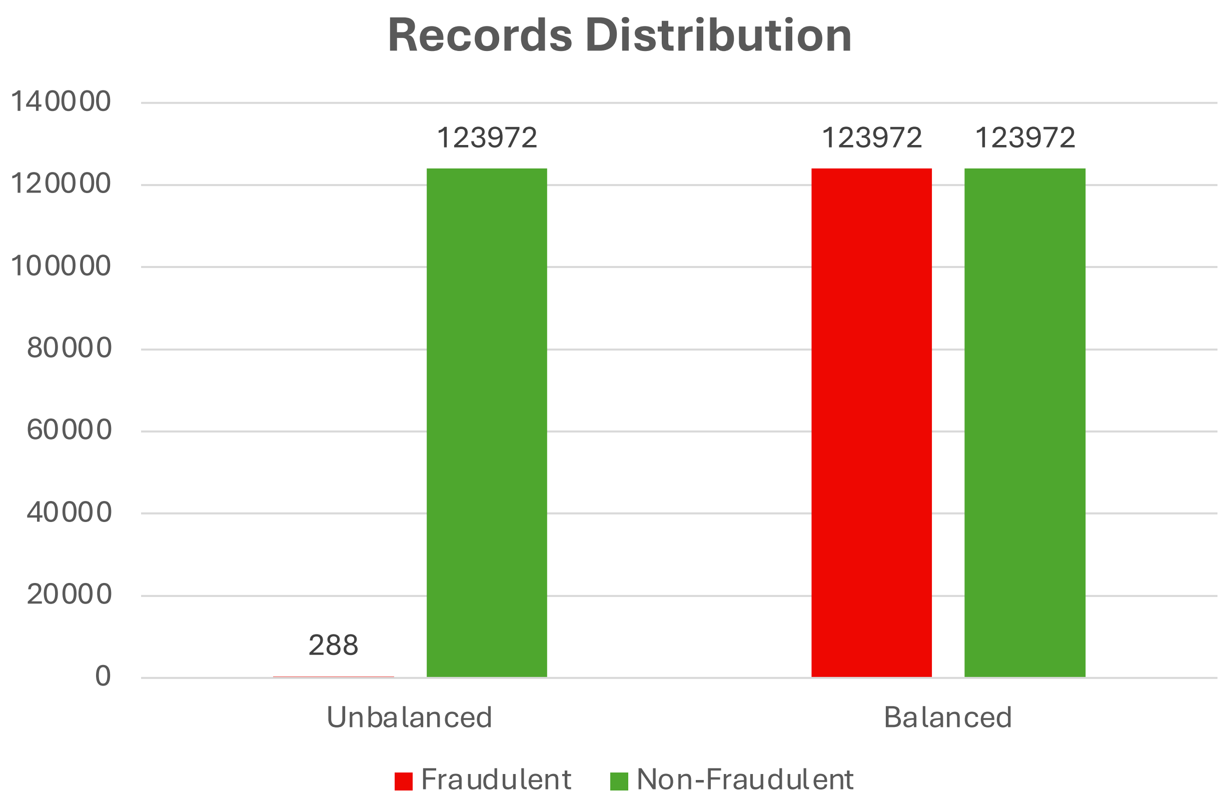
MODELING: BALANCING DATA

Challenges:

2. Balancing data:

Aim: Helping the model to prevent becoming biased towards one class.

Tool: upSample method



CLASSIFICATION MODELS RESULTS

Model selected: Decision Tree

Key aspects

- **Assessment of Performance:** With 78.21% accuracy, 77.36% sensitivity, and 79.06% specificity, the Decision Tree model proved to be an excellent classifier of both fraudulent and non-fraudulent transactions.
- **Simplicity:** The concept is beneficial in real-time contexts where speed and transparency are crucial because it is computationally efficient and easy to apply.

Decision Tree - Accuracy: 78.21%		
	Reference	
Prediction	0	1
0	29456	8405
1	7801	28722

Logistic Regression - Accuracy: 70.56%		
	Reference	
Prediction	0	1
0	26714	11353
1	10543	25774

Random Forest - Accuracy: 100%		
	Reference	
Prediction	0	1
0	37257	0
1	0	37257

RECOMMENDATIONS

- The Decision Tree model is well-suited for business applications where transparency is a priority.
- It offers a good balance between accuracy and ease of interpretation.
- If a model reaches 100% accuracy, it likely indicates overfitting, meaning it won’t generalize well to new data.
- Overfitting can result from excessive model complexity or data leakage. Reason why expecting 100% accuracy in business is unrealistic due to inherent data uncertainty.
- It's more important to emphasize reliable performance metrics like precision and recall than aiming for perfect accuracy.
- Decision Tree provides a more accessible and realistic approach to fraud detection, being essential for maintaining public trust in the financial industry.

THANK YOU

FEATURE SELECTION

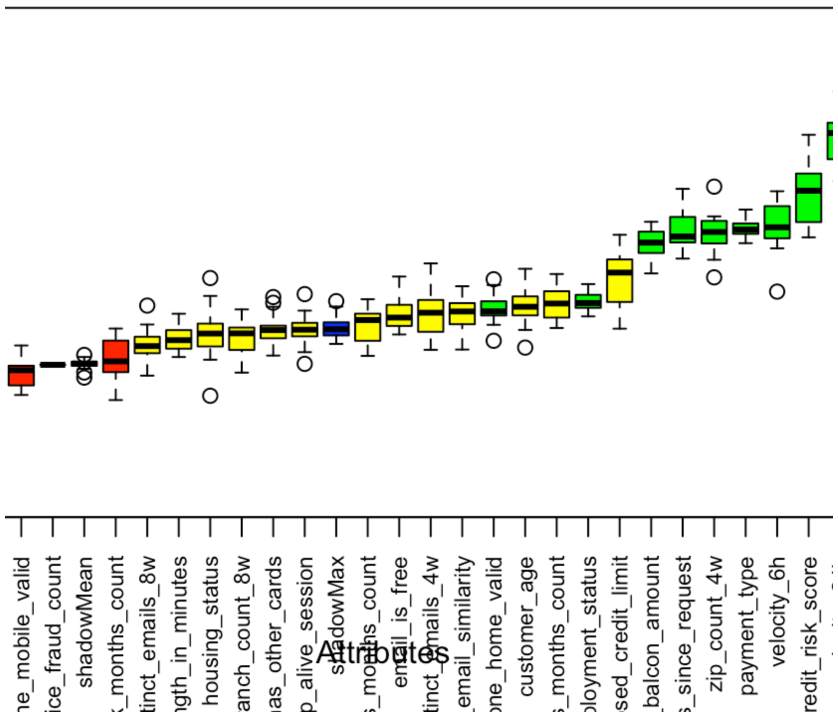
```
install.packages('Boruta')
library(Boruta)
set.seed(555)
bankfraud_selection_br <- Boruta(fraud_bool ~ ., data = bankfraud_base_nomissing, doTrace = 2, maxRuns = 20)
```

The following variables have been selected and marked with green:

- phone_home_valid, employment_status, intended_balcon_amount, days_since_request, zip_count_4w, payment_type, velocity_6h, velocity_6h, credit_risk_score, velocity_24h, month, velocity_4w

Yellow variables where Boruta was undecided:

- proposed_credit_limit, proposed_credit_limit, bank_months_count, customer_age, email_is_free, session_length_in_minutes



DATA PARTITIONING

```
nrows_bankfraud <- nrow(bankfraud_base_nomissing)

set.seed(128) index <- sample(1:nrows_bankfraud, 0.7 * nrows_bankfraud) # We are using a 70-30 rule to approach the partition for training and testing

train_bankfraud <- bankfraud_base_nomissing[index,]

test_bankfraud <- bankfraud_base_nomissing[-index,]

table(train_bankfraud$fraud_bool)
```

Records Distribution			
	Non Fraudulent	Fraudulent	Total Records
Intial dataset	988,971	11,029	1,000,000
Filtered dataset	86,786	196	86,982

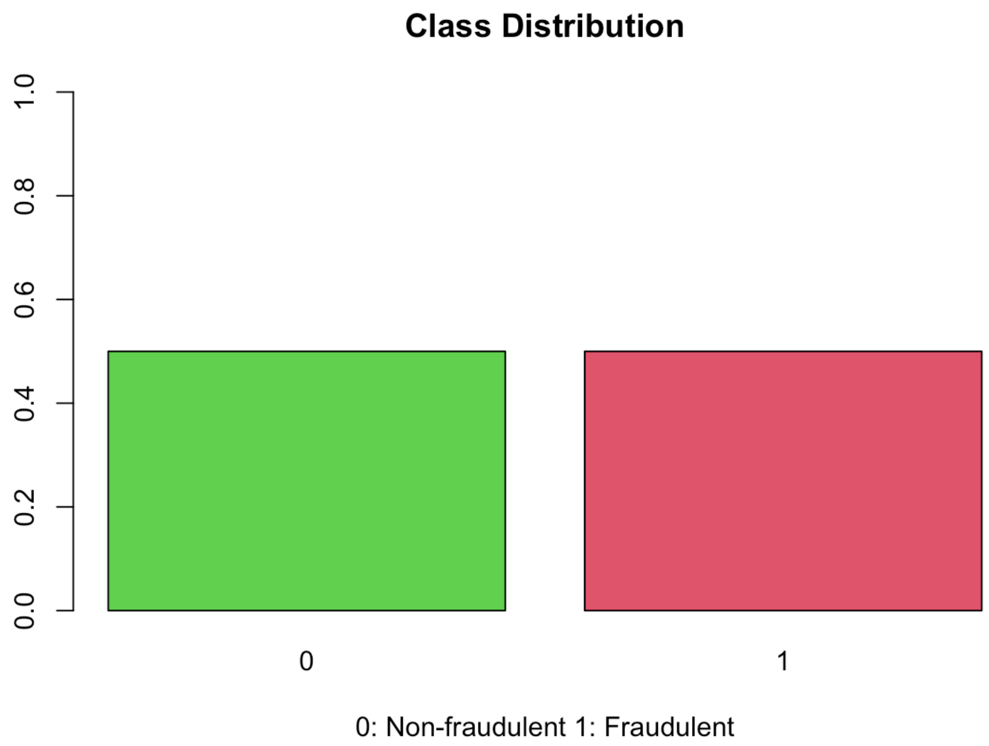
CLASS BALANCING

```r
library(caret)

# First we convert the outcome variable fraud_bool as a factor before using the function upSample

bankfraud_base_nomissing$fraud_bool <- as.factor(bankfraud_base_nomissing$fraud_bool)


bankfraud_upsample <- upSample(
x = bankfraud_base_nomissing,
y = bankfraud_base_nomissing$fraud_bool)
```



MODELING: DECISION TREE

```
decisiontree_bankfraud <- ctree(fraud_bool ~ phone_home_valid +
                                intended_balcon_amount +
                                days_since_request +
                                zip_count_4w +
                                velocity_6h +
                                credit_risk_score +
                                velocity_24h + month +
                                velocity_4w +
                                proposed_credit_limit +
                                proposed_credit_limit +
                                bank_months_count +
                                customer_age +
                                email_is_free +
                                session_length_in_minutes,
data = train_bankfraud_upsample, control = ctree_control(maxdepth = 7))
```

Statistics - Decision Tree	
Accuracy	0.7821
95% CI	(0.7791, 0.7851)
No Information Rate	0.5009
P-Value [Acc > NIR]	< 2.2e-16
Sensitivity	0.7736
Specificity	0.7906

MODELING: LOGISTIC REGRESSION

```r
logistic_bankfraud <- glm(fraud_bool ~ phone_home_valid +
                          employment_status +
                          intended_balcon_amount + days_since_request +
                          zip_count_4w + payment_type +
                          velocity_6h + credit_risk_score +
                          velocity_24h + month + velocity_4w +
                          proposed_credit_limit + proposed_credit_limit +
                          bank_months_count + customer_age +
                          email_is_free + session_length_in_minutes ,
data = train_bankfraud_upsample, family = 'binomial')
```

Statistics - Logistic Regression	
Accuracy	0.7056
95% CI	(0.7023, 0.7089)
No Information Rate	0.5009
P-Value [Acc > NIR]	< 2.2e-16
Sensitivity	0.6942
Specificity	0.717

MODELING: RANDOM FOREST

```
randomforest_bankfraud <- randomForest(fraud_bool ~ phone_home_valid +
                                        employment_status + intended_balcon_amount +
                                        days_since_request + zip_count_4w + payment_type +
                                        velocity_6h + credit_risk_score + velocity_24h +
                                        month + velocity_4w + proposed_credit_limit +
                                        proposed_credit_limit + bank_months_count +
                                        customer_age + email_is_free +
                                        session_length_in_minutes ,
data = train_bankfraud_upsample, ntree = 500, proximity = F, importance = T)
```

Statistics - Random Forest	
Accuracy	1
95% CI	(1, 1)
No Information Rate	0.5009
P-Value [Acc > NIR]	< 2.2e-16
Sensitivity	1
Specificity	1