Credit Card Fraud Data Analysis

Rajeev Thomas Saganty, Manisha Karim, Alonzo Velez

Dataset Overview

Dataset: Credit Card Fraud Prediction [1]

- 555,719 instances and 22 attributes
- Target variable (is_fraud = 1/0)

- trans_date_trans_time
- Cc_num
- merchant
- Category
- amt
- First
- last
- gender
- Street
- city
- State
- Zip
- Lat
- long
- city_pop
- job
- dob
- trans_num
- unix_time
- merch_lat
- merch_long

1. https://www.kaggle.com/datasets/kelvinkelue/credit-card-fraud-prediction/data

Exploratory Data Analysis

- ~~trans_date_trans_time~~
- Category
- amt
- gender
- city_pop
- job
- ~~trans_num~~
- ~~unix_time~~

- ~~Street~~
- ~~city~~
- ~~State~~
- Zip
- ~~Lat~~
- ~~Long~~
- ~~merch_lat~~
- ~~merch_long~~

- ~~Cc_num~~
- merchant
- ~~First~~
- ~~last~~
- ~~dob~~

- Category
- amt
- gender
- city_pop
- job
- Zip

Exploratory Data Analysis

Target Variable

Is_fraud:

1(fraud)	0(not fraud)
2145	553574

TRAIN

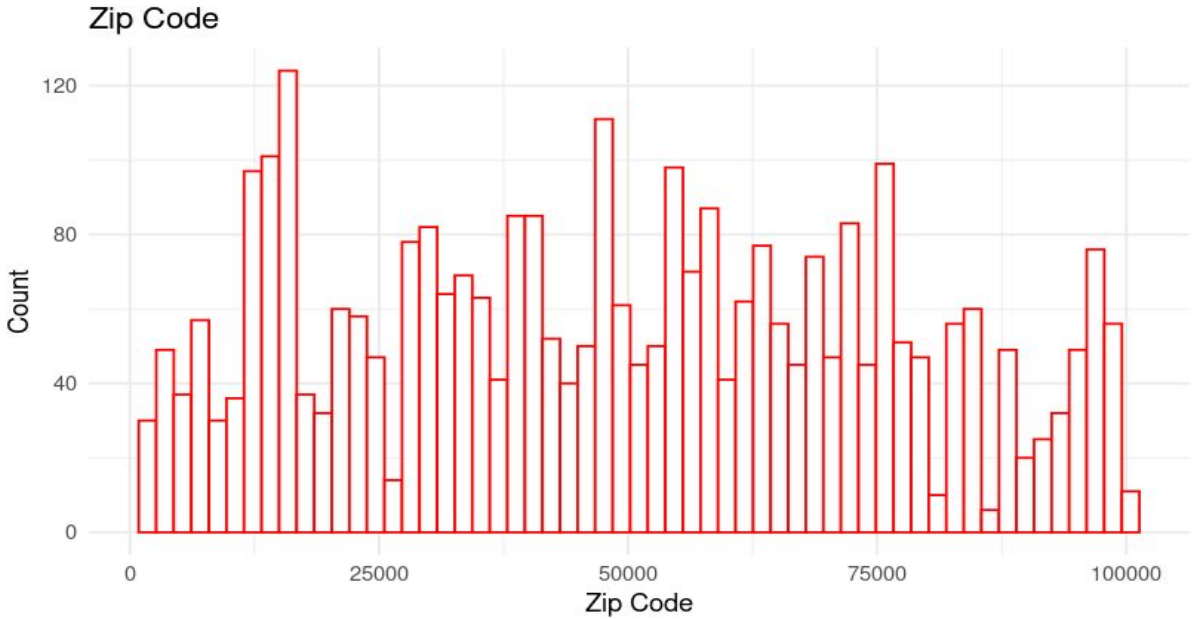
1(fraud)	0(not fraud)
1621	1596

TEST

1(fraud)	0(not fraud)
533	540

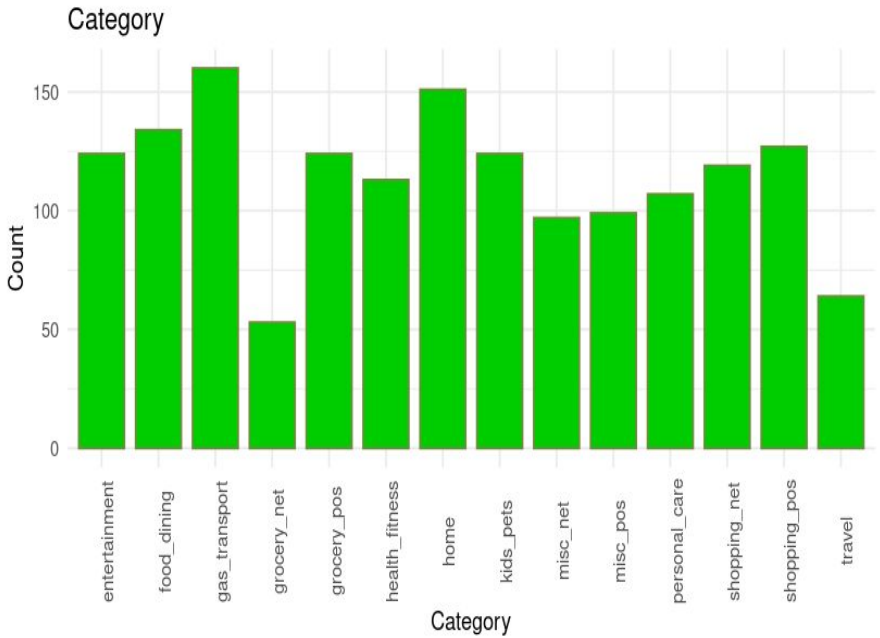
Exploratory Data Analysis

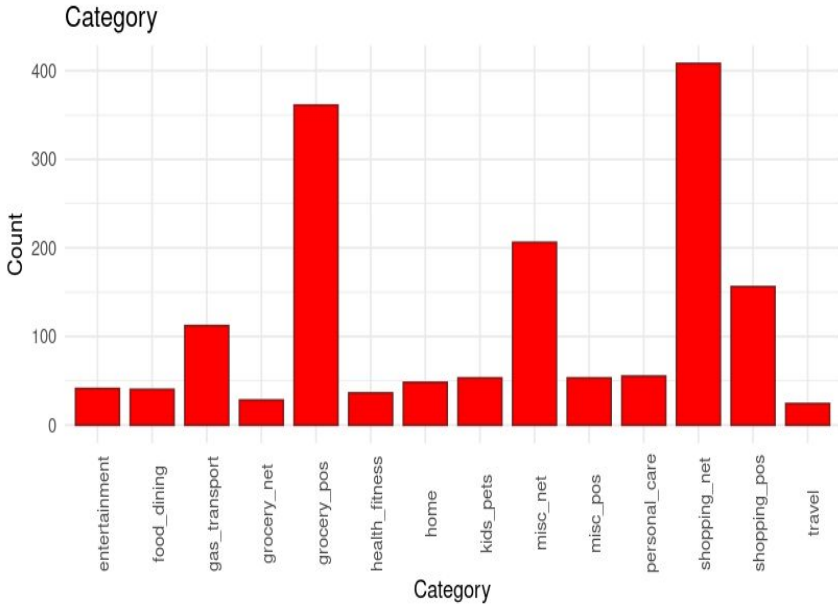
ZIP CODE



Exploratory Data Analysis

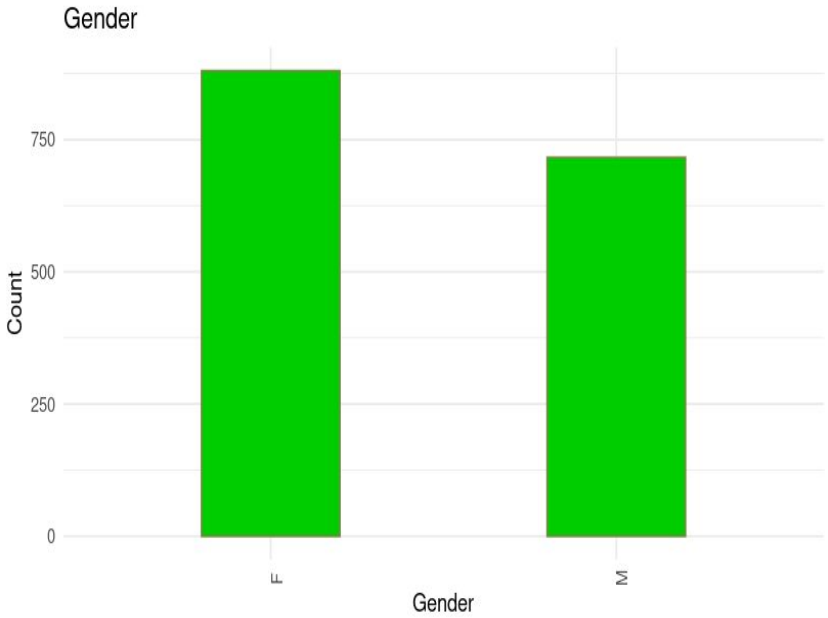
Category

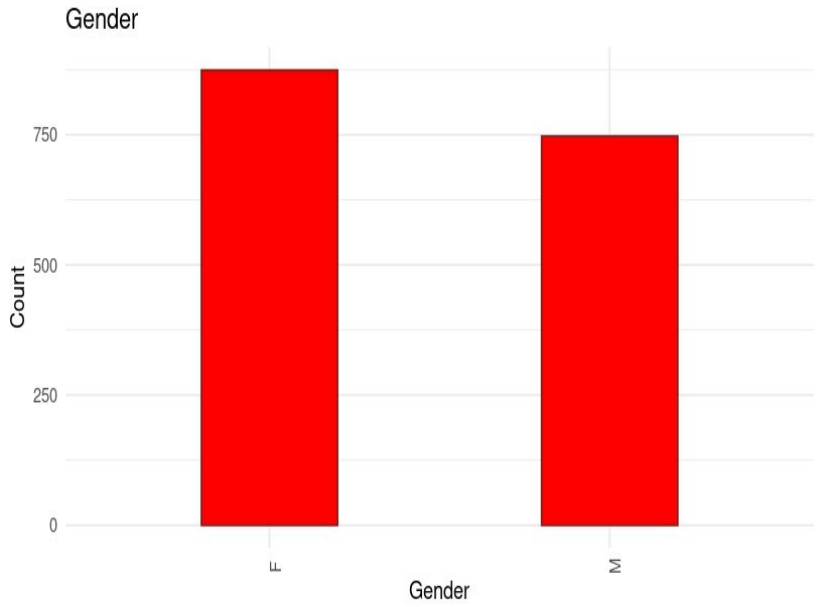




Exploratory Data Analysis

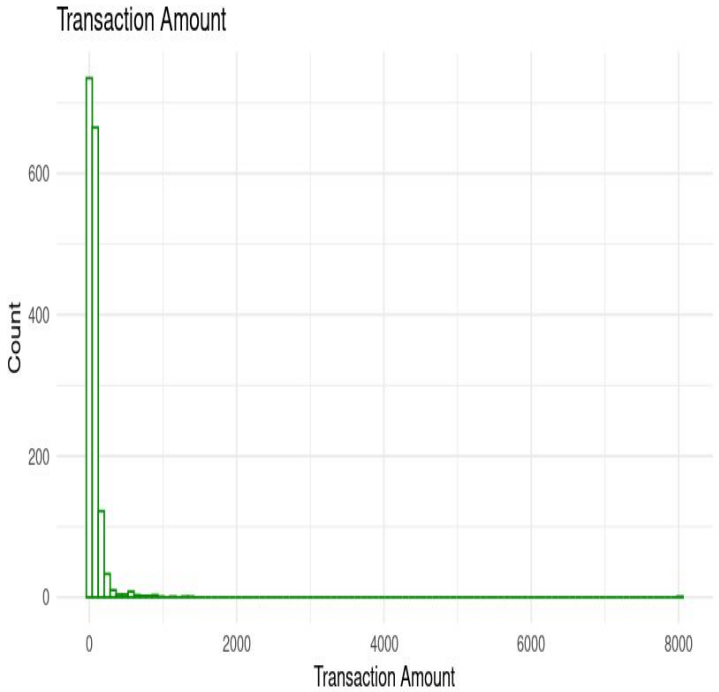
Gender

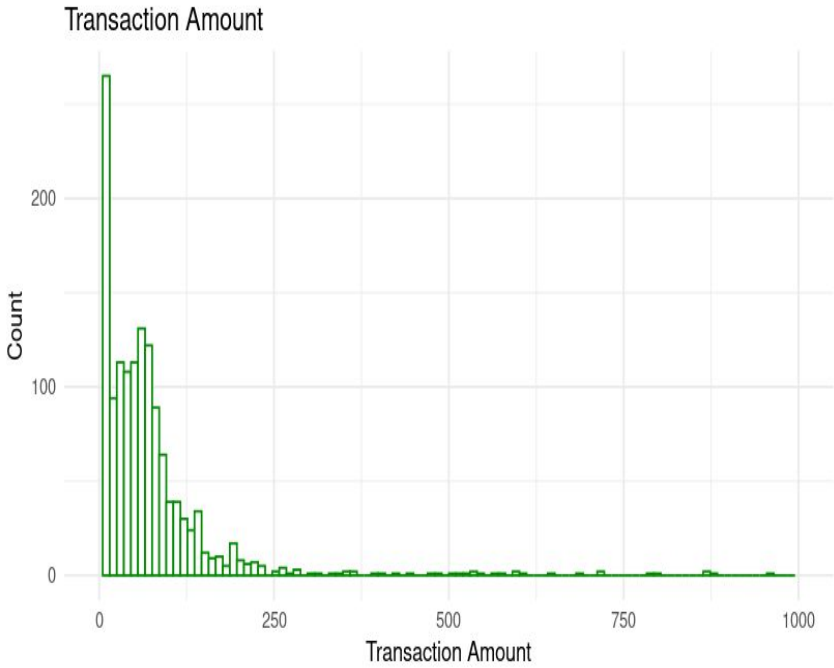




Exploratory Data Analysis

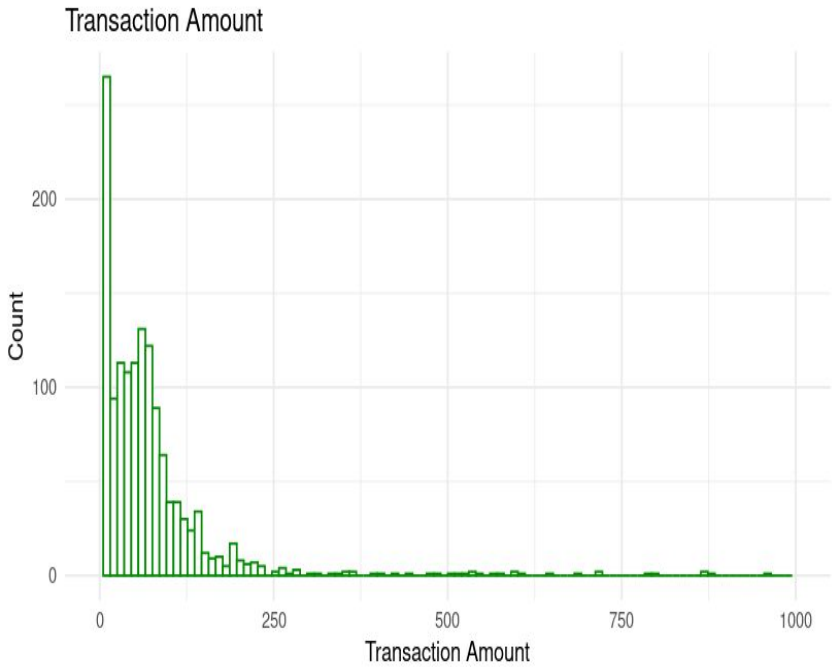
Transaction Amount

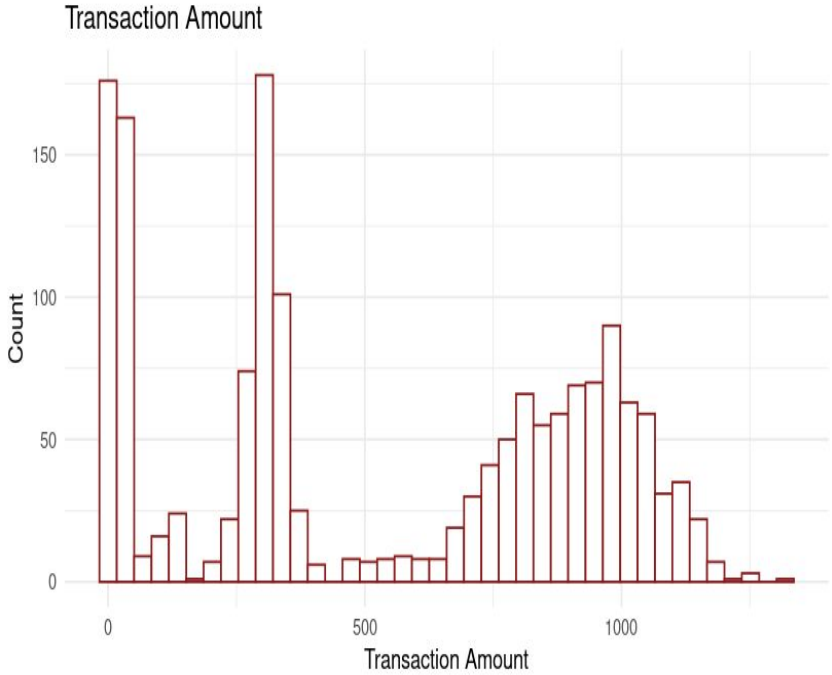




Exploratory Data Analysis

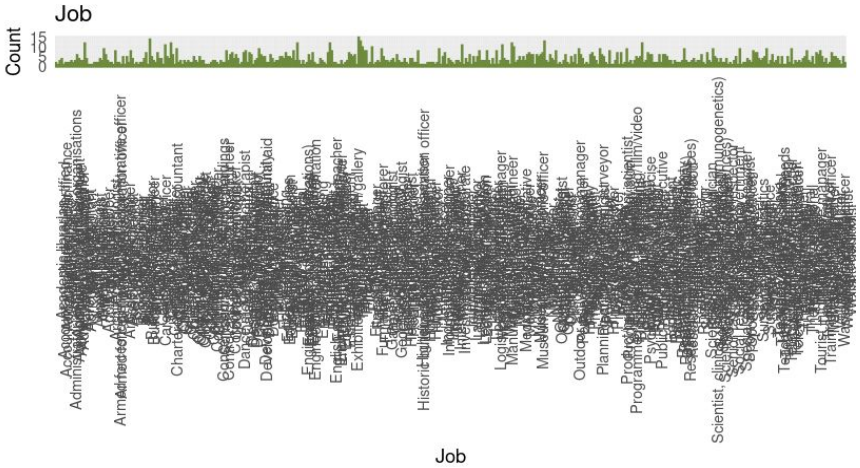
Transaction Amount

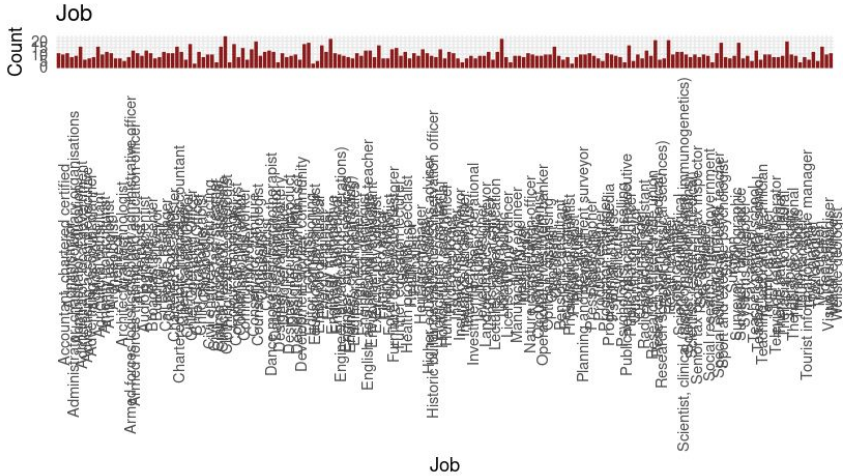




Exploratory Data Analysis

JOB





Number of unique values = 478

Pre-Processing

- One Hot encode categorical data (category).
- Standardise the data

$$x' = \frac{x - \bar{x}}{\sigma(x)}$$

KNN

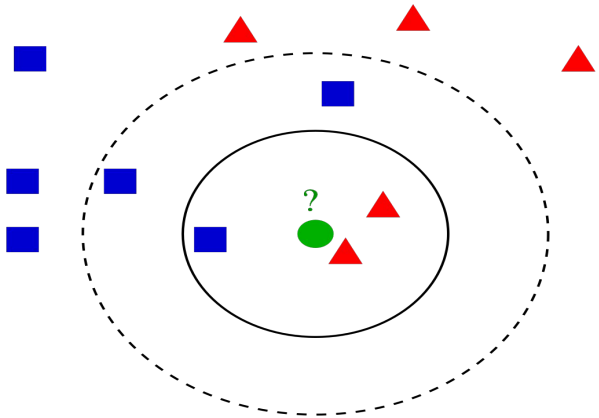
- The KNN algorithm is based on the idea that similar data points tend to belong to the same class or have similar values.
- It calculates the distance between the data point we want to classify and all other data points in the dataset.
- The class of the majority of the K nearest data points is assigned to the data point being classified.

KNN

- Compute the distance between the new data point and all other data points using a chosen distance metric

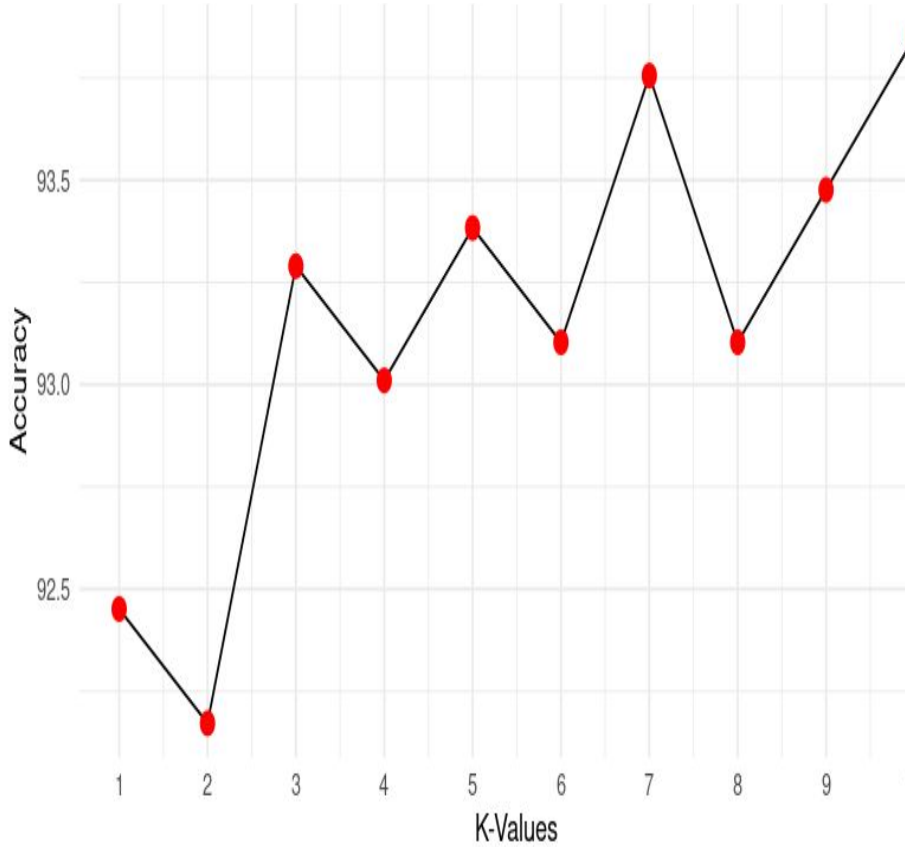
$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

- K is the number of closest neighbors to be considered. Identify the K nearest neighbors to the new data point based on the calculated distances.
- Evaluate the algorithm's performance using metrics like accuracy.
- Select the optimal value of K.



KNN

- Highest Accuracy : 94% (approx.)
- Optimal Number of Neighbors: 7

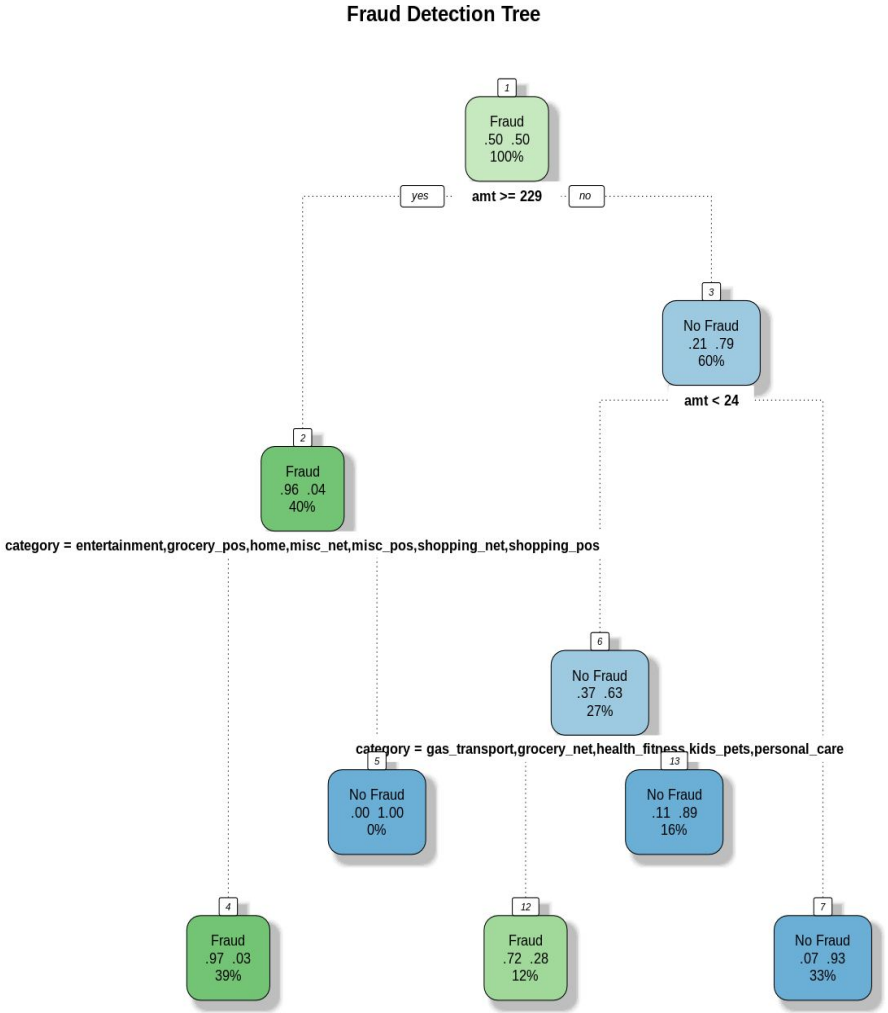


Tree Classifier

- Tree based classifiers use a tree like structure to divide data into separate classes.
- Uses boolean statements based on data variables provided which most accurately separate the classes from one another.
- Example (gender = F) is either true or false. Samples which have gender = F are separated from gender = M

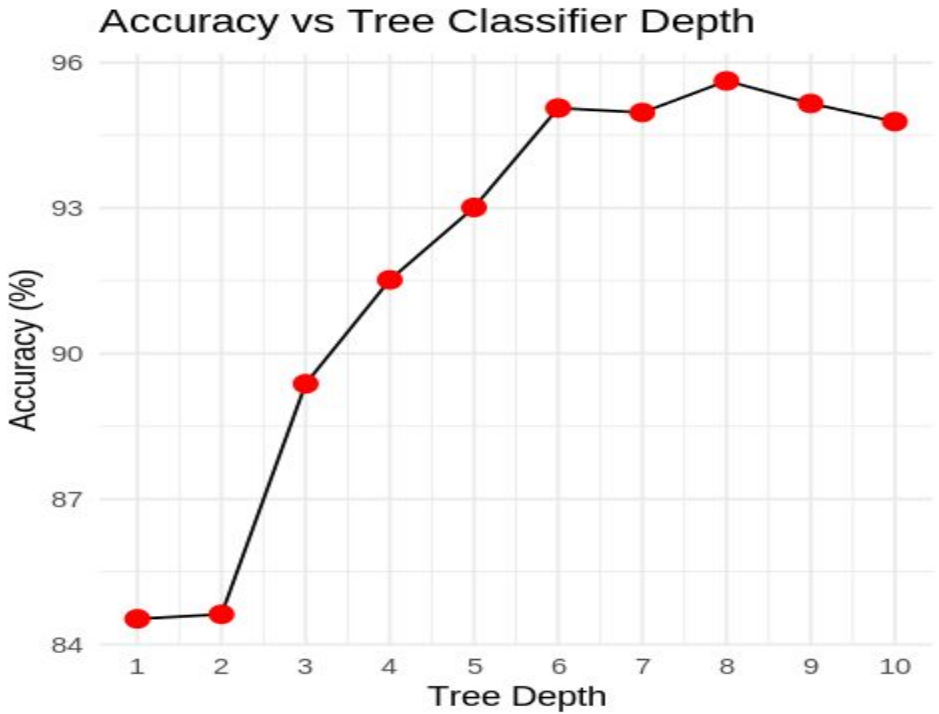
Tree Classifier (Depth = 3)

- Root node splits based on whether the transaction amount is greater than or equal or less than $229.
- 40% of transactions are above $229 are predicted as fraudulent with 96% of those samples being fraudulent.
- 60% of transactions are less than $229. Of those 79% are non fraudulent.
- Of variables available in dataset to predict fraud, transaction amount and category most useful.



Standard Tree

- Optimal tree classifier depth: 8
- Tree depth 3 accuracy: 89.37%
- Tree depth 8 accuracy: 95.61%



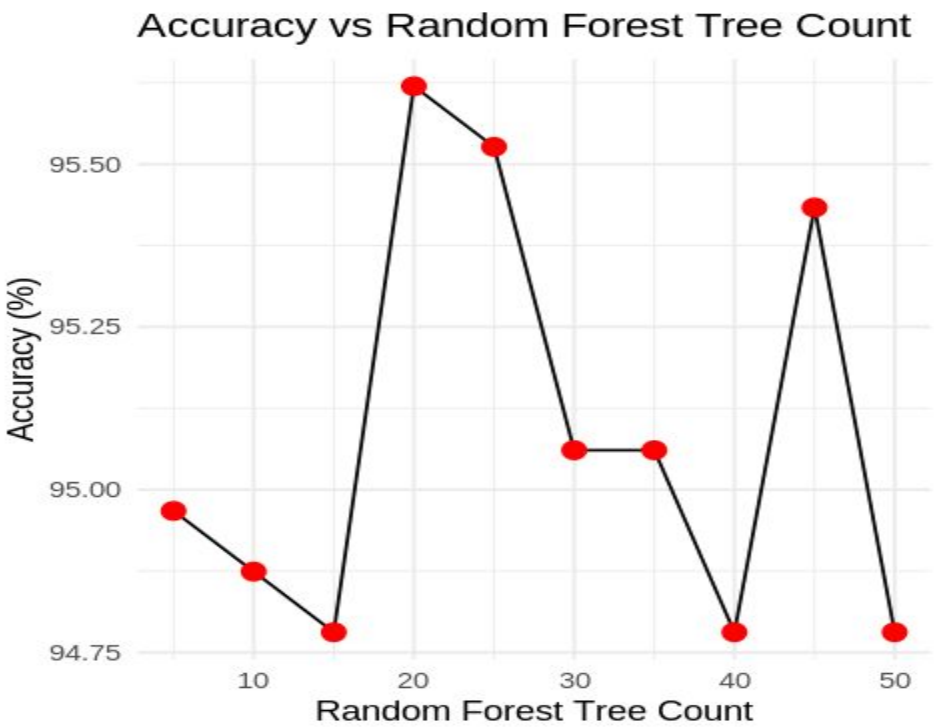
test_predictions	0	1
Fraud	60	479
No Fraud	480	54

Random Forest

- Combines multiple tree based classifiers together to ideally improve classification performance.
- Main parameter to control is the number of trees used in the forest.

Random Forests

- Highest accuracy: 95.75%
- Optimal tree count: 20
- Nearly identical performance to regular tree classifier in this case.



pred.tree.test	0	1
0	520	36
1	20	497

Conclusions

- After accounting for the large class discrepancy between fraudulent and non fraudulent cases the models are able to discriminate between fraudulent and non fraudulent samples to a large extent
- Both KNN and tree based methods have near identical performance
- Tree-based techniques can handle categorical data, therefore the model can be rerun using attributes that were removed because they had a lot of unique values.