Fraud-Detection-in-Online-Transactions Using Machine Learning

**Group Members:**

* **Philip Daniel Chevuri**
* **Thanuja Sirasanambati**
* **Banu Werma Khallepalli.**

1

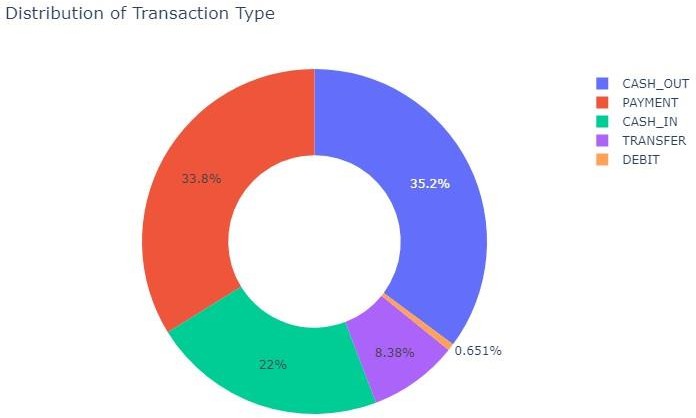
# Table Of Contents

|  |  |  |
| --- | --- | --- |
| **S.NO** | **DESCRIPTION** | **PAGE NO** |
| 1 | Context | 3 |
| 2 | Introduction | 4,5 |
| 3 | Doctoral thesis, comprehensive summary | 5 |
| 4 | Machine Learning | 6-7 |
| 5 | Use Case of Machine Learning in Online Transactions. | 8 |
| 6 | Model Training | 10 |
| 7 | Precision and Recall | 11 |
| 8 | Indicators for Fraud Transactions | 11 |
| 9 | Feature Engineering | 12-18 |
| 10 | Conclusion | 19 |
| 11 | References | 20 |
| 12 | Appendix | 20 |

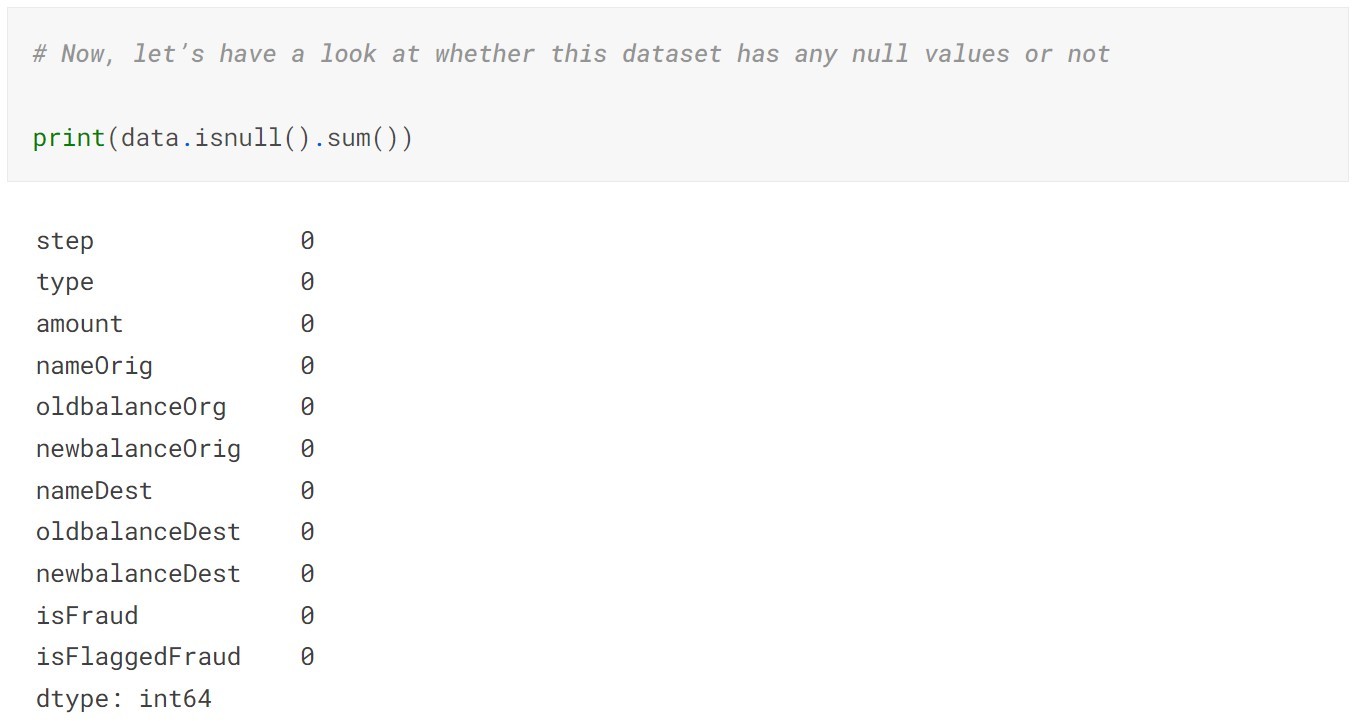
**Project Proposal**

My project is about Online Fraud Detection Using Machine Learning. This is a financial Based Concept which I want to find whether the transaction is Fraud or Not (0 or 1). So, I have searched some of the work of others, I find “Jainil Shah work, I am very Impressed with his work, he has used Exploratory Analysis in a right way, he mentioned donut pie chart, for Transaction analysis. He used Decision Tree algorithm which he got 99% of highest Accuracy. Since the Dataset is clean but the thing is, the records/rows of the dataset is huge. It is very hard to run this dataset due to 6 million records. Although Jainil handled the project in a right way.

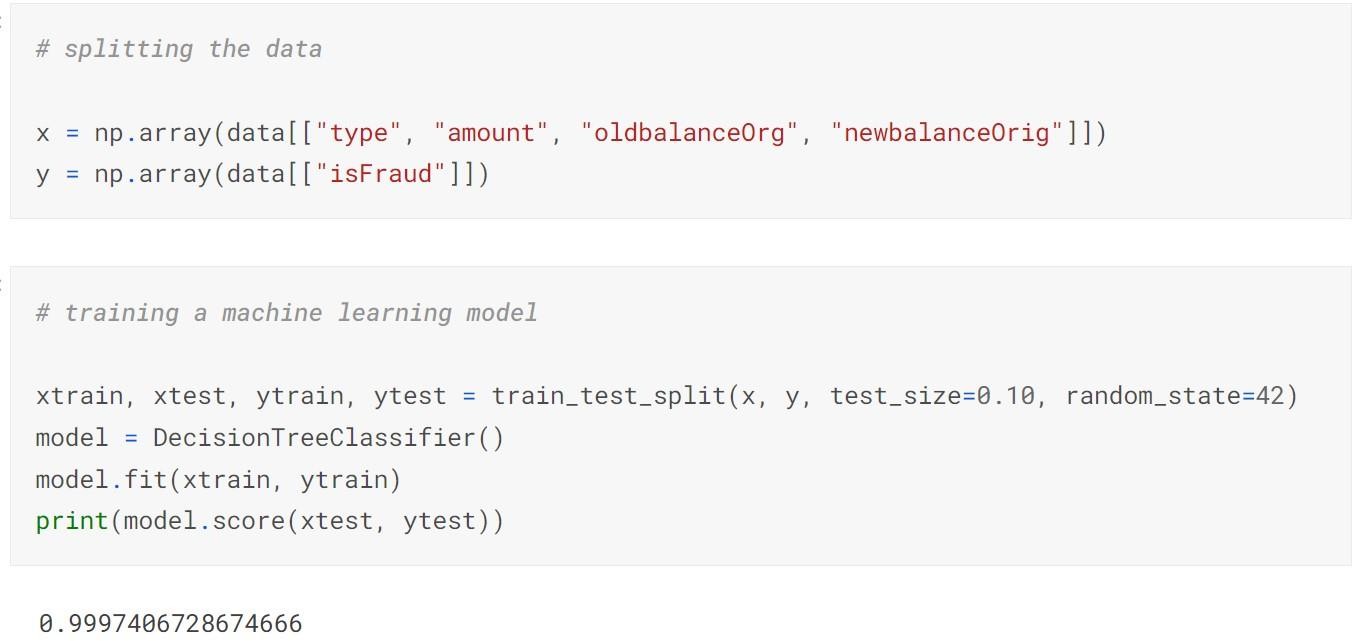




This is a Donut pie chart, here we are analysing the kind of transactions were happened. There are 5 types of transactions were recorded. i.e., Cash out, payment, cash in, transfer, debit.







He Dropped the unwanted attributes, Because I think, if we put the unwanted columns, the algorithm will get biased or will take hard to train the model. In decision tree algorithm he got 99% accuracy.



He also checked the model performance with new data, he got accurate answer.

The link of his Approach: [https://www.kaggle.com/code/jainilcoder/fraud-detection-99-](https://www.kaggle.com/code/jainilcoder/fraud-detection-99-97-accuracy) [97-accuracy](https://www.kaggle.com/code/jainilcoder/fraud-detection-99-97-accuracy)

# Context

There is a lack of publicly available datasets on financial services and specially in the emerging mobile money transactions domain. Financial datasets are important to many researchers and in particular to us performing research in the domain of fraud detection. Part of the problem is the intrinsically private nature of financial transactions, that leads to no publicly available datasets.

We present a synthetic dataset generated using the simulator called PaySim as an approach to such a problem. PaySim uses aggregated data from the private dataset to generate a synthetic dataset that resembles the normal operation of transactions and injects malicious behaviour to later evaluate the performance of fraud detection methods.

**Content**

PaySim simulates mobile money transactions based on a sample of real transactions extracted from one month of financial logs from a mobile money service implemented in an African country. The original logs were provided by a multinational company, who is the provider of the mobile financial service which is currently running in more than 14 countries all around the world.

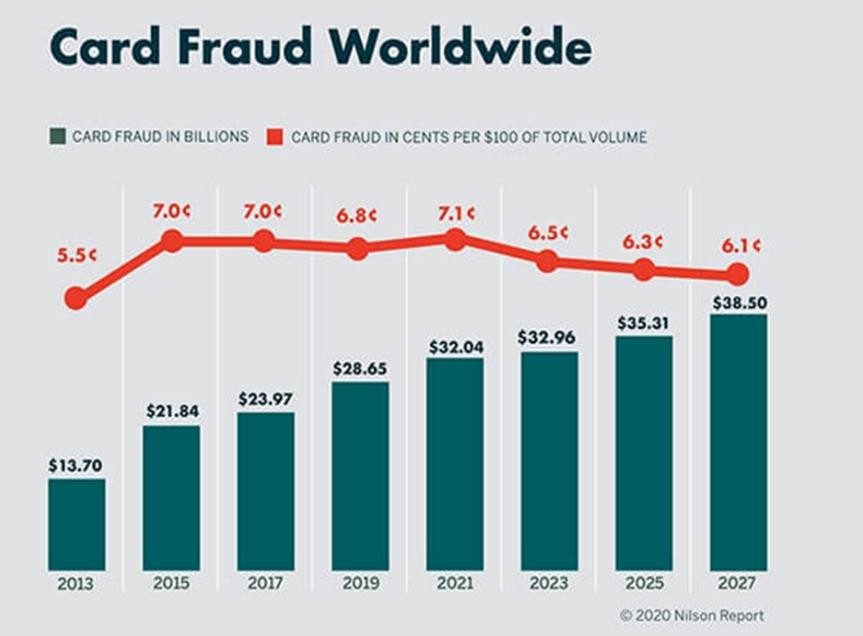
This synthetic dataset is scaled down 1/4 of the original dataset and it is created just for Kaggle.

## INTRODUCTION

The introduction of online payment systems has helped a lot in the ease of payments. But, at the same time, it increased in payment frauds. Online payment frauds can happen with anyone using any payment system, especially while making payments using a credit card. That is why detecting online payment fraud is very important for credit card companies to ensure that the customers are not getting charged for the products and services they never paid.

## Card Fraud World Wide

Payment card fraud affects everyone. Almost 30 billion dollars were lost worldwide in card fraud and identity theft only in 2019. Although financial institutions are locked in an escalating arms race against cybercriminals and scammers, losses still have to be accounted for. Consumers end up paying for money lost to fraud out of pocket, in the form of vendor and transaction fees. While corporations and governments spend more billions investigating and handling fraud cases.



## Doctoral thesis, comprehensive summary (Other academic)

This thesis introduces a financial simulation model covering two related financial domains: Mobile Payments and Retail Stores systems.

The problem we address in these domains is different types of fraud. We limit ourselves to isolated cases of relatively straightforward fraud. However, in this thesis the ultimate aim is to introduce our approach towards the use of computer simulation for fraud detection and its applications in financial domains. Fraud is an important problem that impact the whole economy. Currently, there is a lack of public research into the detection of fraud. One important reason is the lack of transaction data which is often sensitive. To address this problem, we present a mobile money Payment Simulator (PaySim) and Retail Store Simulator (RetSim), which allow us to generate synthetic transactional data that contains both: normal customer behaviour and fraudulent behaviour.



## Machine Learning

Fraud Detection Using Machine Learning deploys a machine learning (ML) model and an example dataset of credit card transactions to train the model to recognize fraud patterns. The model is self-learning which enables it to adapt to new, unknown fraud patterns.

Use this Guidance to automate the detection of potentially fraudulent activity, and the flagging of that activity for review. Fraud Detection Using Machine Learning is easy to deploy and includes an example dataset but you can modify the code to work with any dataset.

## Why is machine learning suited to fraud detection? Super-Fast

When it comes to fraud decisions, you need results FAST! Research shows that the longer a buyer’s journey takes the less likely they are to complete checkout.

Machine learning is like having several teams of analysts running hundreds of thousands of queries and comparing the outcomes to find the best result - this is all done in real-time and only takes milliseconds.

As well as making real-time decisions, machine learning is assessing individual customer behaviour as it happens. It’s constantly analysing ‘normal’ customer activity, so when it spots an anomaly, it can automatically block or flag a payment for analyst review.

## Scalable

Every online business wants to increase its transaction volume. With a rule only system, increasing amounts of payment and customer data puts more pressure on the rules library to expand. But with machine learning it’s the opposite - the more data the better.

Machine learning systems improve with larger datasets because this gives the system more examples of good and bad e.g., genuine and fraudulent customers. This means the model can pick out the differences and similarities between behaviours more quickly and use this to predict fraud in future transactions.

## More Accurate

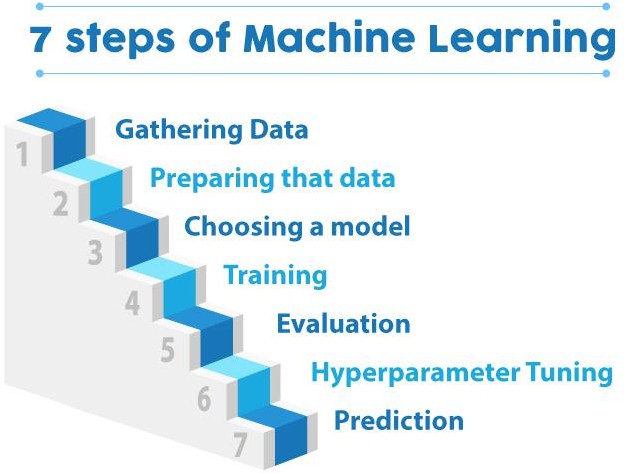
In the same way, machine learning can often be more effective than humans at uncovering non-intuitive patterns or subtle trends which might only be obvious to a fraud analyst much later.

Machine learning models are able to learn from patterns of normal behaviour. They are very fast to adapt to changes in that normal behaviour and can quickly identify patterns of fraud transactions.

This means that the model can identify suspicious customers even when there hasn’t been a chargeback yet. For example, a neural network can look at suspicious signals such as how many pages a customer browses before making an order, determine whether they are copying and pasting information by resizing their windows and flag the customer for review.

## How does a machine learning system work?

We use a few different forms of machine learning at Ravelin - here’s a simple explanation of how a supervised machine learning system works. Listen to this podcast to hear more detail about the process.



## Input Data

When it comes to fraud detection, the more data the better. For supervised machine learning, the data must be labelled as good (genuine customers who have never committed fraud) or bad (customers with a chargeback associated with them or have been manually labelled as fraudsters).

## Extract Features

Features describe customer behaviour, and fraudulent behaviours are known as fraud signals. At Ravelin, we group features into five main categories, each of which has hundreds or thousands of individual features:

## Identity

Number of digits in the customer’s email address, age of their account, number of devices customer was seen on, fraud rate of customer's IP address.

**Orders**

Number of orders they made in their first week, number of failed transactions, average order value, risky basket contents.

**Payment Methods**

Fraud rate of issuing bank, similarity between customer name and billing name, cards from different countries.

**Locations**

Shipping address matches the billing address, shipping country matches country of customer's IP address, fraud rate at customer’s location.

## Network

Number of emails, phone numbers or payment methods shared within a network, age of the customer’s network.

**Train algorithm**

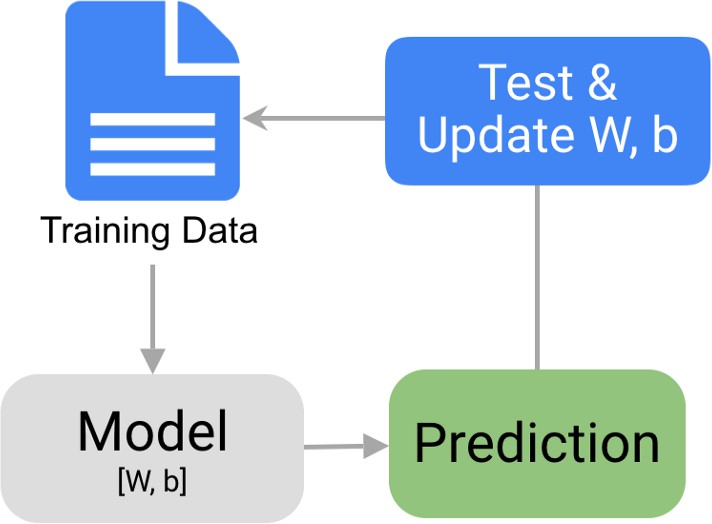
An algorithm is a set of rules to be followed when solving complex problems, like a mathematical equation or even a recipe. The algorithm uses customer data described by our features to learn how to make predictions eg. fraud/not fraud.

In the beginning, we’ll train the algorithm on an online seller’s own historical data, we call this a training set. The more fraud in this training set the better, so that the machine has lots of examples to learn from.

**Create a model**

When training is complete you have a model specific to your business, which can detect fraud in milliseconds.

We constantly keep an eye on the model to make sure it is behaving as it should, and we’re always looking for ways to improve it. We regularly improve, update and upload a new model for every client so that the system will always detect the latest fraud techniques.



## Examples of feature which could be good indicators for fraud are:

**Order rate** - fraudsters order at a much more rapid pace, we quantify this as number of orders per week.

**Email** - fraudster might have a dodgy-looking email, we quantify as % of digits in the email address

**Delivery location** - it could be somewhere typically genuine/unlikely to be fraud like a penthouse apartment, or it could be somewhere fraud like a park. We quantify this as the location fraud rate %

All features are created as a number as the model can’t absorb raw text. We build up our features and categorize them into groups.

## Putting precision & recall in context

If you prevent threshold is at 95, you’re blocking a very small % of customers. You’d have very high precision - you’re only blocking a few customers that you’re fairly sure are fraudsters. You’ll have a very low false-positive rate. However, recall is likely to be low as there are likely to be fraudsters with scores under 95 which you’re not preventing.

If we look at the opposite situation - if you have a block threshold of 5. You’re preventing a huge amount of your traffic and so you’re likely to have very poor precision - and probably end up with lots of false positives. You will have high recall - as you’re going to block most if not all of the fraudsters.

## How we select the right business data for feature engineering

So, we know how we build a model, but how do we decide what data it should look at? Every business has a lot of data, but not all of it is relevant for fraud. Here's how we select specific data 'features' to analyse and get an indication of fraud.

## First, what is a feature and how is it engineered?

At a basic level, a feature is an individual measurable property or characteristic, such as the cost of a transaction. Feature engineering is the process of extracting these meaningful characteristics to use as learning material for the algorithm.

## Building features

We look for features to capture certain aspects that help us predict fraud. We group the types of features into the below categories.

## Traditional features

These are the typical aspects that predict fraud, for example orders, transactions, cards, location, email. These features generally cover the data you would expect to find on your receipt and are customer-centric.

## Behavioral features

We derive behavioural features from the customer session - these are features are based on describing the customer actions e.g., velocity of orders, time spent on the page, length of time between adding a new card and making an order. One purpose of extracting these features is to capture other subversive technology use e.g., if a fraudster is using a script to scrape a webpage vs normal browsing activity.

## Real Time Features

Real-time features are based on the up to date, real-world incidences of fraud. These features are all based on categorical data - give the real-time rate of fraud by category eg. country / ASN card digits / email domain etc. An example feature could be the fraud rate in certain regions/countries.

One purpose of these features is to help merchants to expand into new markets where they have no existing data. We monitor the real-time traffic to help our merchants seamlessly move into new markets, without seeing any adverse effects from the machine learning models eg. bias.

## Individual Customer Features

These features tell us about the similarity with the specific customer’s typical past behaviour. This could be their typical spend, their regular billing address, home IP address etc.

## Sessions Tracking Features

These features are a little more involved than the behavioural features. These features cover the data we get from JavaScript e.g., whether the customer is pasting a card number into the checkout, cookies, if they are using a password vault etc. One purpose of these features is to capture genuine customer behaviour e.g., taking time to change the size of a piece of clothing.

## Network Derived Features

As well as customer-centric and entity-centric features, we also look for network level features. These features focus on network topology (network shape) as a means of enhancing our customer data.

## Online Payments Fraud Detection with Machine Learning

To identify online payment fraud with machine learning, we need to train a machine learning model for classifying fraudulent and non-fraudulent payments. For this, we need a dataset containing information about online payment fraud, so that we can understand what type of transactions lead to fraud. For this task, I collected a dataset from Kaggle, which contains historical information about fraudulent transactions which can be used to detect fraud in online payments. Below are all the columns from the dataset I’m using here:

1. step: represents a unit of time where 1 step equals 1 hour
2. type: type of online transaction
3. amount: the amount of the transaction
4. nameOrig: customer starting the transaction
5. oldbalanceOrg: balance before the transaction
6. newbalanceOrig: balance after the transaction
7. nameDest: recipient of the transaction
8. oldbalanceDest: initial balance of recipient before the transaction
9. newbalanceDest: the new balance of recipient after the transaction
10. isFraud: fraud transaction.

## Online Payments Fraud Detection using Python

I will start this task by importing the necessary Python libraries and the dataset we need for this task:

import pandas as pd import numpy as np

data = pd. read\_csv ("credit card.csv") print(data.head()) print(data.isnull().sum())

step 0

type 0

amount 0

nameOrig 0

oldbalanceOrg 0

newbalanceOrig 0

nameDest 0

oldbalanceDest 0

newbalanceDest 0

isFraud 0

isFlaggedFraud 0

dtype: int64

So, this dataset does not have any null values. Before moving forward, now, let’s have a look at the type of transaction mentioned in the dataset:

18

## Exploring transaction type

print(data.type.value\_counts())

|  |  |
| --- | --- |
| CASH\_OUT | 2237500 |
| PAYMENT | 2151495 |
| CASH\_IN | 1399284 |
| TRANSFER | 532909 |
| DEBIT | 41432 |

Name: type, dtype: int64

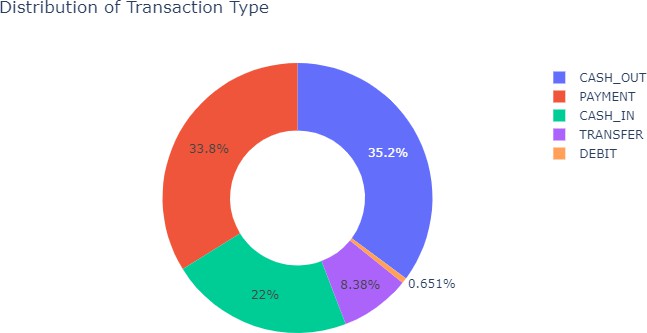
type = data["type"].value\_counts () transactions = type.index

quantity = type.values

import plotly.express as px figure = px.pie (data,

values=quantity, names=transactions,hole = 0.5, title="Distribution of Transaction Type")

figure.show()



Now let’s transform the categorical features into numerical. Here I will also transform the values of the **isFraud** column into No Fraud and Fraud labels to have a better understanding of the output:

data["type"] = data["type"]. map({"CASH\_OUT": 1, "PAYMENT": 2, "CASH\_IN": 3, "TRANSFER": 4,

"DEBIT": 5})

data["isFraud"] = data["isFraud"].map({0: "No Fraud", 1: "Fraud"}) print(data.head())

|  |  |
| --- | --- |
| 1 | 2237500 |
| 3 | 2151495 |
| 0 | 1399284 |
| 4 | 532909 |
| 2 | 41432 |

## Online Payments Fraud Detection Model

Now let’s train a classification model to classify fraud and non-fraud transactions. Before training the model, I will split the data into training and test sets:

## splitting the data

from sklearn.model\_selection import train\_test\_split

x = np.array(data[["type", "amount", "oldbalanceOrg", "newbalanceOrig"]]) y = np.array(data[["isFraud"]])

## Now let’s train the online payments fraud detection model:

#Training a machine learning model

from sklearn.tree import DecisionTreeClassifier

xtrain, xtest, ytrain, ytest = train\_test\_split (x, y, test\_size=0.10, random\_state=42) model = DecisionTreeClassifier ()

model.fit (xtrain, ytrain)

print (model. score (xtest, ytest)) Score: **0.9997391011878755**

Now let’s classify whether a transaction is a fraud or not by feeding about a transaction into the model:

prediction

#features = [type, amount, oldbalanceOrg, newbalanceOrig] features = np. Array ([[4, 9000.60, 9000.60, 0.0]])

print (model. Predict(features)) Output = **['Fraud']**

**Conclusion**

### How human insight complements machine learning

When used successfully, machine learning removes heavy burden of data analysis from your fraud detection team. The results help the team with investigation, insights and reporting. Machine learning doesn’t replace the fraud analyst team, but gives them the ability to reduce the time spent on manual reviews and data analysis. This means analysts can focus on the most urgent cases and assess alerts faster with more accuracy, and also reduce the number of genuine customers declined. Machine learning makes the role of a fraud analyst more efficient, as their time is freed up to do more strategic work. Analysts improve and optimize machine learning fraud detection systems through reviewing and labelling customers and tuning the rules. Machines are exceptionally good at doing the heavy lifting in data analysis, number crunching and output. They work tirelessly through the night and never complain about working weekends.

Machines are less good at dealing with uncertainty. There are cases that are new, or that are difficult, or somehow different. Edge cases are those that require more attention and may be difficult to determine - this is where the human insight comes in and provides massive value.

The expert human intervention here is not just at the point approving a transaction. It’s more a case of analysis after the event and labelling the data in a way that gives rapid feedback to a machine. Remember, labelled data is the ultimate training set for a machine. So, the more confirmed behaviour labels it can receive the more accurate a result there is likely to be.

**REFERENCES**

**INSIGHTS**

* https://[www.ravelin.com/insights/online-payment-fraud](http://www.ravelin.com/insights/online-payment-fraud)
* https://[www.ravelin.com/insights/machine-learning-for-fraud-detection](http://www.ravelin.com/insights/machine-learning-for-fraud-detection)
* https://[www.ravelin.com/insights/link-analysis-and-graph-database-for-fraud-](http://www.ravelin.com/insights/link-analysis-and-graph-database-for-fraud-) detection
* https://pages.ravelin.com/fraud-and-payments-survey-2022
* https://[www.ravelin.com/insights/account-takeover-fraud](http://www.ravelin.com/insights/account-takeover-fraud)
* https://[www.ravelin.com/insights/policy-abuse](http://www.ravelin.com/insights/policy-abuse)
* https://[www.ravelin.com/insights/ultimate-guide-psd2-strong-customer-](http://www.ravelin.com/insights/ultimate-guide-psd2-strong-customer-) authentication

## APPENDIX

**Resources for Fraud Transaction Analysis**

* Kaggle website.
* Python.
* Machine Learning
* Jupyter notebook