Detecting Fraudulent Transactions

Capstone project - Timothy Ong

Introduction

Background

- Fraudulent activities often result in negative economic and social impact to a company
- To prevent this from happening, a company has to conduct regular audit checks
- It is expensive to conduct checks on all transactions and it is not effective to conduct

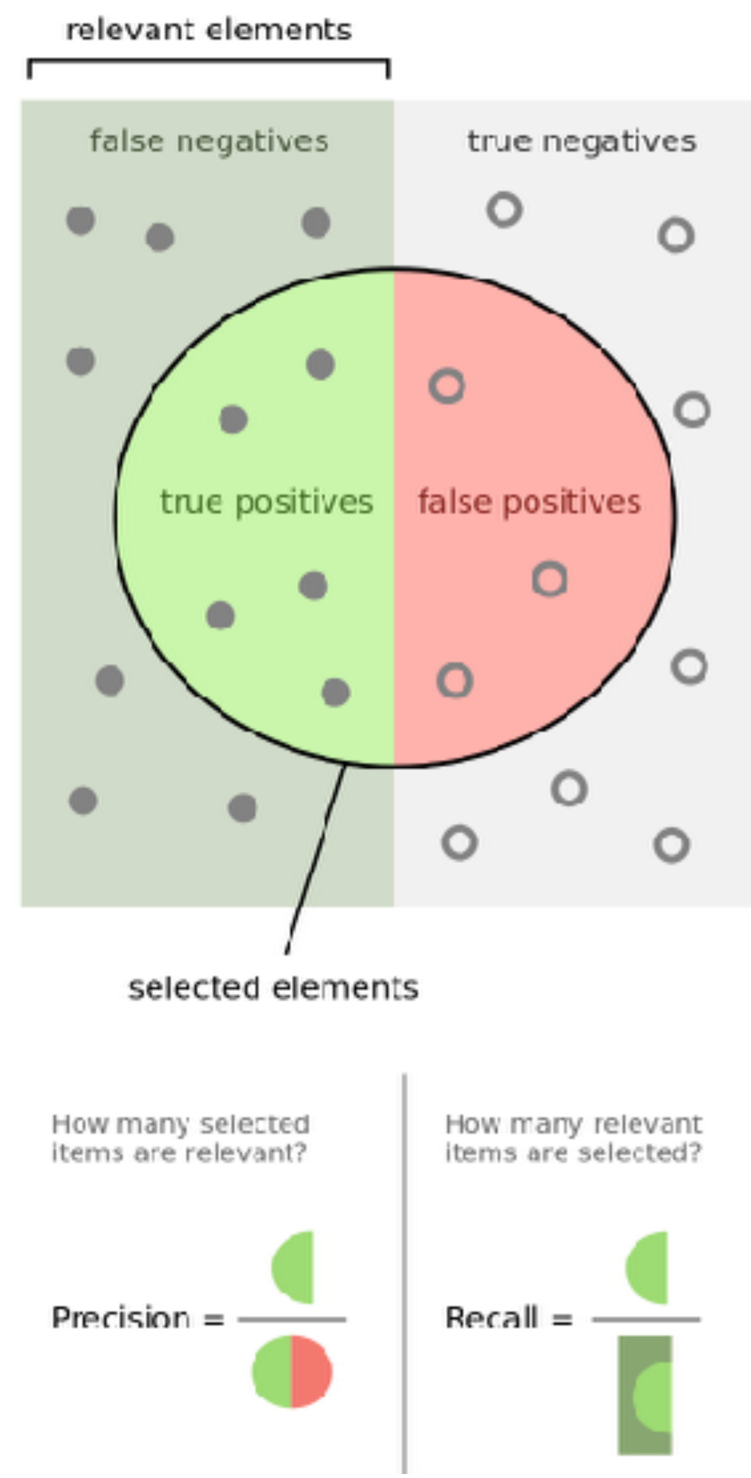
Objectives

1. Generate a predictive model that will classify fraudulent transactions
2. To provide a fraud probability ranking system to maximise company's limited resources

Introduction

Evaluation Metrics

- Accuracy
- Precision
- Recall
- F1 Score
- AUC of ROC



Dataset*

Metadata

- Dataset is based on transactions reported by salespeople of a company
- These salespeople sell a set of products and are free to set the selling price
- At the end of the month, they will report back to the company on their transactions
- Id : salesperson ID
- Quant: quantity sold in that transaction
- Val: total value of transaction
- Insp: 3 different classes - ok, fraud, unknown
- Pdt: product ID

Dataframe of the first 5 transactions in the dataset:

	id	quant	val	insp	pdt
1	v1	182.0	1665.0	unkn	p1
2	v2	3072.0	8780.0	unkn	p1
3	v3	20393.0	76990.0	unkn	p1
4	v4	112.0	1100.0	unkn	p1
5	v5	6164.0	2026.0	unkn	p1

*Dataset taken from a GitHub page

Dataset

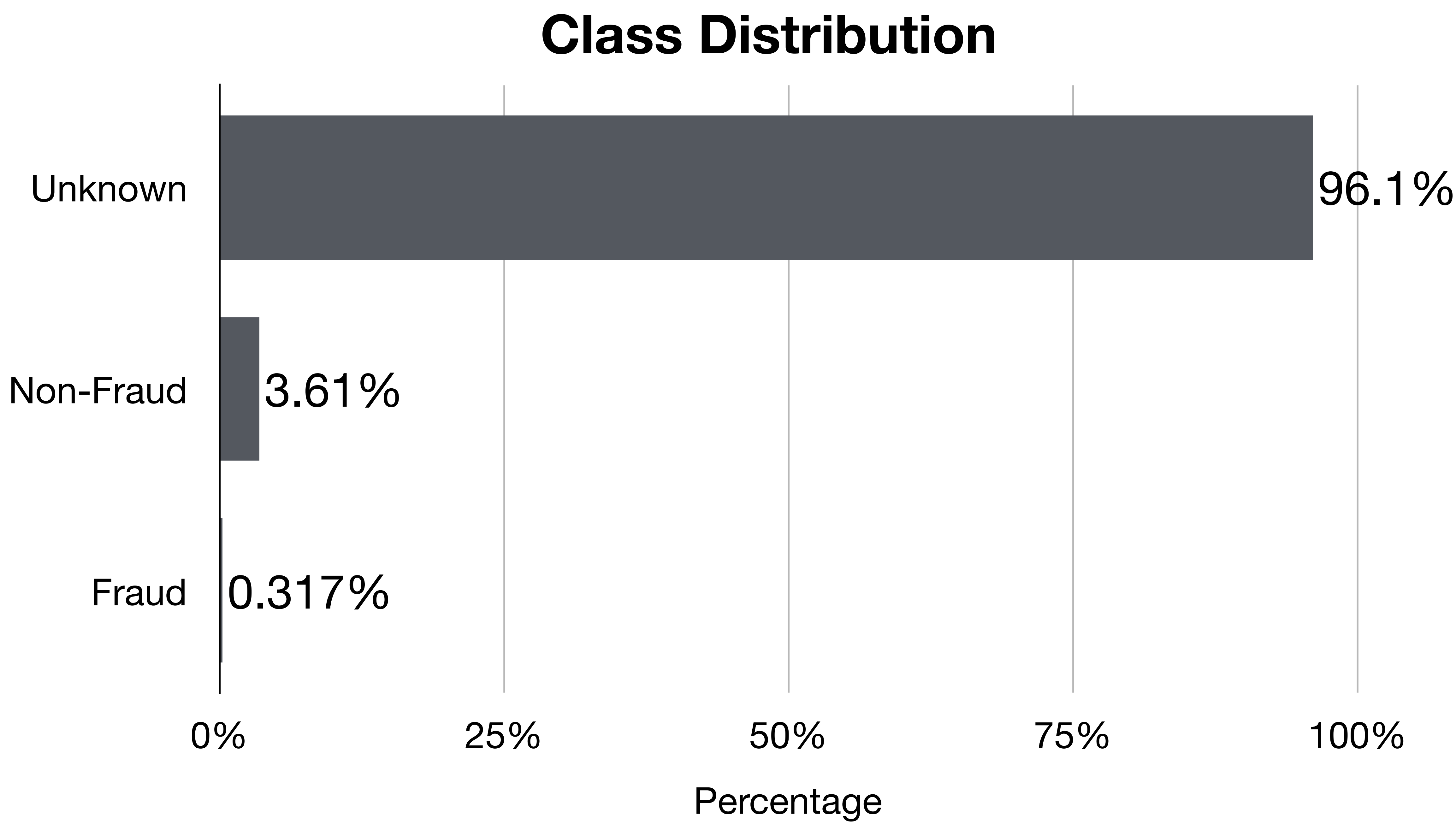
Cleaning

- ~401k rows in dataset
- ~13k rows have missing values
- 2 new features were engineered in the midst of data cleaning
    - unit price of each transaction
    - median unit price of each product
- They were use to impute missing values in the dataset

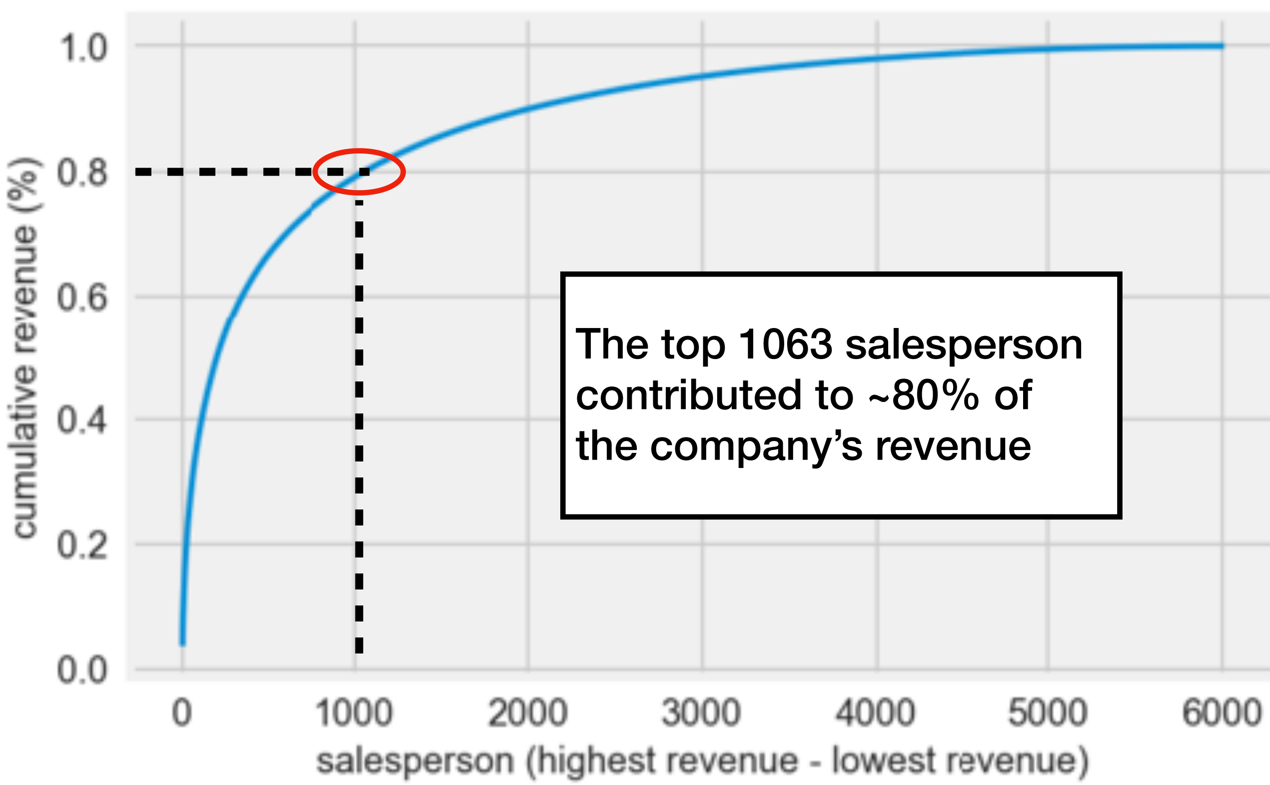
Dataframe of the first 5 transactions in the dataset:

	id	quant	val	insp	pdt	unit price	median price
1	v1	182.0	1665.0	unkn	p1	9.1	11.4
2	v2	3072.0	8780.0	unkn	p1	2.9	11.4
3	v3	20393.0	76990.0	unkn	p1	3.8	11.4
4	v4	112.0	1100.0	unkn	p1	9.8	11.4
5	v5	6164.0	2026.0	unkn	p1	0.3	11.4

Exploratory Data Analysis

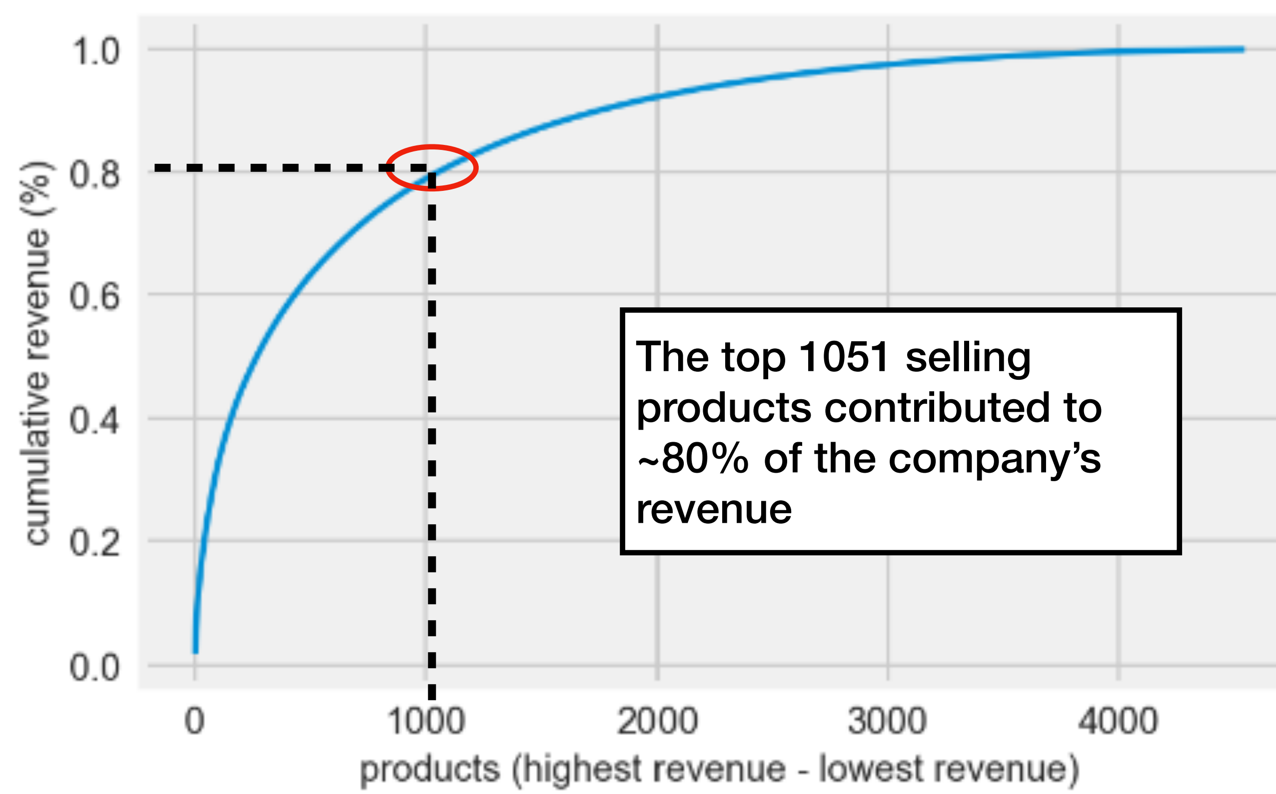


Exploratory Data Analysis

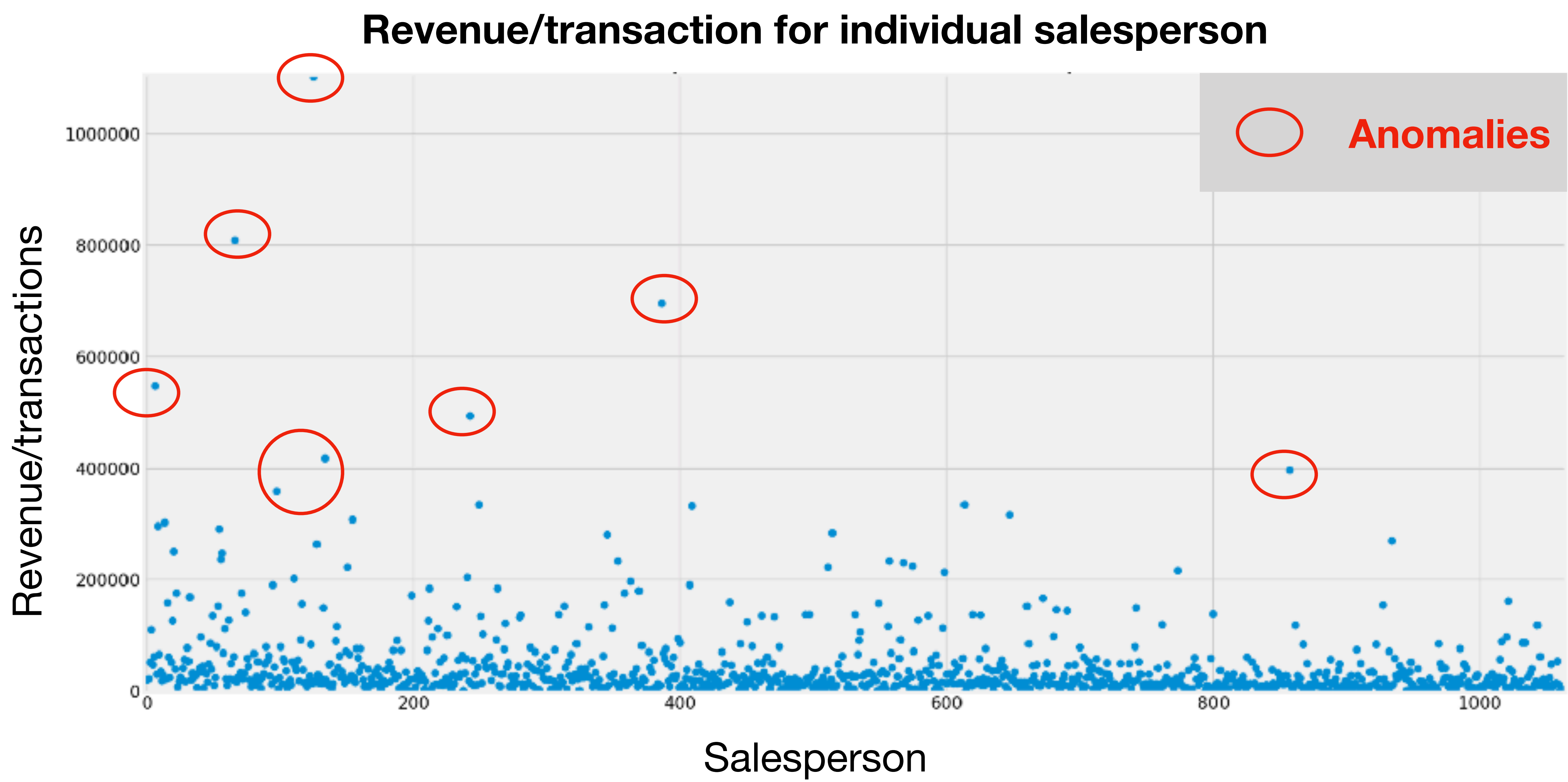


Observations

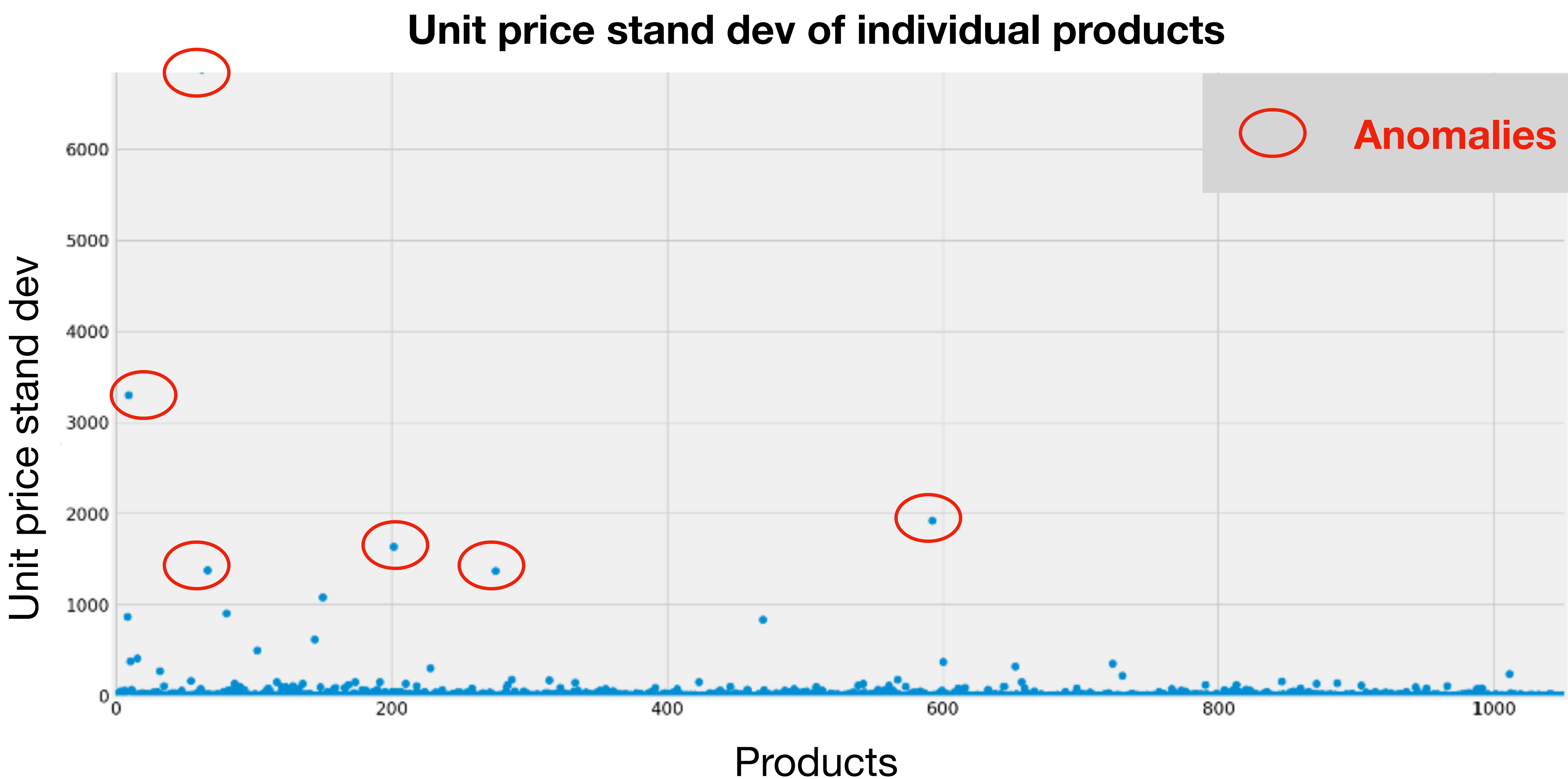
- We can observe the 80/20 rule, whereby the top 20% of the company's salespeople are contributing to 80% of the company's revenue. Similar for the top 20% products
- If any of the top 20% salesperson commit a fraudulent transaction, it will harm the company extensively



Exploratory Data Analysis

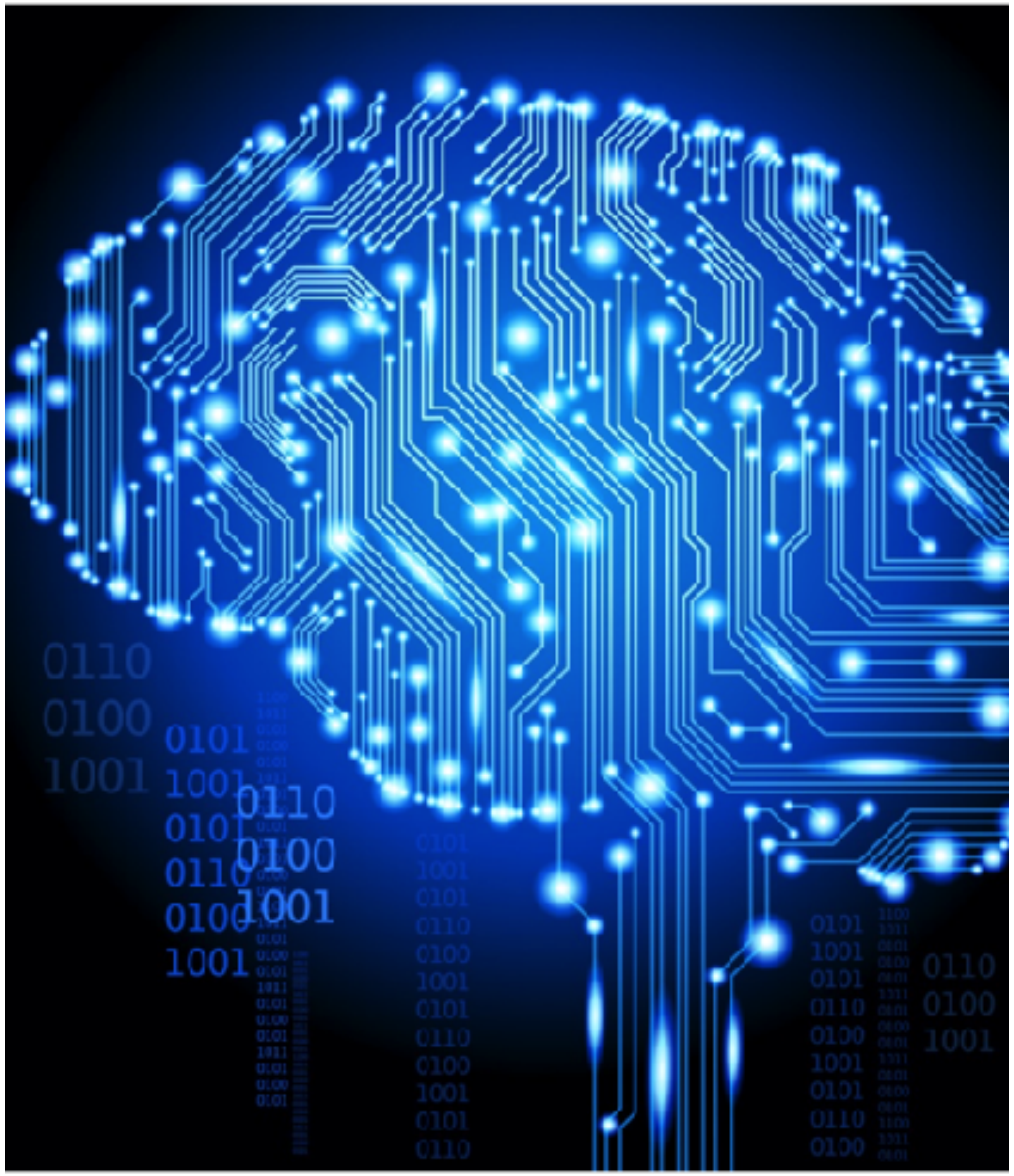


Exploratory Data Analysis



Exploratory Data Analysis

Is there a more definitive way of identifying fraudulent transactions?



Modelling

- 96% of the data are unlabelled
- Is it good enough to just model with 4% of the data?
- Do a train-test split and generate two Random Forest models
  - 30% of training set VS 100% of training set
- Perform a cross-validation to ensure that the results are reliable

Results

	30% train set	100% train set	Difference
Accuracy score	0.726	0.839	0.113
Precision Score	0.990	0.990	0.000
Recall Score	0.679	0.815	0.136
F1 Score	0.791	0.894	0.103

Precision, recall and F1 score are based on the minority class (fraudulent class)

Supervised Models

- Created 3 test set to test for overfitting. Each differs by proportion of fraud labels
  - Test set 1: original proportion, 12 non-fraud : 1 fraud (5,190 samples)
  - Test set 2: 1 non-fraud : 1 fraud (836 samples)
  - Test set 3: 0.25 non-fraud : 1 fraud (501 samples)
- Robust scaling to prevent the model from being influenced by outliers
- SMOTE to create synthetic examples of minority class to solve the imbalanced problem
- Gridsearch

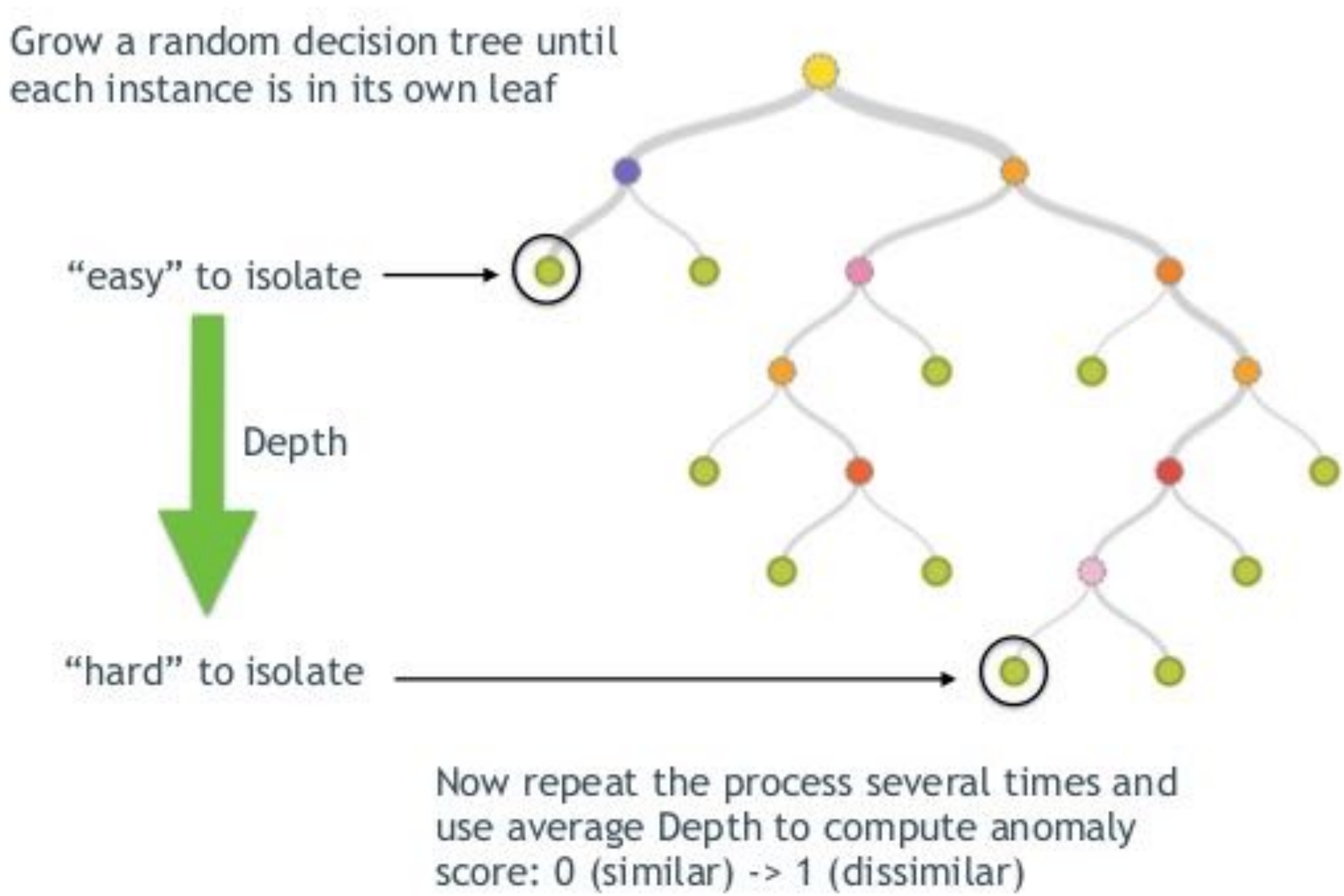
Results

	Logistic Regression	SVM	Random Forest
Accuracy Score	0.734	0.936	0.973
Precision Score	0.167	0.575	0.809
Recall Score	0.579	0.797	0.871
F1 Score	0.260	0.668	0.839
AUC	0.663	0.873	0.926

Precision, recall and F1 score are based on the minority class (fraudulent class)

Unsupervised Model

- Outlier detection - Isolation Forest



Unsupervised Model

- Created a new feature
  - unit price of a transaction - median unit price of the product in that transaction

	Value	Quantity	Product ID	Unit Price	Median Unit Price	UP - Median
1	2585	377	1	6.85	6.43	0.42
2	14180	356	5	39.83	17.14	22.69
3	15175	407	3	37.29	16.71	20.58
4	9950	618	2	16.10	10.9	5.2

- Importance of recall score
  - The higher it is, the lower the number of false negative
  - False negative = predicting a fraud as a non-fraud

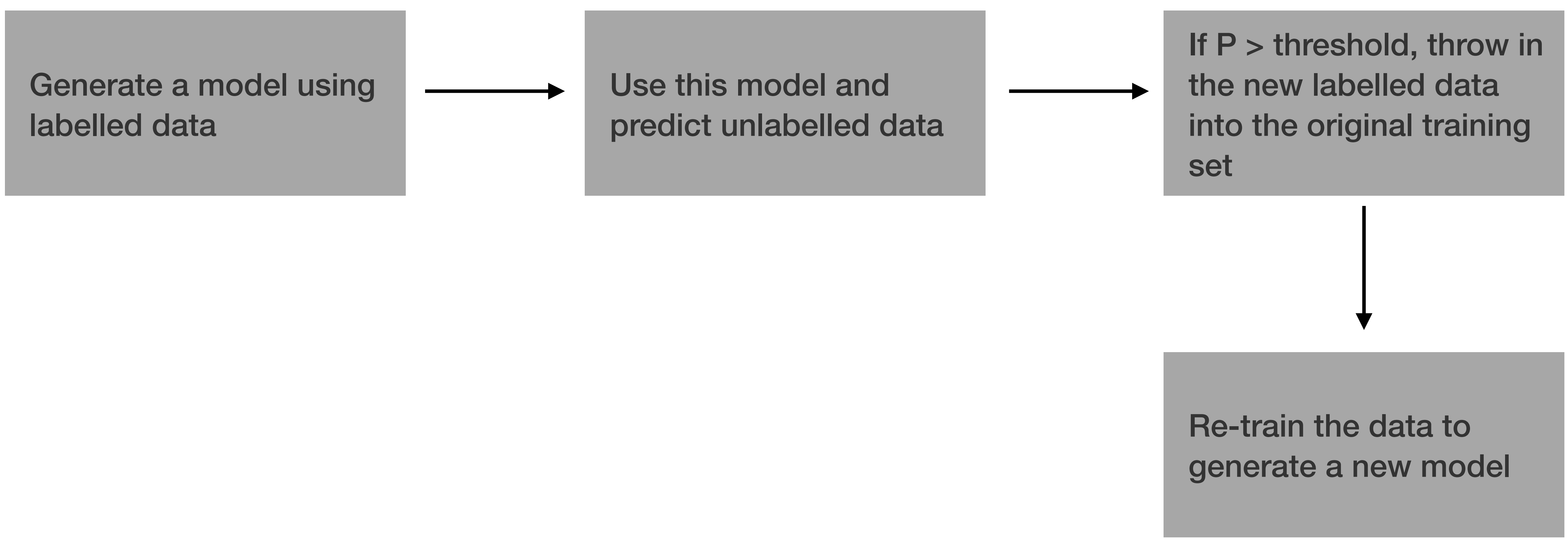
Results

	Isolation Forest	Isolation Forest (recall optimised)	Random Forest
Accuracy Score	0.892	0.689	0.973
Precision Score	0.336	0.165	0.809
Recall Score	0.342	0.706	0.871
F1 Score	0.339	0.268	0.839
AUC	0.641	0.697	0.926

Precision, recall and F1 score are based on the minority class (fraudulent class)

Semi-Supervised Model

- Self-training method
- Using Random Forest as base model



Semi-Supervised Model

Results

Random Forest	Supervised	Semi-Supervised
Accuracy Score	0.973	0.991
Precision Score	0.809	0.930
Recall Score	0.871	0.955
F1 Score	0.839	0.942
AUC	0.926	0.974

Precision, recall and F1 score are based on the minority class (fraudulent class)

Confusion Matrix of test set

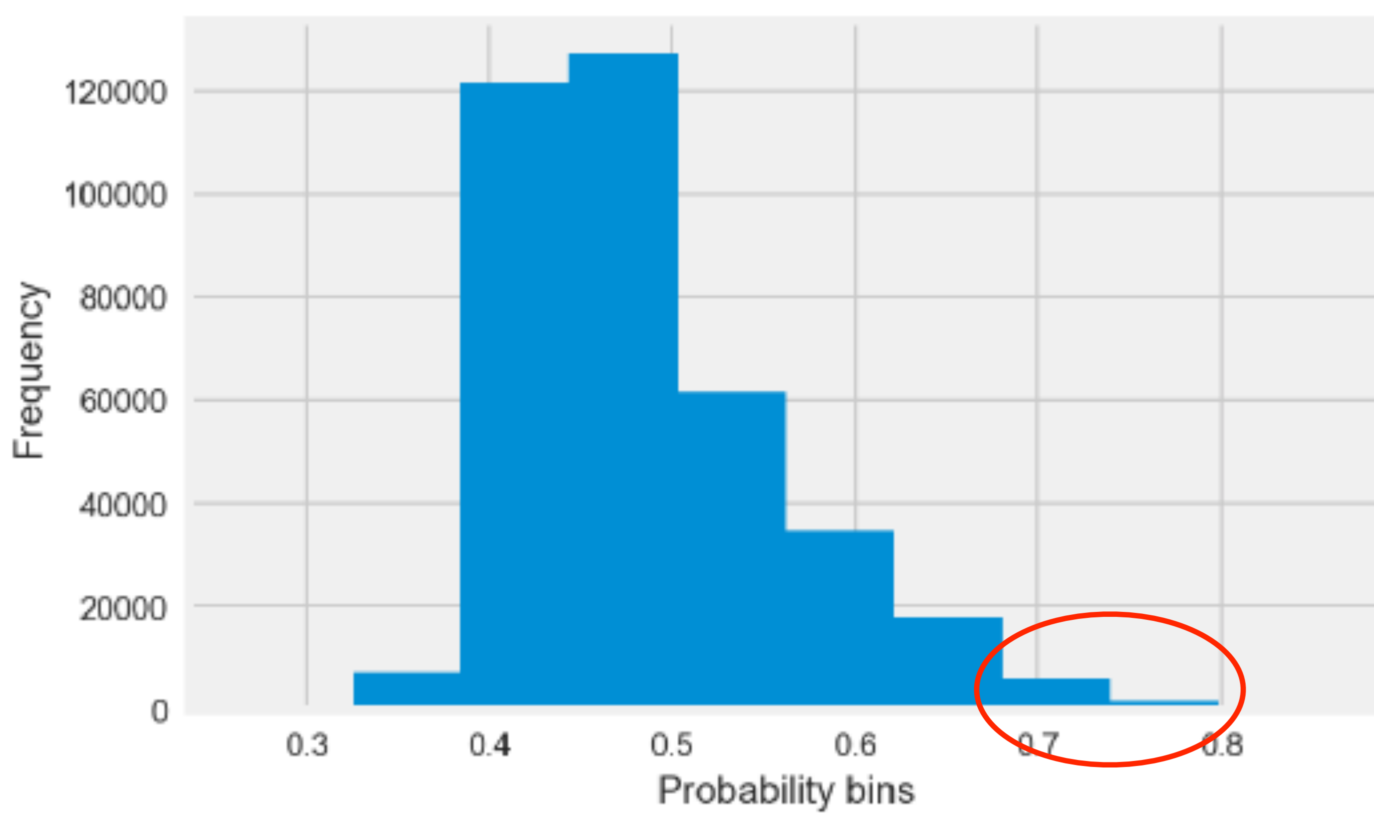
	Is Non-Fraud	Is Fraud
Pred Non-Fraud	4742	30
Pred Fraud	19	399

Semi-Supervised Model

Fraud probability ranking system
(displaying the top 5 probable fraudulent transaction)

Index	Pred Non-Fraud	Pred Fraud
47	0.142	0.858
3123311	0.143	0.857
12314	0.146	0.854
8345	0.148	0.852
438	0.150	0.850

Histogram of predicted probability fraudulent cases



- Company's resources will be focused on auditing these transactions (red circle) as they have a higher probability of being frauds
- Company's resources are maximised

Summary

Conclusion

- Semi-supervised Random Forest model produced the best results
- Beats baseline accuracy - 0.991 vs 0.919
- Excellent precision (0.930) and recall score (0.955) for fraudulent class

Lessons Learnt

- It is always useful to obtain a cost matrix that tells us the cost of a false negative and false positive
- When dealing with a imbalanced dataset, it is always good to create rigour test sets
- Always conduct cross-validation for checks
- Always good to draw up a workflow before jumping into solving the problem