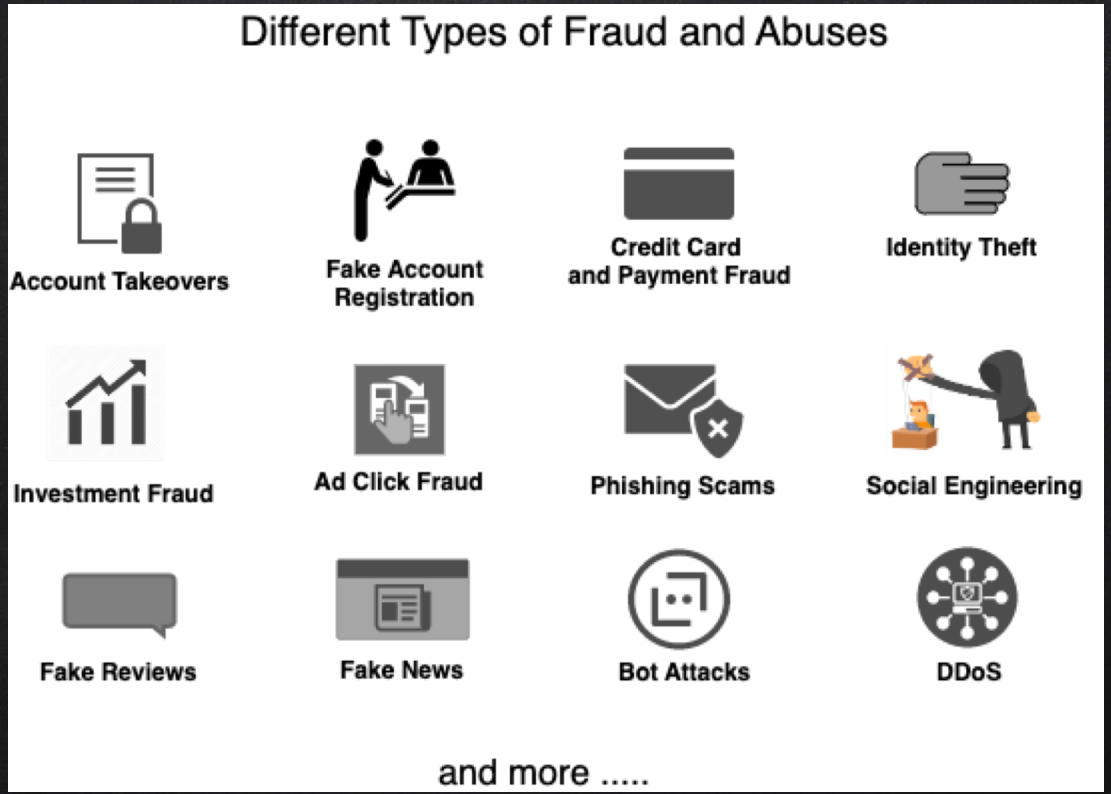
Fraud Detection - Overview

- Roger



Reference: [12 Types of Fraud and Abuse that Everyone Should Know About]

Foundations of a Fraud Detection System

- In the early days of e-commerce, companies often start with manual investigations when fraud emerges.
- Even today, manual reviews and analyzing common fraud behaviors are the foundations for building fraud detection systems.
- Large established organizations still use manual reviews as another layer of validation despite having highly accurate machine learning-based fraud detection systems to reduce customer frictions, for example.

- Manual reviews and behavioral analytics are two foundational ways to detect fraud.
- However, both do not work in real-time.
- It is equally important to use the knowledge previously gained from past fraud patterns and use that knowledge to further stop similar activities from occurring in the future.
- The simplest way to encode the knowledge of patterns is to encode those patterns in the form of rules.
- Rules:
  - Lists: e.g., a list of usernames or IP addresses to deny or allow access
  - Thresholding: i.e., limiting access if a certain pattern is observed. For example, when more than 25 items are bought on an e-commerce within 30 minutes, then block further transactions and send for a manual review.

Summary

1. Manual Investigations: Human investigators manually look at the online activities for suspicious patterns.
   - They are specialists.
   - A very important role that manual investigations play is helping gather data points and labels for building machine learning models later on. The activities marked by human investigators are used as ground truth and are foundations for building most ML backed fraud detection models.
2. Behavior Analytics: Fraud Analysts can create clusters of multiple fraud activities (that are already found by manual investigators) into behavioral patterns. Such patterns can later be encoded into a set of rules and those rules can be embedded into the website or the app to automatically stop those activities from re-occurring.

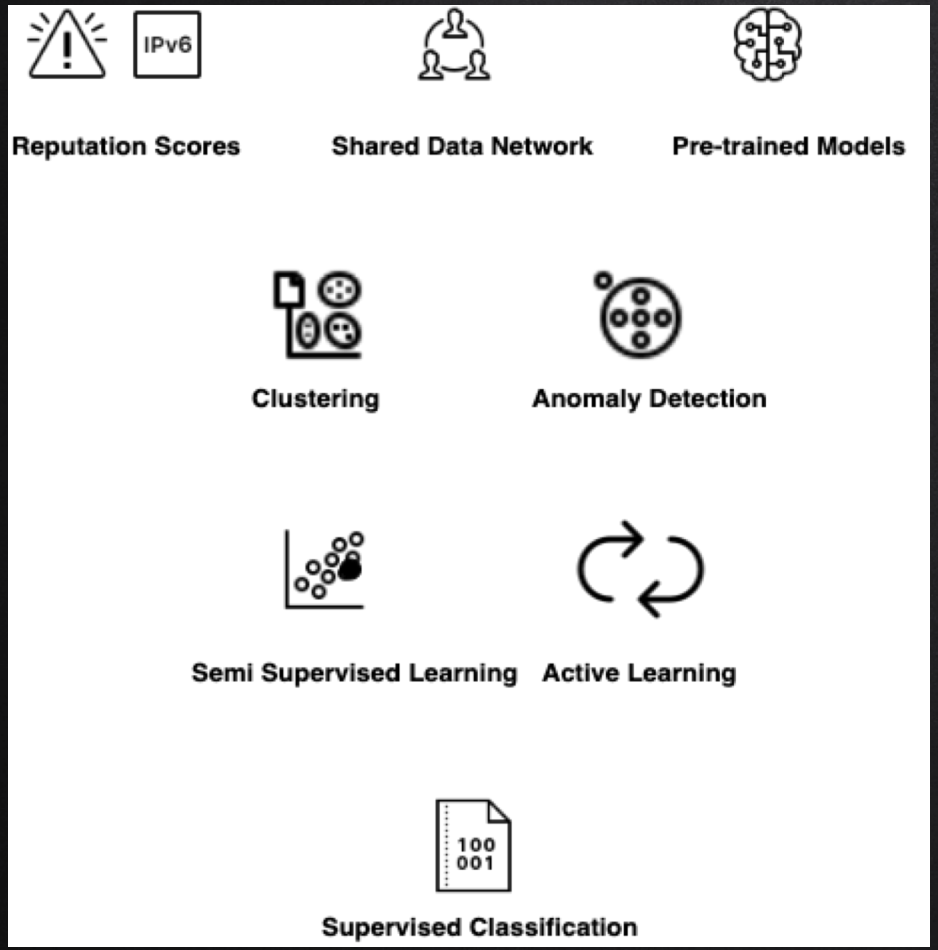
Summary

3. Rules: Rules are the set of logics that can be coded directly into the service by a Software Engineer.
   - Lists: contain the attributes that can be directly used to deny access to a service.
   - Thresholding: Some rules are not tied to specific attributes like a specific IP address. But rather to a specific pattern.
4. Customer Feedbacks: another way of finding fraud and taking some enforcement actions like helping the customer change password. Although, if an organization is only finding fraud with customer feedback, they are not really doing a good job in proactively preventing fraud and might lose customers.

Summary

3. Roles:
   - Investigators.
   - Risk Analysts/Data Scientists.
   - Software Engineers.
4. Infrastructure and tools:
   - Make it easy for investigators to visualize the information they need.
   - Keep the customer's data secure but find a way to make it possible for your risk analysts (with access to confidential data for security purposes) to analyze the data for suspicious behaviors.
   - Make it easy for your customers to reach out in case they see anything suspicious.

Different Design Frameworks for ML Based Fraud Detection



WE WILL DISCUSS A HIGH-LEVEL MENTAL FRAMEWORK FOR TYPES OF ML ALGORITHMS THAT CAN BE IMPLEMENTED AT DIFFERENT STAGES OF ORGANIZATIONAL GROWTH.

The following frameworks are segmented based on organization's capability to retrieve past user activity data and on the presence of known fraudulent activities to extract patterns from. From 1 to 4, there will be an upward trend in terms of capability to tackle fraud, but it will require time and manual efforts to reach there.

Progression of Machine Learning Frameworks for Fraud Detection

#	Collecting user activity data	Have examples of fraudulent activities	Use the following
1	No	No	Reputation Scores, Shared Data Networks/Consortium/Federated Learning, Pre-trained Models
2	Yes	No	Clustering, Anomaly Detection + All from Paradigm 1
3	Yes	Yes, a little bit	Semi-supervised Learning, Active Learning + All from Paradim 2
4	Yes	Yes, quite a bit	Supervised Classification Algorithms + All from Paradigm 3

Group 1. Reputation Scores, Shared Data Networks and Pre-trained Models

The question is, can you do better than traditional solutions like setting up rules or hiring investigators without having access to the historical user activity data and without having knowledge of the activities that have been fraudulent in the past. The answer is, yes. You can try the following options to get the wheel rolling.

High Risk, IPv6, Artificial Intelligence, Collaborating In Circle.

These solutions are not limited to the lack of training data, and are also about the lack of time and budget to train models from scratch, or the absence of systems that store the user activity data.

A. Reputation Scores

Reputation scores are risk scores that represent the general riskiness of attributes associated with an online activity. These scores are provided by security systems that search for early signs of malicious behavior originating from multiple places on the internet.

The common attributes for which you can find reputation scores are IP address, email address, phone number, or the user agent, with IP address being the most common one.

Examples of reputation score providers include DataVisor, TeleSign, Apivoid, Talos, MaxMind minFraud, and Microsoft Defender Threat Intelligence.

Note that the reputation scores have various limitations too and cannot tackle all types of modern fraud vectors. Most importantly, the assumption that all frauds MOs have bad reputation of IP, email or user agent is often violated.

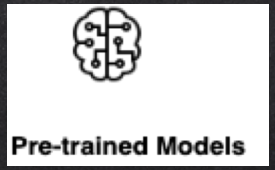
B. Shared Data Network, Consortium and Federated Learning Solutions

Also called **Consortium**, the shared data network is a secure system where a large number (like thousands) of global institutions pool the data of comfirmed malicious activities. If done securely, i.e. without impacting any institions' data privacy, the consortium can be a really powerful tool where organizations come together to tackle fraud together and once one organization sees an attack vector, they share with others. Joining one such consortium is another option to improve your fraud detection capabilities.

You can choose one of the options among Falcon Intelligence Network, Microsoft's fraud prevention network, Stripe's Radar, Kount' Identity Trust Global Network and use the consortium risk score in conjunction with authentication products in your overall fraud deterrence strategy.

For a secure and privacy preserving model training with a shared data network, **Federated learning** is an important emerging field of the machine learning that enables the models to be trained from different datasets located at different servers, without transfering the training data to a central server. It is decentralized model training that allows each organization's data to remain on their local servers, reducing the possibility of data breaches while allowing the transfer of knowledge.

C. Pre-trained Models



The idea of pre-trained models is similar to transfer learning that we commonly discuss in the natural language and computer vision literature. The model trained on one corpus of dataset can be used to transfer the knowledge over another corpus of dataset with some fine tuning. While fine tuning of the pre-trained model on the new data corpus is an important step, if it is not possible to do so, as in the case where we don't have any historical data to fine tune with, the pre-trained model still has a lot of useful signals. For fraud detection, the companies like Ravelin provide solutions on the similar lines with their micro model architecture. You can use one of their pre-trained models that have the knowledge of fraud patterns.

Group 2. Clustering and Anomaly Detection

Clustering, Outliers,

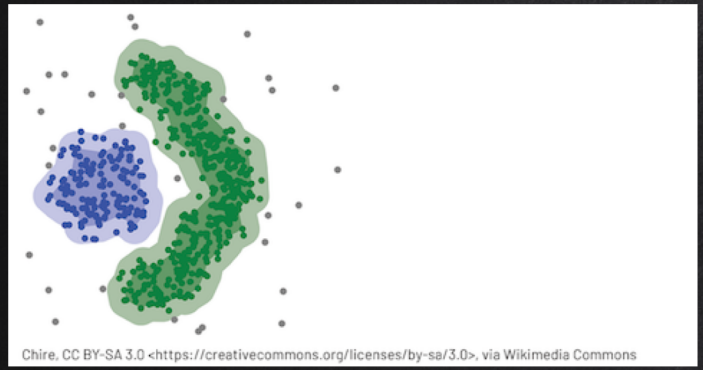
Most companies even without risk/fraud teams collect user activity data for the business purposes like targeted marketing, personalizations and basic account access handling. This category of solutions can be employed when you do not have any fraud labels but have been collecting user activity data.

A. Clustering

In order to find clusters, behavioral (e.g. number of clicks), frequency (e.g. number of times a user logged in within last 1 hour) and profiling (e.g. location of order) attributes of users and their associated activities are used, but any information on confirmed fraud labels is not required. The clustering approaches like Hierarchical clustering, K-means and DBSCAN are common in fraud domain. More recently, Graph based clustering methods are used to capture complex relationships that other approaches would miss.

Clustering can be very useful in certain fraud attack scenarios, as following:

1. Find organized frauds:

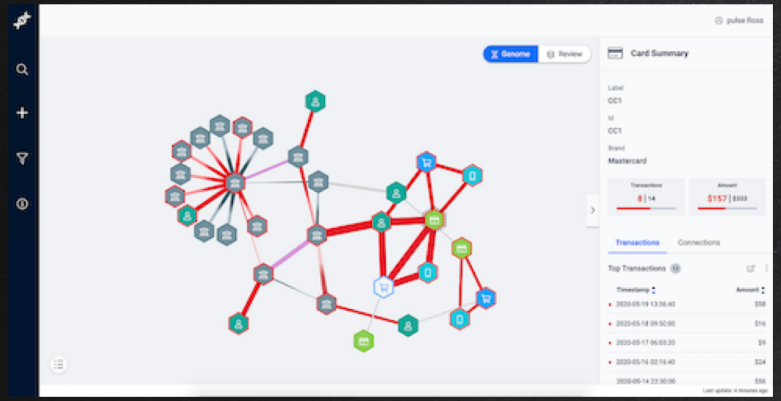


Reference: Identifying Financial Fraud With Geospatial Clustering by Databricks

Fraud coordinated by groups of professional fraudsters that are part of a group and act at scale to maximize their gain is referred to as organized fraud. Organized fraud relies on coordinated events called fraud campaigns. During a fraud campaign, several orders are placed over a limited period of time by a small group of fraudsters using different electronic identities.

In such cases, analyzing each activity in isolation and independent of each other is not an effective solution to find and stop the fraud attacks. Instead, you can create clusters based on user behavior and profiles. Organized attacks would fall into high density clusters compared to the legitimate users. Based on the clustering output, you can apply additional screening steps in abnormally high density clusters (i.e. having a large number of activities in a cluster)

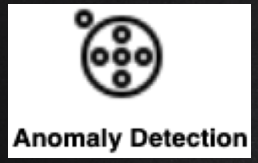
2. Bulk investigate suspicious activities:



Reference: Empowering Fast Graph Visualizations for Fraud Detection (or Why We Built Our Own Graph Database by Francisco Santos

Rather than making your investigators look at each activity independently, you can use clustering to make their lives easier and surface a group of events that investigators can look at. If one event in a group is fraudulent, there are high chances that other events too in the same group are fraudulent and displaying such events as a group makes it easy to investigate at scale and take bulk actions. Such functionality can increase investigator's efficiency and provide an overall boost to your fraud detection capability. You can even go a step further and provide reasons why your clustering algorithms think that certain events are connected. For example, it might show that N orders are connected because they are being generated by same card number.

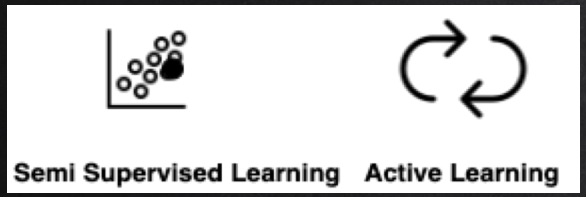
B. Anomaly Detection/ Outlier Detection



Outlier Detection (OD) (or Anomaly Detection) is the task of identifying abnormal objects from the population of normal objects. OD is employed in fraud detection sub-fields like intrution detection, malicious URL detection, backdoor attack detection, fake reviews and credit card fraud detection. Fraudsters are expected to show unusual behaviors like unusually high volume of traffic, unusual patterns of failed logins, multiple promotions from the same device etc.

It is important to note that not all unusual transactions are fraudulent. There may be legitimate reasons for a transaction to exhibit unusual behavior. However, by identifying transactions that are more likely to be fraudulent, outlier detection algorithms can help to reduce the risk of fraud.

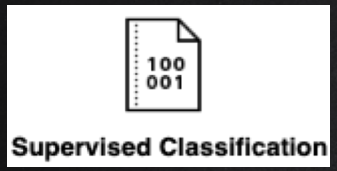
Group 3: Semi-supervised Learning, Active Learning



Scatter Plot

This category of solutions are applicable when you have a large amount of user activity data that hasn't been evaluated for suspicious behavior (i.e. unlabaled data) and a very small amount of confirmed fraud labels. In short, unlabeled data is abundant and easy to get, while labeled data is expensive and scarce. For example, consider the case of Account Takeover fraud: you might not have any team of investigators or risk analysts to find comprimised accounts, but the real user whose account got compromised might reach out to your customer care and complaint about the compromise. This way you will have "Customer Reported Labels". Such labels might be scarce but are useful.

Group 4. Supervised Classification Algorithms



Binary File.

Now we are talking about the most optimistic scenario, i.e. when a large number of ground truth labels are available. Even within single fraud type, there can be different sub-definitions of fraud but the simplest approach is to classify an activity as either fraudulent (1) or legitimate (0). Given the binary label setting, the most obvious and actually most high performant solution among others we discussed today, is supervised classification framework.

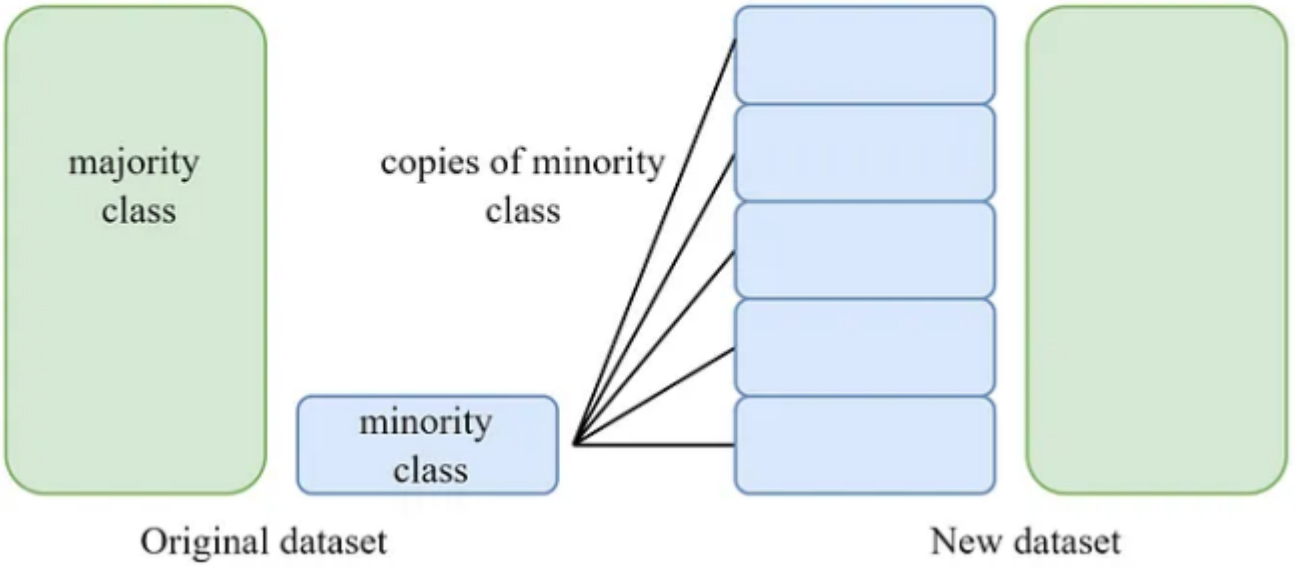
Supervised classification in tabular data setting is modeled most robust and performant with tree based ensembles. Gradient boosting methods like Catboost, LightGBM, XGBoost take the lead. But you need to use some feature engineering tactics to train better models.

Imbalanced Classification

Imbalanced classification is a common problem in machine learning, particularly in the realm of binary classification.

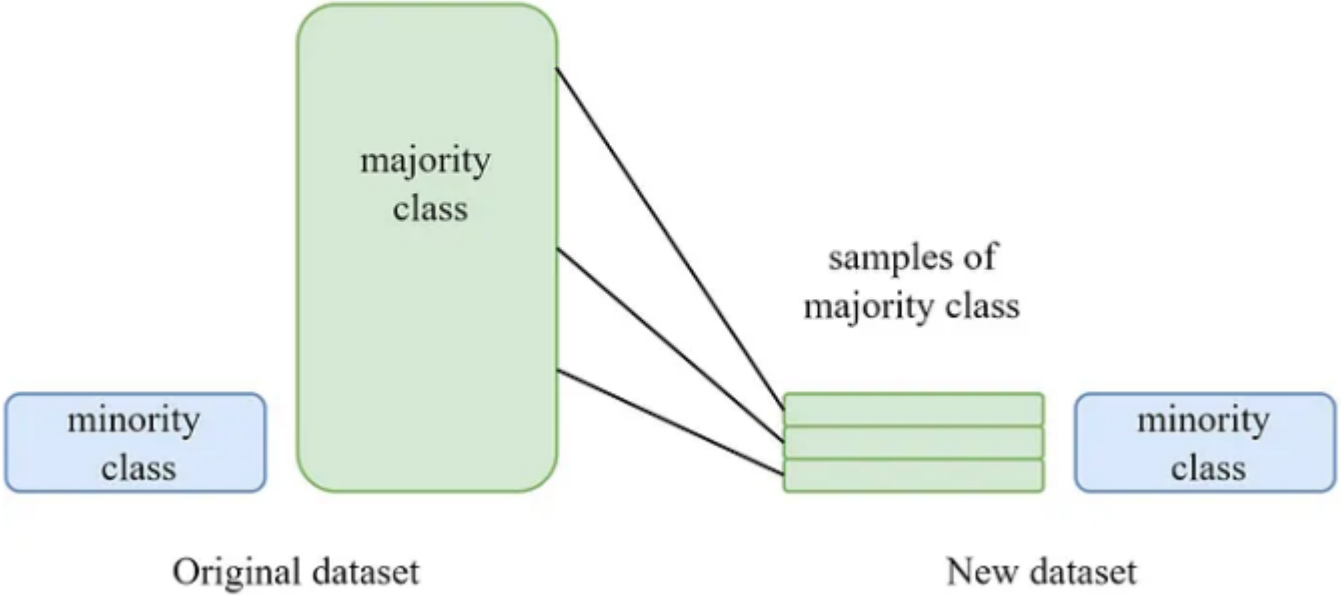
Reference: [Solving The Class Imbalance Problem]

Oversampling



Reference: [Solving The Class Imbalance Problem]

Undersampling



Reference: Solving The Class Imbalance Problem

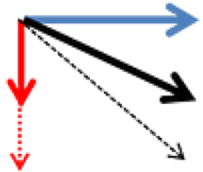
Weights modification on a loss function

$$L = \sum_i l\left(\hat{y}_i, y_i\right) \qquad \frac{\partial L}{\partial \theta} = \sum_i \frac{\partial l\left(\hat{y}_i, y_i\right)}{\partial \theta}$$

The non frequent class samples gradients sum

the frequent class samples gradients sum

Increase non frequent class samples gradients sum by oversampling or weighted loss



Reference: [Solving The Class Imbalance Problem]

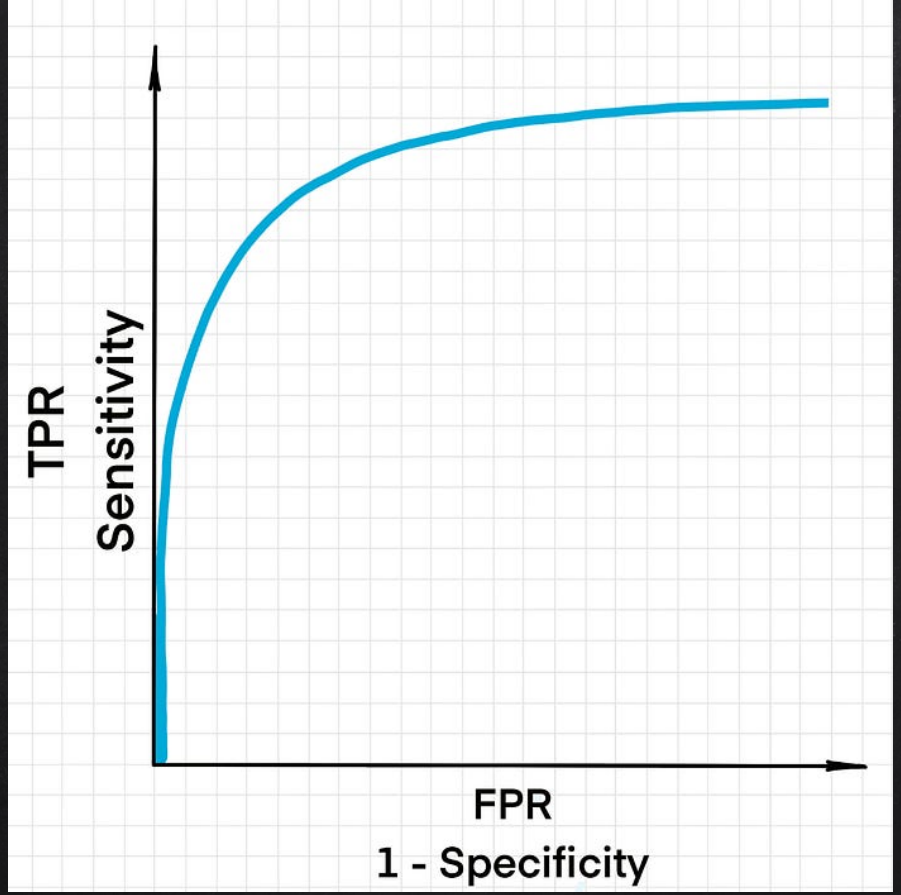
Classification Metrics

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

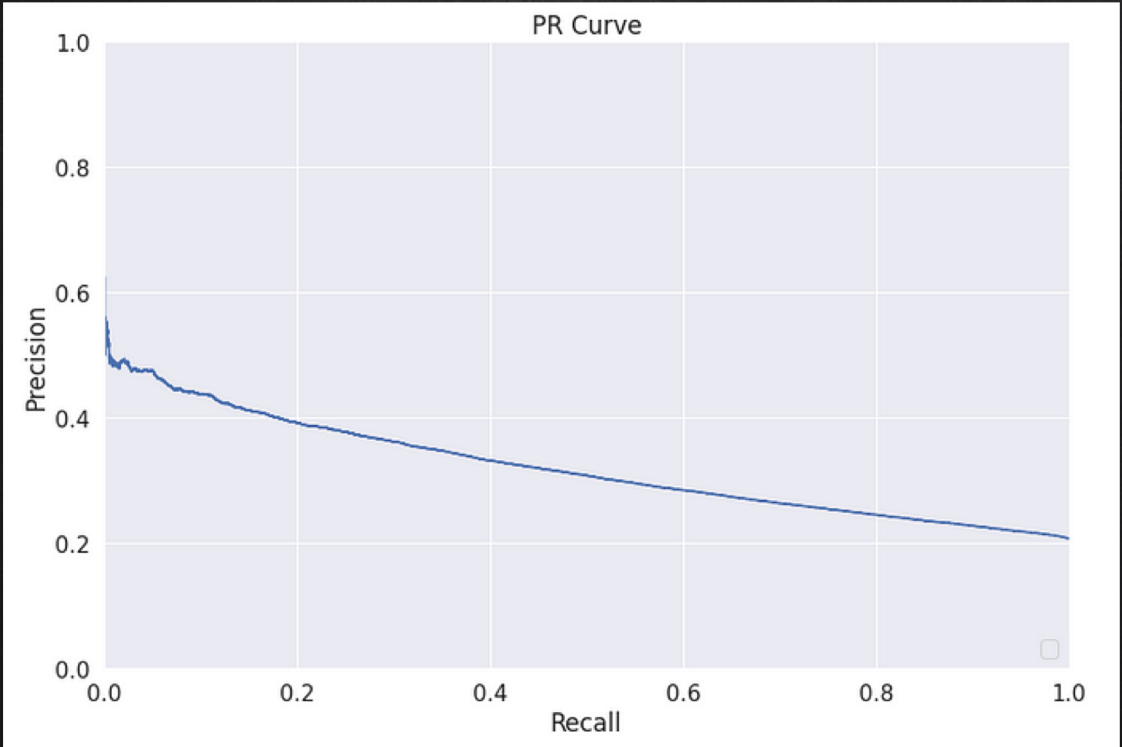
$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$



Reference: Understanding AUC — ROC and Precision-Recall Curves



Reference: [Understanding AUC — ROC and Precision-Recall Curves]

Reference

https://groverpr.github.io/detection/2023-03-07-types-of-fraud-and-abuse

https://groverpr.github.io/detection/2023-04-10-traditional-detection-framework

https://groverpr.github.io/detection/2023-05-02-fraud-detection-frameworks

https://medium.com/metaor-artificial-intelligence/solving-the-class-imbalance-problem-58cb926b5a0f

https://medium.com/@data.science.enthusiast/auc-roc-curve-ae9180eaf4f7