Dirty Laundry: Machine Learning for Financial Corruption Investigations

John Mauricio, Atharva Kulkarni, Alex Makhratchev, Prabina Pokharel
University of California San Diego, Halıcıoğlu Data Science Institute

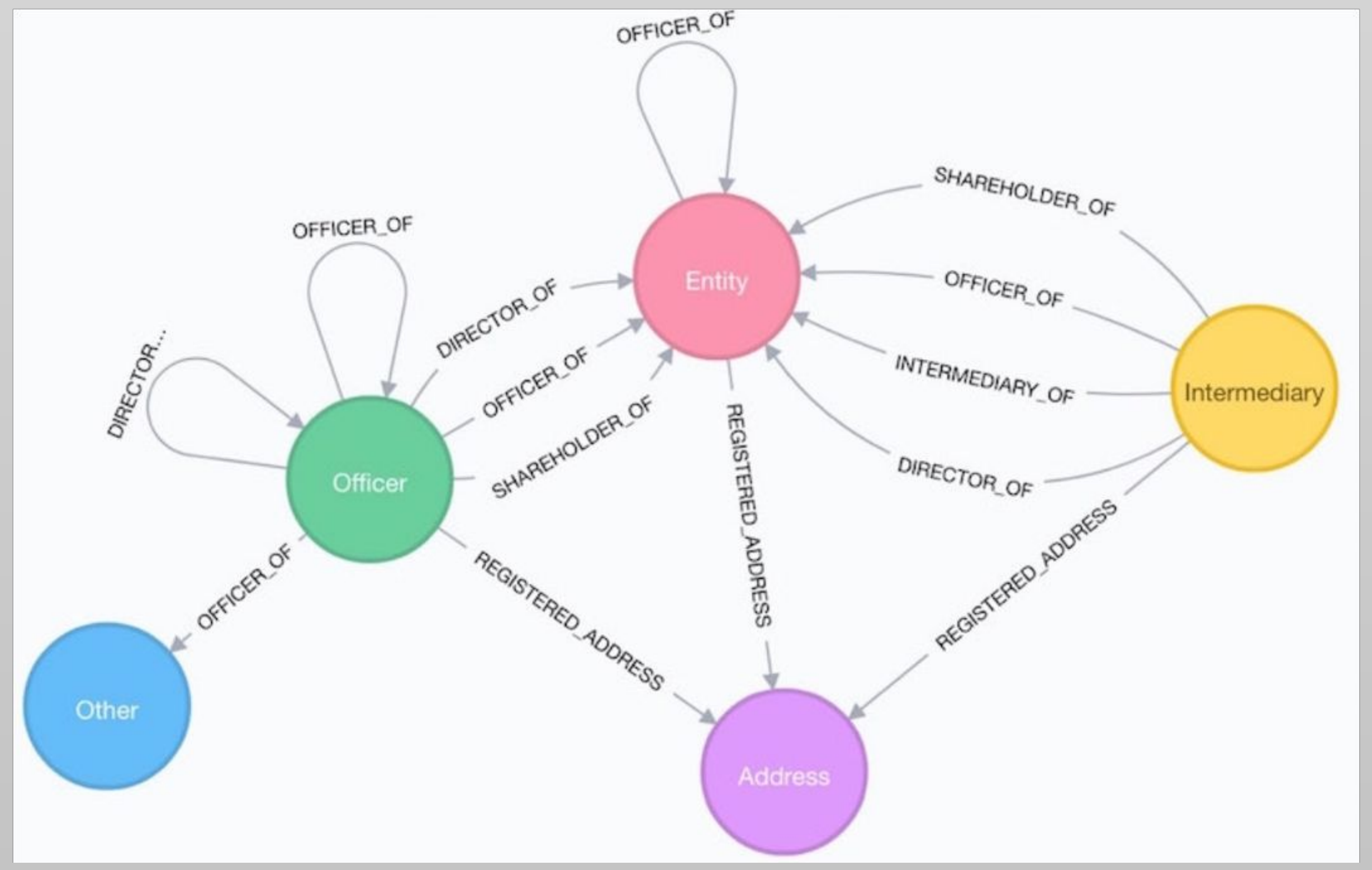
Abstract

Panama Papers and Paradise Papers are leaked documents that expose personal financial information of wealthy individuals and political leaders that take place in offshore companies. With hundreds of thousands of entities involved in these leaks, it becomes tedious for investigators to pick out the companies worth investigating. We propose a solution with our project that is to be able to predict what entities are likely involved in illegal activity. This abstracts the investigator from sifting through thousands of data points and closing the scope to the ones that matter. Our research utilizes XGBoost and feature extraction to find suspicious entities within our dataset

Dataset & Background

A graph database is a database that uses graph structures for semantic queries with nodes, edges, and properties to represent and store data. Each 'node' in our data was either entities, officers, intermediaries, addresses, or other, and each node has several properties depending on the type of node it was. For example, given an entity, it would have info on the entity's name, home country, etc. Each node can have relationships (directed edges) with other nodes, or be a disconnected component (no edges). After labelling, our dataset comprised of 6% suspicious and 94% unsuspicious entities.

With this in mind, we begin to think about what important data points we need to extract for the machine learning model.

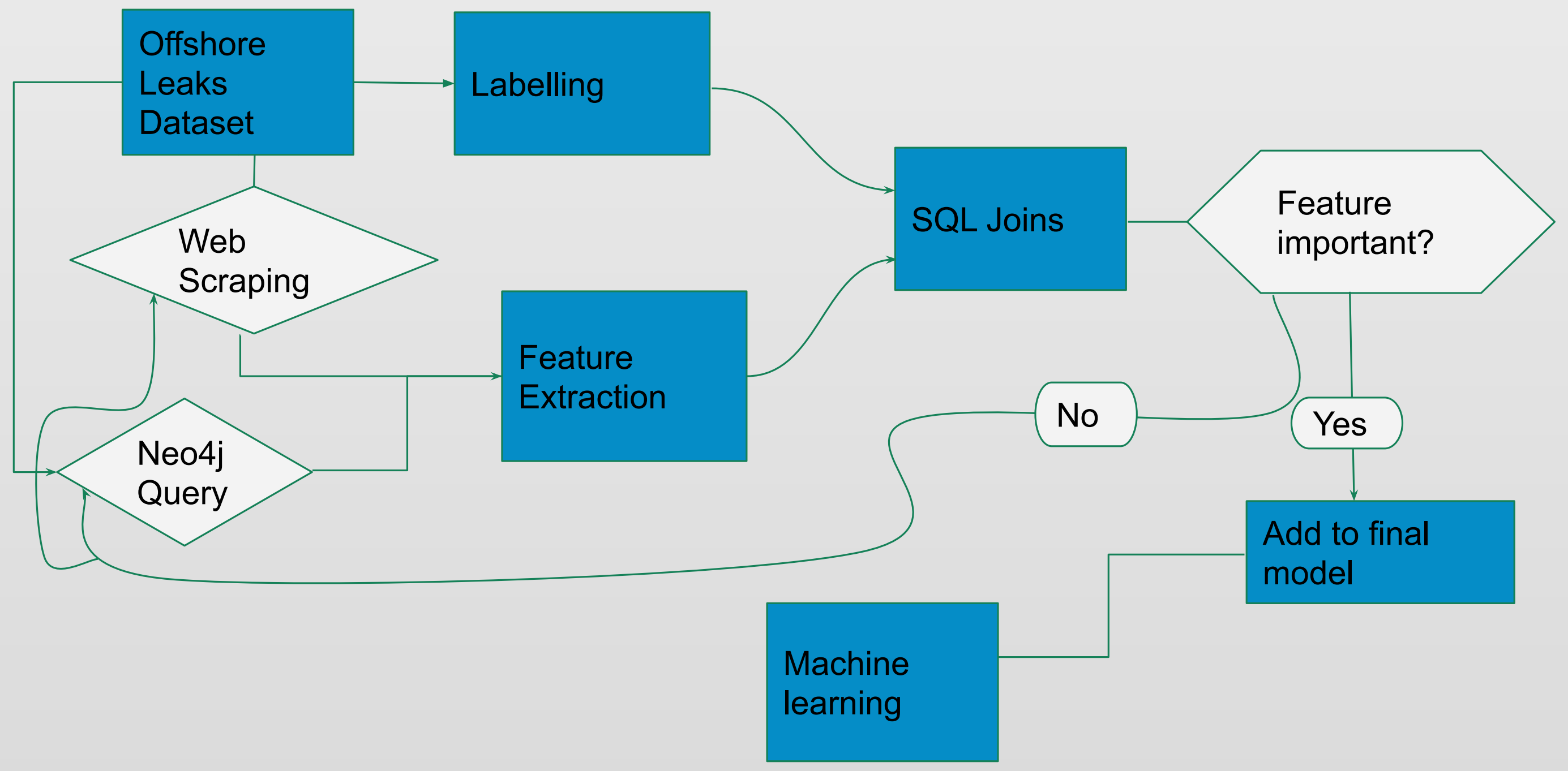


Labelling

We joined the entities present in ICIJ with other knowledge bases that contain blacklisted entities such as the Special Designated Persons List and Blocked Persons List from treasury.gov. Using SQL, we were able to merge our Paradise Papers and Panama Papers data and found 44 labels across both leaks accounting for about .002% of our dataset being labeled. In order to add more labels that we considered suspicious, we used a "friend of friend" approach.

The "friend of friend" approach assumes that those entities connected in the 2nd degree are also assumed to be involved in illegal activities. For example, if company A is blacklisted and that company is connected to Officer B who is also connected to other companies C, D, and E, then those three companies are also labeled as illegal, thus giving us 6% of our data that is labeled illegal, allowing for easier training.

Data workflow



Offshore Leaks Dataset

Labelling

Web Scraping

SQL Joins

Feature important?

Feature Extraction

No

Yes

Neo4j Query

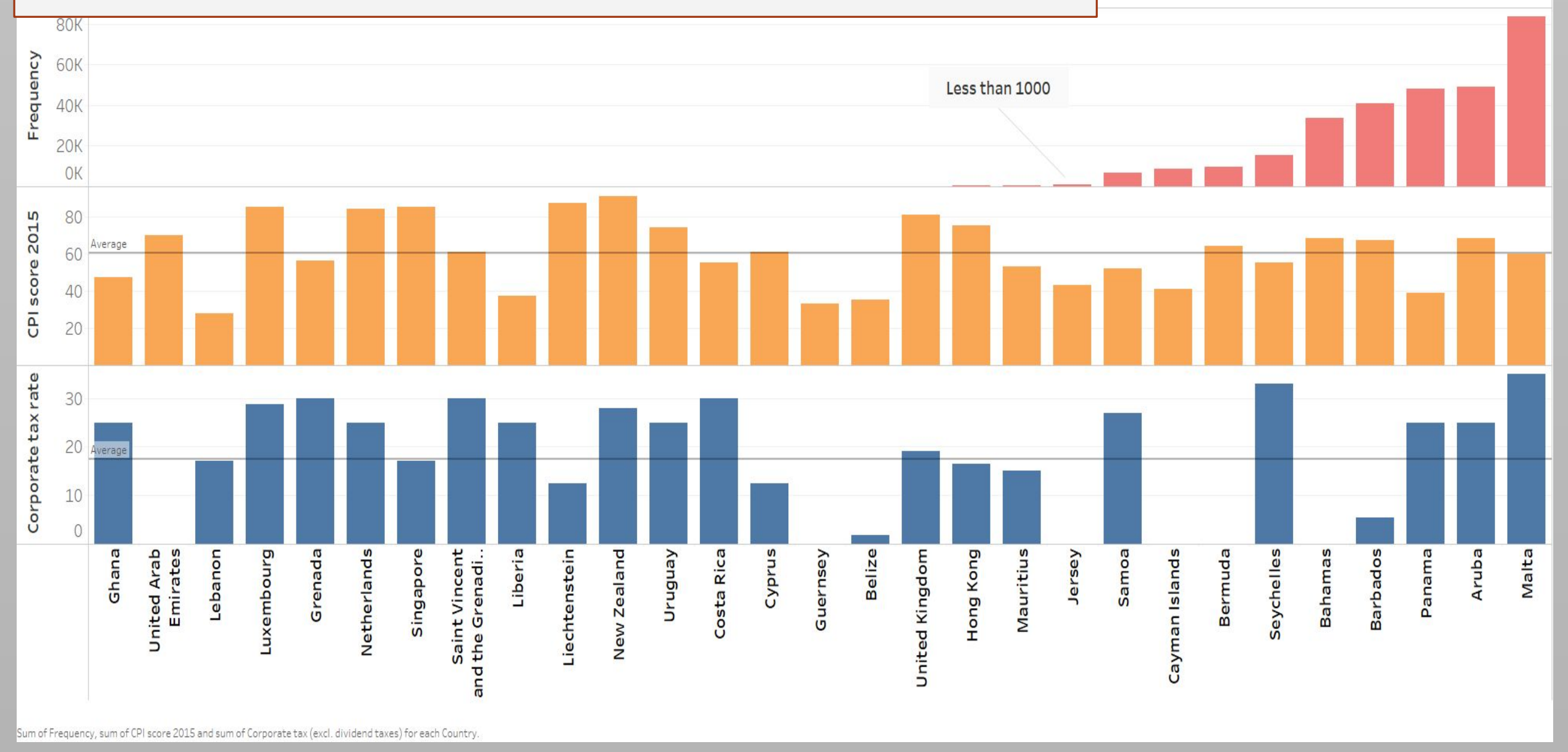
Add to final model

Machine learning

Feature Selection

- Corruption Perceptions Index (CPI) measures the corruption level using metrics like bribery and the diversion of public funds for each country's public sector. We used the CPI scores of the countries where the entities were situated to find a correlation between that entity and the corruption level. Our hypothesis was that if a country where an entity was located had a low CPI score, that entity was more suspicious.
- Corporate tax is imposed on corporations' income. One may set up an offshore company in tax haven countries where corporate tax rates are very low, or even 0. Our hypothesis was that if a country had a very low corporate tax rate, the offshore companies situated there are likely to be suspicious.
- Google developed the PageRank algorithm in order to determine the importance of each web page relative to others. The PageRank score is determined by the number of incoming connections and the corresponding importance for each. Our assumption here is that the entities are as important as the people or intermediaries that connect them and there might be a correlation with suspicion. Eigenvector Centrality follows the same principle but does not care about the direction of the connections.
- The purpose of the degree centrality algorithm is that it allows us to find the most "popular" nodes with the most edges connected to them. We believe that the number of connections will be relevant to determining if an entity is suspicious.

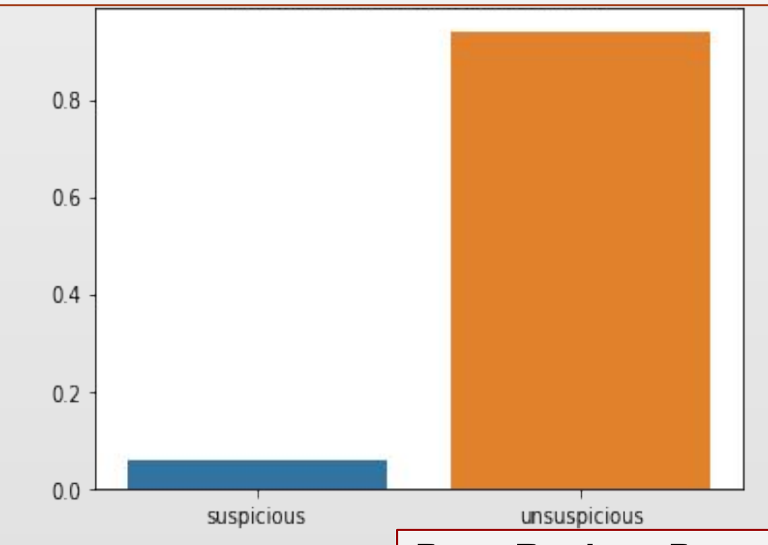
Frequency of Countries vs CPI And Corporate Tax Rate



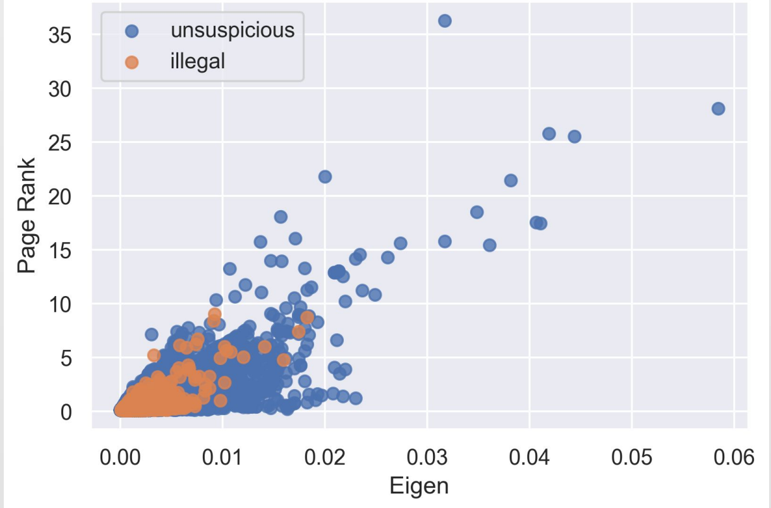
Sum of Frequency, sum of CPI score 2015 and sum of Corporate tax (excl. dividend taxes) for each Country.

Exploratory Data Analysis

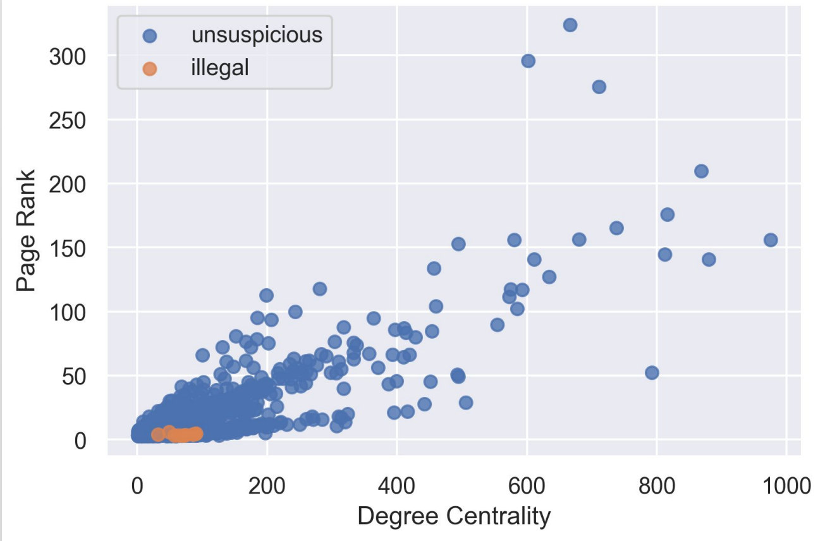
No. of suspicious vs unsuspicious classes



Pagerank vs Eigen for each label



Page Rank vs Degree Centrality for each label



Our Model

Logistic Regression

Began experimenting with logistic regression because it is known to do well for binary classification. After playing around with the hyperamaters, we were unable to significantly improve the results on best guess, so we had to move on to different model.

Random Forest

Following the assumption that suspicious entities have outlying values, we decided to use a Random Forest model. Each feature is taken one at a time and the most optimal threshold is determined by how to split the data and that is done until all of the points are classified. This model proved to perform better.

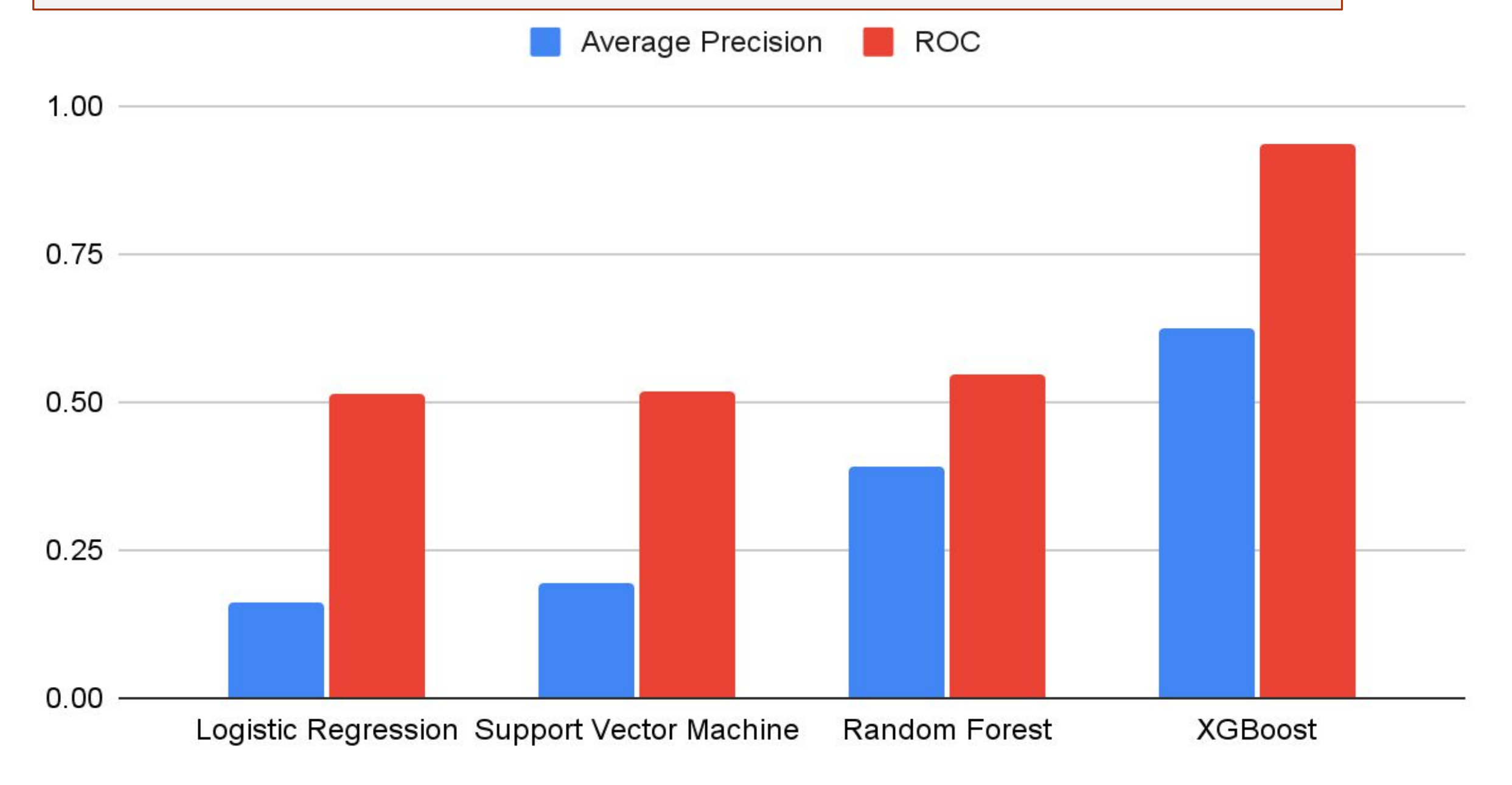
Support Vector Machine

A support vector machine classifier took the longest to train by far. We believed that suspicious entities had outlying values, so a hyperplane would be able to separate the two classes. However, due to the poor class balance, this model did not achieve significantly better results.

XGBoost

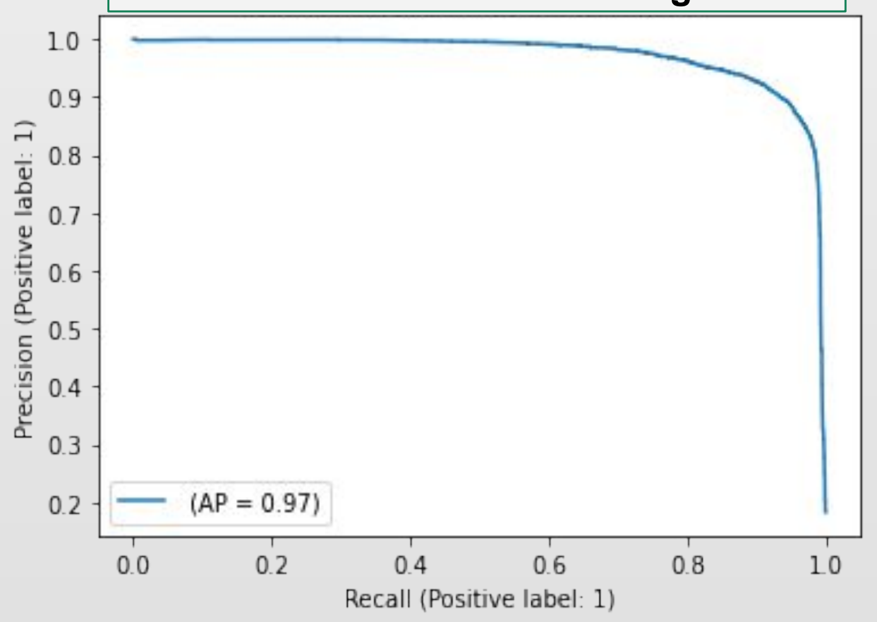
Extreme Gradient boosting received the best average precision score because of the fact that it's a regularized ensemble method where each tree boosts attributes that lead to misclassification. Hyperparameter tuning was used and positive labels were scaled to the ratio of positive to negative labels which is 1:17 because of the huge class imbalance.

Average Precision and AUC ROC For Each Machine Learning Model

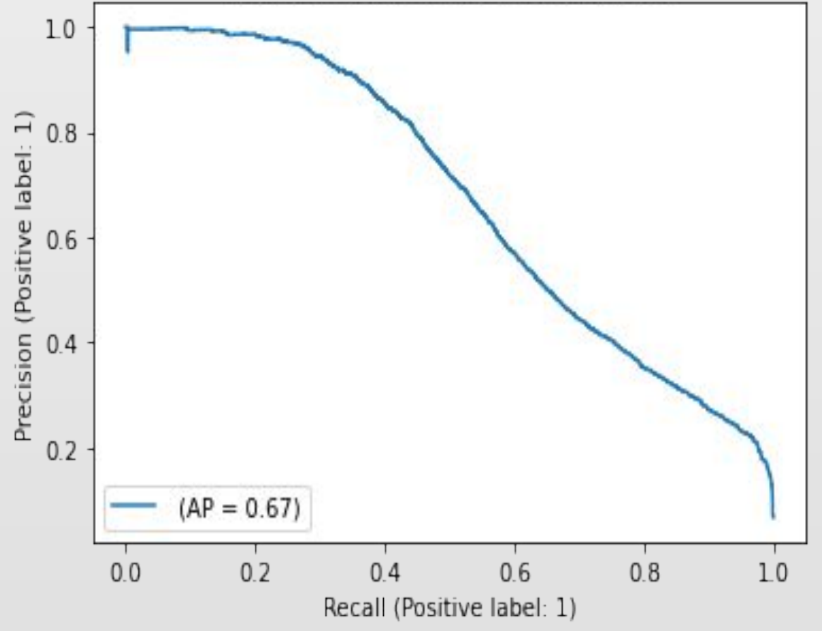


Results

Precision vs Recall For Training Data



Precision vs Recall For Testing Data



The precision-recall curve shows the tradeoff between precision and recall for different threshold. A high area under the curve represents both high recall and high precision, where high precision relates to a low false positive rate, and high recall relates to a low false negative rate.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Why is this important?

The ICIJ offshore leaks database consists of over 850,000 offshore entities and over 800,000 people that each have their own relationships. How can investigators know what key players to look for? Which entities are likely to be involved in illegal activities? Even the most basic question of where do you start? Using our algorithm, investigators can cut the time sifting through the database and be pointed to the entities that should be looked into for further investigation. Our model should be thought of as a tool that assists investigators in potentially reducing tax evasion, which deprives the government of money needed to carry out laws and initiatives, reduces the effectiveness of government, and increases budget deficits. Additionally, our model could reduce money laundering where financial assets are disguised and used for criminal activity such as nuclear proliferation.

Future Work

Ideally in the future, we would like to utilize big data tools such as Apache Spark to join our features engineered from the Neo4j database on all of the 1.9 million data points consisting of the other leaks which were Pandora, Offshore, and Bahamas as well as integrating data from open corporates that contains data such as incorporation date, dissolution date, etc. on 3 million companies worldwide. Another improvement could have been keeping the data in network form. The data is naturally in network type and making it tabular and running machine learning algorithms on the tabular form may have caused a decrease in performance, as opposed to making our own ranking algorithm that ran on the Neo4j database and predicted suspiciousness in that flow without any extra pre-processing.

References

Github