Determining Cryptocurrency Scams: A Machine Learning Approach

Michael Schultes

Group Member: Michelle Palumbo

CIS 4914 Senior Project

Advisor: Dr. Tao Li

tao.li@ufl.edu

4/22/2019

Abstract

With the rise of Cryptocurrencies in both popularity and market capitalization, the influx of new Initial Coin Offerings (ICOs) has led to a dangerous ecosystem of fraudulent and scam coins [1]. As opposed to other traditional banking systems, cryptocurrencies are decentralized and therefore they are not subject to government regulations. This has created a lack of a reliable credit rating system as the means for detecting a scam coin are not yet defined.

In response to this situation, we have developed multiple machine learning models to analyze ICO white papers to determine a pattern for recognizing scam coins. The models we successfully implemented are Naïve Bayes Classifier (supervised learning), K-means Clustering (unsupervised learning), and a Latent Dirichlet Allocation (LDA) Topic Modeler using Python 3.7.1. On a sample of 250 labeled whitepapers, the proposed supervised system can identify scam ICO projects with 0.678 precision.

We hope this work will help investors identify scam ICOs and bolster efforts in learning-based pattern recognition of ICO projects.

1.0 – Introduction

1.1 – Problem Statement

An increase in the creation of crypto coins in a domain where there is a lack of a reliable credit rating system for detecting fraudulent coins has created an influx of scam ICOs [1]. Machine learning is rising in popularity and is the most appropriate solution for detecting a pattern within the scam coin communities because it can analyze the growing volumes of white paper data that are being generated in an efficient manner [2].

1.2 – Related Work

The ICO white paper was chosen as the sample in our project because it is a document which includes an outline of the coin's project as well as a detailed description of the coin, its architecture, its interaction with users, and its team members [1]. This type of approach is like IcoRating, a project with relatable work that aimed to establish a credit rating system based on a coin's team, website, white paper, and GitHub repositories using an unsupervised model approach in conjunction with LDA topic modeling.

1.4 – Solution Statement

In this project, we propose a machine learning approach to classifying ICOs as "potential scams" and "not potential scams" using Naïve Bayes Classifier (supervised learning), K-means Clustering (unsupervised learning), and a Latent Dirichlet Allocation (LDA) Topic Modeler. This classification would inform potential investors about the risks involved with ICOs, as well as provide a sample set of labeled whitepaper data for use in similar machine learning models.

1.3 - Contribution

There are large databases of coin whitepapers, but none are labeled in a fashion that is suitable for use with machine learning [3]. In our pursuit to build our various models we have developed a sample of labeled, raw text whitepapers that did not exist beforehand. This contribution is invaluable to the cryptocurrency domain as well as similar machine learning projects in the future.

2.0 - Problem Domain

Cryptocurrencies utilize blockchain technology and rely on computer science advancements to operate. A trending tech buzzword, blockchain technology is defined as a growing list of records, called blocks, which are linked using cryptography. Companies are looking to invest into blockchain technology, and a large market is being generated for cryptocurrency coin. Analyzing the scams associated with the ICOs using machine learning also fits in the computer science domain as machine learning is rising in popularity and is becoming a factor in modern day business actions. The benefits of the proposed solution include a scalable approach that will take an unbiased look at the current scam problem to determine a pattern.

3.0 - Literature Search

3.1 - Parallel Research

Our main inspiration was from IcoRating as it was a large-scale project that aimed to establish a credit rating system based on a coin's team, website, white paper, and GitHub repositories [1]. The approach used machine learning and generated topic LDA to provide interesting results for coin investors as well as machine modelers.

3.2 - Library Research

The process of designing and developing the models involved searching for textbooks related to this subject. *Python Machine Learning by Example* [2] was the perfect resource that helped us to initially design and implement our models

3.3 - Internet Resources

The forum bitcointalk.org [3] provided us with a basic understanding of the bitcoin community and facilitated some of our initial questions about the scam ICO domain. It also relayed valuable information to us. Most of our sample set white papers were downloaded from coinmarketcap.com [4].

4.0 - Solution

The machine learning models were used to determine if an Initial Coin Offering (ICO) whitepaper is a "potential scam" or "not potential scam" (binary classification) using both supervised (training the machine on labeled data) and unsupervised learning. The LDA topic model was used to generate a list of topics within each classification.

4.1 - Preprocessing the Data

A library of 250 labeled whitepapers was generated with an even split of classifications. The data was labeled by the team based on their assessment utilizing litigated ICOs by the Securities and Exchange Commission (SEC), exploring crowdsourced technologies to find potential scams [1], targeting aggressively advertised coins on forums [3], and analyzing the ICO's website and white paper manually [4].

The data was then preprocessed to account for number and punctuation removal, human name removal, stop words removal, and lemmatization [2]. The preprocessing was assisted by the Python Natural Language Toolkit library. The preprocessed text is then tokenized into unigrams, bigrams, and trigrams (three consecutive words) and is turned into to a word vector using the Scikit-learn Python library.

A document frequency analyzer was then used to ensure that any word vectors that appeared in only a single document would be excluded as they would be too insignificant. In the same line of thought, any terms that appeared in over 50% of the documents were also removed as they would be ubiquitous.

4.2 - Naïve Bayes Classifier (Supervised)

The supervised machine model was built and trained using a Naïve Bayes model algorithm developed within Python. The word vectors from the preprocessing step were used as input for the model, but they first had to be distributed into various training and testing sets.

The data population was randomly divided into 3 samples (maintaining 50% classification split) named Training, Validation, and Testing. The standard practice for the size of each sample is to follow the 80/20 rule (80/20 training/testing, 80/20 training/validation) [5]. Therefore, the samples were of size 160 for Training, 40 for Validation, and 50 for testing.

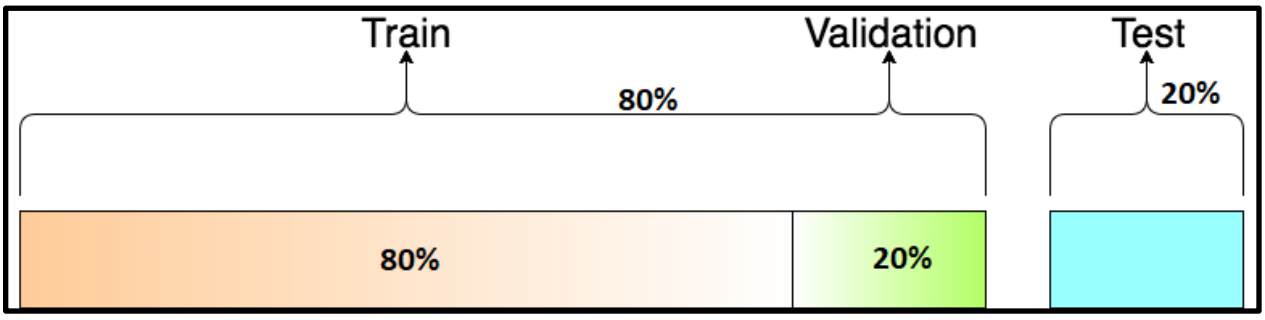

Figure 1: 80/20 rule [5]

After training, the model used the pattern it developed to predict the classification of the testing set. The results of the testing set prediction were used to score the accuracy of the model.

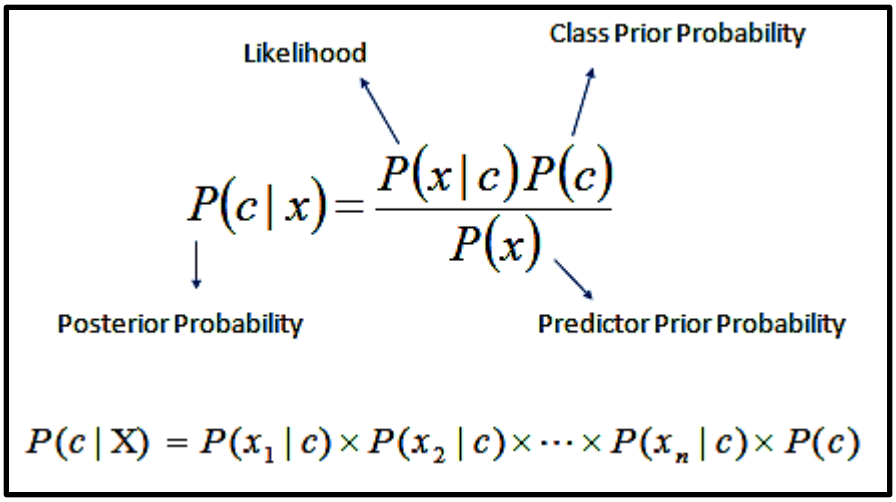

Figure 2: Naïve Bayes Classifier algorithm [6]

4.3 - K-means Clustering (Unsupervised)

The unsupervised model was built using a K-means clustering algorithm developed within python. The data was not labeled this time and instead the model had to come up with the classifications without any training.

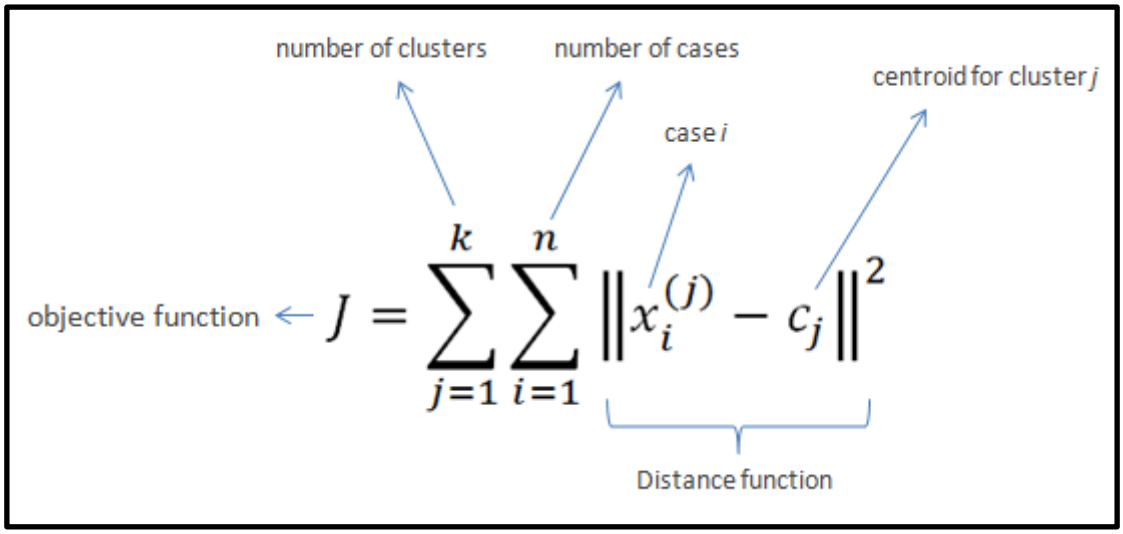

Figure 3: K-means clustering algorithm [7]

Again, the words vectors were fed into the model and a prediction of classification was made. The results of this prediction were used to score the accuracy of the model.

4.4 - LDA Topic Modeling

The LDA topic modeler was used to generate "Topics" from the whitepapers and output them so that the user will be able to see what common topics are within the different classifications of white papers. LDA topic modeling is a type of statistical modeling for discovering the abstract “topics” that occur in a collection of documents based on the word vectors of preprocessed text.

The topics generated could be used by an investor to make a smarter investment by providing a list of topics that are more prevalent in potential scam whitepapers that can be spotted when reading through them. The topics can also be integrated into a researcher’s machine learning model to improve the accuracy of the model’s prediction by providing additional keywords for the model’s vocabulary.

5.0 - Results

5.1 - Naïve Bayes Classifier (Supervised)

On a sample of 250 labeled whitepapers, the supervised machine learning model identified scam ICO projects with 0.678 precision.

```
The accuracy on 87 testing samples is: 67.8%
The accuracy using MultinomialNB is: 67.8%
              precision    recall  f1-score   support

           0       0.67      0.70      0.68        43
           1       0.69      0.66      0.67        44

   micro avg       0.68      0.68      0.68        87
   macro avg       0.68      0.68      0.68        87
weighted avg       0.68      0.68      0.68        87
```

Figure 4: Classification report from Scikit-Learn on the Naïve Bayes model

5.2 - K-means Clustering (Unsupervised)

On a sample of 250 whitepapers, the unsupervised machine learning model could not detect a pattern and the classifications were too inconsistent. Instead of a prediction, the model generated two clusters in which it grouped the white papers based on the word vectors.

Cluster 1 contents	50 not potential scams, 22 potential scams
Cluster 2 contents	75 not potential scams, 103 potential scams

Figure 5: Clustering results

In the process of generating the above model, valuable classification data was extracted. This data includes stop words, tokenized and preprocessed n-grams, and term frequency–inverse document frequency (TF-IDF) mappings. This data was compiled into spreadsheets at the request of our advisor in order to be analyzed later to improve the classification process with a scaled sample set.

Terms	tf_idfs
signature	8.203685562
game	7.859291151
card	7.75595949
token sale	7.244518009
hash	6.910360495
message	6.098513943
figure	5.827791046
layer	5.76895413
credit	5.762762637
fiat	5.706493208

Figure 6: Top 10 tf_idf terms used in cluster modeling

5.3 - LDA Topic Modeling

The topics generated by the model were organized into potential scam and not potential scam topics. These topics provide a machine's approach to the classification of the whitepapers and highlight some of the key topics relevant in each.

Potential Scam Topics	Game, player, sponsor, electric, win, explore, match, prize
Not Potential Scam Topics	Energy, credit, marketplace, api, retail, investor, host, fiat, whitepaper, copyright

Figure 7: Table of topics generated by the LDA topic modeler

6.0 - Conclusion

6.1 - Summary

Machine learning can be powerful, but it takes considerable effort to come close to the assessment of a human who is trained within the domain. Our models were good at generating a list of stop words and relevant topics and word vectors, but the prediction accuracies were

unfortunately low. A large sample set would undoubtedly increase prediction accuracy, but the means for attaining such a set are not yet plausible.

6.2 - Conclusions

As my first experience with machine learning and cryptocurrencies, this project taught me the importance of generating a good sample set for my models, as the models are severely reliant on the quality of the sample set. In this specific domain of scam ICOs, there is too much grey area to effectively analyze whitepapers and generate a classification of potential scam versus not scam, which is probably why no such sample set exists. Our models were designed to be scalable so that in the future if larger sample sets are generated our code could be reused to design a more accurate prediction.

6.3 - Future Work and Direction

Future work includes increasing the size of the sample set (to at least over 1000 whitepapers) while still maintaining the 50/50 split in classification. Also, our classifications are inherently biased and require further refinement by experts in the scam ICO domain. The results of our models can even guide this classification process as the data derived is relevant to the task.

7.0 - Standards and Constraints

All software was developed using Python 3.7.1 (32-bit), utilizing the libraries for NLTK 3.4 and Scikit-Learn 0.20.1 installed through the Anaconda 2018.2 package.

8.0 - Acknowledgment

Dr. Li, thank you so much for volunteering your time, energy and patience when working with us. You always contributed to our project with ideas and tips about how we should progress. I am extremely appreciative that you willingly donated time out of your busy schedule to be our advisor. You listened to what I had to say and took my recommendations seriously. The fact that you did this for several other teams at the same time is just a testament of your devotion to us as students. Thank you so much.

I'd also like to thank Dr. Schmalz for making this senior project experience as streamlined and enjoyable as the process allows for. Your professional and organizational skills resonated with me and have helped me to further develop my own.

Lastly, my senior project group member Michelle, thank you for working hard and keeping me accountable. You helped make this project exciting and I appreciate your dedication.

References

1. Bian, Shuqing. *IcoRating: A Deep-Learning System for Scam ICO Identification.* Cornell University, ArXiv Publishing, March 2018.
2. Liu, Yuxi. *Python Machine Learning by Example.* Packt Publishing, May 2017.
3. "Bitcoin Forum." *Bitcoin Talk*, https://bitcointalk.org/, Date Accessed: April 17, 2019
4. "CoinMarketCap." *Cryptocurrency Market Capitalizations,* https://coinmarketcap.com/, Date Accessed: April 17, 2019
5. "Naive Bayes Classifier." *UC Business Analytics R Programming Guide*, http://uc-r.github.io/naive_bayes/, Date Accessed: April 17, 2019
6. Dr. Saed Sayad. *K-Means*, https://www.saedsayad.com/clustering_kmeans.htm/, Date Accessed: April 17, 2019
7. "KDnuggets." *Big Data, Data Mining, and Data Science*, https://www.kdnuggets.com/2016/07/text-mining-101-topic-modeling.html/, Date Accessed: April 17, 2019

Appendix

```
# Transform texts to Tf-Idf (term frequency-inverse document frequency) coordinates
def cluster_texts(texts, clusters):
    vectorizer = TfidfVectorizer(tokenizer=process_text,
                                 lowercase=True,
                                 max_df=0.5,
                                 min_df=0.1,
                                 max_features=500)

    # Transform texts to Tf-Idf (term frequency-inverse document frequency) coordinates
    tfidf_model = vectorizer.fit_transform(texts)

    # Cluster using k-means
    km_model = KMeans(n_clusters=clusters)
    km_model.fit(tfidf_model)

    clustering = collections.defaultdict(list)

    # Output data obtained from the process
    feature_names = vectorizer.get_feature_names() # all words
    feature_mappings = vectorizer.vocabulary_ # words and the
    feature_stopwords = vectorizer.stop_words_
```

Figure: K-means clustering vectorizer

Biography

Michael is pursuing a Bachelor of Science in Computer Science Engineering at the University of Florida (UF) with an expected graduation date of May 2019. Hoping to find a career in the defense industry, Michael has a strong passion for national security and wishes to support the country's military through combat simulation training software.

Michael has spent his last three summers working as a Software Engineering Intern and Database Engineer at Lockheed Martin – Rotary Missions Systems in Orlando, FL. He supported the F-35 Pilot Training Devices (PTD) team with their current database contracts for the F-35 pilot flight simulator. Michael's tasks involved working with Geographical Information System (GIS) python toolkits within ArcMap where he updated legacy airfield content to the latest standards for F-35 PTD databases.

Michael currently has an offer for a full-time position with Lockheed as an Embedded Systems Software Engineer for the latest hypersonic missile contract.