Group Assignment Cover Page

Group Number/Team Name: 4

Last Name (Alphabetically)	First Name	Student #	Section #
Aquino	David	10134911	1
Horvath	Aidan	10132966	1
Krakovsky	Michael	10134030	1
Sahota	Abhit	10147531	1
			1

Course Title: Data Science for Business

Course Number: COMM 461

Assignment Title: Final Report

Professor: Mikhail Nediak

Due Date: April 26, 2018

I (We) certify that this submission is my (our) original work. I (We) have read the guidelines on Academic Integrity at https://smith.queensu.ca/about/academic_integrity/, and understand that any violation of academic integrity is a serious offense within the university community.

Using Online Sentiment and Other Quantitative Metrics to Predict Initial Coin Offering Success via a K-Nearest Neighbours Classification Model

COMM 461 Final Project
April 26th, 2018

David Aquino	10134911
Aidan Horvath	10132966
Michael Krakovsky	10134030
Abhit Sahota	10147531

Table of Contents

PROBLEM DESCRIPTION

Overall, a classification model was developed to predict the success of the Initial Coin Offerings (ICOs) of individual cryptocurrencies, based on features which are a blend of online sentiment metrics, and characteristics of the ICO itself. Thus, supervised learning was used. To understand the problem addressed by the model, a description of cryptocurrencies and ICOs is needed.

Cryptocurrencies and ICO's: An Overview

A cryptocurrency is a digital currency, that can be used to transfer assets and secure transactions. Cryptocurrencies feature a decentralized control system supported by a blockchain, a distributed ledger. Over the past year there has been a substantial increase in trading volume, with popular coins such as a bitcoin peaking at market capitalizations of 300 billion US dollars.

When a start-up firm wants to raise money, it can do so through an ICO instead of using traditional financing methods. To do so, the firm usually publicizes a white paper stating the details of the project and the purpose of the capital being raised. During the ICO campaign, enthusiasts and supporters of the firm's initiative are able to buy some of the distributed cryptocoins with fiat or virtual currency. These coins differ from the shares of a company sold to investors in an Initial Public Offering (IPO) in that they do not necessarily confer any ownership rights (Schleifer, 2017). Investors may only be able to profit via capital gains if the coin rises in value. If the money raised does not meet the minimum funds required by the firm, the money is returned to the backers and the ICO is deemed to be unsuccessful. If the funds requirements are met within the specified timeframe, the money raised is used to either initiate the new scheme or to complete it.

In 2017, 165 companies have raised more than $6.5 billion via ICOs, up from $226 million in 2016 (Richter, 2017). Often, companies promise investors something of perceived value for their money other than ownership, such as a discount on future services if the project succeeds. This aspect of ICOs is somewhat analogous to a kickstarter campaign: an ICO investment is similar to a donation when assessing the likelihood of success, or rather a form of speculative investment. Please note that while the terms "coin," "token" and "cryptocurrency" have taken on slightly different meanings within the crypto community, this report will use them interchangeably to refer to a digital currency with an upcoming ICO.

Description of Problem

Accurately predicting if a given ICO will successfully reach its funding goal would be extremely valuable for both potential investors and the organizers of the ICO itself, as noted in the "Business Implications" section. However, unlike for traditional financial securities or even more established cryptocurrencies, there is relatively minimal quantitative data available for ICO tokens on which to base investment decisions due to the premature nature of the firms raising funds and of the cryptocurrency market in and of itself. Due to this financial ambiguity, ICO tokens, perhaps unlike larger established coins, are heavily influenced by attention and marketing surrounding the ICO. But, without any proven measures to track a coin's popularity, pre-ICO investors must rely on accumulated knowledge and domain expertise to decipher probable ICO success and relative success as compared to other offerings.

With this in mind, the goal of this project was to create a model predicting whether an ICO will hit its funding goal or not, based on online sentiment towards the ICO, and characteristics of the ICO itself. The outcome of the model for a given ICO would thus be a binary classification of whether or not an ICO will hit its goal. An important distinction must be made here between an ICO's "hard cap" and "soft cap." A hard cap is the maximum target amount of funds an ICO will receive (above which funds will be returned to investors) while a soft cap is a far lower target which if not achieved results in all contributed funds being returned to investors and the cancellation of the project (Coinist, 2018). As a result soft caps are typically set extremely low by ICOs, and investors generally view a project's ability to hit its *hard cap* as a better signal of future success. Hence, this model defines ICO success as hitting the hard cap, not soft cap.

Data from the online forum Reddit was used as a proxy for *overall* sentiment facing a token, due to the platform's immense popularity within the cryptocurrency community. While other sources such as Twitter or private forums may also influence sentiment, these have the potential to have a more biased outlook due to their individual-based nature. The organization of Reddit content is also conducive to sentiment analysis: user posts and comments about the same topic are grouped into "sub-reddits" which can easily be scraped (most notable tokens would have their own subreddit). Further, comments are "up-voted" and "down-voted" by users to create an aggregate proxy for popularity within the community. Characteristics of each ICO were obtained directly from their respective white papers.

DATA COLLECTION AND PREPARATION

Overall, 9 features across 54 coins were used to train the model. The following image shows these features for the first 3 coins in our dataset (the complete set can be found in Appendix 1)

Name	Number of Users	Total # of Posts & Comments	Sentiment Mulitply Score	Weighted Average Sentiment	ICO Run Time	ICO Price	Supply Ratio	Largest Bonus	Hard Cap (USD millions)	Success
0xProject	11151	5053	1617	0.40142565	5	$ 0.05	50%	0%	$ 24.00	1
Accord	435	56	1.071429	0.33259525	32	$ 0.23	60%	30%	$ 10.00	0
AdHive	128	62	10.5119	0.15912328	15	$ 0.00	30%	10%	$ 17.50	1

Attributes Used

Target attribute
The target attribute ("Success" in the above table) was a binary variable, with 1 representing if an ICO reached its hard cap goal, and 0 representing if it did not.

Feature set
The features used to train the model fell into two groups: those relating specifically to online sentiment, and those relating to the characteristics of the ICO itself.

Sentiment-based features: As noted, the central hypothesis motivating the model is the notion that ICO successes are heavily impacted by online sentiment. The idea was inspired by investment banks who utilise sentiment analysis from Twitter to predict movements in stock prices.

1. *Number of Users:* The total number of online users subscribed to the subreddit of a particular coin as of April 10th, 2018. Obviously, the ICO dates of coins used for training all were prior to this date. However, the additional manual effort needed to determine the user number *as of each ICO's date* was deemed unnecessary since the majority of all chosen coins ICOed within the last three months and it seems unlikely that their followings would have changed substantially.
2. *Total Number of Posts and Comments:* The total number of posts and comments appearing on a coin's subreddit throughout the coin's ICO run time (see below).
3. *Weighted Average Sentiment:* A normalized metric on a [-1:1] scale (inclusive) representing the aggregate sentiment score for each coin (-1 = 100% negative, 1 = 100% positive) over the timeframe of one month before the ICO start date, to its closing. Its calculation is detailed in the "Sentiment Score Calculation" section below.
4. *Sentiment Multiply Score:* Weighted Average Sentiment multiplied by the total # of posts and comments. This was used to account for the fact that the average positivity or negativity of sentiment facing a coin, would be magnified by the sheer volume of comments about that coin. While it is likely this metric will be somewhat correlated with Weighted Average Sentiment, the use of regularization techniques in model training should mitigate this risk.

ICO characteristics: Beyond purely sentiment-based metrics, metrics relating to the ICO itself were included due to their likely impact on ICO success.

1. *ICO Runtime:* The number of days over which investors can contribute to an ICO. The hypothesis was that longer runtimes would enable more money to be raised.
2. *ICO Price:* The price per coin (USD) of the ICO; the rationale being that a lower price per coin may make the coins more liquid and hence increase the success probability.
3. *Supply Ratio*: The percentage of total available coins which are "released" by the ICO.
4. *Largest Bonus:* The percentage of "bonus" coins given to investors of the ICO (ex: 30% would mean that an investor paying for one coin would receive 1.3 coins). A larger bonus was hypothesized to incentivize investment and therefore contribute to success.
5. *Hard Cap:* The actual ICO hard cap target; a higher target was expected to make success more difficult, all else equal.

Notable features not included

One feature which was considered for inclusion was a metric representing the aggregation of ICO ratings provided by independent websites (similar to Buy/Sell ratings provided by Equity Analysts). Ultimately, this was not included, for two reasons. First, only certain coins are covered by certain sites, meaning data would be incomplete for the training set. Second, objectivity of the ratings is doubtful due to a lack of clarity regarding methodology of public ratings and possible influence from ICOs to receive a favourable rating.

Coins Used for Training

The aforementioned attributes were collected for 54 coins, creating the training dataset. The complete list of coins can be found in Appendix 1. The coins were chosen based on a random sampling of coins whose ICOs ended in approximately February 2018, beginning earliest in November 2017. Chosen ICOs were manually sourced from several different sources and feature data was cross-referenced across available sources and typically ICO white papers published by

the ICO team. This set was then adjusted to correct for its extremely unbalanced classes due to both market exuberance and the tendency of online ICOs ledgers and websites to feature more successful ICOs on their list. To account for this, coins with failed ICOs over the same timeframe were manually chosen, and used to randomly replace coins from the initial set. Ultimately, the final coin set contained 12 coins with failed ICOs, and 42 coins with successful ICOs.

Weighted Average Sentiment Score Calculation

The calculation of Weighted Average Sentiment Score for each coin consisted of three main steps: querying the subreddit pages of sampled coins, 'scrubbing' the data obtained, and training the sentiment generator model.

Querying subreddit pages
Once the sample set of coins was created, it was necessary to pull posts and comments from each of these coins' related subreddit pages over the timeframe analyzed. Since 54 coins were used, a custom tool was created to automate this process. Specifically, a program was written which utilized an API called pushshift.io, enabling the scraping of reddit content. This API requires the user to create a URL that specifies the data they wish to receive. The example below illustrates how the URL was structured:

https://api.pushshift.io/reddit/search/submission/?subreddit=NEO&after=1523269172&before=1523272772&size=500

Four parameters needed to be hard-coded for each coin: the date to start the scrub ('after'), the date to end the scrub ('before'), the token's subreddit name ('subreddit'), and the number of comments to pull ('size'). The start time of every token was adjusted to begin roughly one month before the ICO start date, while the end date was kept constant at April 10, 2018. We later decided to only include the final 20 days after the end of each ICO to standardize the testing process. The token's subreddit was found online through a Google search. 'Size' was limited to only 500 submissions at a time by the API itself. Please see Appendix 3 for the complete program code used.

The data was transmitted in a JSON format and was inputted into a dictionary. For each comment or post, the following data was obtained:

Data Field	**Description**
ID	A tag that is unique to the Reddit submission.
Score	The popularity of the submission measured by net up/down votes. More popular comments receive a higher score while non-popular comments are closer to 0.
Date Created	When the submission was created.
Comment	The user's post or comment.
Author	Who the user was.

Data scrubbing
Once all the data was obtained, it was "scrubbed" to remove redundancies and format it correctly for sentiment analysis. The python package NLTK (Natural Language Toolkit) was used to remove punctuation and make minor formatting changes. Submissions which had been deleted on Reddit,

but were still captured by the API, were also removed. Deleted submissions are still shown on Reddit's forums in the format '[deleted]'. While these do likely somewhat contribute to online sentiment, their impact is probably far less than that of non-deleted content.

Model training

In order to properly classify the comments and posts obtained for each coin, a classifier model needed to be trained. The use of IBM Watson's built-in sentiment classifier was considered but ultimately rejected due to (1) the "black-box" nature of this program, and (2) the extremely domain-specific vocabulary used to describe cryptocurrencies on Reddit. Hence, a custom model was created from scratch. This consisted of two steps: (1) creating the training dataset, and (2) training the model.

Since there were no Reddit sentiment training datasets online, one needed to be created. This process required manually pulling content from subreddits relating to cryptocurrencies not in the sample set, classifying the content as either positive or negative, and thus creating two text files containing all positive and all negative content, respectively. The positive training set contained over 750 posts and comments while the negative set contained over 900 posts and comments. From here, the NLTK package was used to create a collection of the 5000 most common adjectives, adverbs, or verbs that appeared in the entire collection of posts and comments. Next, a dictionary containing these 5000 words as keys, and either "True" or "False" as values, was generated *for each post / comment* ("True" indicated the word was included in the comment, "False" indicated it was not). This dictionary was combined with either a "pos" or "neg" tag (indicating whether the post / comment was classified as positive or negative) in a two-element tuple. This set of tuples made up the dataset used to train and test the sentiment classifier. A snippet of the tuple can be visualized as follows:

{'offered': False, 'crashing': False, 'issuing': False, 'thin': False, 'changed': False, 'decided': False, 'losing': False, 'clean': False, 'exaggerating': False}, 'pos')

Seven classifier models were trained by splitting this dataset into training and testing portions. These models were Naive Bayes, Multinomial Naive Bayes, Bernoulli Naive Bayes, Logistic Regression, SGD, Linear SVC, and Nu-Support Vector Classification. Each model was given a equal-weighed hard vote on whether a specific comment displayed positive or negative sentiment, creating two outputs: (1) the sentiment category (either positive or negative) as determined by majority voting, and (2) the percentage of models in agreement with this category. This percentage was multiplied by -1 for negative sentiment content, resulting in a final sentiment score within the range [-1, 1] for each post and comment. Each sentiment weighting was multiplied with a normalized version of the 'popularity' score of the reddit submission. Therefore, Reddit content with higher scores would receive a higher weighting. The final ICO sentiment equals the summation of all the weighted scores throughout the time frame of the ICO indicating the overall sentiment for the token's associative subreddit.

MODEL TRAINING METHODOLOGY

Overall Approach

Model training consisted of four high-level steps. First, a set of candidate model types were chosen based on their applicability to the problem and dataset. Second, each of these models were trained using python. Third, the individual performances of each of these models were enhanced where possible using various techniques (ex: ensemble learning and regularization). Finally, the highest performing of these models was chosen. Note that other approaches to the final step were considered (ex: voting, model stacking); the reasoning for *not* using these approaches is detailed below. Access to the complete code used for model training can be found in Appendix 5.

Initial Model Set

Five model types were chosen to train: K-nearest neighbours (KNN), a decision tree, logistic regression, linear discriminant analysis (LDA), and Gaussian Naive-Bayes (NB). Each of these models is appropriate for a classification problem such as this. Further specifications related to each model are outlined in "Initial Training."

Certain model types were considered but not used. Specifically, given the relatively small number of instances in the dataset (54 coins), neural networks were rejected since they tend to be more effective on large datasets. Further, Bernoulli Naive Bayes was considered however research indicated it is generally more applicable to text analysis problems. Finally, Multinomial Naive Bayes was not appropriate since only one target attribute was considered.

Initial Training

Initial training on each of the chosen models was completed using the scikit learn python package. Sample code for the KNN classifier is shown below:

```
# 1. Creating a K Nearest Neighbors
clf = KNeighborsClassifier(n_neighbors=6)
clf.fit(xTrain, yTrain)
print("KNeighbors: " + str(clf.score(xTest, yTest)))
```

Specific considerations were needed for certain model types:

- *KNN:* Euclidean distance was used instead of manhattan or jaccard. Manhattan distance is not applicable to the problem while jaccard distance was impossible to use since the features of each coin could not be expected to exactly match those of others. Further, a nearest neighbours cut-off of 6 was used. By iterating through k-values of 3-10, 6 was determined to deliver the highest testing-set accuracy.
- *Decision Tree:* A maximum depth of 3 was used. Again, this was determined to be optimal by iterating through a variety of potential depth values. Please see Appendix 6 for a visualization of the decision tree.

Of the entire dataset, ~20% of the coins (11) were randomly chosen as a holdout set; the remaining 43 coins were used to train the models. This approach yielded the following accuracies on the testing (holdout) set:

Model	Testing Set Accuracy
Decision Tree	0.8182
KNN	0.9091
Logistic Regression	0.7273
Gaussian NB	0.3636
LDA	0.8182

Considering that classifying *all* coins in the dataset as successful would yield an accuracy of 0.778 (42 successful coins out of 54 in total), these results were somewhat promising, especially those of the Decision Tree, KNN, and LDA models.

One interesting result of this analysis pertained to the decision tree classifier. As visualized in Appendix 6, this model classified coins with *longer* ICO run times as unsuccessful, contradicting the initial hypothesis that longer run times would equate to a higher likelihood of success. A potential reason for this is the impact of signalling and confidence. It is possible that ICOs with stronger fundamental use cases or technologies will be more confident in their ability to obtain their desired funding, and hence will set a shorter ICO runtime in order to secure this funding faster. Compounding this effect is that potential investors may as a result see shorter runtimes as a signal of a worthwhile investment, and therefore be more likely to contribute to the ICO. Both factors support the observed result that shorter runtimes increase the likelihood of success.

Fine-Tuning of Each Model, and Model Comparison

In order to further increase the testing accuracy of each model, certain techniques were used, each of which are described below. Then, the resulting performances of each model were averaged over 1000 iterations before ultimately being compared.

Bagging (applied to all trained models)

A bagging classifier imported from scikit learn was applied to each of the models. Bagging trains each model iteratively on different random subsets of the overall datasets, before aggregating the results. It hence reduces the portion of generalization error arising due to variance in the training data (i.e. overfitting). This ensemble learning method was chosen over boosting (which reduces generalization error due to bias / underfitting) because of the small size of the dataset. A dataset of 54 coins means that any significant outliers could be expected to have a relatively large impact on model training, compared to if a far larger dataset was used. Hence, there was deemed to be a substantial risk of overfitting, supporting the use of bagging as opposed to boosting. Inspection of the training data supported this choice since there appeared to be certain coins which were obvious outliers (ex: FundRequest had the 3rd highest Weighted Average Sentiment Score but did not successfully hit its ICO target). Sample code of the bagging classifier used is as follows (for KNN):

```
from sklearn.ensemble import BaggingClassifier
kNeighboursBagging = BaggingClassifier(KNeighborsClassifier(), max_samples=0.5, max_features=1.0, n_estimators=20)
kNeighboursBagging.fit(xTrain, yTrain)
print("KNeighbors Classifier: " + str(kNeighboursBagging.score(xTest, yTest)))
```

Max_samples indicates the number of samples to draw from the dataset to train each base estimator. A value of 0.5 indicates that half the sample set was used. Max_features indicates the maximum proportion of features used in a sample. Since there are relatively few features in the

sample set, max_features was kept at 1.0 (i.e. 100%) because using all features will not lead to performance issues. N_estimators is the number of base estimators in the ensemble, and was increased to 20 so as to improve accuracy without significantly slowing code execution.

Bagging was effective at increasing the accuracy of four of the models (decision tree, logistic regression, Gaussian NB and LDA), while KNN was unaffected:

Model	**Testing Set Accuracy (Pre-bagging)**	**Testing Set Accuracy (Post-bagging)**
Decision Tree	0.8182	0.9091
KNN	0.9091	0.9091
Logistic Regression	0.7273	0.9091
Guassian NB	0.3636	0.6363
LDA	0.8182	0.9091

Regularization (applied to Logistic Regression only)
Lasso regularization was applied to the logistic regression model. This was deemed important since only a small number of features were used, meaning there was the potential for certain specific features to have a disproportionate impact on model performance. Regularization would correct for this by penalizing features given very high weightings. A regularization parameter built in to the scikit learn logistic regression algorithm was used to control this.

Approaches considered but not used
Like boosting, certain approaches for improving model accuracy were considered but not used. First, a random forest approach was attempted for the decision tree, however resulted in considerably lower test set accuracy than the bagged model (0.8182 vs 0.9091). One potential reason for this is the relatively small set of features in the dataset, limiting the benefits of randomization. Second, dimensionality reduction (PCA) was not utilized. Given the small number of features to begin with (9), the cost/benefit trade-off between the computational power and time required, and the potential benefit from any reduction in features, was deemed to be unfavourable.

Average Model Performance over 1000 Iterations
Following the above adjustments, each model type was ran 1000 times on the training data, to account for variances in the random samples chosen for bagging. The mean and standard deviation of each model's accuracy over this set was then determined, shown below.

RESULTS AND FINAL MODEL CHOICE

The final results of the above methodology are as follows (over 1000 iterations):

Model	**Mean Test Set Accuracy**	**Standard Deviation of Test Set Accuracy**
Decision Tree	0.8091	0.0858
KNN	0.9091	1.11e-16
Logistic Regression	0.8909	0.0364
Gaussian NB	0.5909	0.0838
LDA	0.8545	0.0603

These results clearly indicate that KNN is the superior model of the group. It has the highest mean accuracy (1.82% higher than logistic regression) and the lowest standard deviation. Standard deviation of model accuracy was deemed an important consideration because a higher standard deviation would indicate significant sensitivity to changes in the training or testing sets, making real-world model performance less reliable.

Final Model Choice and Rationale

Based on the above factors, it was decided that utilizing specifically the KNN model as the ultimate predictor of the target attribute was the optimal strategy, as opposed to other approaches such as model stacking or voting. Model stacking would have necessitated splitting the dataset into two subsections, one for training the initial set of models, and one for training the higher-level "blender" model. Given the limited dataset size, this approach would likely have resulted in an increase in generalization error of the lower-level models due to over-fitting (the training set would only have been 27 x 80% = ~22 coins), ostensibly negating any benefit from model stacking. Voting techniques were also considered, however when tested resulted in significantly poorer performance than KNN on its own. Specifically, soft voting weighted by each model's test set accuracy produced a mean accuracy of 0.857 with a standard deviation of 0.189 when ran 1000 times, which is clearly inferior to KNN. It is hypothesized that this result is due to the individual models being somewhat correlated because of the small dataset size, which would negate any advantages deriving from integrating weaker learners (ex: LDA, Gaussian Naive Bayes) into the final classification. Hard voting and non-weighted voting variants of the above approach both produced similarly inferior results. When compared to the accuracy which would have been obtained had each ICO been merely classified as "successful" (0.778), it is clear that KNN does provide valuable predictive power.

In addition to its quantitatively superior performance, certain other factors support the choice of KNN as the overall predictor. First, it is relatively insensitive to outliers (Bronshtein, 2017). This is very important for this application, since cryptocurrency markets are characterized by large swings and a few outlying coins (ex: bitcoin, ethereum). Hence, were the training set to be supplemented in the future with more ICOs as they occur, KNN would be less affected than other models by market variations at the time of each ICO. Furthermore, KNN is a very simple model to visually understand even for someone with a non-technical background (Bronshtein, 2017), which is important given that many of the business implications discussed later on in this paper would involve investors as opposed to data scientists or statisticians.

One disadvantage of KNN compared to other models is that because it stores the entire training set in memory, it is computationally expensive (Bronshtein, 2017). However, mitigating this is the relatively small size of the training set used. Furthermore, an average of only ~113 ICOs have occured per month over the last year (ICO Bench, 2018), meaning even if the training set was grown over time with new ICOs, its size would remain manageable for the foreseeable future.

DISCUSSION OF RESULTS

While a mean accuracy of 0.9091 is relatively strong, a few factors must be kept in mind when considering the robustness of this value. First and foremost is the aforementioned relatively small sample size used, in terms of both coins, and comments used to train the sentiment generator. Since

there was no pre-existing dataset which would have properly enabled model training in either of these cases, much of the effort put into this assignment related to the construction of these datasets. Hence, their sizes were largely limited by time. Larger training sets would make the achieved accuracy more robust. Furthermore, it is noted that the sampled ICOs predominantly occurred within a timeframe over which the general cryptocurrency market was experiencing record activity and valuations. Since KNN will compare any new ICOs to this training set in order to predict their success, reduced accuracy may result from a market correction that somewhat alters the parameters important for ICO success.

To improve the dataset size and account for any future changes in cryptocurrency markets, it is recommended that the training sets for both the sentiment generator and the overall model be continually updated as more data becomes available. To truly account for any material changes to market dynamics, one avenue which could be explored is to use a *rolling* set of the 'x' most recent ICOs, as opposed to keeping *all* historic ICOs in the training set indefinitely. This would ensure that nearest neighbours classification is not unduly influenced by ICOs occuring at a time when success factors were different from those impacting the market at the time of classification.

BUSINESS IMPLICATIONS

Both potential investors in ICOs, and the teams running ICOs themselves, would have certain use cases for the model.

Cryptocurrency Investors / Crypto-Investment Funds
Due to the immaturity of the cryptocurrency market, individual investors and established crypto-funds all leverage unique methodologies to ICO and coin valuation. Furthermore, due to the lack of fund analysts and regulation regarding financial reporting, investors typically rely on ICO directories for aggregated information of ICO campaigns. As a part of the ICO directory business model they typically prescribe to ratings and evaluations of listed ICOs under a *freemium* pricing plan. We have identified that ratings for ICOs all feature a 'hype' rating but have very superficial scoring methodologies that provide little insight into the confidence within the investing community. Current rating systems are also static: they are unable to dynamically adapt to negative or positive news on the ICO. Via our model we can provide the following services which together would address this gap in the market:

Pre-screening ICOs for further analysis. With limited funds to invest and so many cryptocurrencies available, investors must make difficult decisions regarding which ICOs are worthy of their investment. At the current rate there are over 160 ICOs that conclude monthly, cumulatively averaging over a billion dollars in capital raised. Deciding which of these to invest in is important because an investment in an ICO which is ultimately unsuccessful has two potential consequences. At best, valuable capital is tied up in an unproductive venture for the duration of the ICO and thus creates opportunity cost. At worst, the capital may be lost entirely if the underlying project faces bankruptcy as a result of the failed ICO. This model can help investors identify ICOs likely to succeed and therefore may act as a "first pass" with which to evaluate prospective investments.

Detecting exit scams. An exit scam is a strategy where a cryptocurrency disappears from the market soon after collecting money from its ICO. Confido and LoopX are two examples of coins

which did this in recent months, scamming investors out of $175,000 and $4,500,000 respectively (Seth, 2018). There is evidence that retrospective sentiment analysis seemed effective at uncovering many of the warning signs leading up to these scams, and that these indicators negatively impacted the sentiment facing these coins (Seth, 2018). Hence a modified version of the model more heavily focused on sentiment could act as a predictor for exit scams.

Identification of non-typical investment opportunities. The cryptocurrency market is a global phenomenon with activity concentrated in countries such as China, Russia, Korea, Japan and the United States. Language and cultural barriers exclude many investors from understanding relative sentiment of non-native ICOs. With extensions to the sentiment model and language modifications to the data scraper, investors could better understand the community confidence of an international ICO and therefore expand their range of potential investments.

Cryptocurrency Leadership Teams / ICO Consultancy Firms

Further development and analysis of contributing factors of ICO success could be used as the foundations of a ICO campaign consultancy firm. It is common practice for start-up firms to outsource their marketing for ICOs to specialized firms to increase probability of ICO success; BlitzCrypto is one successful example of such a firm. The developed model could serve these firms, or the ICO teams directly, by enabling them to do the following:

Understanding factors which lead to ICO success. By observing the impact of each of the features used by the model on the target attribute, ICO teams could adjust the terms of their offerings accordingly in order to increase the probability of success. For example, if ICO runtime was negatively correlated to success, then ICO teams could shorten their runtime to the extent possible.

More accurately predicting of whether their ICO will successfully reach its goal. By using the developed model on their own token as its ICO date approaches, ICO teams could predict the likelihood of their ICO hitting its hard cap funding goal. This is important because ICO funds generally are used to finance future expenditures for the project. Hence, the more accurately a project can predict its ICO success, the better suited it will be to plan expenditures accordingly and pre-emptively seek out other funding sources (ex: debt or traditional equity investors) if needed.

Understanding the importance of Public Relations. More generally, the high accuracy of the model illustrates the importance of public relations to ICO success, a fact that may often be overlooked by leadership teams. With such a strong correlation between public sentiment and ICO success, cryptocurrency projects should be investing heavily in public relations to keep the online sentiment towards their coin positive. Teams should also avoid making decisions that they know will be viewed negatively by the public, as the resulting negative sentiment may inhibit their ability to raise funds via ICO.

CONCLUSION

Overall, a supervised classification model based on the K-Nearest Neighbours algorithm was developed to predict whether or not a given Initial Coin Offering would successfully reach its hard cap funding goal. This model utilized bagging in order to increase its accuracy, and outperformed other modelling techniques including other classification algorithms, model stacking, and

weighted voting. Its mean testing set accuracy over 1000 iterations was 0.90 with a standard deviation of 1.11e-16.

The model's primary use cases are as an analysis tool for prospective ICO investments, and as an input into decisions by ICO teams. Hence, it fills the need for a sentiment- and data-driven predictor of ICO success in a space which is largely devoid of such tools to date. Major next steps include the continued expansion of the training sets used to train both the sentiment predictor and the overall classification model, development of the natural language classifier, and extension to other online communities besides Reddit.

APPENDICES

Appendix 1: Complete Training Dataset

Name	Number of Users	Total # of Posts & Comments	Sentiment Mulitply Score	Weighted Average Sentiment	ICO Run Time	ICO Price	Supply Ratio	Largest Bonus	Hard Cap (USD millions)	Success
0xProject	11151	5053	1617	0.40142565	5	$ 0.05	50%	0%	$ 24.00	1
Accord	435	56	1.071429	0.33259525	32	$ 0.23	60%	30%	$ 10.00	0
AdHive	128	62	10.5119	0.15912328	15	$ 0.00	30%	10%	$ 17.50	1
AionNetwork	4069	428	215.2684	0.41468901	1	$ 1.00	27%	25%	$ 20.00	1
ArcBlock	1960	5995	2647.749	0.44058664	2	$ 0.50	45%	8%	$ 45.00	1
BABB	626	876	235.5666	0.25416066	16	$ 0.00	60%	0%	$ 20.00	1
Banca	40	6	38.57143	0.511978	32	$ 0.00	35%	0%	$ 20.00	1
BankEra	1112	2323	412.3479	0.1667171	94	$ 0.02	30%	0%	$ 215.90	0
CLN	2	0	0	0	5	$ 0.10	35%	0%	$ 50.00	0
CoinLion	51	25	14.02041	0.48404	70	$ 0.45	50%	0%	$ 18.00	0
Copytrack	37	27	0.314286	-0.01712	73	$ 1.19	60%	0%	$ 62.90	0
CREDITS	1124	1711	-11.5967	0.02049985	12	$ 0.19	60%	30%	$ 20.00	1
Current	589	474	277.5828	0.58060463	57	$ 0.24	35%	0%	$ 36.00	1
Datawallet	74	31	15.67857	0.50297619	31	$ 0.18	33%	0%	$ 39.44	1
Debitum Network	356	456	48.7752	0.14573599	32	$ 0.13	60%	20%	$ 17.20	1
Dether	378	179	35.74297	0.19593537	3	$ 0.23	60%	17%	$ 13.40	1
dock.io	11716	725	110.2046	0.16407651	15	$ 0.08	30%	20%	$ 20.00	1
EBCoin	1	0	0	0	29	$ 0.04	50%	20%	$ 25.30	0
Electrify.Asia	671	485	202.1923	0.4598277	8	$ 0.08	50%	0%	$ 30.00	1
Electroneum	18353	8477	2654.036	0.30336179	40	$ 0.01	29%	0%	$ 40.00	1
EximChain	16	8	5.142857	0.492063	30	$ 0.33	40%	0%	$ 20.00	1
FintruX	159	49	13.42091	0.20434032	22	$ 0.58	75%	10%	$ 25.00	1
FundRequest	187	20	13.1	0.60784246	12	$ 0.48	40%	0%	$ 15.00	0
Fusion	346	219	36.67738	0.22954599	11	$ 2.06	25%	50%	$ 42.20	1
FuzeX	70	32	11.52381	0.36772487	32	$ 0.08	60%	25%	$ 36.20	1
GBX	273	122	73.97719	0.61136599	31	$ 0.10	35%	0%	$ 27.00	1
Globitex	265	246	67.92663	0.34287369	29	$ 0.12	65%	0%	$ 12.30	1
GRAFT	865	2208	664.7002	0.29542633	32	$ 0.42	10%	0%	$ 25.00	0
Havven	1104	321	118.7954	0.44728491	9	$ 0.50	60%	50%	$ 30.00	1
HERO Token	143	70	31.34624	0.47824004	28	$ 4.35	80%	30%	$ 173.90	0
Huobi token	329	765	114.5101	0.13441392	36	$ 1.52	60%	0%	$ 300.00	1
InsightsNetwork	23	3	0.857143	0.309524	71	$ 0.18	40%	5%	$ 17.40	1
iungo	234	106	16.55505	0.20362892	57	$ 1.15	64%	50%	$ 45.00	1
Lendroid	19	0	0	0	7	$ 0.02	40%	0%	$ 47.50	1
Lympo	405	417	190.4524	0.44081985	12	$ 0.02	65%	20%	$ 12.70	1
NaPoleonX	481	935	394.7196	0.41670784	38	$ 0.87	70%	20%	$ 18.30	0
nebulas	4378	2705	1222.369	0.47382415	9	$ 2.00	30%	0%	$ 60.00	1
NucleusVision	60414	1450	859.2043	0.51966602	37	$ 0.01	40%	25%	$ 40.00	1
Quantocoin	1	0	0	0	80	$ 1.00	56%	0%	$ 15.00	0
Refereum	20538	1806	247.6088	0.16632331	29	$ 0.02	40%	0%	$ 32.30	1
REMME	667	359	105.8293	0.24018804	3	$ 0.04	50%	10%	$ 17.50	1
Rentberry	555	379	24.46507	0.02822404	86	$ 0.35	70%	33%	$ 30.00	1
Republic Protocol	229	49	25.22562	0.49711234	1	$ 0.06	60%	15%	$ 34.30	1
SALT	1172	0	0	0.27935576	15	$ 0.89	45%	0%	$ 48.50	1
selfKey	2152	1314	376.4808	0.32158409	31	$ 0.02	33%	30%	$ 21.80	1
SyncFab	302	198	110.8093	0.48637891	46	$ 0.12	30%	0%	$ 31.20	0
TE-FOOD	579	227	58.03591	0.29185835	29	$ 0.04	56%	36%	$ 19.10	1
The Bee Token	5706	2400	225.9538	0.10888867	3	$ 0.20	30%	30%	$ 15.00	1
THEKEYOFFICIAL	5100	850	324.5074	0.37271683	76	$ 0.35	51%	100%	$ 22.00	1
Tomochain	323	170	104.4084	0.63445852	20	$ 0.25	40%	0%	$ 8.50	1
WePower	1450	762	147.3019	0.20378827	2	$ 0.13	45%	25%	$ 40.00	1
Winding Tree	868	543	115.6313	0.22130441	15	$ 1.05	75%	30%	$ 10.00	1
Zeepin	9957	821	339.5846	0.4168393	1	$ 0.13	50%	30%	$ 42.00	1
zilliqa	5099	400	138.0056	0.42141979	3	$ 0.00	30%	0%	$ 22.00	1

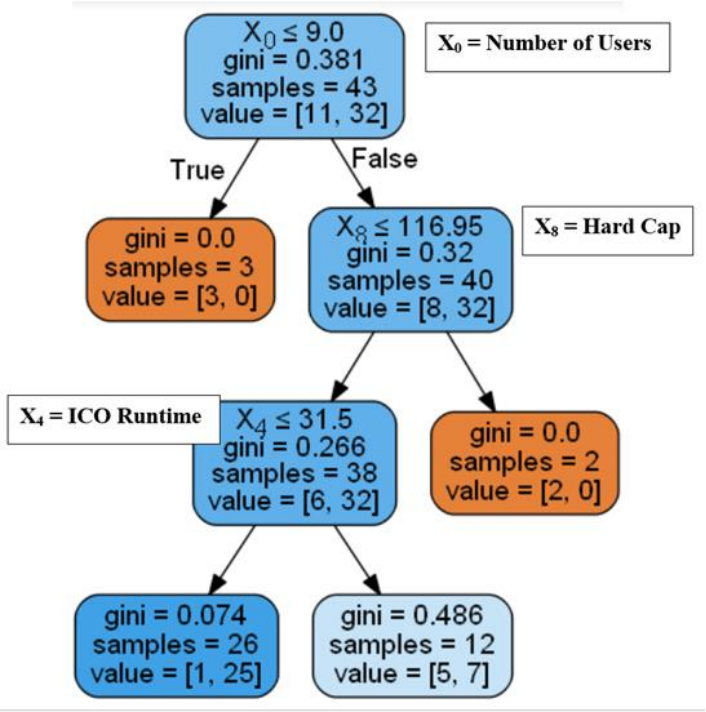
Appendix 2: Link to Feature Set and ICO List
https://docs.google.com/spreadsheets/d/1KXBK4ZCIuKTOYGCUOmQ48v9E4U0JDexQWhU4mgoDJO8/edit?usp=sharing

Appendix 3: Link to Sentiment Scrubbing Code:
https://github.com/michaelkrakovsky/CommentScrubbing/tree/master/PreProcessing_And_Analysis

Appendix 4: Link to Reddit Comments Training Set:
https://github.com/michaelkrakovsky/CommentScrubbing/tree/master/Training_Set

Appendix 5: Link to Model Creation:
https://github.com/michaelkrakovsky/CommentScrubbing/tree/master/Model_Creation

Appendix 6: Decision Tree Visualization:



WORKS CITED

Bronshtein, A. (2017, April 11). A quick introduction to k-nearest neighbours algorithm. In *Medium*. Retrieved April 19, 2018, from https://medium.com/@adi.bronshtein/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7

Raised funds and number of ICOs (2018, April). In *ICO Bench*. Retrieved April 21, 2018, from https://icobench.com/stats

Richter, Wolf. (2017, December 20). The Most Mind-Blowing Cryptocurrency ICO of All Time Is Going on Right Now. In *Business Insider*. Retrieved April 22, 2018, from www.businessinsider.com/biggest-cryptocurrency-ico-ever-going-on-right-now-2017-12?curator=thereformedbroker&utm_source=thereformedbroker

Seth, S (2018, March). What's a Cryptocurrency exit scam? How do you spot one? Retrieved April 25th, 2018 https://www.investopedia.com/tech/whats-cryptocurrency-exit-scam-how-spot-one/

Schleifer, T. (2017, September 19). Silicon Valley is obsessed with ICOs — here's why. In *Recode*. Retrieved April 22, 2018, from https://www.recode.net/2017/9/19/16243110/initial-coin-offering-ico-explained-what-is-money-bitcoin-digital-currency

Peterson, R. L. (2016). *Trading on Sentiment: The Power of Minds Over Markets*. John Wiley & Sons.

Understanding Soft Caps, Hard Caps & Emission Schedules (2018). In *Coinist*. Retrieved April 21, 2018, from https://www.coinist.io/crypto-hard-caps-soft-caps/

1.9. Naive Bayes. (n.d.). Retrieved from http://scikit-learn.org/stable/modules/naive_bayes.html