Using Twitter Sentiment Classification to Predict Hourly Changes in XRP Price

Braden Fineberg (bfine@seas.upenn.edu), Matt Oslin (moslin@seas.upenn.edu)
Sam Weintraub (sweint@seas.upenn.edu)

Summary

This project uses machine learning to predict the change in the crypto currency Ripple (XRP) based on Twitter data. Crypto currencies are highly speculative, and hence depend strongly on market sentiment. One way to capture market sentiment is by observing individual Tweets relating to the currency and combining all related Tweets to predict the future price. Data is collected and processed, and Tweet trustworthiness is evaluated to remove spam. Each good Tweet is classified based on its sentiment using a learned sentiment classifier. A count bag of words feature space is generated for each hour and classifiers are tuned to learn price change from the given features. In addition, other features such as exclamation points, ellipses, and capitalization were used to determine sentiment.

Data Collection & Pre-processing

To collect the tweets, we deployed a scrappy scraper on a Twitter search. Initially, our goal was to determine the difference between quality and poor data so we looked at the top 30 crypto currency accounts and then a series of crypto bots. Each set of data was used to construct a search that allowed us to target very specific tweets within a 12 day window from April 13-Arpil 15. After constructing our search to include the terms 'XRP or Ripple,' but excluding some terms like "ICO, airdrop, and token sale," we collected just over 87k tweets from the 12 days.

While we had a large sample size, the variability within the quality in the tweets was very high. We used a series of tokenizers, stemmers, and transformations to turn text in to features that our models could process. In addition, we normalized the times by shifting all tweets to the next closest hour.

We then paired this tweet data with XRP pricing. Finally, we classified each of our 287 hourly epochs into one of 3 block: "positive", "neutral", or "negative" depending on the change in price. Tweets with a greater change that ±0.05% were classified as "positive" or "negative" while anything in between was "neutral."

Examples

Raw Tweet:
"'# XRP '100 Percent Not a #Security ,' #Ripple Claims\n\n http:// bit.ly/2v5gC26 \xa0"

Post-Processing:
"'XRP 100 Percent Not a Security Ripple Claims _URL_"

Spam Filter

Generating Training Data:
After loading the data, URLs and Mentions were regularized using to Regex patterns to '_url_' and '@' respectively and hashtags were removed. Using a DictVectorizer, the most frequent words were identified and analyzed for links to spam. For example a top word identified was 'airdrop' which was associated with tweets like:

"stockchain (scc) final airdrop 500 scc bonus don't miss! #airdrop #btc #neo #eth #freetoken #crypto #xrp #blockchain #ripple #trx"

From this analysis we labeled any tweet containing the following words as spam:

Top Spam Words		
ICO	Freetoken	Token
Airdrop	Bigpumpgroup	Current Price
Free	Cryptobot	XRPticker

For spam classification a Naïve-Bayes filter was applied with TF-IDF tokenization. A Naïve-Bayes classifier generates predictions according to:

$$\hat{y} = \max_{k \in 0,1} p(y_k) \prod_{i=1}^{n} p(x_i|y_k)$$

We only used data from our Top30 sources as positive examples and spam as negative. Despite, training and testing accuracies of 99.5 and 98% respectively, the probability distribution showed that there was extreme overfitting due to similar tweets from the bots and that future data would be hard to filter.

Measuring Trustworthiness:
An expanded feature set was applied to measure trustworthiness. This feature set included the number of favorites, replies, mentions, hashtags, exclamations, questions, words, characters, and tokens as well as binary features representing the presence of media, replies, or urls.

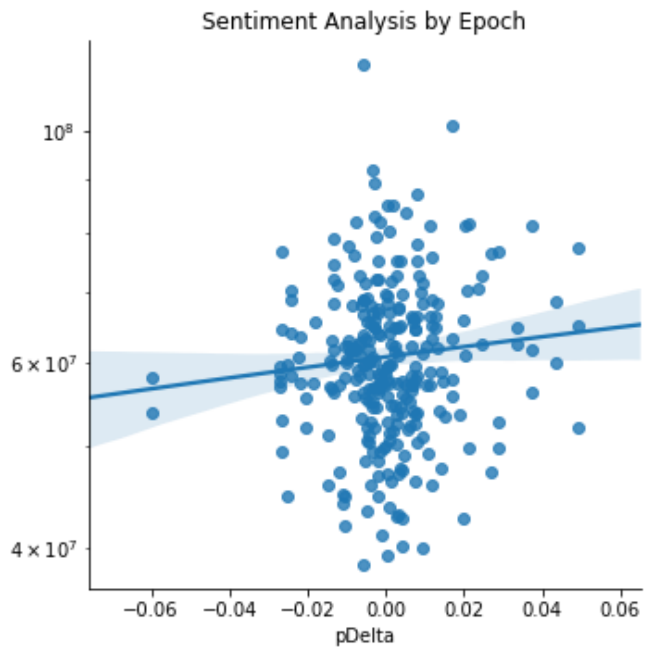
A Bernouli Naïve Bayes classifier was applied to the expanded set as well as a Random Forrest Classifier with maximum depth of 7. The expanded feature set showed significant improvement in the bimodality of the probability distribution as seen in the attached graphs.

Sentiment Analysis

To compute sentiment analysis, several techniques were deployed including traditional tokenization and sentiment methods using NLTK, as well as bag of words and variable weighting trained during reinforced learning. Initially pre-process was completed using a tokenizer, stemmer, and a lower-caser. After the initial pre-processing, a perceptron was used as a way to update the various word weight vectors.

```
If word_sentiment == wordDict_sentiment:
        Tweet Sentiment = Tweet_sentiment + wordDict[word]
Else:
        wordDict[word] = wordDict[word] + weightUpdate[word]
```

Classify each tweet based on the words in the tweet then average the tweet sentiment within each 281 hour epochs.

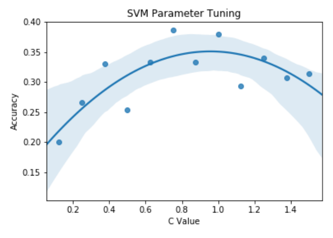


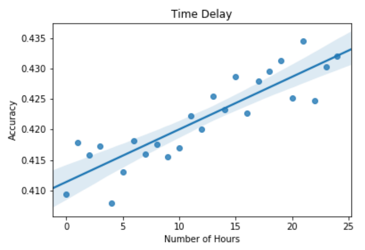
Models

A variety of models were tuned on a count bag of words feature space. This baseline feature space has the number of tweets a stemmed word or bigram appears in each hour. Two hour windows were also explored.

Model Accuracies			
Classifier	1 Hour	2 Hours	Equation
SVM	50.88%	51.72%	$\min \|\|w\|\|^2 + C \sum \max[0,1 - y_i(w^T x_i + b)]$
Logistic Regression	52.63%	50.26%	$\max \prod \sigma(x_i)^{y_i} (1 - \sigma(x_i))^{1-y_i}, \sigma(x) = \frac{1}{1 + e^{-w^T x}}$
Naïve Bayes	47.37%	48.30%	$\hat{y} = \max_{k \in 0,1,2} p(y_k) \prod p(x_i\|y_k)$
Neural Net	47.39%	48.28%	Two hidden fully collected layers, followed by Relu

Temporal relations in the dataset were also considered. One surprising result shows that delaying the hour in which price change is measured from when the tweet data is collected can actually improve classifier accuracy. This indicates that the market may have a delay in price response to twitter activity.





Results

Initial classifier results are promising, even with only a simple count bag of words feature space. Filtering out misleading tweets and adding sentiment based features could build on this foundation to greatly improve the accuracy of the tuned classifiers.

SVM and Logistic Regression performed the best, as they are both very similar linear classifiers and can be expected to behave similarly. The Naïve Bayes classifier did not perform as well as it relied more on the prior distribution of the y values than the features that differed between them, biasing its predictions. The four layer neural net faced a similar struggle in differentiating its outputs, but this was likely due to the low number of labeled examples, 281.

While a sentiment analysis appears to show some correlation with the market change, we don't believe there is enough data to truly compute what will happen; therefore, we believe that the sentiment analysis will best be used as a feature for a SVM or Logistic Regression

Future Work

Spam Filter Improvements:
One of the underlying assumptions with a Naïve is Bayes classifier is:

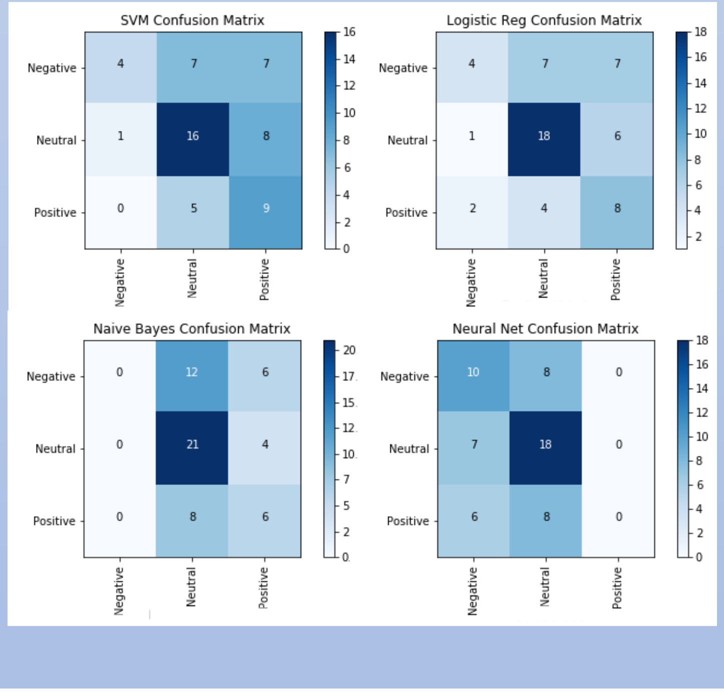
$$P(A \cap B|Y) = P(A|Y) * P(B|Y)$$

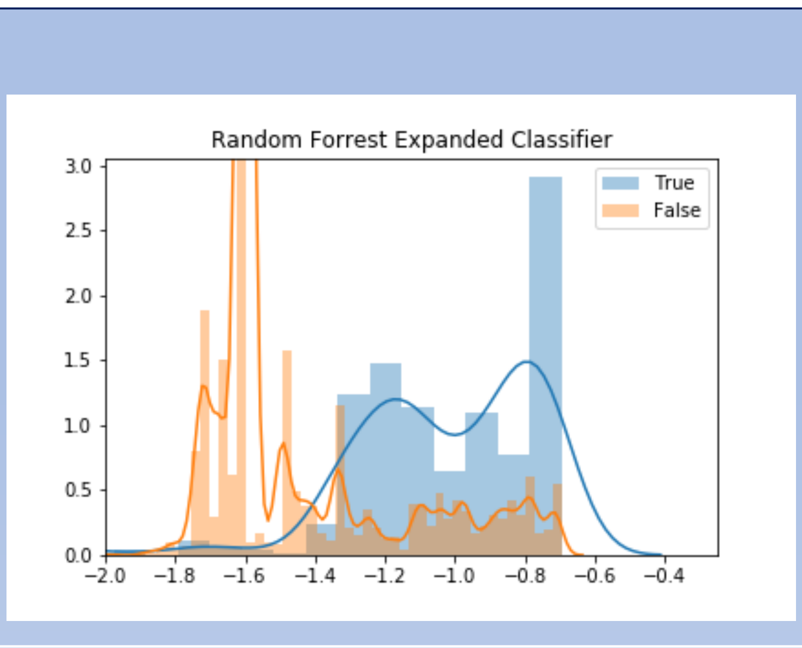
Meaning that A is conditionally independent of B given Y. For our Spam Classifier, this may be a particularly bad assumption because the relationship between words i.e. bigrams and trigrams may be important. For example a 'good' tweet may contain the phrase 'With this momentum I believe the price will hit $10,000' and a 'spam' tweet may contain 'Price Update: XRP = $0.90'. Both tweets contain the word price, but the important distinction for the Spam tweet is the bigram of 'price update'. These relationships need to be considered to improve the Spam Filter.

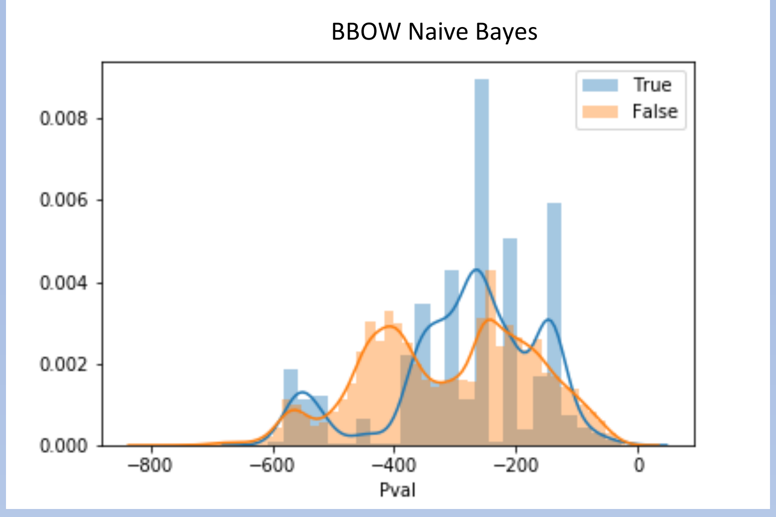
Another method to supplement the Spam filter is to evaluate the properties of each user that tweets come from. Studies have shown that lexical diversity is a good indicator to distinguish bots from humans. This approach requires collecting all of the tweets for a given user and then associating a LD score with their user id. We could not implement this because we did not have enough tweets to limit to only repeat tweeters. In the future we would like to implement this.

To improve accuracy, we hope to implement the Spam Classifier with the Expanded Feature set as discussed in Spam Filtering.

Sentiment Analysis:
The largest drawback to the sentiment analysis was the lack of quality data. Unfortunately, a tweet of 140 characters isn't enough to understand a market epoch. Instead, we hope to aggregate thousands of tweets for each epoch to create a word list long enough to capture sentiment. In addition, we would need a training set that includes slang frequently used on twitter as well as a finance dictionary.







Randomly Sampled Tweets from Various Data Sets	
Data Set	**Tweet**
Trusted Sources	I think he's arguing from a value or commonality standpoint. Definitely doesn't make you stupid, but we all know there's no real connection
Trusted Sources	Good overview and perspective on #blockchain + #bitcoin from @balajis via @WSJ . #tech #innovation #digital #economy _url_
Spam Tweets	1 ripple = 0.6589 usd. ripple has changed by -0.0026 usd in 30 mins. live price: _url_
Spam Tweets	ripple price alert. the last ask price for $ xrp in usd is $0.857858 xrp ripplebot_cs