# 

Neural Networks & Natural Language Processing for an Altcoin Trading Strategy

Ignace Loomans

Relevant Web Scraping Files & More Can be found at:

<https://github.com/ILoomans/Alt-Coin-Applied-Project>

Word Count: 2515

# 

[**Introduction**](#_mbjlus93fhie) **3**

[**Motivation**](#_jfz9l4x9an0n) **4**

[**Data Aggregation**](#_gw1rhjksf8cj) **5**

[Sources](#_97ce9qrjtsjc) 5

[Web Scraping BeautifulSoup and Selenium](#_9fsne7a1hj2) 5

[Article Selection](#_8xkxcyf5njr0) 5

[Price Selection](#_9fti6eytkhmo) 6

[**PreProcessing**](#_bk075w9psf1b) **6**

[Stemming](#_c3m2hxaqq211) 6

[Stop Words](#_2bvy4lwbvzx6) 6

[Countvectorizer](#_p8hkmtwxwvwc) 7

[**Neural Network**](#_78g7pb9q2wzr) **7**

[Classification](#_6xr056ockp32) 7

[Train Test Split](#_264dpdb3723l) 7

[Cross Validation](#_wgu3wtifbtf1) 7

[Training the model](#_21symy7igyku) 8

[**Results**](#_1fjbn6a6uppj) **8**

[The Trading Strategy](#_1y2p2upcv3cs) 8

[The Neural Network](#_t78rmqkip3rr) 9

[Logistic Regression](#_fciup6wl4mi3) 9

[**Conclusion**](#_vb5zvdn6r5e1) **9**

[**Bibliography**](#_hrvw2eyj61e3) **10**

# 

# Introduction

This project will look at formulating a trading strategy using altcoins based solely on the text data produced by the world's leading cryptocurrency and blockchain news sites. Altcoins are commonly defined as ‘alternatives to bitcoin’, however, given the rising prominence of ethereum in becoming a major cryptocurrency, we will also exclude this from the list of altcoins (Frankenfield, 2020). We will fetch as many articles as possible from top sites, and filter it out to the articles referring specifically to a given altcoin. From this article, we will then divide whether it produces a buy, hold, or sell signal.

The altcoin market is of strong interest as it is a novel and highly volatile market which is heavily impacted by social media, news, and prominent figures. This has most recently been seen with the dramatic rise of Dogecoin following activity on the popular social media platform TikTok and posts made by Elon Musk. The price of DogeCoin was driven up by a popular trend where Tik Tok users attempted to push the price of DogeCoin from $0.042 to $1 dollar with hashtags such as ‘#dogecointiktokchallenge’ (Suberg, 2020).

With investors seemingly being heavily influenced by non-structured data, as cryptocurrencies, unlike equities and debt, do not have an underlying productive asset. It is for this reason that applying natural language processing techniques to this market should prove to render interesting and hopefully successful results. The aim is then to see if there is the possibility to generate strong positive returns solely based on this text data.

We will select the altcoins on which we will make trades based on two factors. First, after aggregating all articles we will filter articles relating to the top 100 cryptocurrencies by market size according to coinmarketcap. From this list, we will then look at articles referring solely to this cryptocurrency. Following this, for the remaining cryptocurrencies, we will use these cryptocurrencies for which we can find reliable price data. It is in this subsection of cryptocurrencies and their corresponding articles that we will develop our trading strategy.

These articles will be processed through standardized methods of text pre-processing. Consequently this data and feed it into a neural network classifier whose hyperparameters we will tweak and train the model. From the buy/sell/hold recommendations made by our neural network, we will hopefully generate a profitable trading strategy.

# 

# Motivation

One strong motivation for the project is to evaluate the efficient market hypothesis of altcoins. The efficient market hypothesis suggests that there are three forms of market efficiency, strong, semi-strong, and weak form. Specifically, the semi-strong form of market efficiency suggests that the price of the asset only reflects public information. Given this, if we are able to generate an excess return based solely on text data, we would surely be able to make a case for altcoins market efficiency to be in a semi-strong form. This is due to our ability to make returns off article data not yet interpreted and priced in by the investor community (Maverick, 2020).

The inability to generate a return from this strategy may lead us to believe that the altcoin market is in a weak form, as only previous historic price data is priced into the current price of the asset. Exploring this topic will be of use as our ability to evaluate the altcoin market more holistically will provide valuable insights into the state of this novel market.

Lastly, another strong motivation is to have a functional trading algorithm which on its own can generate positive returns. Specifically, a trading strategy that is able to make strong predictions in one of the more novel and volatile markets that is continuously evolving. It is because of the volatile nature of this market that makes it probable that we will be able to generate positive returns by having an algorithm that is able to process news articles and make trades before traditional investors catch on to the information.

# 

# Data Aggregation

## Sources

The sourcing of crypto news sites was taken from news.crypterium, which included a list of the ‘top 15 crypto news sites’. From these websites, a smaller subsection was collected based on additional criteria. Namely, when selecting altcoins news sites, the website must provide a minimum of information for it to be useful for our model. The minimum data requirements we needed include having an author name, an article title, relevant text data, and a time posted with the minute specified. Gaining accurate time data is of the essence to our model as we need insight in when our trading strategy would buy the altcoin. If a site included articles that met these standards, it became an eligible source of data for training and testing the model.

## Web Scraping BeautifulSoup and Selenium

In order to get all relevant information from these crypto news sites, a mixture of BeautifulSoup and Selenium was used. Beautifulsoup is a useful tool to parse Html data, to find the specific elements that are needed to feed into our model. Selenium was used when the data we needed was not navigable via a web link but instead required recursive actions such as pressing the ‘load more’ button until there were no more articles to be rendered. All of the web scraping files can be found on the following Github page (<https://github.com/ILoomans/Alt-Coin-Applied-Project>).

## Article Selection

From the subsection of all aggregated articles, an additional filter had to be made, namely, the articles should reference one and only one altcoin in the title. This would give a reasonable degree of certainty that the article was about a given altcoin and that we could train the model on the text data of that article and the returns produced by the relevant altcoin. Practically, this was done by looking if the title contained one and only one altcoin name or one and only one altcoin symbol.

## Price Selection

The aforementioned article subselection gave us a list of altcoins that were of interest. We then evaluated whether our data source could provide the data of the prices at the necessary time. The price data was sourced from CryptoCompare, from which we fetched all the price data of the altcoins of interest for which they had the price data available. The most up to date price data we could select for the necessary time period was in an hourly format, so when associating an article time to the price of the relevant altcoin we rounded to the closest hour.

# PreProcessing

In order to be able to interpret and feed the text data which had been collected, we had to effectively create tokens of the word included in the text. Through this methodology, we will create a taxonomy of all the words which are included in the text. The process of tokenization is quite simple as it would translate a sentence such as ‘imperial college is in london’, to the tokens ‘imperial’, ‘college’, ‘is’,’in’ and ‘london’. Following this we will make some adjustments to these tokens to ‘standardize’ these tokens more appropriately (Stanford, 2008).

## Stemming

In order to effectively create tokens to be used by our neural network, we have to extract the root of each word. As an example, create and creating would be reduced to ‘crea’. The words create and creating have the exact same meaning, therefore when creating tokens of words we would want these grouped together. Because of this, it is of utmost importance to include this process in our pre-processing of text before feeding this data to our machine learning model.

## Stop Words

There are also words which we can estimate have no tangible meaning or value in making a prediction of whether the price of the cryptocurrency will increase or decrease. These are words that are commonly referred to as stop words in natural language processing. This includes words such as ‘and’, ‘that’, ‘a’, ‘this’ etc. This will also allow our model to place more importance on words with actual meaning rather than the insignificant ones. For the purpose of this project, we will use the stopwords from the python nltk.corpus package (Ponugoti, 2020) .

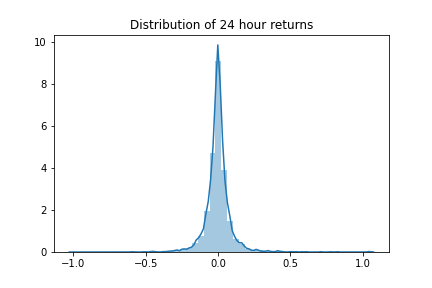
## Countvectorizer

A count vectorizer is a unique and easy to understand representation of text data. It looks at the number of times a word appears in a document. This translates the number of times that a word appears in a sentence into a dictionary of the word and the number of times that appeared in the document. For the purpose of training our data, we use the sklearn python package to complete the count vectorization of the various documents collected (2018, Heidenreich).

# Neural Network

## Classification

In order to give our model an outcome to predict, we had to assign the different returns with a different classification. This was done by collecting all 24 hour returns of altcoins after an article specifically referencing them came out. From this, we created a distribution of returns as seen in the plot below. From these datasets we then took the 20th and 80th quartile of returns which corresponded to -4.33% return and 4.01%. Anything below -4.33% was classified as a sell recommendation, anything above 4.01% was a buy recommendation and everything in between became a hold recommendation.



## Train Test Split

For time-series data, it is important that we train our model on past data, and then complete the tests on future data. Although in the case of our model, historic price data is not incorporated and we are only focusing on classifying text data, we still chose to train test split using this method. For this purpose the train test split was done manually on the time series ordered list of data. We split the data with 70% for the training data set and 30% for the validation data set.

## Cross-Validation

For the purpose of tuning the model to produce the best out of sample accuracy, the model uses cross-validation to tune and select which amount of iterations we use, the hidden layer size, the random state of the model, and finally also the learning rate of the model. Specifically, to complete this cross-validation, we used the GridSearchCV function provided by sklearn. Through this, we are aiming to optimize the predictive ability of the model in making buy/sell recommendations.

## Training the model

The neural network which is used to make these predictions is the multilayer perceptron. The Multi-Layer perceptron provides a standard but highly effective neural network. This model provides a simple and intuitive representation of how our brain makes connections between different events. These models have been highly effective in predicting ranging from financial predictions and speech recognition etc.

The model consists of multiple layers that contain ‘neurons’ within them. The data provided as an input is pushed through these layers, connected to all neurons which have an activation function and weight, and then these values are passed along to the next layer of neurons. The first layer is a ‘visible layer’, because this is where we provide the inputs for the model. No computations are done in this layer as this is merely where the data is introduced into the model and passes this data in the next hidden layers where the actual computations are being completed.

Namely, they are passed through to the hidden layers of the model, where the computations of the model are being produced. The hidden layer contains neurons that produce computations based on inputs, weights, and an activation function to provide an ultimate output that will be provided to the next hidden layer or to the final output layer. The weight given to a neuron gives the model a certain bias, as some neurons will be unimportant and unuseful in making an accurate final prediction, as a result of this that neuron would contain a small weight. The activation function takes the weights of all inputs provided by all neurons of the previous layer and then decides what the output value and what weight this neuron will have and passes on to the next layer.

The final layer is a ‘visible layer’ which simply points out the classification that the model predicts. Which is the case of the altcoin prediction model is 0,1 and 2 for a sell, hold, and buy recommendation (Brownlee, 2016).

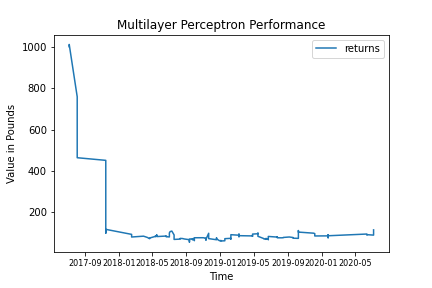
# Results

## The Trading Strategy

For the purpose of our trading strategy, we will simply follow the recommendation made by our model on the test set of data. The trading strategy follows the recommendation made by the model if the money isn't locked into another trade. We then invest all of the available funds in the recommended trade and sell it after 24 hours. By then we calculate the returns made over this time period and are able to make further trades when the short or buy action on a given altcoin has been completed.

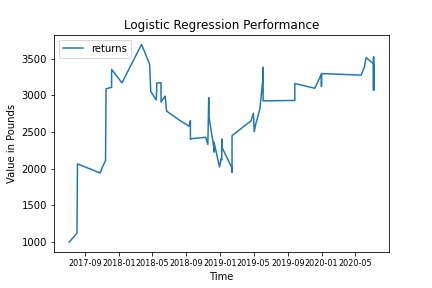
## The Neural Network

The neural network clearly overfits the training set data, resulting in poor data in the consequential trades made on the validation set of our data. As seen in the plot below, there are trades being made which are resulting in heavy losses, meaning that the classification errors on the test set are highly expensive. In this case, the initial balance was £1000 and by the end, all that remained was £115 pounds. As a result, it could be deemed that a neural network with a significant amount of hidden layers is not desirable for the model. Regardless of the fact, the model was most optimally trained using GridSearchCV which deemed two hidden layers to generate the best predictive ability. As a result, a more desirable algorithm to complete this analysis may be a logistic regression model which includes a ridge or a lasso regression.



## Logistic Regression

The logistic regression model proves to be a significantly more useful model than that of the neural network as it is more able to place a penalty on redundant variables. Through the use of cross-validation of hyperparameters, we deemed that the ridge regression was the most appropriate penalty to place on the model. As seen in the plot below, it makes trades with very strong returns, although highly volatile. However, with the results growing from £1000 to over £3000 in a period than less than three years, this degree of risk seems acceptable.



# Conclusion

There is a strong possibility to make excess returns via altcoin trades based on articles corresponding to an altcoin. With a properly trained model that does not overfit the training set data, the model can analyze the text data and make trades before the general investing community can react to the information. The use of a neural network did not prove to be useful in completing this, as it overfits the training set data. However, penalized regression such as the logistic regression model proves to generate strong returns as it penalizes the introduction of additional parameters. Through this, the logistic regression model is more able to select key parameters to make predictions on whether an article about an altcoin will make the price of that asset increase, decrease, or stay the same.

# 

# Bibliography

Brownlee, J., 2020. Crash Course On Multi-Layer Perceptron Neural Networks. *Machine Learning Mastery*. Available at: https://machinelearningmastery.com/neural-networks-crash-course/ [Accessed August 13, 2020].

Frankenfield, J., 2020. Altcoin. *Investopedia*. Available at: https://www.investopedia.com/terms/a/altcoin.asp [Accessed August 12, 2020].

Heidenreich, H., 2018. Natural Language Processing: Count Vectorization with scikit-learn. *Towards Data Science*. Available at: https://towardsdatascience.com/natural-language-processing-count-vectorization-with-scikit-learn-e7804269bb5e [Accessed August 12, 2020].

Maverick, J.B., 2020. The Weak, Strong, and Semi-Strong Efficient Market Hypotheses. *Investopedia*. Available at: https://www.investopedia.com/ask/answers/032615/what-are-differences-between-weak-strong-and-semistrong-versions-efficient-market-hypothesis.asp [Accessed August 12, 2020].

Ponugoti, S.T., 2020. *Stop Words in NLP*, Towards Data Science.

Stanford, 2008. *Tokenization*. Available at: https://nlp.stanford.edu/IR-book/html/htmledition/tokenization-1.html [Accessed August 13, 2020].

Suberg, W., 2020. TikTok Sends Dogecoin Price to 2-Year High as Twitter Says 'Be Smart'. *Cointelegraph*. Available at: https://cointelegraph.com/news/tiktok-sends-dogecoin-price-to-2-year-high-as-twitter-says-be-smart [Accessed August 12, 2020].