Blockchain ICO Forcasting by Using Machine Learning

# Introduction

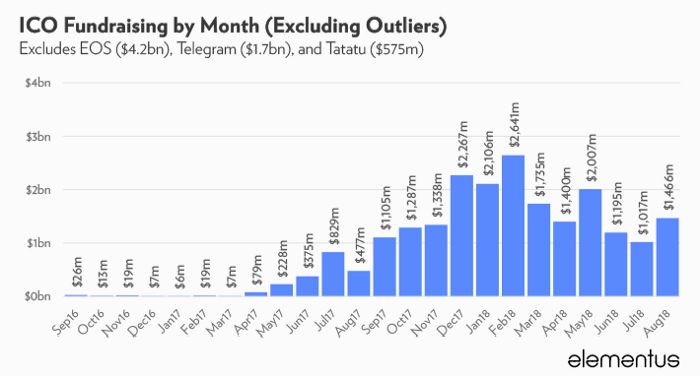
Cryptocurrencies need no introduction. What was once reserved to the nerd communities on online chat forums, Cryptocurrencies have now almost gone mainstream. On the other hand, ICOs are less known.

Blockchain A decentralized network, built from a continuous chain of code segments of predetermined size (blocks). All transactions on the network are stored on a public ledger, which exists throughout the network, meaning there is no need for a central server to authorize transactions on the network.

An Initial Coin Offering, commonly referred to as an ICO or a token sale, is a means of fundraising, where [tokens in a newly issued cryptocurrency are exchanged to the public](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3080098) for other cryptocurrencies such as Bitcoin or Ethereum.

These are intended to be used to fund early stage cryptocurrency projects, however they are quite risky investments as more than [10% of money raised in ICOs have been lost or hacked](https://www.ey.com/Publication/vwLUAssets/ey-research-initial-coin-offerings-icos/%24File/ey-research-initial-coin-offerings-icos.pdf) and only [10% of ICO tokens issued are used post ICO](https://www.bloomberg.com/news/articles/2017-10-23/only-one-in-10-tokens-is-in-use-following-initial-coin-offerings) in the actual token ecosystems, while the rest are simply traded.

Until recently, there was very little regulation in ICOs and the market was flourishing, with ICO funding growing rapidly. This was especially the case in late 2017 and early 2018, as can be seen below. This growth has since slowed down rapidly, as the whole cryptocurrency market has become quite bearish.



Source: [Elementus.io](https://elementus.io/blog/ico-market-august-2018/)

The unpredictability of ICOs and cryptocurrencies in general, make for an interesting and difficult problem — can we model these behaviours and even predict them?

More specifically:

***Can we use Machine Learning to predict the success/price of ICOs?***

Let’s find out.

# Methodology

1. Choose inputs and outputs.
2. Collect and aggregate the data.
3. Prepare the data.
4. Explore and attempt to understand the data.
5. Choose a Machine Learning Model.
6. Measure the performance of the Model.
7. Save the Model.
8. Use the Model to make predictions.

**1. Choosing Inputs and Outputs**

**Inputs**

Choosing the right inputs and outputs (in the case of supervised ML) are critical for the success of Machine Learning algorithms. There needs to be some sort of correlation between the inputs and what you trying to predict-output. The best way to do this is to try and gain some domain knowledge. I did some research into various aspects of ICOs as to gauge what might affect their future value.

The main proposed factors of an ICO affecting their performance were the following:

* **The quality of the development team —***better team, better quality of ICO product.*
* **Popularity on Social Media such as the number of followers on Twitter —***more publicity an ICO has, the better chance of it gaining funding.*
* **Factors relating to an ICO specifically-***price, total supply, market cap…*
* **Factors relating to the market at the period of ICO launch —***was the cryptocurrency market doing well at the time when an ICO was launched?*

I then created a list of features that I thought might affect an ICO’s value and checked if I could collect this data from any sources and created a final list of Inputs.

1. Month released
2. Day of the week released
3. Type of ICO industry - finance, retail, gambling, entertainment etc
4. Type of token - usage, equity, currency, community token.
5. Quality of white paper
6. Individual cap
7. Team - team bios, age, degree *IcoRating*
8. POC
9. Social Media - slack, telegram, twitter, reddit
10. Github - does it exist.
11. Country (*Exploring Signals for Investing in an Initial Coin Offering (ICO))*
12. Duration of company existence before ICO (*Exploring Signals for Investing in an Initial Coin Offering (ICO))*
13. Price\_BTC *(ICO Stats)*
14. Price\_USD     *(ICO Stats)*
15. Market\_cap\_usd *(ICO Stats)*
16. Total\_supply *(ICO Stats)*
17. Eth\_price\_at\_launch *(ICO Stats)*
18. Btc\_price\_at\_launch *(ICO Stats)*
19. Token distribution *(*[*https://www.benzinga.com/general/education/17/10/10187104/the-definitive-ico-checklist-what-to-look-for-and-what-to-avoid*](https://www.benzinga.com/general/education/17/10/10187104/the-definitive-ico-checklist-what-to-look-for-and-what-to-avoid) *&& Exploring Signals for Investing in an Initial Coin Offering )*
20. Google trends *(Cryptoeconomics: Data Application for Token Sales Analysis)*
21. *Twitter influencers*

Unfortunately, it can be difficult to collect certain data numerically — for example, measuring the quality of the development team that has worked on an ICO — do you measure years of experience? Is that a valid measure of quality?

Furthermore, other metrics such as popularity of ICO when it was launched is quite difficult to find data on. For example, it is difficult to get historical Twitter data, even using the [Wayback Machine](https://archive.org/web/), which doesn’t index every page, which makes getting the number of Twitter followers when an ICO launched difficult.

**Final List of Features Used:**

* **Price in USD.**
* **Price in BTC.**
* **Total Supply.**
* **Market Cap — the number of tokens that can be purchased during token sale.**
* **Available Supply.**
* **USD raised.**
* **Ethereum price at launch.**
* **Bitcoin price at launch.**
* **Month ICO was launched.**
* **Date ICO was launched — for example the 8th day of the month.**
* **Country an ICO was launched from.**
* **ICO Duration in Days.**

# Outputs

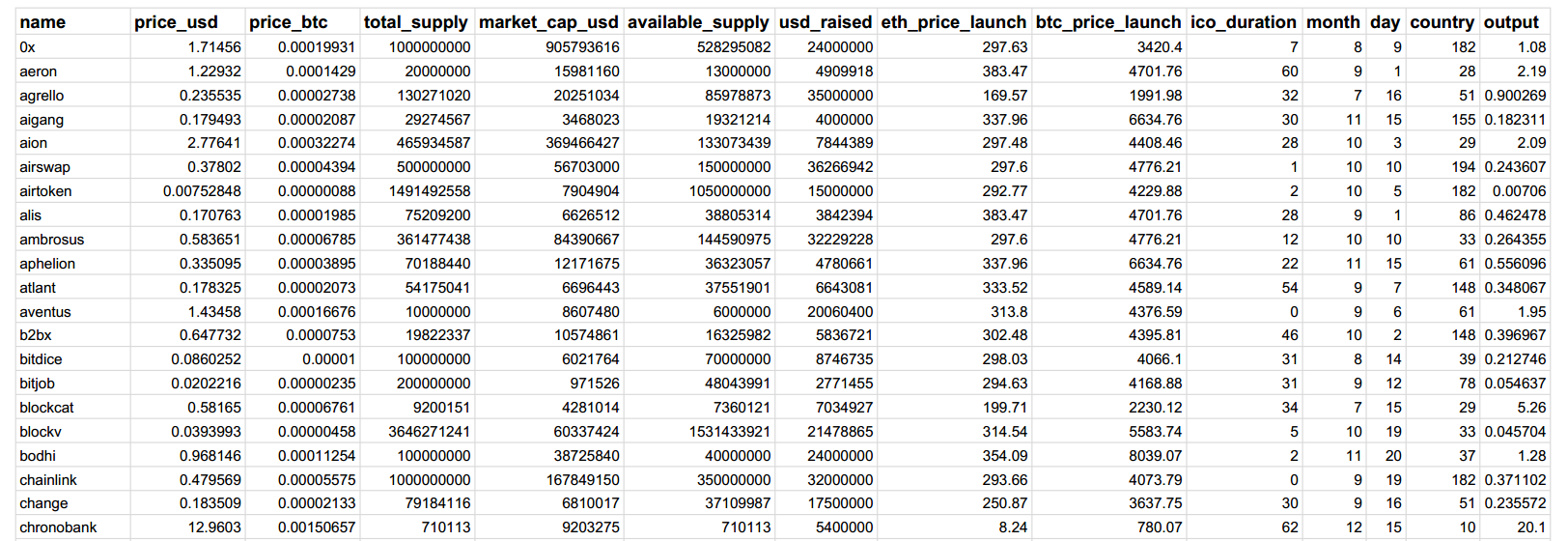
ICOs are still generally a new means of fundraising so there aren’t many ICOs available to collect data from. A six-month period was chosen, as it is sufficient to observe a potential growth in the price of an ICO, and there were enough ICOs older than six months from the available data.

**inal Output:**

**Price of an ICO after six months.**

# ****Examples of some of the data Collected on ICOs****

Full dataset available here.



# 2. Collecting and aggregating the data

A large part of this project was data collection. There was no available dataset on ICOs and so a dataset had to be created.

A restful API was used, written in C#, following the dotnet core framework. This API brings together data from various sites and stores it in a Mongo database.

Some of the data, such as country an ICO was launched from, had to be collected manually as this was not consistently available from a specific source and so multiple websites had to be used. Data was collected on **2 885 ICOs**.

# 3. Preparing the data

Once the data was collected and stored in a Mongo Database, the data had to be converted into a useable dataset. All the data collected from various sources were aggregated using database views.

As mentioned, data was collected on 2 885 ICOs, but this data was sparse and hence data preparation was required.

Once the data was cleaned, we had information on 189 ICOs. Finally, this was further filtered to only include data on ICOs that have existed for longer than six months and we ended up with 109 ICOs in our dataset.

**Problems that were encountered while preparing the data:**

* **Unclean Data**

Invalid values for Start and End Dates of ICOs, ICO duration and ICO month were encountered, these were cleaned by manually correcting these fields.

* **Missing Data**

A usual approach to handle missing data is often to use either the mean or median. However due to the sparsity of the data (less than 10% of the data had most of the required fields), this approach was not followed.

Rather, data with most of the required fields were chosen and then the missing values, such as country ICO was launched in, were manually collected from various websites.

* **Conflicting data**

Some of the data was conflicting, for example websites had different start and end dates for an ICO or Country it was launched in. This was handled by checking multiple sources and taking the values that was in consensus.

* **Categorical data**

One hot encoding was applied to the categorical data fields — ICO  
Date, ICO month launched and ICO country.

# The workflow for the Data collection and Preparation phases



# 4. Exploring and attempting to understand the data

We can explore the relationships between the inputs and the outputs by calculating their correlation coefficients and drawing scatter plots to visualise these relationships.

The Price of an ICO in USD/BTC when launched was closely correlated with the future price of an ICO in six months (obviously!), while other inputs were not closely correlated to the output.

Furthermore, it was critical to ensure that the distribution of the data of an ICOs was representative of the current market in order to have a model that generalises well. This was achieved and the data collected correlates to statistics related to ICO distributions by [country](https://medium.com/@kerya/ico-statistics-countries-traffic-and-investors-79d934591b4b).

Distribution of ICO Data Per Country in ICO.

**5. Choosing a Machine Learning Model**

**Ridge Regression**

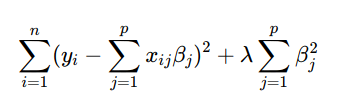
Once one-hot-encoding had been applied to the data, the input matrix formed what is known as an undetermined or a fat matrix. This essentially means that there are more features than examples (128 features versus 109 data points).

This means that our regression model is susceptible to overfitting and multicollinearity.

*“Multicollinearity is a state of very high intercorrelations or inter-associations among the independent variables. It is therefore a type of disturbance in the data, and if present in the data the statistical inferences made about the data may not be reliable.” —*[*Statistics Solutions*](https://www.statisticssolutions.com/multicollinearity/)

To avoid this, we apply regularisation. In our case we apply Ridge Regression (L2 regularisation), which penalises very large weights. You can read more about Ridge Regression [here](https://onlinecourses.science.psu.edu/stat857/node/155/).

**Ridge Regression Formula:**



Source : [Penn State](https://onlinecourses.science.psu.edu/stat857/node/155/)

**Neural Network**

We also used a Neural Network to compare the results with those achieved by the Regression model.

For the Neural Network, we used the tanh activation function, and for solvers we used the Adam Solver and also the Gradient Descent Solver.

# 6. Measuring the performance of the Model

Two measures of performance were used —**RSquared(R²)** and **Root Mean Squared Error**(**rMSE).**

**R²**, measures the [“percentage of the variance](https://www.r-bloggers.com/assessing-the-accuracy-of-our-models-r-squared-adjusted-r-squared-rmse-mae-aic/)” that the model can explain. A high R² score is usually good, but not always, as it could mean your model is overfitting to the data. You can read more about R² [here](http://blog.minitab.com/blog/adventures-in-statistics-2/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit).

**rMSE,** measures the root of the average squared error our model achieves (error — difference between the values our model predicted and the actual values).

**Results**

All the results were computed using the holdout method — performance is measured on test data (i.e. data the model has never seen before).

# 7. Saving the Model

Once we have a model that we are happy with, we should save the model so that we can re-use it to make predictions later.

I used [Joblib](https://joblib.readthedocs.io/en/latest/), which is a python helper class that is part of the SciPy package, that provides utilities for pipelining operations. It makes saving and loading models simple.

# 8. Using the Model to make predictions

Once you have a saved model, you can load this model and make predictions without having to retrain your model.

# ****Conclusion and Final Thoughts****

Machine learning in the real world is largely dependent on your data. You often spend more time cleaning, preparing and aggregating the data than you do working with your model.

In this article, we managed to create a model, which can predict the price of ICO’s reasonably well. We also showed general steps one could follow to apply Machine Learning to real world problems.