Crypt Predict
Final Project Report

BSc (Hons) in Software Development

Name: Adam Eaton

Student ID: C00179859

Year: 4th Year

Supervisor: Lei Shi

Due-Date: 18 - 04 - 2018

Table of Contents

1. Abstract

The purpose of this document is to provide an overview of the project. Outlining areas including but not limited to the learning outcomes of the project, what was achieved throughout as well as the challenges encountered throughout development. This document aims to objectively reflect the final version of the project completed within the project timeframe.

2. Introduction

This document aims to reflect the final version of the application that was developed during the project timeline which ran between October 2017 to late April 2018. It will hopefully provide the reader an understanding of the development involved in this project.

This document will be broken down into three core sections. The first acting as a detailed description and overview of the finished project. The second being a description of the learning outcomes found through the completion of the project, both on a personal level and a technical level. The third and final core section of the document will outline a review of the development process including what I found worked out, or what went right, additionally what I found challenging throughout development and finally what I would do differently given the opportunity to attempt the same project.

3. Project Description

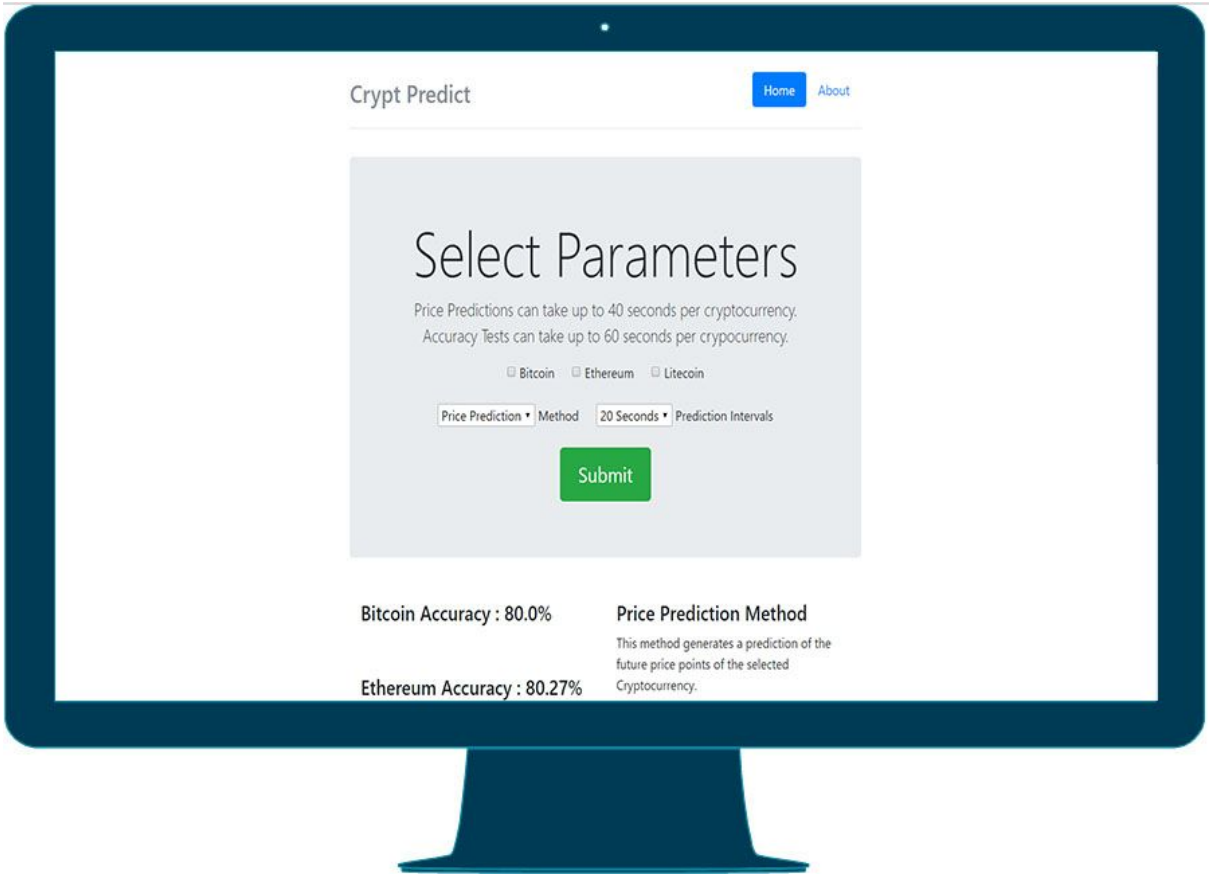
The overarching goal of the Crypt Predict application was to assist existing or potential cryptocurrency owners and traders in making better educated decisions in terms of trading.

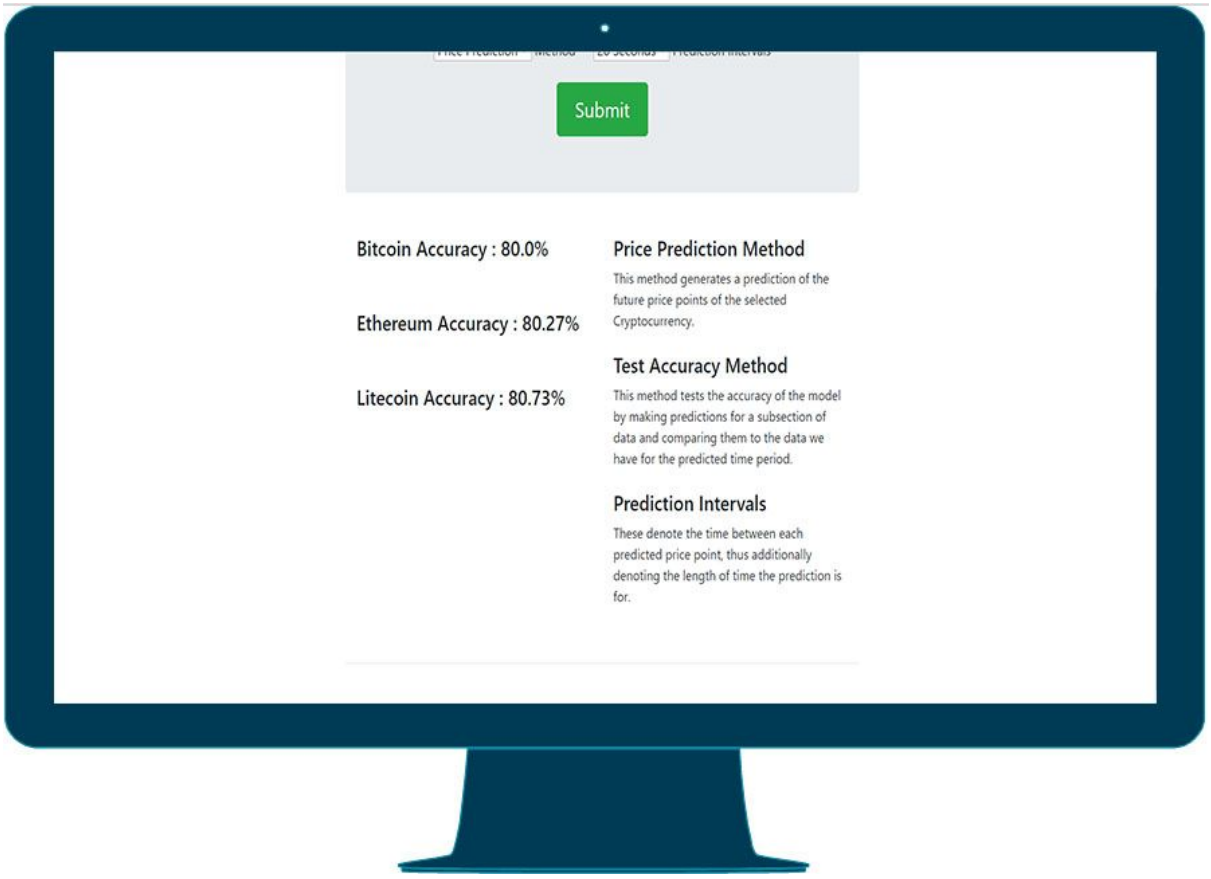
Crypt Predict is implemented as a Web Application to allow for ease of use when either running it locally or hosting the application online. The application itself can be broken down into several key components. Below I will describe each of these components along with screenshots where applicable.

3.1. Main Screen

The main screen of the application is created largely using a template from Bootstrap with the addition of the parameters form, the accuracy scores and the information regarding the different parameters being added to the template.

Below are two screenshots displaying how the design and the content of the main screen.




Submit
Bitcoin Accuracy : 80.0%
Ethereum Accuracy : 80.27%
Litecoin Accuracy : 80.73%
Price Prediction Method
This method generates a prediction of the future price points of the selected Cryptocurrency.
Test Accuracy Method
This method tests the accuracy of the model by making predictions for a subsection of data and comparing them to the data we have for the predicted time period.
Prediction Intervals
These denote the time between each predicted price point, thus additionally denoting the length of time the prediction is for.

3.2. Results Screen

The main screen of the application is effectively the same as the main screen but with the contents of the page aside from the navigation bar removed. Once the user has submitted a request to the application and the process has completed the application will navigate to the results page which will contain the graph(s) for each of the selected Cryptocurrencies.

The following screenshots display the results page for both the Price Prediction method as well as the Test Accuracy method, for the purpose of these screenshots I have used a single Cryptocurrency for each method. The first screenshot displays the generated page when running the Price Prediction method, with the second being of the Test Accuracy method

.


Crypt Predict
Home
About
Litecoin
Predicted
Actual
129
128.8
128.6
128.4
128.2
128
Export to plot.ly »

3.3. Predictive Algorithm

The algorithm utilised in the project is heavily inspired by the ideas provided in the paper 'Bayesian Regression and Bitcoin' published by Devavrat Shah and Kang Zhang.[1] The algorithm itself is a combination of a Bayesian Linear Regression and an Ordinary Least Squared algorithm.

The first step involved is taking the dataset passed to the function and splitting it into three equal arrays, each containing 9000 entries. Following this three sets of dataframes are created, each containing three dataframes. The three dataframes created within the sets each contain fifty rows with varying column lengths, those being 45, 90 and 180, Each of the three sets of dataframes uses one of the aforementioned arrays to populate it's dataframes. For example: The first set of dataframes uses the first of the three arrays, so on, so forth.

The next step involves calculating the training delta value for each time period, those being the dataframes with a column count of 45, 90 and 180 as each row within these dataframes equates to a time span, those being 15 minutes, 30 minutes and 60 minutes accordingly.

Following this we combine all of the training delta values into a dictionary, which is used as the data to fit our Ordinary Least Squared model. Using this model we generate a series of fifty predictions of the future delta values. These predicted values are made more understandable by converting them into a monetary value based on the data which had been previously collected.

3.4. Data Collection

Data is collected through the use of the Data Collection script, this script is ran as a separate process to the main application. When the user runs the script it will enter into an infinite state of collecting data every twenty seconds. The data collection process itself involves making two queries to the OKCoin API for each cryptocurrency, one for Ticker data and the other for Depth data, following this the script parses the JSON response and extracts the following data:

- The date and time that the query was made as epoch value.
- The price of the cryptocurrency at the time of the query.
- The amount of bids at the time of the query.
- The amount of asks at the time of the query.

Once this information has been parsed it is stored in a list and formatted as a string, after which it is stored in the associated cryptocurrencies datastore.

3.5. Slack Error Notification

If an exception occurs during the runtime of either the application or the data collection script the send_notification function will be called with the name of the file and function the exception occured in being passed as a string to the function.

Within the notification function an instance of the Slack Client is created using the stored token. Following this the most recent exception is pulled from the stack as a string and appended to the end of the message passed to the function. The slack API is then queried, passing the channel and the message text as parameters. This will post the supplied message in the given channel as a bot user.

Below is a screenshot demonstrating how the generated message would appear to users within the Slack channel.

4. Learning Outcomes

Throughout this project there has been ample opportunity for learning on both a personal and a technical level. Below I will discuss what I believe were the most significant learning outcomes I felt I have taken away from this project.

4.1. Personal

4.1.1. Cryptocurrencies

Throughout this project I have undoubtedly gained an incredibly valuable insight into cryptocurrency and the cryptocurrency economy. Prior to undertaking this project I realistically had little knowledge of cryptocurrency outside of Bitcoin, even such my knowledge of the technology behind it and the influence within the economy that it has was next to nothing.

I found the research conducted throughout this project in relation to cryptocurrency quite eye opening, the extensive list of existing cryptocurrencies within the market as well as the overall market value as astounding. Outside of monetary value, the technology behind the individual cryptocurrencies was astoundingly impressive and interesting to learn about.

One interesting anecdote that I have found after this project is the volatility of cryptocurrency value over such a relatively short period of time. When this project began Bitcoin was valued at roughly €5000, spiking to the dizzying height of €16,200 in mid December, following this the price plummeting back to €5000 in mid february. While the price may be seemingly on the rise once again it certainly demonstrated truly how volatile such a new economy is.

4.1.2. Time Management / Self Discipline

This was by far the largest software based project I have ever undertaken. Despite having a relatively clear schedule during the earlier months of development I found myself procrastinating the work, partially under the assumption of the workload remaining roughly the same throughout the year. By doing so I found myself paying the price during the earlier months of 2018 when the workload was significantly increased and I found myself with less than adequate time to meet my planned iteration goals.

Forcing myself to stop procrastinating and find a balance between the multitude of projects and deadlines that were presented throughout the term was certainly difficult but was certainly rewarding when it was ultimately achieved. Due to my earlier procrastination I found I was forced to make relatively quick decisions regarding important aspects of the project, this lead to finding that I was unable to implement functions I would have otherwise liked to make an addition to the application.

4.2. Technical

4.2.1. Python

Despite having prior experience using Python throughout development I feel as though i've gained a deeper insight into more efficient means of coding in the language in addition to the availability and use of the packages used within the project.

4.2.2. REST APIs

Within the application I use several APIs to perform tasks including data collection and error notification. Through my usage of these APIs I feel as though i've gained a significantly better understanding of the process of interactions with them, not only the process if sending a query but the process of parsing data in such a way as to make it usable for both the application and the developer.

4.2.3. Plotly

My usage of Plotly within the application has done nothing but made their product even more impressive to me, the plethora of different attributes available when generating a graph is nothing shy of staggering. Having both the option to generate a graph and display it through the Plotly site itself and to generate the graph through their offline API was incredibly useful for the purpose of this application, especially when it came to understanding the format the attributes had to be passed to the API.

5. Project Review

5.1. What went right?

5.1.1. Data Collection

I found that while developing the data collection aspect of the project that the process of querying APIs and storing data was relatively straightforward, the only issue discovered during the development was how to run the script at all times. This was quickly remedied by dedicating a previous laptop to the task through configuring it to prevent it shutting down when the lid was closed, while not optimal, for the purpose of this project it was more than suitable.

5.1.2. Transition to Web Application

As mentioned in the Functional Specification, during the earlier iterations I opted to postpone the development of the web application in place of developing more core functionality such as the algorithm and graphing abilities.

As a result of this when reaching the final iteration I was tasked with implementing the already existing functionality into one cohesive application in place of individual files. While I was unable to implement the data collection process into this application I feel as though the transition of the other functionality into the web application was relatively smooth and successful.

5.2. Challenges Encountered

5.2.1. Data Storage

Originally I had intended to use MongoDB as a means to store all the related data due to my past experience using it alongside flask applications, however when resolving the issue of running the data collection script continuously I found that transferring data between the two computers and simply viewing the existing data was going to prove troublesome.

As a result of this I made the early decision to switch from using MongoDB to using a series of CSV file in its place. While not the optimal solution I feel as though it was certainly the correct decision to make given the circumstances and relative time constraints involved in researching a new means of storing data.

5.2.2. Algorithm Development

During the early stage of developing the algorithm I found myself having profound difficulty actually understanding the concepts put forward in the aforementioned paper. Ultimately this forced me to truly sit down and get a better understanding of what was being suggested in the paper and slowing the rate of my development to understand what exactly I was doing along with the reasons why I was doing it.

5.2.3. Time Constraints

As previously mentioned, time was a considerable factor throughout the lifecycle of this project. I found myself having to make rapid decisions as to what functionality I needed to prioritise over others and which idealistic features I had to cut. A considerable worry was the period of time between developing the data collection script, beginning to build my own datasets and when I finally had enough data to begin testing the algorithm, prior to this I had utilised a historical dataset in its place.

5.3. What would I do differently?

5.3.1. Hosting Platform

If selecting a platform on which to host the application again I would ideally put in more time to adequately research any restrictions of the platform. What I found when hosting the application on PythonAnywhere was that they have restrictions implemented as to prevent the usage of WebSockets, additionally their restriction of multi-threading applications would have been a hypothetically issue had I been able to implement said feature.

5.3.2. Documentation

While the documentation was not simply left throughout the development cycle, it was far from updated regularly. If undertaking such a project again I would certainly make it a point to update documentation more frequently. Ultimately this comes down to the aforementioned issue of unfortunate time constraints.

6. Future Features

6.1. Multi-Threaded Application

By implementing multi-threading into the application the process of running the individual methods for each cryptocurrency would be drastically sped up. Additionally the ability to run multiple generation tasks simultaneously it would allow for a significant increase in the amount of data used in each generation, ideally leading to improved accuracy. The addition of multi-threading in the Web Application would also allow for multiple users to be interacting with the application simultaneously, which leads me on to the next features.

6.2. Alternative Hosting Service

Ideally I would like to switch to a hosting service which allows for both the use of websockets as well as multi-threading as they are both issues pertained with PythonAnywhere.

6.3. In-Application Data Collection

This ties in with both the previously mentioned future features, primarily the addition of multi-threading however. Having the data collection process implemented into the application itself would be considerably simpler overall for any given user.

6.4. Additional Cryptocurrencies

While I am happy with my selection of cryptocurrencies to gather data for, I feel that by gathering additional data on cryptocurrencies within different price ranges would allow for even more extensive testing on how the algorithm functions with varying sized values.

7. Acknowledgements

I would like to thank Dr. Lei Shi for both his help, advice and support throughout the course of this project. His reaffirmation of ideas and implemented functionality was invaluable.

Additionally, I would like to thank both the lecturers as well as classmates who provided their feedback and insight into the project during development.

Finally, I would thank both my family and friends for being an unwavering source of support and encouragement throughout this year, you undoubtedly played a significant role in my ability to persist through challenges faced.

8. Bibliography

[1] Devavrat Shah, Kang Zhang. 2014. Bayesian regression and Bitcoin. [ONLINE] Available at: https://arxiv.org/pdf/1410.1231v1.pdf.