

Bitcoin Price Analysis Product

C964 - Computer Science Capstone

Western Governors University

LETTER OF TRANSMITTAL

March 20, 2020

John Smith, CTO
ABC Financial
555 Financial Rd
Miami, Florida

Dear Mr. Smith,

Our company has always prided itself on being at the forefront of the financial advising market. However, with the rise of digital assets, such as Bitcoin, we find ourselves lacking in an area I believe critical to the future success of the company. Many of our clients have been asking about diversifying their portfolios into this new cryptocurrency market. As of today, we are not equipped to properly service these customers. Our company's current focus is the stocks and bonds markets. We currently don't have the resources to tap into this new and booming market.

I believe we can break into this new market, and service our customers with a new data product. This product will enable our employees to get the information they need to provide our clients with the best financial advice and decision-making. It will be a single dashboard with all the information and tools they need. The program will chart the price history of Bitcoin with the ability to filter the results. A bar graph will show the volume of daily trades. It will also incorporate a Linear Regression algorithm to make predictions on future prices based on price history data.

This product will benefit the client in several ways. It will enable the client to get the best information from us to determine whether he or she should add Bitcoin to their investment portfolios. This will potentially lead to higher returns for our customers and improved customer satisfaction. The objectives of the product ensure this. The product's objectives are to ensure the users of the software have up to date and accurate data related to Bitcoin pricing. This leads into the second objective which is to take this data and visualize it to turn it into useful information for decision-making. This will then help accomplish our ultimate objective which is to provide the highest level of customer service by providing our clients with the highest returns on their investments.

The funding required to build and maintain the project consists of an upfront cost of \$21,000. Then an additional \$2,400 per year is required to maintain the software. The developer chosen for this assignment has three years of experience building data applications that implement machine learning algorithms. The developer also has a Bachelor's degree in Computer Science. I believe the developer will be able to build this data product in a timely fashion and on budget.

Thank you for taking the time to read my proposal. I'm looking forward to hearing your response. If you have any questions, please feel free to reach me at my office.

Sincerely,

Project Proposal

Problem

ABC Financial has helped its clients find appropriate investments for many years. With the sudden rise of cryptocurrencies, such as Bitcoin, many of our clients have been asking about adding this digital asset to their portfolios. A recent article by Forbes offered some insight the future outlook of Bitcoin in 2020. The three key points of the article were the Bitcoin halving, the maturing Bitcoin market, and central bank-issued digital currencies (Torpey, 2020). These events could have a positive effect on the value of Bitcoin, thus increasing demand for it. It's important for our company to be equipped and ready for this potential new revenue stream.

As of today, ABC Financial is not equipped with the decision-making tools required to guide our clients into this new realm of investments. Currently, our main offerings within the financial advising space is focused on stocks and bonds. Many of our customers are more risk tolerant and would like to get into higher risk/higher reward investments. This is an area we are currently lacking.

Solution

I believe we can expand our business's offerings with a new data product. This data product will enable the analysis of Bitcoin prices. The Bitcoin price analysis product will enable our employees to get the information they need to successfully recommend the best times for our customers to make Bitcoin investments. They will be able to make better informed decisions that will lead to higher quality financial advisement.

This tool will display Bitcoin price data in several ways that will give a clear look at the current price trends. Based on these trends, we can make recommendations to those customers looking to break into the cryptocurrency market. I believe this product will enable our company to branch out and create new revenue streams. This will also give us a competitive advantage over our competitors that don't offer cryptocurrency advisement.

Outline

The data product will be a standalone application written in the open source programming language Python. It will support industry appropriate security features such as a login page. The home page will be a dashboard with all the visual data in one place. A line graph will chart the Bitcoin price over time. A bar graph will display the daily volume of Bitcoin trades. There will be a section that displays potential future prices. This section will get its data from training a machine learning algorithm on past prices. It will use Linear Regression to plot future data points based on price history.

Data Description

The data used to construct this product will be from the Bitfinex trading platform made available to the public by Quandl.com. This data includes daily price data such as:

- High (Integer)
- Low (Integer)
- Mid (Integer)
- Last (Integer)
- High/Low Percent (Real)
- Volume (Integer)

- Date (Text)

The product will pull fresh data every time it is launched, and then will store this data into a local database. The data is limited in that it is single day price data. The product won't have access to highly detailed data, such as price changes every minute or second. Thus, high frequency trading is out of the scope of this project.

Objectives and Hypothesis

There are several objectives of this product. One of the main objectives is to ensure the user of the software has the most up to date and accurate data related to Bitcoin pricing. Another important objective is to process and then visualize this raw data to make it useful. This will help to accomplish our other main objective which is to improve our employees decision-making abilities with the processed information. These all work to solve the ultimate priority of our business, which is to provide the highest level of customer service by providing our clients with the highest return on their investments.

My hypothesis is that through Linear Regression semi-accurate prices predictions can be obtained. This future price data will enable our company to make better decisions for our clients. I believe if this product is built and maintained, then both our company's revenue and our client's satisfaction level will greatly improve.

Project Methodology

This project will use the Waterfall Methodology. This methodology was chosen because the requirements for the project are well defined. The products feature set is not overly complicated, so a straight forward and linear methodology will work best. This will keep development on track and enable delivery of the completed software on time and within budget. The following describes which parts of the project align with this methodology's phases:

1. Requirements – During this phase, we will meet with the end users of the proposed product and gather all the requirements.
2. Design – Then, a design will be created that ensures all the requirements from the previous phase are met.
3. Implementation – During this phase, the code will be written ensuring it adheres to both the requirements and design.
4. Verification – Once the code is complete, and working version will be given to the end users to ensure the product does what it is supposed to do.
5. Maintenance – As the end users use the product, any bugs, inadequate features or other errors will be fixed as needed.

Funding requirements

The majority of the product is created using free open source software and tools. However, costs will arise from several areas. The software developer team will consist of one software developer. The estimated hours the project will take is ~110 hours. At \$100/hour the developer's salary will cost \$11,000. Upfront costs for acquiring the equipment the developer will need is estimated at \$10,000. Additionally, maintenance costs will be roughly \$100/hour for 24 hours per year onward. So, this product will require a total of \$21,000 of upfront costs to get a working product plus an additional \$2,400 per year.

Stakeholders Impact

There are many individuals with a vested interest into our company. This proposed product will affect them in several ways. Our customers will be able to further diversify their investments with Bitcoin through the new product. The company's value will increase through the addition of a new revenue stream which will positively affect our shareholders. The increase in the company's value will lead to job stability and job growth for our employees.

Data Precautions

The data used in the product is made freely available to the public by Quandl.com. Therefore, there is no sensitive or protected data. The application makes no use of personal information from the users. The product is in full compliance with standard privacy laws. Typically, in situations where an application is working with sensitive data, such as in health care, education, or payment processing, there is both an ethical and legal requirement to protect that data. Many precautions need to be in place to ensure the data remains private and does not leak out. Since our product doesn't involve the use of such data, we don't need to take such heavy precautions.

Developer's Expertise

The developer chosen to build this new product has plenty of experience in building and maintaining software. The developer has three years experience developing data applications that apply machine learning algorithms. On top of all the developer's experience, the developer also has a Bachelor of Science in Computer Science. I believe the experience coupled with a proven track record make this developer an ideal candidate to develop our new data product.

EXECUTIVE SUMMARY

Problem Statement

Within the past several years, a new form of financial assets has emerged. This new breed of digital currency has taken the world by storm. Bitcoin is a cryptocurrency that allows anyone to send and receive payments without passing through a central authority, such as a bank (Acheson, 2018). This has lead to a new type of investment for those looking to diversify their portfolio. As a financial advising company, we have a duty to ensure we have the appropriate tools to guide our clients into this new, potentially risky endeavor.

As it stands today, we are not equipped to provide our customers with this. We need a way to gather more data and transform that data into usable information. This information will empower our employees with the information and knowledge they need to make the best possible investment decisions for our clients.

Customer Summary

The “customers” that will be affected most by this new data product will actually be two separate groups. The first is the users of this new data product, our employees. These people will be using the product on a day to day basis. They work with our clients to find investments most appropriate for their level of needs. The data product will be installed on to the employee’s computer along side their other financial data tools. The data product will provide information similar to what the employee has worked with before. Some examples include: a line graph that plots price history over time, a bar graph that charts volume data daily, and a price forecast section derived from a machine learning algorithm. Thus, our employees will already have the skill sets needed to use this new product and get the most utility out of it.

The second “customers” of this new data product is the actual customers, our clients. This group consists of the people coming into our office and seeking out financial investing help. This includes retirees, people blessed with a sudden windfall, and just average Joes looking to put their money to work. Many of these clients have been asking to invest into this new Bitcoin market. Our new data product will enable us to fulfill these customer’s needs.

Existing System Analysis

As of today, our systems and tools are targeted towards the stock and bond markets. These tools pull and display data focused only on those areas. They do not provide any information on cryptocurrencies. This new data product is designed to work along side the existing system. It will be installed on the same computers that run our existing software. This will allow us to expand into a new market while still supporting our current offerings.

The existing physical systems won’t need to be altered too much for the new product. Our employee’s existing machines will be able to run the program considering they are able to run similar programs already. I suggest a small upgrade to the RAM of each system to support the additional concurrent

running programs. I would also recommend a dual monitor setup for each workstation. This will enable the employee more screen space to display all the information required to do their jobs. All of these suggested upgrades have been accounted for in the funding requirements.

Data

The data our product will be using is originally sourced from Bitfinex. Bitfinex is a trading platform for Bitcoin. People can buy and sell Bitcoin through this exchange. This makes Bitfinex an excellent source of data for determining Bitcoin's value. This data provides a lot of useful information such as the price people are willing to pay for Bitcoin along with data pertaining to how much and how often people are buying and selling it. Bitfinex makes this data available to the public for free through Quandl.com. Quandl.com is a website that provides financial and economic datasets to anyone with an account.

Our product pulls this data from Quandl through their public facing API. Quandl also provides a third party Python module that coalesces nicely with our code. We import this library and use it to pull the data off of Quandl's servers each time the application is launched. We then execute the pullAndCleanData function to create a new dataset with only the data relevant to our uses. This dataset includes the columns High, Low, Last, Volume, and High/Low Percent (HL_PCT). Then any null data from the original data set is replaced with the integer -99999. This is done so that null data will be treated as an outlier, instead of deleting a whole row that may contain other useful data. This data is then stored locally in a SQLite3 database. We store the data locally so that if Quandl's servers are ever down, we still have data to work with.

Project Methodology

The industry standard methodology that will be used for this project is the Waterfall Model. This a good choice for this type of project because the requirements will be well known in advanced. Since a single developer will be working on this project the linear sequential flow will make it easier for the developer to guide and support the design and development of the product. Each phase will produce a deliverable that will act as the input to the next phase. The way the aspects of this project will be managed by each phase is detailed as follows:

1. Requirement Analysis -

- A meeting with the end users will occur to discuss the what the program will need to do to effectively support decision-making.
- A list of these requirements will be captured for use in the next phase.
- The scope of the program will be defined to ensure no unnecessary features will be included in the final design.

2. System Design -

- The documents produced from the previous phase will be analyzed during this step.
- The requirements will be studied to ensure the developer has a clear understanding of what the product need to accomplish.
- A system design, based on these requirements, will be prepared. This design will detail the structure of the program and how modules will interact with each other.

3. Implementation -

- The system design from the previous phase will be used to begin building the code base.
- The code for each module will be written and tested individually.
- This is the phase where Unit Testing will occur to ensure each piece of software works as intended before integrating them with the rest of the code.

4. Integration and Testing -

- Once the individual modules are written and tested, they will be assimilated one module at a time.
- After each module is added to the code base, additionally Integration Testing will occur to ensure each module interacts with the others properly.
- Once all modules are integrated, System Testing will take place. The entire system will be tested as a whole to ensure there are no faults or failures.

5. Deployment -

- Once the testing has taken place and any discovered bugs have been fixed, the product will be deployed onto our employee's computers.
- Acceptance Testing is done at this stage to ensure the software works as intended.
- The software will be installed along side the existing financial data tools.
- Any hardware upgrades that support the addition of the new software will be done during this time.

6. Maintenance -

- Once the software is out and being used daily, issues are bound to arise.
- This stage is a continuous stage over the life of the software because there is no such thing as a 100% bug free program.
- During this phase, patches are released as issues arise and are fixed.

Project Outcomes

The project will produce many deliverables. These deliverables will fall into two distinct categories: Project Deliverables and Product Deliverables. Within the first category, each phase of the project methodology will produce one or more deliverables that will typically be used as input into the next phase. During the first phase, a requirements document detailing a list of requirements the final project must have will be produced. A scope statement will also be created to ensure all parties understand what features will be implemented and also what features are out of the scope of the project, such as high frequency trading. The next phases will produce documents associated with the design of the program. One of these documents include a flow chart describing the structure of the code and how the modules will interact. A wire frame will also be produced to get an idea of how the GUI will look. A testing plan will be created here to ensure the next phase is ready to begin producing the code.

The following phases will produce deliverables associated with the second category, Product Deliverables. The program's source code modules will be built during the Implementation phase and the final program will be the output of the Integration and Testing phase. At that point, we will have a

fully functioning GUI. A database full of up to date Bitcoin pricing data will also be produced after those phases.

Implementation Plan

Once the product has been produced, it will be time to put it into the production environment. The plan for which is detailed by the following:

- **Strategy for implementation** – A critical step for any software product is ensuring an effective integration with existing systems and workflows. The software will be designed from the start to work along side existing software and systems. This ensures that once it's time to implement the new data product, it can be easily installed onto existing computers without interrupting the existing programs. Another key factor when implementing a new software solution is making sure the new software integrates well with existing workflows. Since the data product is designed to function in a similar manner as our employees existing tools, it will be a much smoother transition for our employees once the software rolls out.
- **Phases of roll out** – First, the product will be installed onto a small number, such as three or four, of our employee's computers. They will do acceptance testing to ensure the program fulfills all of its original requirements. Then, we will install the product on a small group of employee's, between 10 to 15, computers. These employees will work with a select few clients to use the product and begin investing in Bitcoin. After a few weeks, any problems that are uncovered will be fixed. Finally, the product will be rolled out to all remaining employees and the product will officially be in production.
- **Levels of testing and final distribution** – Testing will be done at each stage of the roll out. The initial stage will see acceptance testing to ensure the original requirements are fully met. The following stage will test how the program works in a controlled production environment. A few clients will begin to work with our employees and the program to begin using the product in a live setting. During each of these steps, any bugs discovered will be fixed and patches issued. Once all the issues are resolved, the final roll out will be implemented and the product will be distributed to all our employees.
- **Milestones** – Each stage of the product roll out will mark a milestone. Each milestone will bring the product closer to the final distribution. These milestones will be planned and scheduled to ensure that the project stays on track and the final distribution release date doesn't get pushed back.
- **Deliverables** – There will be a few documents produced during the implementation. A document detailing and ultimately accepting the product will be created during the acceptance testing of the roll out. This will verify the product meets the original requirements. Bug reports will be produced during the testing portions of the roll out. A project closure document will be signed off by the CTO ultimately declaring the project complete and finished.

- **User testing** – User testing will be done at each stage of the roll out, as detailed above. Any bugs discovered will be logged in a bug report. These will be subsequently resolved and patches to the program will be issued.

Evaluation Plan

Verifying and validating the application meets all the requirements is a crucial part of any project. Testing will be an integral part of ensuring the application does what it is supposed to do. There will be tests done at each step of the development life cycle. Unit testing will be done upon completion of each module of code. Each module must pass individual tests before writing the next one. Integration testing will occur each time a module is added to the code base. This will ensure the modules interact with each other properly. Once the product is fully written, system testing will commence to ensure that, as a whole, the program is executing properly. Finally, a subset of the end users will initiate the acceptance testing to verify the product meets all the requirements that were originally laid out in the beginning.

Further verification and validating will occur outside of the regular testing. Once the product is fully rolled out, we need to ensure that the product meets its most important objective, earning meaningful return on investment for our clients. This will be done by closely monitoring the performance of our client's portfolios that have invested in Bitcoin with the help of our product. We can consider our product successful if at least 75% of these clients increase the value of their investment portfolios by a minimum of 5% over the first three months of the product's roll out. This will be a metric to measure whether the product is successful in accomplishing its primary objective, an increase in ROI for our client's portfolios.

Resources and Costs

Costs are a major factor for any project. Luckily, our project relies heavily on free, open source software and tools. Also, a majority of the hardware requirements already exist within our business. Here is a breakdown of the costs associated with the project:

- **Programming Environment** – Hiring a developer isn't the only expense needed to build this project. We need to make sure that the developer has the tools they need. As far as hardware is concerned, we need to provide the developer with a laptop to work on. A budget of ~\$1,200 will be allotted for the purchase of the developer's work laptop. The majority of the software needed to complete the project will be free. Python, SQLite3, Git, Python's Integrated Developer Environment, and the third party libraries for Python are all open source and free. A Windows 10 license will need to be purchased if the laptop doesn't come with Windows preinstalled.
- **Environment Costs** – The developer will be sharing office space with our other employees so the costs typically associated with renting office space will be spread out among the already existing employees. So, these costs will be very minimal. A central server to house the database will be acquired at the cost of \$2,000. Electricity and internet costs will also be spread out among the existing employees, so these costs are also very minimal.

- **Human Resource Requirements** – The bulk of the costs for the project come from the salaries of the employees. Our project requires only one developer, so we can keep our costs at a minimum. The cost breakdown for human resource requirements are as follows:

Description	Hourly Rate	Time	Total
Planning	\$100.00	17 hours	\$1,700.00
Design	\$100.00	17 hours	\$1,700.00
Implementation and Integration	\$100.00	60 hours	\$6,000.00
Testing	\$50.00	20 hours	\$1,000.00
Totals		114 hours	~\$10,400.00

Timeline and Milestones

The project will take one and a half months to complete. The planned completion date is May 1, 2020. Roughly 110 hours will be spent over the course of the time completing milestones for the project. A breakdown of the timeline and planned milestones is as follows:

Milestone	Start and End Dates	Duration	Resources
Requirements Analysis	March 23 – March 25	15 hours	End Users, Stakeholders
GUI Design and Mock-up	March 26 – March 27	7 hours	Software Developer
Code Architecture and Flow Design	March 30 – March 31	8 hours	Software Developer
Module Development and Testing	April 1 – April 17	40 hours	Software Developer, QA Engineer
Module Integration and Testing	April 20 – April 24	20 hours	Software Developer, QA Engineer
Stage 1 Deployment and Acceptance Testing	April 27 – April 28	8 hours	Software Developer, End Users
Final Deployment	April 29 – May 1	12 hours	Software Developer, End Users, Stakeholders

Sources

- Torpey, K. (2020, January 17). The Bull Case For Bitcoin In 2020. Forbes. Retrieved from <https://www.forbes.com/sites/ktorpey/2020/01/17/the-bull-case-for-bitcoin-in-2020/#ac739bf3878a>
- Acheson, Noelle. (2018, January 20). What is Bitcoin. Coindesk. Retrieved from <https://www.coindesk.com/learn/bitcoin-101/what-is-bitcoin>
- Li, Yun. (2019, November 25). Bitcoin sinks to lowest level since May, falling \$3,000 in a month as China accelerates crackdown. CNBC. Retrieved from <https://www.cnbc.com/2019/11/25/bitcoin-sinks-to-a-6-month-low-as-china-accelerates-crackdown.html>

Post-Implementation Report

C964 - Computer Science Capstone

Western Governors University

Table of Contents

Project Purpose 3

Datasets 3

Data Product Code 4

Hypothesis Verification 5

Effective Visualizations and Reporting 5

Accuracy Analysis 6

Application Testing 6

Application Files 8

User’s Guide 8

Summation of Learning Experience 9

Project Purpose

Before the completion of this product, we didn't have any way of offering financial advising in the cryptocurrency space for our clients. Our software tools were focused solely on the stocks and bonds markets. We were ill equipped to get the data and information we needed to suggest when to buy and sell Bitcoin. This left us at a disadvantage against other companies that were beginning to offer these services. It also left a portion of our customers that wanted to invest in Bitcoin frustrated that we couldn't accommodate their requests.

The application that was built was able to address these concerns and expectations of our clients. Our employees were now equipped with a Bitcoin analysis tool that allowed them to get the information they needed to make appropriate decisions for our clients. This met the expectations of the client because they were now able to diversify their portfolios through Bitcoin investments. The final program was able to meet all the requirements of the end users of the software, our employees. They needed a way to visualize Bitcoin price history to spot trends in the data. This was accomplished by plotting out Bitcoin's prices on a line graph with the Python library Matplotlib. Additionally daily volume data was another requirement for the program to see the amount of people buying and selling Bitcoin. This shows if interest in Bitcoin is either growing or waning over time. The objective was accomplished through the use of bar graphs to easily see whether the volume over time was increasing or decreasing.

Datasets

The program uses freely available data from Quandl.com. It pulls this raw data, and then cleans it for our purposes. A sample of the raw data is shown below.

Date	High	Low	Mid	Last	Bid	Ask	Volume
2014-04-15	513.900000	452.00	504.23500	505.0000	503.5000	504.97	21013.584774
2014-04-16	547.000000	495.00	537.50000	538.0000	537.0000	538.00	29633.358705
2014-04-17	538.500000	486.10	507.02000	508.0000	506.0400	508.00	20709.783819
2014-04-18	509.000000	474.25	483.77000	482.7500	482.7500	484.79	10458.045243
2014-04-19	513.989900	473.83	505.01065	507.4999	502.5313	507.49	8963.618369
...
2020-03-05	9186.500000	8788.60	9075.05000	9075.0000	9075.0000	9075.10	5663.171625
2020-03-06	9199.700000	9011.30	9172.85000	9174.5000	9172.8000	9172.90	3113.235936
2020-03-07	9241.500000	8867.40	8918.95000	8919.0000	8918.9000	8919.00	4483.884533
2020-03-08	8919.000000	8060.00	8062.25000	8060.6000	8060.2000	8064.30	24945.376311
2020-03-09	8287.213641	7700.00	7902.65000	7902.7000	7902.6000	7902.70	26024.364534

This raw data included more information than was needed for our purposes. Thus, a function was created to clean this dataset and create a new one with just the data that was useful for our purposes. This function is the 'pullAndCleanData()' function. It firsts creates a new column that calculates the High/Low percentage for each row and adds that column to the dataset.

```
dataFrame['HL_PCT'] = (dataFrame['High'] - dataFrame['Last']) / dataFrame['Last'] * 100
```

It then creates a column using the data index. This was done to make it easier to plot the data with the dates in Matplotlib. After, it creates a new dataframe using only the columns we need for the program and discarding the columns we do not need.

```
dataFrame['Date'] = dataFrame.index
dataFrame = dataFrame[['Date', 'High', 'Low', 'HL_PCT', 'Last', 'Volume']]
```

Finally, it replaces any null data with the integer '-99999'. It's important that we don't have any null data when passing this data to Matplotlib and Sci-kit Learn. This ensures that any null data will be treated as an outlier and we don't get an errors at runtime.

```
dataFrame.fillna(-99999, inplace=True)
```

Once the data has been cleaned and ready for our uses, the function returns the new dataset. At this point we have the data we need and can insert it into our database. A sample of the cleaned code is shown below.

Date	High	Low	HL_PCT	Last	Volume
2014-04-15	513.900000	452.00	1.762376	505.0000	21013.584774
2014-04-16	547.000000	495.00	1.672862	538.0000	29633.358705
2014-04-17	538.500000	486.10	6.003937	508.0000	20709.783819
2014-04-18	509.000000	474.25	5.437597	482.7500	10458.045243
2014-04-19	513.989900	473.83	1.278818	507.4999	8963.618369
...
2020-03-05	9186.500000	8788.60	1.228650	9075.0000	5663.171625
2020-03-06	9199.700000	9011.30	0.274674	9174.5000	3113.235936
2020-03-07	9241.500000	8867.40	3.615876	8919.0000	4483.884533
2020-03-08	8919.000000	8060.00	10.649331	8060.6000	24945.376311
2020-03-09	8287.213641	7700.00	4.865598	7902.7000	26024.364534

Data Product Code

The data was analyzed through several methods. My descriptive method is what I used for variable selection and variable elimination. The pullAndClean data function was used to determine which data features would be used to train the classifier. This was accomplished by disregarding useless features and creating a new feature based on the existing data:

```
dataFrame['HL_PCT'] = (dataFrame['High'] - dataFrame['Last']) / dataFrame['Last'] * 100
dataFrame['Date'] = dataFrame.index
dataFrame = dataFrame[['Date', 'High', 'Low', 'HL_PCT', 'Last', 'Volume']]
dataFrame.fillna(-99999, inplace=True)
```

My predictive method was where I created and trained the classifier. The forecastPrices function accomplished the bulk of the analysis done with the machine learning algorithm. I determined my 'label', which was my 'y' value, would be the Last Price data column. That would be what I was predicting, i.e. my future prices. My 'X' value was the features that I would use to train my classifier.

```
X = numpy.array(dataFrame.drop(['label'],1))
X = preprocessing.scale(X)
recentX = X[-forecastOut:]
X = X[:-forecastOut]
y = numpy.array(dataFrame['label'])
```


Then this function splits the data into training and testing data. It uses this to train the classifier and test the model to see it's accuracy. Finally, it outputs predicted data into the 'futurePrices' variable to be later visualized onto the application's dashboard.

```
X_train, X_test, y_train, y_test = model_selection.train_test_split(X, y, test_size=0.2)
classifier.fit(X_train, y_train)
score = classifier.score(X_test, y_test)
futurePrices = classifier.predict(recentX)
```

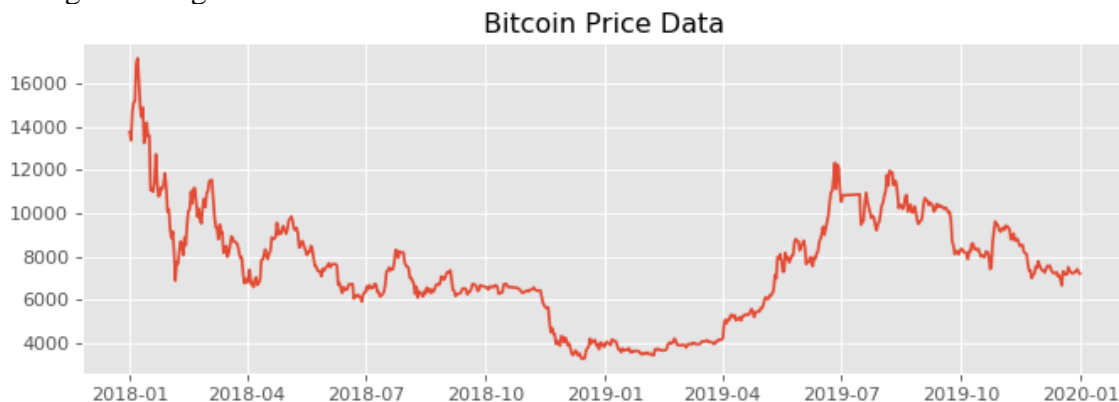
Hypothesis Verification

Our original hypothesis stated that through Linear Regression we would semi-accurately predict future Bitcoin prices. Based on the results, this section of the program is best used minimally for decision making. There are many reasons for this. Linear Regression attempts to find a “best fit” line through the data. This can then be used to predict future data based on past data. It uses the trend of the data to predict future value if the trend continues. This isn't always a reliable indicator of the future value of financial assets. There are many factors that contribute to the rise or fall of Bitcoin prices. Many of these factors have nothing to do with financial data at all. Political agendas can have a large effect on the price of Bitcoin. China has historically introduced strict regulation on Bitcoin businesses. This has even led to Bitcoin prices falling by \$3,000 in a single month (Li, 2019).

These factors can greatly influence Bitcoin prices, which can't be predicted through our current Linear Regression algorithm. Including more appropriate “features” to train the algorithm can help to increase the price prediction accuracy. As it stands now, this section is best used complementarily to the other decision-making visualizations in the application, and not used solely to determine the best times to buy and sell.

Effective Visualizations and Reporting

My visualizations were chosen because of their usefulness in accomplishing our objectives. Our employees can glean a lot of information through the line graph. This graph shows an accurate representation of the history of Bitcoin's prices. Displayed in this way, it makes it a lot easier to spot pricing trends within the data. Our employees will be able to determine whether the overall trend in price is rising or falling.



Graphing the data this way gives our employees useful information for whether they should be recommending buying or selling Bitcoin. The graph also supports filtering dates to zoom into the graph to get more detailed data for those specific dates. The bar graph gives a lot of useful information about

the demand for Bitcoin and how popular it is overall. The bar graph charts the daily volume of Bitcoin trades. So, a trend of higher volumes over time show that Bitcoin is rising in popularity and it could be a good opportunity to increase our clients holdings.

The application supports data preparation by taking the raw data and organizing it by keeping useful data and discarding the rest. It displays this clean data in a way that allows our employees to do data analysis. The application uses Linear Regression to analyze the data and make predictions based upon the data. This gives the users more information to help make purchasing decisions. All this preparation and analysis is summarized nicely within the three visualizations: the line graph, the bar graph, and the future prices chart.

Accuracy Analysis

Scikit-Learn includes function that provides a measure of how well observed outcomes are replicated by the model. This function is called `score()` and it returns the coefficient of determination. Once the training and testing dataset is determined, the classifier fits the training data and runs the score function on the test data:

```
X_train, X_test, y_train, y_test = model_selection.train_test_split(X, y, test_size=0.2)
classifier.fit(X_train, y_train)
score = classifier.score(X_test, y_test)
```

This score will give you a number that shows how accurate the model is at predicting observed outcomes. Bitcoin is a very volatile asset. Using this model to try and predict future prices is difficult considering the environment. This causes the future price set to somewhat resemble the price set that the classifier was trained on.

Application Testing

Different levels of testing was performed throughout the development life cycle of the application. As each of the individual modules were finished, unit testing took place. For some modules, this was accomplished by providing a sample input to the module. Then the module was executed and the output was observed. The output was compared to the original input. If the module changed the output in the way that was expected, then the module passed its unit test. An example of this was the unit testing for the `pullAndCleanData` function. The input is a string of a ticker for the Quandl API. This pulled the data from Quandl into a dataframe. The dataframe was printed so it could be compared later. The function did some cleaning of the data and returned a new dataframe. These two dataframes were compared to ensure the function successfully altered the dataframe the way that was intended.

Integration testing was done as multiple modules were completed. One example of this includes the `insertData` function. This function relies on the output of the `pullAndCleanData` function to work properly. The testing commenced by first observing the data within the existing database. Then we feed the output of `pullAndCleanData` to the `insertData` function. After execution, we observed the changes within the database. If new data was added to the database properly, the functions passed the integration testing. The before data:

priceID	date	high	low	HL_PCT	last	volume
Filter	Filter	Filter	Filter	Filter	Filter	Filter
2102	2020-02-29 00:00:00	8829	8581	2.88	8581	2390
2103	2020-03-01 00:00:00	8775	8460	2.53	8558	5383
2104	2020-03-02 00:00:00	8991	8525	0.73	8926	5164
2105	2020-03-03 00:00:00	8940	8686	1.44	8813	5316
2106	2020-03-04 00:00:00	8878	8718	1.04	8786	4168
2107	2020-03-05 00:00:00	9186	8789	1.23	9075	5663
2108	2020-03-06 00:00:00	9200	9011	0.27	9174	3113
2109	2020-03-07 00:00:00	9242	8867	3.62	8919	4484
2110	2020-03-08 00:00:00	8919	8060	10.65	8061	24945

The database after running the integration tests:

priceID	date	high	low	HL_PCT	last	volume
Filter	Filter	Filter	Filter	Filter	Filter	Filter
2104	2020-03-02 00:00:00	8991	8525	0.73	8926	5164
2105	2020-03-03 00:00:00	8940	8686	1.44	8813	5316
2106	2020-03-04 00:00:00	8878	8718	1.04	8786	4168
2107	2020-03-05 00:00:00	9186	8789	1.23	9075	5663
2108	2020-03-06 00:00:00	9200	9011	0.27	9174	3113
2109	2020-03-07 00:00:00	9242	8867	3.62	8919	4484
2110	2020-03-08 00:00:00	8919	8060	10.65	8061	24945
2111	2020-03-09 00:00:00	8287	7700	4.87	7903	26024
2112	2020-03-10 00:00:00	8180	7777	2.48	7982	10230

There is new data from the next two days, so we know the functions successfully executed. The system testing was done similarly. However, the application as a whole was tested. This was done by walking through each step of running the program. Ensuring that every feature worked as expected.

Once the program was fully built and tested, the acceptance testing took place. We took a small portion of the final users, our employees, and sat down with them at their workstations. Once the program was installed and running, they began testing it to ensure that all the original requirements had been met. The results of all these tests were used to improve the program. Anytime something wasn't working as expected the developer used those errors to pinpoint bugs in the code. These bugs were corrected and the program improved each time.

Application Files

The Python programming language allows you create nice, GUI based programs very easily with a small number of files. All the source code exists in a single '.py' file. This file is named BitcoinAnalysisProduct.py. Another file used by the application is the database file. This file is named

BitcoinPricing.db. The program stores all the pricing data it pulls from the cloud into this file. If this file is not present when the program launches it will be created. The file hierarchy is described by the following:

```
./C_DataProgram
  BitcoinAnalysisProduct.py
  BitcoinPricing.db
  healthLog.txt
```

Python enables us to build impressive applications while using the bare minimum of the number of files. A log file is also included that stores information about how the program is running. This is useful for debugging issues that may arise within the application.

User's Guide

The following details the steps required to install and use the application.

1. Ensure that these programs are installed onto your Windows computer:
 - Python 3
 - Pip (To install third party Python libraries)
 - Matplotlib
 - Scikit-learn
 - Quandl (Python library)
 - Numpy
2. Open a command prompt and navigate to the project directory. This directory will contain both 'BitcoinAnalysisProduct.py' file and the 'BitcoinPricing.db' file.
3. Run the command: C:\path\to\python.exe
C:\Users\Username\script_directory\BitcoinAnalysisProduct.py

*Important Note: Ensure that the command prompt directory is in the same location as the BitcoinPricing.db file. Otherwise you will get this error ->

sqlite3.OperationalError: no such table: prices

*Alternatively, you can open the BitcoinAnalysisProduct.py file in Python's IDLE program and run it from there.

4. Log into the program with these credentials:
 - Username = test
 - Password = test
5. View the dashboard made up of three sections:
 - The main line graph displaying Bitcoin price history.
 - The bar graph displaying daily volume information.
 - The side chart displaying future price information derived from a Linear Regression algorithm.

Summation of Learning Experience

There were many past experiences that assisted me in the completion of this project. My prior experience with learning Python programming helped me greatly during this project. I was able to use the skills I've learned on my own to produce working Python code. I had completed free, online courses that used Python to build graphical user interfaces, graphs with Matplotlib, and data analysis programs. These skills were essential to build the data product required for this capstone.

The largest contributor to my success, though, has been the Computer Science program from Western Governors University. Through this degree program I have received a solid foundation of computer programming. A big part of this project was about working with and analyzing data. So, SQL was an important language to use to complete all the requirements. I was very comfortable using SQL after completing a previous class in this degree program. That class gave me the knowledge and skill to work with data the way I needed to for this project.

There was one requirement for this project that required me to seek out help. I didn't know very much about machine learning algorithms when I started this project. However, I was able to find a video series on Linear Regression from a very skilled, competent developer. Through his tutorials, I was able to learn a lot about Linear Regression and how to apply that to financial data. I took the knowledge I learned from him and used it to build my machine learning algorithm to predict future Bitcoin prices. This experience has really showed me that no matter how much knowledge you will acquire in the future, there will always be something you don't know. Learning is a life-long process. I will now be more open to seeking help and knowledge from others. This project has shown that technology is a fast-paced environment and we computer engineers will always need to seek new knowledge and help to keep up with these changes.