

Functional Requirements Document

DATA 515 - Course Project

University and Department:



W **MASTER OF SCIENCE IN DATA SCIENCE**
UNIVERSITY *of* WASHINGTON

Topic of Interest:



VERSION HISTORY

Version	Description of Change	Author	Date
Version 1	First Draft	All	04/25/2018
Version 2	Addressed review comments from Joe	Tejas	05/12/2018

CONTENTS

1	INTRODUCTION	3
1.1	Purpose	3
1.2	Scope	3
1.3	References	3
1.4	Assumptions and Constraints	3
2	FUNCTIONAL REQUIREMENTS	4
2.2	Detailed Requirements	4
3	COMPONENTS	5
4	DATA SOURCES AND RETRIEVAL	5
5	DATA FLOW	7
6	DATA PREPROCESSING	7
7	MACHINE LEARNING	8
8	VISUALIZATION	8
9	OTHER REQUIREMENTS	9
9.1	Interface Requirements	9
9.2	Data Conversion Requirements	9
9.3	Hardware/Software Requirements	9
9.4	Operational Requirements	9
	APPENDIX A - GLOSSARY	13

1 INTRODUCTION

Even though there is no worldwide acceptance of crypto currencies, we are at point in time in its evolution where there is significant endorsement for it at the highest levels of Government and Financial Organizations in multiple large economies of the world. To a large extent, this has quelled the apprehensions of its legitimacy. As a result, we have around 23.7 million and more people who trade in Bitcoins alone as of today. Some proportion of these users are day traders and would be highly benefitted with a daily prediction of Bitcoin price, even if it was going to be a binary prediction of up/down. As we have realized, there are many indicators for Bitcoin price movements akin to stocks in the complex global financial system. There have been some strong indicators identified like US Dollar price, Gold Bullion price, etc. What is lacking is a system that can combine many of these indicators and create a provide a consolidated and reasonably accurate prediction.

1.1 Purpose

The goal of this project is to create a daily binary prediction system for Bitcoin price. This document details the entire design specification for achieving the same. The project will be submitted to satisfy the requirement of DATA 515 – Software Engineering Course – Spring Quarter 2018. The project name is “Bitcoin Predict”. It will be hosted on GitHub (Link – TBD).

1.2 Scope

This document specifies the functional requirements and the technical design of the project. It provides the use cases, user personas, user interaction points, the datasets being considered, the Machine Learning model used, the deliverables, dependencies, assumptions and constraints.

1.3 References

GitHub Link: TBD

Meeting Summaries: TBD

1.4 Assumptions and Constraints

1.4.1 Assumptions

- The datasets have no bias in collection of the data.
- etc

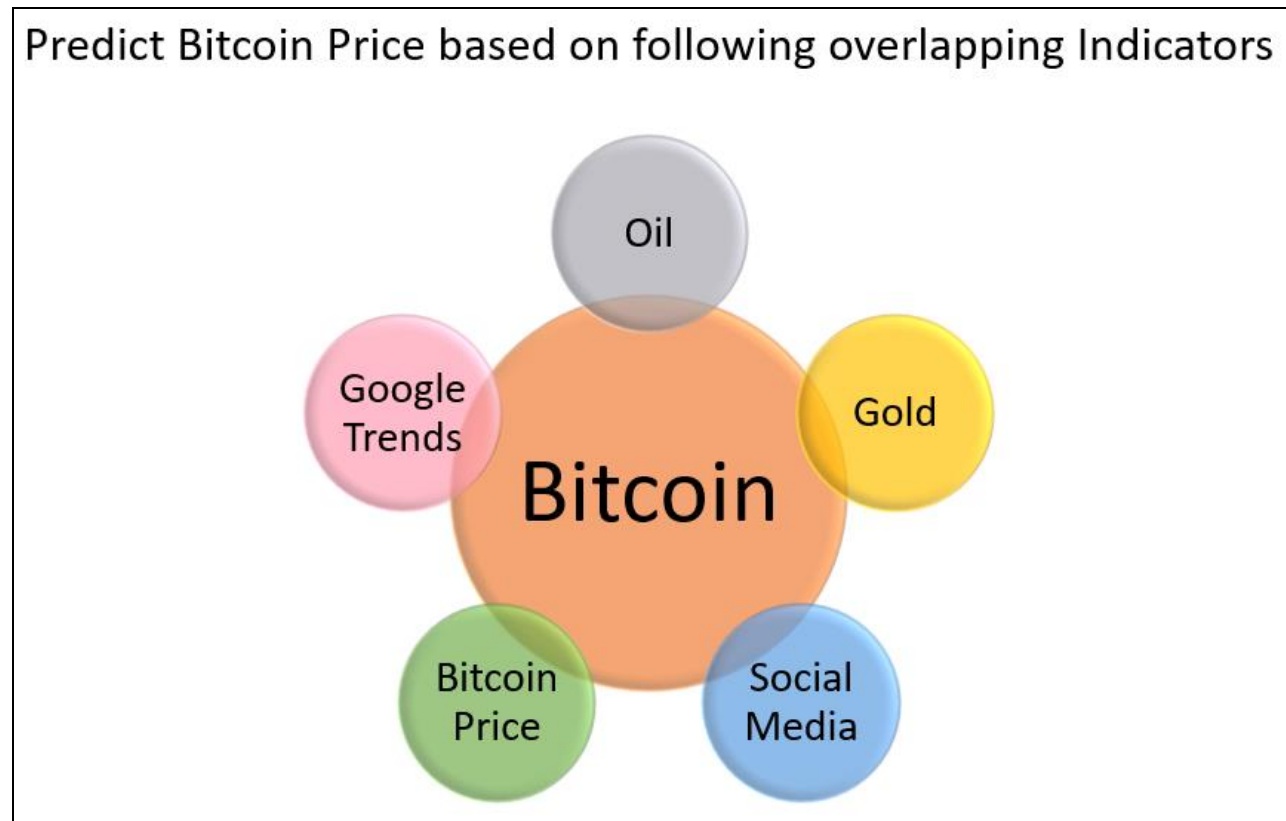
1.4.2 Constraints

- Out of box models being used are tested for accuracy.
- etc

2 FUNCTIONAL REQUIREMENTS

2.1 High Level Requirements

The final product of this project is an interactive visualization and the intended end users of this product are cryptocurrency day traders and enthusiasts.



2.2 Detailed Requirements

2.2.1 Website

Create a website for the product where the user can access and use it.

2.2.2 Binary Prediction

User should be able to see a binary (Up/Down) prediction for Bitcoin Price which updates every 24-hour window.

2.2.3 Binary Prediction Confidence

User should be able to see the confidence level of binary prediction.

2.2.4 Past Performance Interactive Visualization

Users should be able to view past performance in an interactive visualization that allows him/her to filter by a date range. Users should be able to zoom in on the visualization

2.2.5 Social Media Buzz Word-cloud

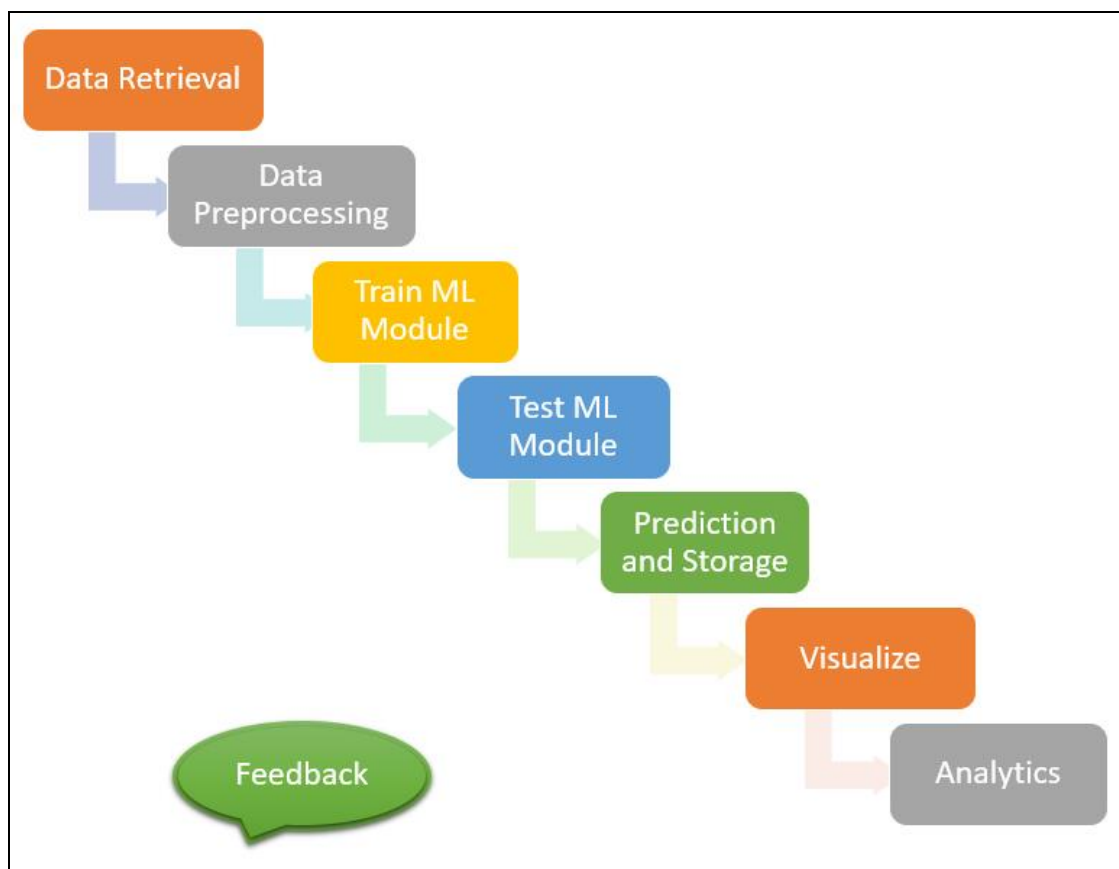
Users should be shown the word-cloud for relevant social media buzz for the last 24-hour window.

2.2.6 Skill for Alexa (Optional)

Add a skill to Amazon Alexa to provide this prediction for suitable questions.

3 COMPONENTS

3.1 High Level Architecture



4 DATA SOURCES AND RETRIEVAL

The section lists all the data sources to be used and corresponding retrieval strategies.

4.1 Input Data Sources

Input data sources are brainstormed amongst the team, to identify the data sources that can have sufficient predictive power for the use-case. After identifying the sources, we are relying on API calls from different sources to fetch data.

The data required for prediction is:

- Bitcoin price/memory-pool/transactions etc.
- Social media data (Twitter, Reddit etc.)
- Google search trends
- Oil price
- Gold price

Bitcoin price history has been volatile, and over the years gone through huge increases and drops on news related to hacks, regulations etc. However, the aim is to create a list of hypotheses that can lead us to make an educated “guess” about the price movement the next day.

4.2 Bitcoin price history

Based on Metcalfe’s law, the value of a network is proportional to the square of its number of members. For Bitcoin, it just means that the dollar value attributed to a bitcoin is directly related to the total number of unique users (or wallets). In the absence of concrete numbers related to the same, network transactions are used as a proxy for the number of users. Bitcoin mempool contains all the unconfirmed transactions on the network. With a lot of unconfirmed transactions, the mempool gets clogged.

4.3 Social media data

Twitter and Reddit are the major social media sources that we are considering for the analysis. Sentiment analysis needs to be performed on the incoming data, so that we can decipher the general perception among the enthusiasts related to bitcoin and cryptocurrency in general. The number of mentions will serve as the barometer for the interest and hype surrounding bitcoin, that should be an important daily indicator.

4.4 Google search trends

While it is generally a lagging indicator for bitcoin prices, a very high number of Google searches will point to a bubble and should lead to subsequent selling.

4.5 Oil price

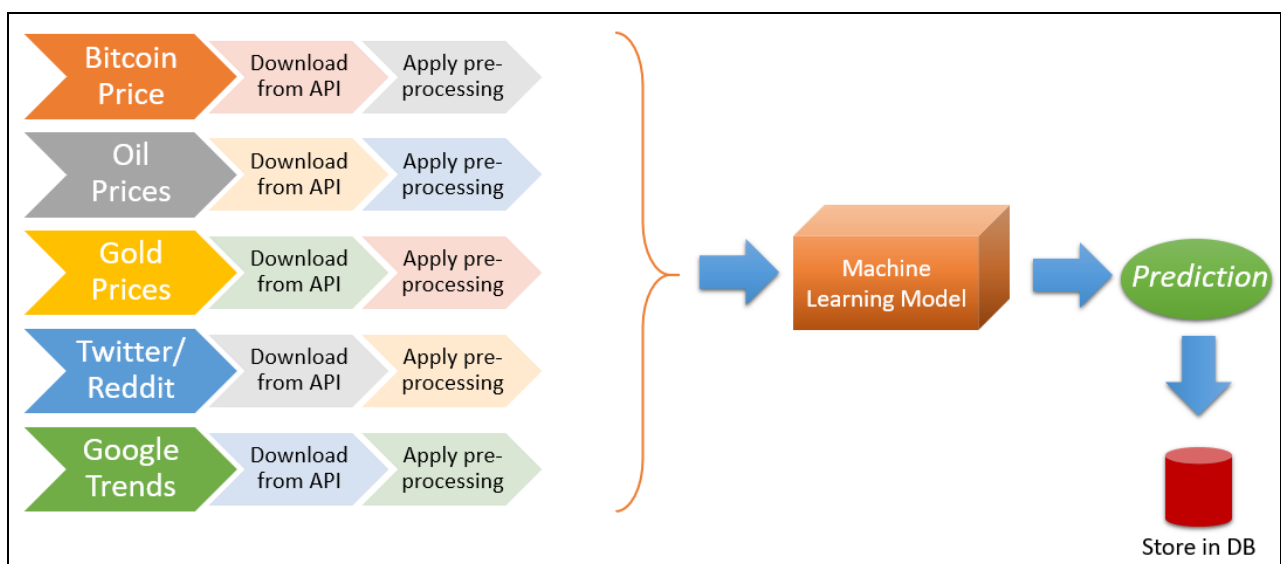
Increase in oil prices is correlated to general unrest in geopolitics. Bitcoin should have a positive correlation with the oil prices, as an increase in unrest signals a lowered confidence in the traditional financial centers.

4.6 Gold prices

Bitcoin is hailed as a digital gold, and there should be a relationship between gold prices and bitcoin. However, the relationship is much less clear and needs to be explored before deciding if it can be added as a predictive variable.

The ideal time-period after which the whole “cycle” will be run needs to be decided based on the consideration related to time for the weakest link.

5 DATA FLOW



6 DATA PREPROCESSING

6.1 Processing Reddit Data files

The data is in json format. The following fields are created by aggregating the data daily.

- # of daily reddit posts related to cryptocurrency.
- # of comments.
- # of new additions in past week in crypto subreddits.

6.2 Processing Twitter Data files

Historical data is in csv format. Current data obtained from tweepy API is in json format. Both data are processed to obtain the following aggregated metrics:

- Total # of crypto keyword mentions.
- Total # of tweets.
- # of positive and negative tweets after sentiment analysis and their ratio as well.

6.3 Auxiliary data

Google trend, Dollar forex rates, Oil and Gold prices. The data is present in csv format. We will consolidate and merge these data into a single auxiliary data file.

7 MACHINE LEARNING

For this component, we ingest the data from Data processing module. This module consists of two sub modules – Train and Test.

7.1 Train Module

This module comprises of Machine learning model, we'll be using sklearn to implement model architecture which would comprise of Model selection, Parameter Tuning and Cross validation.

7.2 Test Module

This module fetches the data for prediction using pre-trained model from Train module.

7.3 Programming Packages

We'll be using Sklearn and Statsmodel Packages from python to build our Machine learning pipeline.

[Scikit-learn](#) : Scikit-learn is a Python module for machine learning built on top of SciPy.

[Statsmodel](#) : Statsmodels is a Python package that provides a complement to scipy for statistical computations including descriptive statistics and estimation and inference for statistical models.

8 VISUALIZATION

This component interacts with the Data processing and ML module to make the results more comprehensible. The example visualization includes the following:

- A primary visualization showing daily price trend for Cryptocurrency.
- Sentiment Trend over the timeline based on the comments and texts obtained from Reddit and Twitter.
- Word cloud of the most important word features from text corpus.

We'll be using interactive plots based on Plotly for the Visualization module.

[PLOTLY](#) : is an interactive, browser-based graphing library for Python.

9 OTHER REQUIREMENTS

[Describe the non-behavioral requirements.]

9.1 Interface Requirements

[Describe the user interfaces that are to be implemented by the system.]

9.1.5 Hardware Interfaces

[Define hardware interfaces supported by the system, including logical structure, physical addresses, and expected behavior.]

9.1.6 Software Interfaces

[Name the applications with which the subject application must interface. State the following for each such application: name of application, external owner of application, interface details (only if determined by the other application).]

It is acceptable to reference an interface control document for details of the interface interactions.]

9.1.7 Communications Interfaces

[Describe communications interfaces to other systems or devices, such as local area networks.]

9.2 Data Conversion Requirements

[Describe the requirements needed for conversion of legacy data into the system.]

9.3 Hardware/Software Requirements

[Provide a description of the hardware and software platforms needed to support the system.]

9.4 Operational Requirements

[Provide the operational requirements in this section.]

Do not state how these requirements will be satisfied. For example, in the Reliability section, answer the question, “How reliable must the system be”? Do not state what steps will be taken to provide reliability.

Distinguish preferences from requirements. Requirements are based on business needs, preferences are not. If, for example, the user requires a special response but does not have a business-related reason for it, that requirement is a preference.

Other applicable requirements on system attributes may be added to the list of subsections below.]

Operational requirements describe how the system will run and communicate with operations personnel.

9.4.5 Security and Privacy

[Provide a list of the security requirements using the following criteria:

- A. State the consequences of the following breaches of security in the subject application:
 - 1. Loss or corruption of data
 - 2. Disclosure of secrets or sensitive information
 - 3. Disclosure of privileged/privacy information about individuals
 - 4. Corruption of software or introduction of malware, such as viruses
- B. State the type(s) of security required. Include the need for the following as appropriate:
 - 1. Physical security.
 - 2. Access by user role or types.
 - 3. State access control requirements by data attribute. For example, one group of users has permission to view an attribute but not update it while another group of users has permissions to update or view it.
 - 4. State access requirements based on system function. For example, if there is a need to grant access to certain system functions to one group of users, but not to another. For example, "The system shall make Function X available to the System Administrator only".
 - 5. State if there is a need for certification and accreditation of the security measures adopted for this application]

The Security Section describes the need to control access to the data. This includes controlling who may view and alter application data.

9.4.6 Audit Trail

[List the activities recorded in the application's audit trail. For each activity, list the data recorded.]

9.4.7 Reliability

- A. [State the following in this section:
 - 1. State the damage can result from failure of this system—indicate the criticality of the software, such as:
 - a) Loss of human life
 - b) Complete or partial loss of the ability to perform a mission-critical function
 - c) Loss of revenue
 - d) Loss of employee productivity
 - 2. What is the minimum acceptable level of reliability?

B. State required reliability:

1. Mean-Time-Between-Failure is the number of time units the system is operable before the first failure occurs.
2. Mean-Time-To-Failure is the number of time units before the system is operable divided by the number of failures during the time period.
3. Mean-Time-To-Repair is the number of time units required to perform system repair divided by the number of repairs during the time period.]

Reliability is the probability that the system processes work correctly and completely without being aborted.

9.4.8 Recoverability

[Answer the following questions in this section:

- A. In the event the application is unavailable to users (down) because of a system failure, how soon after the failure is detected must function be restored?
- B. In the event the database is corrupted, to what level of currency must it be restored? For example, “The database must be capable of being restored to its condition of no more than 1 hour before the corruption occurred”.
- C. If the processing site (hardware, data, and onsite backup) is destroyed, how soon must the application be able to be restored?]

Recoverability is the ability to restore function and data in the event of a failure.

9.4.9 System Availability

[State the period during which the application must be available to users. For example, “*The application must be available to users Monday through Friday between the hours of 6:30 a.m. and 5:30 p.m. EST.* If the application must be available to users in more than one-time zone, state the earliest start time and the latest stop time. Consider daylight savings time, too.

Include use peak times. These are times when system unavailability is least acceptable.]

System availability is the time when the application must be available for use. Required system availability is used in determining when maintenance may be performed.

9.4.10 General Performance

[Describe the requirements for the following:

- A. Response time for queries and updates
- B. Throughput
- C. Expected rate of user activity (for example, number of transactions per hour, day, or month, or cyclical periods)

Specific performance requirements, related to a specific functional requirement, should be listed with that functional requirement.

9.4.11 Capacity

[List the required capacities and expected volumes of data in business terms. Do not state capacities in terms of system memory requirements or disk space—if growth trends or projections are available, provide them]

9.4.12 Data Retention

[Describe the length of time various forms of data must be retained and the requirements for its destruction.

For example, “The system shall retain application information for 3 years”. Different forms of data include: system documentation, audit records, database records, access records.]

9.4.13 Error Handling

[Describe system error handling.]

9.4.14 Validation Rules

[Describe System Validation Rules.]

9.4.15 Conventions/Standards

[Describe system conventions and standards followed.

For example: Microsoft standards are followed for windows, Institute of Electrical and Electronics Engineers (IEEE) for data formats, etc.]

APPENDIX A - GLOSSARY

[Define terms, acronyms, and abbreviations used in the FRD.]