**A Dissertation Interim Report on**

**Trading Strategy in Cryto Currency Market**

**Daniel Sapkota**

**ID: 16401960**

**BSc Computing**

**Supervisor: Himalaya Kakshyapati**

**NAMI College**

**University of Northampton**

**Jorpati, Kathmandu, Nepal**

Table of Contents

[1. Introduction 4](#_Toc503720602)

[1.1. Introduction to the Topic 4](#_Toc503720603)

[1.2. The Problem Domain 4](#_Toc503720604)

[1.3. Hypothesis 5](#_Toc503720605)

[2. Background 5](#_Toc503720606)

[2.1. Forward Propagation 5](#_Toc503720607)

[2.2. Deep Neural Network 6](#_Toc503720608)

[2.3. Cost Function 6](#_Toc503720609)

[2.4. Backpropagation 7](#_Toc503720610)

[2.5. Variance and Bias 8](#_Toc503720611)

[2.6. Regularization and Dropout 8](#_Toc503720612)

[2.7. Vanishing Gradients Problem 8](#_Toc503720613)

[3. Literature Review - General 8](#_Toc503720614)

[3.1. Methods in the Stock Market 8](#_Toc503720615)

[3.2. Studies on Social Media Effects 9](#_Toc503720616)

[3.3. Studies focusing on Bitcoin Market 9](#_Toc503720617)

[4. Literature Review – Machine Learning 10](#_Toc503720618)

[4.1. Stochastic Gradient Descent 10](#_Toc503720619)

[4.2. Mini Batch Gradient Descent 10](#_Toc503720620)

[4.3. Momentum, RMSProp and Adam Optimization Algorithm 10](#_Toc503720621)

[4.4. The Initialization Problem 11](#_Toc503720622)

[5. Aims and Objectives 12](#_Toc503720623)

[5.1. Aims 12](#_Toc503720624)

[5.2. Objectives 12](#_Toc503720625)

[6. Design 12](#_Toc503720626)

[6.1. Pseudocode 12](#_Toc503720627)

[6.2. Deep Neural Network Structure 13](#_Toc503720628)

[7. Methodology 13](#_Toc503720629)

[7.1. Exchange Data Collection 13](#_Toc503720630)

[7.2. Social Media Data 15](#_Toc503720631)

[7.3. Technical Data Calculation 16](#_Toc503720632)

[7.4. Machine Learning Backbone 17](#_Toc503720633)

[7.5. Initialization 17](#_Toc503720634)

[7.6. Forward Propagation 18](#_Toc503720635)

[7.7. Cost Function 18](#_Toc503720636)

[7.8. Backward Propagation with Regularization 19](#_Toc503720637)

[7.9. Adam Optimization 19](#_Toc503720638)

[7.10. Adding it all together 20](#_Toc503720639)

[8. Testing 20](#_Toc503720640)

[8.1. Current approach on Neural Network 20](#_Toc503720641)

[8.2. Testing Back Propagation 21](#_Toc503720642)

[8.3. Testing Algorithm Performance 22](#_Toc503720643)

[8.4. Testing Python Scripts 22](#_Toc503720644)

[9. Further Work 23](#_Toc503720645)

[10. References 24](#_Toc503720646)

[11. Gantt Chart 27](#_Toc503720647)

[12. Old Gannt Chart 29](#_Toc503720648)

[13. Appendix I – Meeting Logs 30](#_Toc503720649)

[14. Appendix 2 – Presentation Slides 32](#_Toc503720650)

# Introduction

The Initial proposal focused on the mechanisms which needed to be learned to create a prediction model and systems used to create such a model. This Interim Report focuses on Machine Learning as it is the biggest part of the Network.

Additionally, some changes have been made on the future Gantt Chart. The old schedule was successfully completed. However, some changes are made in the new schedule as new stuffs has been learned. The new Gantt Chart reflects the aims in a better way than the old one. This report contains the relevant information from the Proposal and adds other information with primary focus on Machine Learning (as per Gantt Chart schedule).

## Introduction to the Topic

In 2008, whitepaper about a decentralized digital currency that claimed to solve the problem of double spending was published. Double Spending had been the biggest constrain for the creation of a decentralized digital currency. Double Spending takes place when the same amount of currency is spent more than once. To solve this problem - Digital currencies used to rely on a server (Like Liberty Reserve, Perfect Money). Addition of a server added central point of failure. Bitcoin solved this problem by creating a public transaction ledger called blockchain and by using Proof of Work (PoW) system. A Bitcoin transaction occurs when calculations (work – thus PoW) is done to find a nonce (a random string) that results in a block (contains list of transactions) hash smaller than a given hash (called difficulty which is automatically calculated). When it is found, the block is added next to the previous blocks in a blockchain and distributed. Any blocks that contain a double spend transaction is rejected by the network when the blocks are distributed. The inner working of bitcoin is longer and beyond the scope of this thesis, so the cited materials should be considered for further information. (Antonopoulos, 2014; Nakamoto, 2009)

After the creation of Bitcoin, there has been several other implementations of Blockchain – with major as well as minor changes. These different coins are traded with each other like stocks.

## The Problem Domain

Most of the profitable trading strategies (in any market) are proprietary because there is no incentive to share them. Sharing them will make the methods unprofitable very fast. The Crypto Currency Market has been an extremely under explored topic (by academics).

Most Non-Academic research in this field has only focused on creating a trading algorithm and not on risk management. Most such strategies work when the market is bullish (rising). But a system that can profit from both bullish and bearish (falling) market with low risk (lower than the market average) is extremely uncommon. Computer based trading dominates wall street, but it is not as common in this market – leaving an opportunity.

This is an attempt to create a computerized trading system that is profitable in the crypto currency market by modifying strategies from the traditional market or by creating new ones.

## Hypothesis

This thesis attempts to prove the hypothesis that a programmatic trading system can be created on the crypto currency market to profit during any Market Trends. To prove this hypothesis this thesis will work on proving/disproving several other hypotheses:

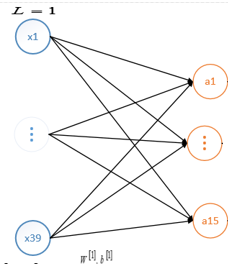
1. Patterns can be derived from past data using Machine Learning to find how likely coins can perform in similar circumstances.
2. Public Data from places like Social Media, News Sites and Google and Wikipedia Trends can help determine general sentiment in a coin.
3. Blockchain based data like transactions can help determining interest in a coin.
4. Technical Analysis data can be combined with Statistical Functions and other data to make a system better.
5. The general feeling of traders can be grabbed from Social Media. Also, Bots are used in Social Media to change general sentiment.
6. Features can be created from these data and be fed into a deep neural network to perform predictions
7. The output from the Deep Neural Network can be used to create a trading model.

# Background

Machine Learning is done on big sets of data to make sense of the data and provide predictions. Supervised Learning is the field of Machine Learning which deals with taking labelled data and using it to classify or perform prediction in unclassified data. Deep Learning – a supervised learning algorithm is being used in this dissertation to make predictions from huge set of data. Deep Learning uses a lot of concepts. The basics concepts are explained below while different algorithms are explained in Literature Review.

## Forward Propagation

The Forward Propagation Algorithm is an expansion of Logistic Regression. In logistic regression, an output classification is calculated by multiplying the features (measurable value that influence the classification outcome) with different weights. Weights are initialized with any random number (or by using some random generation method explained below) and they are modified with the aim of minimizing the total error. Derivative is used to minimize the error function. In Forward Propagation, Logistic Regression like function is used in different places of the Network.

+

As showing in the figure above, each input layer is multiplied by different values of Weights (W) in forward propagation. This is the general representation:

Zn = X1 \* W1n + X2 \* W2n + ….. + X39 \* W39n

A bias variable b is also added. There role is to shift the outcome of the function (on left or right). Once the value of Z is calculated, the value of Activation Function (A) is calculated. This is done to remove linearity from the function. There are many functions to remove linearity like Sigmoid Function, ReLU Function and others. In this dissertation, ReLU function is used whenever possible because it’s derivative can be calculated faster and efficiency is of enormous importance in Machine Learning.

The ReLU Function is:

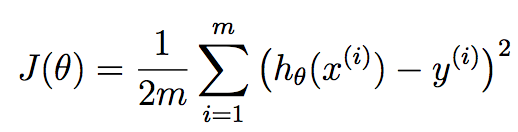
Max(0,Z)

## Deep Neural Network

Forward Propagation can be done for multiple steps. After calculating Forward propagation for a given set of inputs, we can use that output as input and do it again. Performing Forward (and backward) propagation in a deep network is slower but a deep network can show complex functions (Goodfellow, et al., 2016). In our model, our final output will be a number (which will likely denote the predicted price. But it cannot be certain as addition of new features can make a different approach better). So ReLU activation will be calculated on all layers except the final.

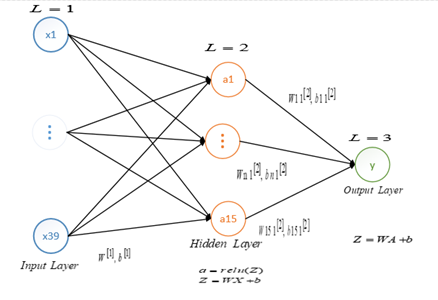
## Cost Function

The Cost Function is the sum of all errors in the model. In a regression model, where output is a number, the cost is the squared sum of all errors.

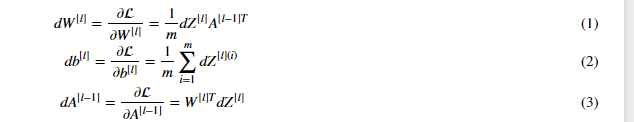


h(x) is the predicted value while y is the actual value in the figure above. Also, 1/2 is added because it is a constant and it makes calculation after taking derivative easier. The sum is taken of all data in the training set (data whose output and input are given) to get a single number as cost (Friedman, et al., 2009).

## Backpropagation



The purpose of backward propagation is to minimize the cost by making changes to the weights and move the network to an optimum. However, a deep network has a lot of weights which need to be changed so the process is longer. Error contribution of each weight is calculated by checking how off the prediction is.



For backpropagation in a regression based model, the differences between the predicted value and actual value is first calculated (which is the derivate of loss with respect to the last layer). Then it used to calculate the derivative of loss function (single cost) with respect to W in the last layer. That is used to calculate the derivate of loss with respect to b and finally the activation of last layer. Then that is used to calculate dZ (It is equal to dA of the previous layer times the derivative of ReLU function in that layer. The derivative of ReLU is undefined at exactly 0 (which is rare), 0 at X<=0 and 1 at X > 0). Then this process is repeated till the first layer. For the interpretation of these, (Goodfellow, et al., 2016; Friedman, et al., 2009) should be consulted. After calculating this the gradients (Z, b) are updated.



While updating, the value is multiplied by alpha. Alpha is a hyper parameter whose value can increase or decrease the learning rate. Making it too high will cause the network to miss optima and making it too low will make the learning too slow. Its value is different for every model and should be tuned with careful testing.

## Variance and Bias

We train a Neural Network by providing it inputs and outputs. After training a network for a certain number of iterations (which is another hyperparameter which we must tune) we send untrained data in the network to see how well it can perform in it. If the network excellently well on trained data but performs poorly on unseen data – we call it Variance problem. To get rid of it, we change value of the hyperparameters (like the alpha value, number of hidden layers, number of layers in hidden layer, number of iterations and other tunable values). We can also perform Regularization and other techniques like Dropout (explained below) (Friedman, et al., 2009).

Similarly, when the network performs poorly in both training and test set, we must add new features, make the model more complex and tune the hyperparameters. This is the High bias problem (Friedman, et al., 2009).

## Regularization and Dropout

Regularization is a technique to reduce over fitting. In regularization, the cost is increased by adding lambda (a regularization hyper parameter – which needs to be tuned) times the sum of parameters to the cost function (and thus in its derivative too). Adding it will make the gradients small (as cost will be larger and this value cannot be changed, the value of parameters will be changed). And mathematically, it has been proved that having small parameters works great in reducing overfitting (Friedman, et al., 2009).

Dropout is another technique to reduce cost function. The idea of dropout is to random drop units in the layer. Doing this prevents units from co depending too much and reduces over fitting (Srivastava, et al., 2014).

Like hyperparameters it is hard to know what works best for a given network. So, one of them or their combination will be tried to prevent overfitting when it occurs.

## Vanishing Gradients Problem

In some cases, some gradients (value of weight) will become very small. This can lead to stop training as multiplication keeps decreasing the value. This is the Vanishing Gradients Problem.

# Literature Review - General

There has been attempts to “beat the market” as soon as the Financial Markets started. Speculative Investors have tried to beat the market for centuries. In initial days, speculation was most of the daily volume (Andersen, 2012). Jesse Livermore, a legendary stock investor used to remember pattern’s and use them before computers was invented in the 1920’s (Lefevre, 2004).

## Methods in the Stock Market

Time Anomaly like January Effect (Increase in stock prices during January) has been observed in previous works on stock markets. These Anomalies can exist in crypto markets too and because it is active across multiple time zones, there might be distinct characters for certain time zones. These Anomaly create trading opportunities.

Neglected Firm Effect (Lesser Known Firms producing abnormally high return) is another anomaly that can be easily imported to the crypto currency Market. Here too the smallest coins have gained the most in times (Bartos, 2015). Most of the public Machine Learning systems using these features in the US Stock Market have not achieved better returns but have achieved lower Standard Deviation (The measurement of Risk) than the S&P 500 Index. Using Standard Deviation as a measurement of risk is not always the best because it treats upside and downside the same. Other systems Like Value at Risk and Omega Ratio exist if the need is different. Because there are a lot of features, Kahn Processing Networks have been used in previous studies to perform parallel calculations. In most of the studies in the US Stock Market, SVN performed better than ANN (Andersen, 2012).

## Studies on Social Media Effects

Another Important place where features (to predict the Crypto Currency price) can be obtained is Social Media. The possibility that Social Media is being used to manipulate public opinion has been proved by various studies (Ratkiewicz, et al., 2011; Ferrara, 2015). For manipulations, bots (explained in Appendix 4) are typically used. BotOrNot is a framework developed for detecting if a twitter account is bot or not using 1000 features. It gives an account score between 0 and 1 where a score above 0.5 can indicate the account is a bot. The features with biggest influence were - Absence of geographical meta data, Activity statistics, Account creation date (bots had more recent accounts) and the Randomness of username. Bot or Not is around 95% accurate. (Davis, et al., 2016)

SentiStrength is a python module which assigns positive and negative polarity score between one and five for a given text. The sentiment score is difference between them where 0 indicates neutrality (Thelwall, et al., 2010). They can be used on reddit and twitter data to find out what mood bots are trying to spread and that can be used to profit. Some features for the model can be grabbed from here. A use of BotOrNot and SentiStrength in the 2016 Election Twitter Data showed that most of the positive bots were supporting Trump. It reviewed 20.7 million Tweets (Bessi & Ferrara, 2016). The value of Social Media can be seen by the fact that A Political Company in Mexico paid 80k $ in election campaign to create 22 trending topics in Twitter (Forelle, et al., 2015). During the 2016 US Election Campaign, The Trump Campaign worked with Cambridge Analytica, a firm whose investor is Robert Mercer - the CEO of Renaissance Technologies mentioned in Appendix 3 for Social Media boosting which probably involved the use of bots (Mayer, 2017).

## Studies focusing on Bitcoin Market

Studies using trivial features like Daily price change, Close Price, Volume, Average, SD, RSI (A technical indicators), Daily Return, % Oscillation, Average true range, Vortex and On Balance Volume have performed better than buying and holding in case of Bitcoin. A Machine Learning algorithm using these data had lesser draw down and greater return than buy and hold strategy (22.88 DD and 33.52 return compared to 4.86% return and 30.9 DD) (Żbikowski, 2016). Although the timeframe is small to conclude anything, the promises are bright. Another study used Block time, Difficulty, Hashrate, no of transactions (non market data), trade volume and transaction to trade ratio as features using Binomial Logical Regressing, SVN and Random walk to get considerable returns. Random walk turned out to bring the best results. (Madan, et al., 2015).

Studies have found significant correlation between google trend and bitcoin price and the volume of positive tweets (Matta, et al., 2015; Georgoula, et al., 2015). Statistical functions like Wavelet Coherence Analysis and Fourier Spectrum have also been used independently (Kristoufek, 2015; Bouoiyour, et al., 2016). Such result shows that addition of other variables like manipulation factors, social media sentiment, altcoin movement brings huge promises.

# Literature Review – Machine Learning

In this sections, literatures and algorithms created from them that is relevant for the Machine Learning purpose is explained.

While calculating the cost function, the algorithm shown in background calculate sum over all example. It is computationally expensive process. That algorithm is Batch Gradient Descent. Modifications has been done in that algorithm to create other algorithms which are faster and in a lot cases more accurate too.

## Stochastic Gradient Descent

For large scale problems, stochastic gradient descent performs extremely well. In stochastic gradient descent while finding the cost, instead of summing all the dataset at each iteration, only a single dataset is used. This makes the process much faster. Using Stochastic Gradient Descent, the data may not find the global optima but it will reach around there and the difference will not be high. Other algorithms are made based on this logic which will help performing better in our solution (Bottou, 2010)

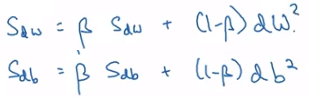
## Mini Batch Gradient Descent

In Mini Batch gradient descent, batches of a particular size is used while calculating the cost instead of everything or a single data. The batch size will be another hyper parameter. Mini Batch gradient descent can also help solving other problems like unwanted values (combined cost is low but all data are far for needed) (Konečný, et al., 2016).

## Momentum, RMSProp and Adam Optimization Algorithm

During Mini Batch Gradient Descent, the data will oscillate too much while converging. Momentum uses old gradients to create a smoother movement in the vertical direction by using the Exponentially Weighted Averages on the old steps. If we just take average, all data will have the same influence on the output but in Exponentially Weighted average, newer the data is, more influence it will have in the weight (Qian, 1999).

RMSProp is used in Mini Batches while trying to decrease oscillation in the vertical direction when it is converging to the horizontal direction (Hinton & Tieleman, 2012). Initially the value of S as shown below is zero.

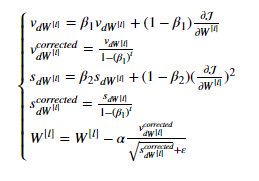


(Ng, 2017)



(Ng, 2017)

So, we create big value, Sdw and a smaller one Sdb to increase progress in one axis while decreasing in another. Beta in the figure above can be tuned as a hyper parameter but a constant value is also common as show by (Kingma & Ba, 2014).



(Ng, 2017)

Adam Optimization is a power algorithm that combines both RMSProp and Momentum and gives good results fast. Adam optimization algorithm will possibly be used on features in this project (unless other give better result). Adam algorithm is the combination of both Momentum and RMSProp. The calculations used in it is shown above. Initially, the velocity v and s are both initialized to zero (Kingma & Ba, 2014). (Kingma & Ba, 2014) recommend 0.99 to be the value B2 and 0.9 is the common choice for B1 (Ng, 2017).

## The Initialization Problem

Using some different values of weights initialization can cause a huge difference in the final solution. In “He Initialization”, the random number is multiplied by √(2/dimension of previous layer) during initialization. This initialization technique solves the problem of Vanish gradient problem as the value of w is a huge factor in creating the vanishing gradient. If the value never reaches there, the vanishing gradient problem will not take place (He, et al., 2015).

(Koturwar & Merchant, 2017) pointed out that different problems have different method of initialization that works best. A system that uses statistic to determine the best initialization method is purposed.

# Aims and Objectives

## Aims

The Aim of this project is to create a trading system which can profit in both bullish and bearish markets.

## Objectives

The following objectives will be performed to reach the aim:

* Create a script to detect Sentiment in Reddit and Twitter for a coin.
* Find ways of determining the flow of news and the general sentiment conveyed by them.
* Track big movements across the blockchain
* Find Patterns from past data
* Find Correlation between the coins and find the optimal way to exploit them.
* Use basic data like transactions and search trends to find the change in interest
* Use Technical Analysis and Statistical Function as features for determining the Market Movement.
* Find time related correlation and how different markets react to an event
* Analyze News to find out what is happening
* Combine some data and use Machine Learning to forecast the price change
* Create models to find the best diversification and mange risks
* Create separate scripts for each task for easy management and troubleshooting

The target audience of the deliverable is someone with basic knowledge about programming, mathematics and the Internet who is willing to learn other topics. To understand the report well and repeat the results, someone with knowledge of Machine Learning is needed. Although knowledge of Crypto Currencies and Financial Markets is a huge plus, it can be acquired while the report is being read.

# Design

## Pseudocode

Python Script is started

Python Script grabs bitcoin and altcoin data from exchanges

Python Script starts grabbing relevant Social Media data

A Python script has been made to store relevant data

Python Script starts grabbing Historical News and Trend Data

Another python script has been used to collect and store relevant news titles and their sentiment data

Python Script starts grabbing blockchain data

Relevant blockchain based data is stored in another file through another python script

The different features are stored in pandas and normalized to same scale

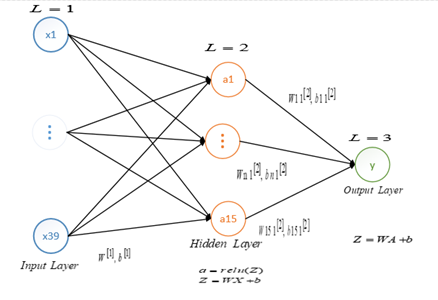
For a given coin, machine learning algorithm is started

Guesses are made and accuracy on untrained data is shown

Models are made to show the risk and how the investment will do on various Market Events.

## Deep Neural Network Structure

A Deep Learning system backbone that can create variable number of layers and hidden layers is created. As the features are tested and changed, the structure will change, but the basic structure is as follows:



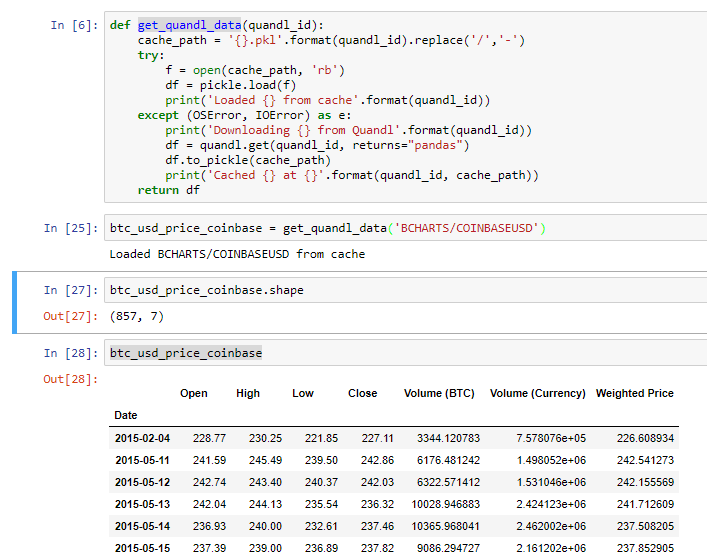
It is likely that the Number of features will be different. Also, the Number of parameters in the Hidden Layer will be different in the final model. The number of hidden layers might also be more as this figure just shows a single hidden layer.

# Methodology

Data has been collected from Exchanges through their API using python. (Ng, 2017) recommends creating an initial implementation of the deep learning model and making changes to it as the most effective strategy. So, an initial implementation has been completed as of today. All functionalities have been created done in Python.

## Exchange Data Collection

To collect Bitcoin data, quandl API in python is used.



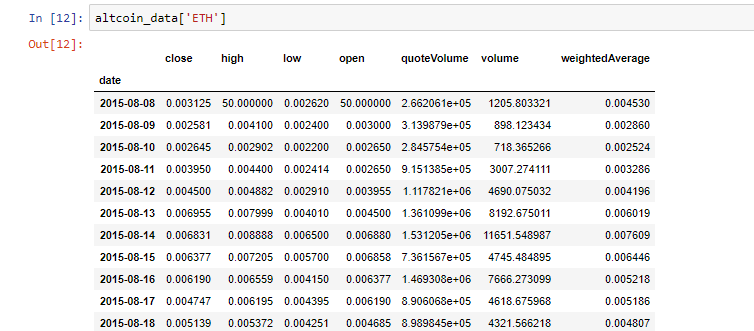
Data is currently collected from Bitfinex, Bittrex, Coinbase and Kraken and the average value is used.

For data of other altcoins, Poloniex API is used in python.

Similar function has been used to grab data from poloniex using their API



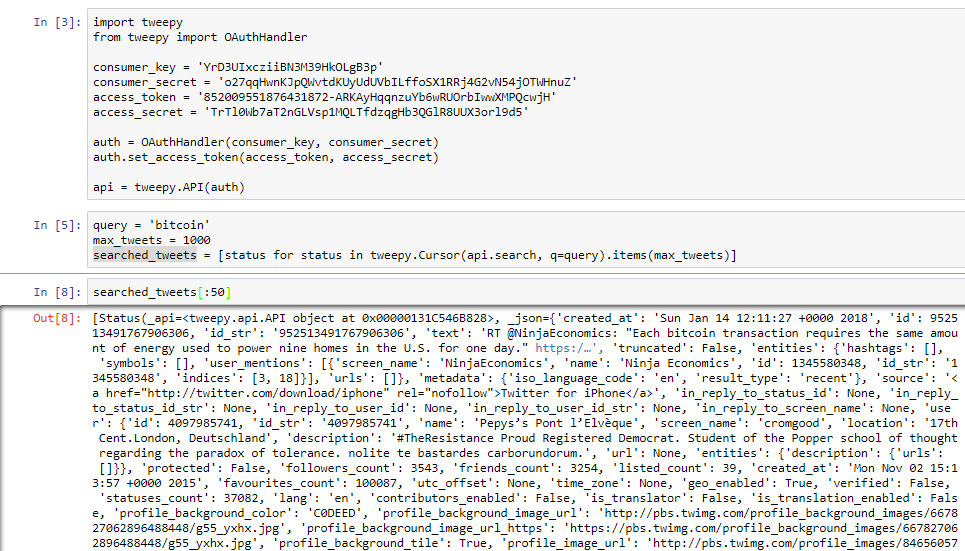
A function to download data and save it in cache for later purpose was created in output window 2. Then the Poloniex API end point was defined and the data was converted into a pandas dataframe inside a dictionary.



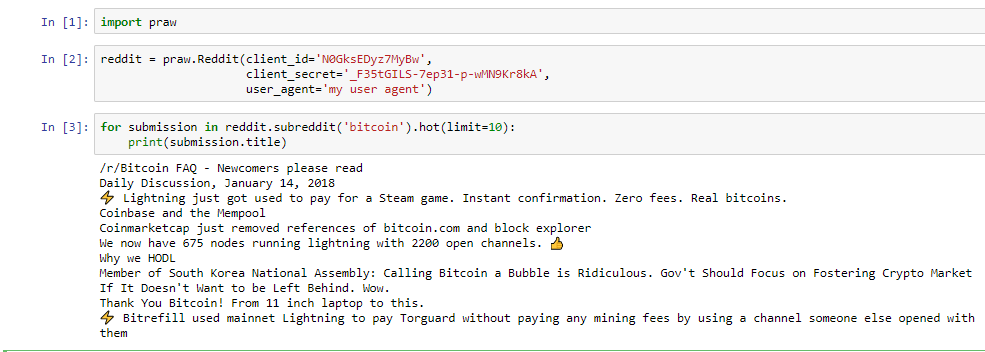
The data looked like this.

## Social Media Data

Tweepy module was used in order to grab twitter data after registration was made to Twitter API.



For reddit, praw module was used after registering to the Python API



Now the relevant information here in past will be searched in relevant subreddits and twitter. Then every day’s information will be converted into numbers which will be used in the model. The numbers will denote how much activity there was in the subreddit/twitter and the sentiment conveyed by them. Then other numbers will be created from bot activity based on the BotOrNot module explained above.

## Technical Data Calculation

The Technical Analysis information that is currently being considered is RSI. More indicators like Exponential Average can be added if adding these two gives better result in the Machine learning algorithm.

RSI = 100 - 100 / (1 + RS)

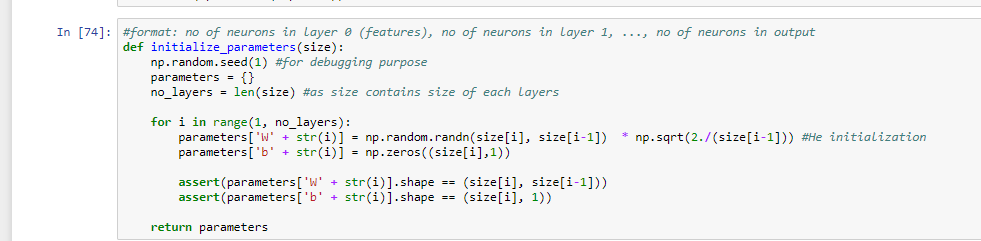
Where RS is the Average gain in up period or loss in down period. This shows the strength of uptrend or downtrend. RSI of different timeframe will be tested in the model to show which will give better result. (Levy, 1967)

Other such technical data will be added depending on how they affect the outcome.

## Machine Learning Backbone

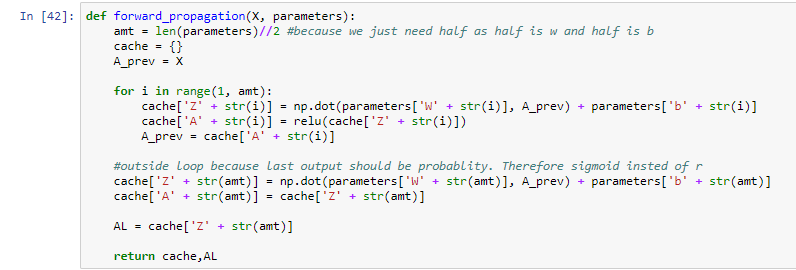
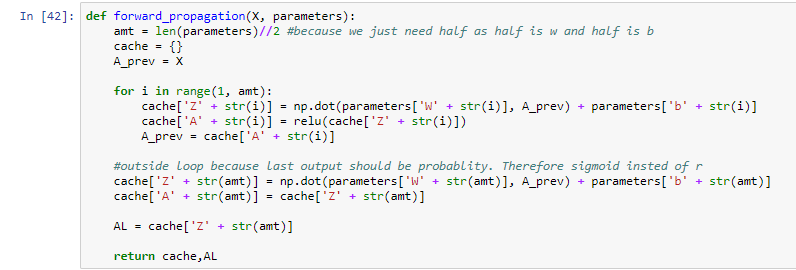
A Machine Learning algorithm to predict Bitcoin price has been created. The backbone will remain the same however changes will be made later by testing which algorithm gives the best result. The Network was based on stuffs learned from (Ng, 2017). This implementation gives linear output of predicted price. That may be changed to show percentage of price changed if that gives better results. A similar network was used on classification problem in (Sapkota, 2018)

## Initialization



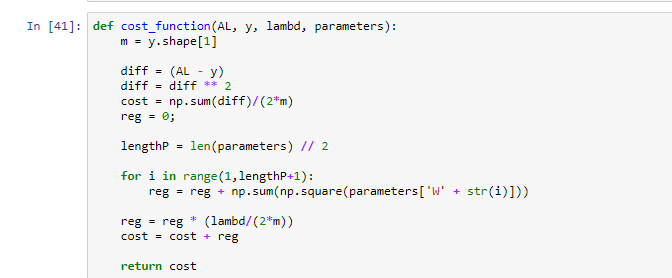
The size of the Neural Network is passed to the Initialization function. The parameter [4,2,4,1] denotes a Network with 4 input features which has 2 hidden layers of size 2 and 4 and a single output layer. Parameters W and b are created. He initialization is used to initiate W while b in initiated to zero.

## Forward Propagation



For forward propagation function, the input and the parameters are passed. Then vectorization is done to calculate the product of weights and inputs. Vectorized implementation is faster and easier than for loop implementations. ReLU activation is used in all layers except the last and the values are returned in a dictionary.

## Cost Function



The squared error cost function shown above is used with regularization. Later modifications may be made to use dropout instead and the regularization term may be dropped.

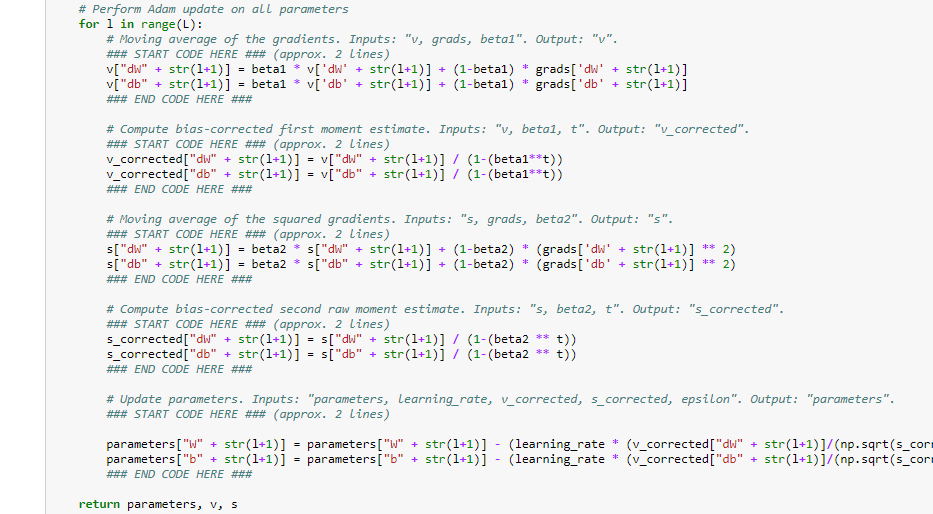
## Backward Propagation with Regularization



Although attempt was made to use as little for loop as possible, it had to be used during back propagation. The equations shown above has been calculated and gradients is returned.

## Adam Optimization

Adam Optimization was tested on a sperate application and has not been added in the main implementation yet.



Adam algorithm will be added and tested in the main network

## Adding it all together

Features will be created from the Social Media and other data collected above and it will be added as features in the Machine Learning Model. Then after prediction is made, tests results will be shown to how well the model will perform. Finally, to show how the model will do in various circumstance, hypothetical circumstances will be created from python.

# Testing

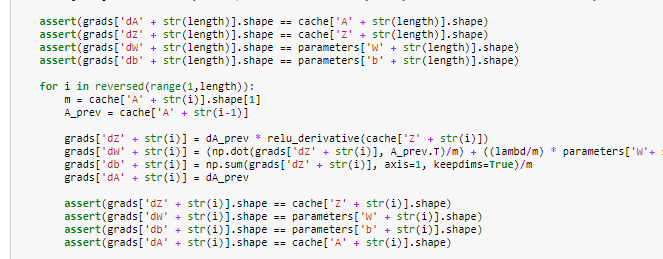
## Current approach on Neural Network

Currently, the dimensions of data are checked to find if there is any error in the model. The dimension of the derivates must be the same as that of initialized vector. Vectorization is done so if the dimensions are not equal an error will have been detected.

The dimension check is added in all functions:



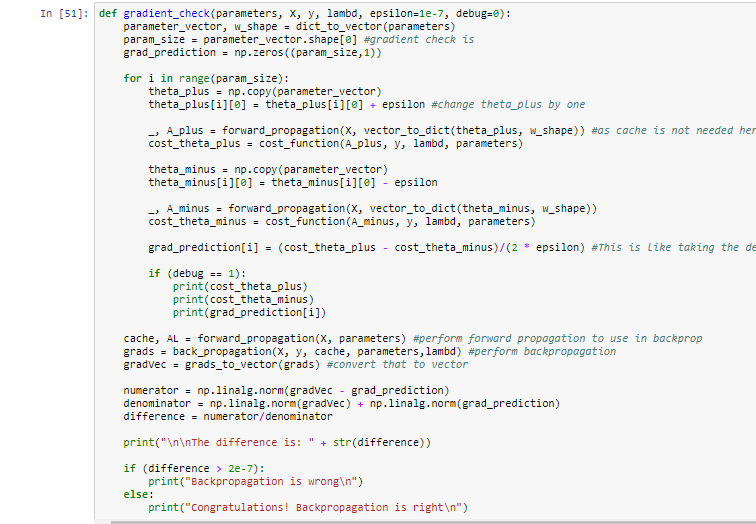
Initialization



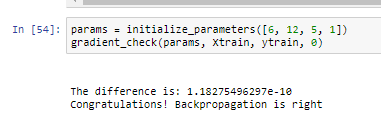
Backpropagation

## Testing Back Propagation

Backpropagation is a complex algorithm as is has many steps. Its goal is to minimize the weights. Gradient Checking Algorithm is used to verify back propagation. As derivative calculates change when a very small change is made in the function, gradient checking is used to make very small change in each of the weights. Then the relative difference is calculated.

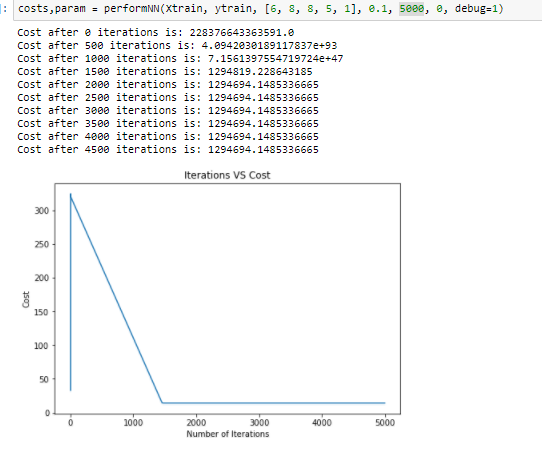


After gradient checking was done, a very small difference was observed.



## Testing Algorithm Performance

To calculate the accuracy of the algorithm, relative difference between the actual value and real value will be calculated on both training and test data. Also cost after every 1000 iterations is currently shown and it is graphed.



The results show that there is a problem and changes should be made in hyperparameters, the learning algorithm and the features. This test was done in 3 year data of Bitcoin.

## Testing Python Scripts

Python Unit Testing will be used in each scripts and functions to show if they are working well. The unit testing function is a built-in python function. Independent unit tests will be created.

# Further Work

First the learning algorithm must be changed, and new data should be added from chart history to get a better result. Technical analysis data also must be calculated. Then predictions should be done on altcoins too. Then the collected social media, news and other public data should be used to add them as features. Then changes should be made in the model depending on the output. Finally, after a satisfactory model is made (which does good on test) – risk calculations should be done to make changes.

# References

Andersen, A., 2012. *A Novel Algorithmic Trading Framework Applying Evolution and Machine Learning for Portfolio Optimization.* s.l.:(Doctoral dissertation, Doctoral dissertation, Masters Thesis, Faculty of Social Science and Technology Management, Department of Industrial Economics and Technology Management)..

Antonopoulos, A. M., 2014. *Mastering Bitcoin: Unlocking Digital Cryptocurrencies.* 1st ed. s.l.:O'Reilly Media.

Bartos, J., 2015. Does Bitcoin follow the hypothesis of efficient market?. *International Journal of Economic Sciences,* 4(2), pp. 10-23.

Bessi, A. & Ferrara, E., 2016. Social Bots Distort the 2016 US Presidential Election Online Discussion. *First Monday,* 21(11).

Bottou, L., 2010. *Large-scale machine learning with stochastic gradient descent.* Physica-Verlag, Springer.

Bouoiyour, J., Selmi, R., Tiwari, A. K. & Olayeni, O. R., 2016. What drives Bitcoin price. *Economics Bulletin,* 36(2), pp. 843-850.

Davis, C. A. et al., 2016. *Botornot: A system to evaluate social bots.* s.l., 25th International Conference Companion on World Wide Web.

Ferrara, E., 2015. Manipulation and abuse on social media by emilio ferrara with ching-man au yeung as coordinator. *ACM SIGWEB Newsletter,* p. 4.

Forelle, M. C., Howard, P. N., Monroy-Hernández, A. & Savage, S., 2015. *Political bots and the manipulation of public opinion in Venezuela.* [Online]   
Available at: https://ssrn.com/abstract=2635800  
[Accessed 4 11 2017].

Friedman, J., Hastie, T. & Tibshirani, R., 2009. *The Elements of Statistical Learning.* 2nd ed. New York: Springer-Verlag .

Georgoula, I. et al., 2015. *Using Time-Series and Sentiment Analysis to Detect the Determinants of Bitcoin Prices.* [Online]   
Available at: https://ssrn.com/abstract=2607167  
[Accessed 11 4 2017].

Goodfellow, I. J., Bengio, Y. & Courville, A., 2016. *Deep Learning.* 1st ed. s.l.:MIT Press.

He, K., Zhang, X., Ren, S. & Sun, J., 2015. *Delving deep into rectifiers: Surpassing human-level performance on imagenet classification.* s.l., Proceedings of the IEEE international conference on computer vision.

Hinton, G. & Tieleman, T., 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. In: *Neural networks for machine learning.* s.l.:Coursera, pp. 26-31.

Kingma, D. & Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint.*

Konečný, J., Liu, J., Richtárik, P. & Takáč, M., 2016. Mini-batch semi-stochastic gradient descent in the proximal setting. *IEEE Journal of Selected Topics in Signal Processing,* 10(2), pp. 242-255.

Koturwar, S. & Merchant, S., 2017. Weight Initialization of Deep Neural Networks (DNNs) using Data Statistics. *arXiv preprint.*

Kristoufek, L., 2015. What are the main drivers of the Bitcoin price? Evidence from wavelet coherence analysis. *PloS one,* 10(4), p. e0123923.

Lefevre, E., 2004. *Reminiscences of a stock operator.* s.l.:John Wiley & Sons.

Levy, R. A., 1967. Relative strength as a criterion for investment selection. *The Journal of Finance,* 22(4), pp. 595-610.

Madan, I., Saluja, S. & Zhao, A., 2015. *Automated Bitcoin Trading via Machine Learning Algorithms.* [Online]   
Available at: https://pdfs.semanticscholar.org/e065/3631b4a476abf5276a264f6bbff40b132061.pdf  
[Accessed 11 4 2017].

Matta, M., Lunesu, I. & Marchesi, M., 2015. *Bitcoin Spread Prediction Using Social and Web Search Media.* s.l., UMAP Workshops.

Mayer, J., 2017. *The Reclusive Hedge-Fund Tycoon Behind the Trump Presidency.* [Online]   
Available at: https://www.newyorker.com/magazine/2017/03/27/the-reclusive-hedge-fund-tycoon-behind-the-trump-presidency  
[Accessed 4 11 2017].

Nakamoto, S., 2009. *Bitcoin: A peer-to-peer electronic cash system.* [Online]   
Available at: https://bitcoin.org/bitcoin.pdf  
[Accessed 1 11 2017].

Ng, A., 2017. *Improving Deep Neural Networks: Hyperparameter tuning, Regularization and Optimization.* s.l.:Coursera.

Ng, A., 2017. *Structuring Machine Learning Projects.* s.l.:Coursera.

Qian, N., 1999. On the momentum term in gradient descent learning algorithms. *Neural networks,* 12(1), pp. 145-151.

Ratkiewicz, J. et al., 2011. Detecting and Tracking Political Abuse in Social Media. *ICWSM,* Volume 11, pp. 297-304.

Sapkota, D., 2018. *Artifical Intillegence Assignment.* s.l.:University of Northampton.

Srivastava, N. et al., 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research,* 15(1), pp. 1929-1958.

Thelwall, M. et al., 2010. Sentiment strength detection in short informal text. *Journal of the Association for Information Science and Technology,* 61(12), pp. 2544-2558.

Żbikowski, K., 2016. Application of Machine Learning Algorithms for Bitcoin Automated Trading. *Machine Intelligence and Big Data in Industry 2016,* pp. 161-168.

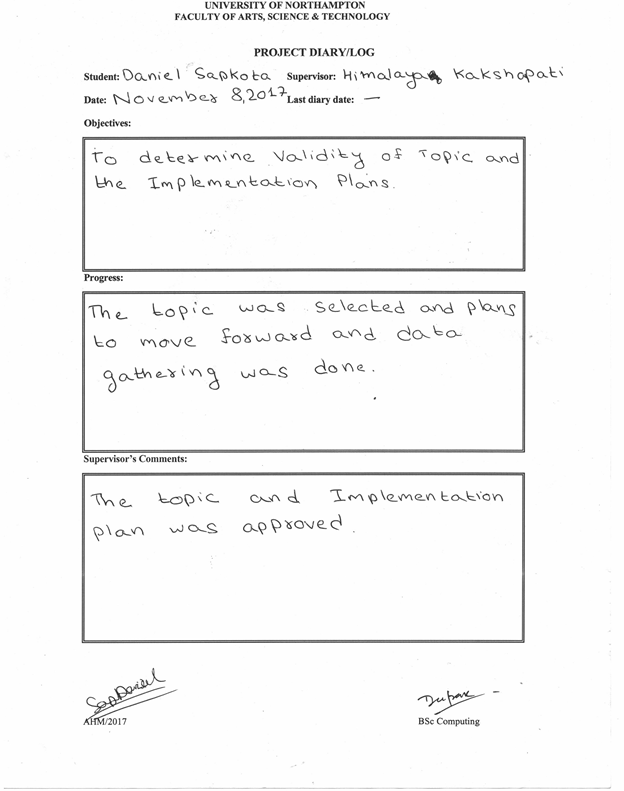
# Appendix 1 - Gantt Chart

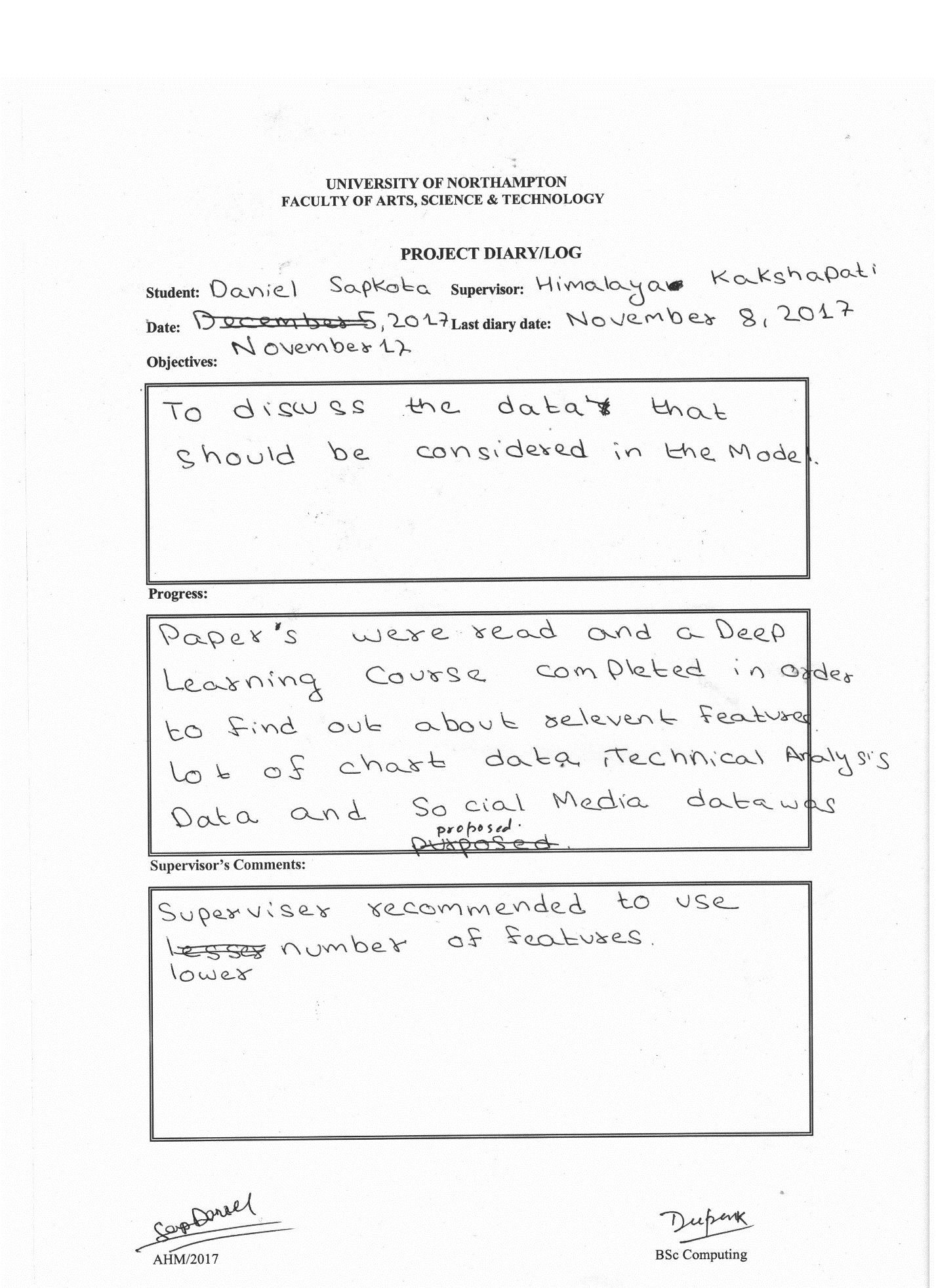
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task** | **November** | | **December** | | **January** | | **February** | | **March** | | **April** | |
| Learn More About Machine Learning. |  |  |  |  |  |  |  |  |  |  |  |  |
| Start Logging Data from Exchanges |  |  |  |  |  |  |  |  |  |  |  |  |
| Read Papers and Books on Machine Learning Models |  |  |  |  |  |  |  |  |  |  |  |  |
| Start working on creating and testing Features |  |  |  |  |  |  |  |  |  |  |  |  |
| Grab the Reddit and Twitter Social Media Data |  |  |  |  |  |  |  |  |  |  |  |  |
| Create and test features from old price data, correlations and Technical Data |  |  |  |  |  |  |  |  |  |  |  |  |
| Analyze the sentiment from twitter, reddit and google trends to create features based on them to add to the model |  |  |  |  |  |  |  |  |  |  |  |  |
| Test the Hyper Parameters |  |  |  |  |  |  |  |  |  |  |  |  |
| Finalize the Model and create the final script. Create forecasts and analyze risks |  |  |  |  |  |  |  |  |  |  |  |  |
| Fixing Bugs and Making the Thesis Better |  |  |  |  |  |  |  |  |  |  |  |  |

# Appendix 2 - Old Gannt Chart

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task** | **November** | | **December** | | **January** | | **February** | | **March** | | **April** | |
| Learn More About Machine Learning. |  |  |  |  |  |  |  |  |  |  |  |  |
| Start Logging Data from Exchanges |  |  |  |  |  |  |  |  |  |  |  |  |
| Read Papers and Books on Machine Learning Models |  |  |  |  |  |  |  |  |  |  |  |  |
| Start working on creating and testing Features |  |  |  |  |  |  |  |  |  |  |  |  |
| Grab the Reddit and Twitter Social Media Data and analyze them |  |  |  |  |  |  |  |  |  |  |  |  |
| Explore the Bitcoin Blockchain and grab features from it |  |  |  |  |  |  |  |  |  |  |  |  |
| Create the model |  |  |  |  |  |  |  |  |  |  |  |  |
| Improve the Model. |  |  |  |  |  |  |  |  |  |  |  |  |
| Fixing Bugs and Making the Thesis Better |  |  |  |  |  |  |  |  |  |  |  |  |

# Appendix 3 – Meeting Logs





# Appendix 4 – Presentation Slides

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |