Predicting the Price Movements of Bitcoin

1. Abstract:

Bitcoin is a form of digital currency known as a cryptocurrency, it was invented in 2008, by an unknown person/group of people who are now referred to by the name Satoshi Nakamoto.

Bitcoin is Decentralized meaning there is no single administrator or authority monitoring the network, it is monitored by a large peer to peer network.

Transactions are verified by this network and recorded on a public ledger known as the blockchain.

This paper explores the accuracy of predicting price movements of XBTUSD (bitcoin against the US Dollar) on the cryptocurrency exchange Bitmex, using machine learning methods.

This is achieved through use of Machine Learning classifiers namely Support Vector Machines, Decision Trees, Neural Networks, Random Forests & the K-Nearest Neighbour algorithm.

The Aim of this paper is to combine Machine Learning & Technical Analysis, in order to predict the price movements of bitcoin and assess the profitability of the Machine Learning models I have developed.

2. Introduction:

2.1 Motivation

Bitcoin is an extremely volatile asset, its price against the USD often fluctuates thousands of Dollars in a matter of hours. Predicting these moves could be very profitable.

This project aims to do just this and profit on the Margin of increase/decrease the market undergoes.

2.2 Problem Statement

At an abstract level, the goal of this project is to classify movement up or down as a '1' or a '0', respectively.

Then using these classifications the goal is to simulate trading on live data & hopefully profit off the margin by going short (predicting a decrease in price) or going long (predicting an increase in price) if the prediction is correct, we will profit off the margin.

2.3 Approach

The approach of this project was to utilise Technical Analysis which is a methodology for forecasting the direction of prices through the study of past market data.

I collected between 2 & 3 months of data in 1-minute intervals, the data consisted of the Opening price, Closing price, Highest price, Lowest Price & the total Volume over each 1-minute period.

Using this data, I generated a csv file with outputs from 12 technical indicators for each minute interval over the three months & trained various machine learning algorithms on these outputs in order to predict a '1' or a '0' indicating an expected increase or decrease, respectively.

Models were then tested not on their accuracy of predicting 1 minute ahead but instead on their profit if we were to use the '1' classification as a signal to buy/go long & the '0' classification as a signal to sell/go short.

2.4 Metrics

Model Accuracy:

The predictions when classifying up or down were compared against the actual movements to give an accuracy score.

Model Profit:

By simulating the flow of data as if it were live, I evaluated the profitability of each model on the three months of data & chose models which had the highest PV value. Models were trained on 30% of the dataset to avoid overfitting.

The PV value is a good estimate of the average returns as a percent of your investment.

$$\text{PV value} = \frac{\frac{Total\ Profit}{Number\ of\ Trades} \times 100}{Volume\ Traded}$$

3. Technical Background:

Technical Analysis

Technical Analysis is an approach to trading which was first introduced by Charles Dow with the Dow Theory in the late 1800's.

Assumptions of Technical Analysis introduced by Dow:

- Markets are efficient with values representing factors that influence a security's price.
- Even random market price movements appear to move in identifiable patterns and trends that tend to repeat over time.

There are three general assumptions that are widely accepted today by technical analysts:

1. The market discounts everything:
   This assumes that everything from a company's fundamentals to broad market factors such as market psychology are already accounted for in the price of a stock.

2. Price moves in trends:
   Prices tend to move in trends and are likely to continue a past trend rather than move erratically.

3. History repeats itself:
   The repetitive behaviour of a markets price movement is believed to reflect the market psychology.

How can Technical Analysis be used?

Using a broad spectrum of indicators that look at different features of the market.

This project utilises Volume & momentum indicators, Oscillators & moving averages.

A single indicator alone cannot predict market moves very well, but together they confirm each other and provide a more accurate prediction.

Encoding of technical indicators

Moving Averages:

- Simple Moving Average: An n-period moving average is calculated by averaging the last n minutes of data for each moment in time. This gives a good idea of trend.
- Exponential Moving Average: An n-period exponential moving average is calculated by averaging the last n minutes of data but weighting each data point exponentially for each moment in time. This gives a good idea of trend & is more sensitive to recent data.

With both the Simple and Exponential moving averages I generated signals with moving average cross detection. The idea is to pair moving averages together so one of them has a shorter time frame than the other. If the shorter moving average rises above the longer one it is a bullish signal that we should buy/ go long. Whereas if the shorter moving average crosses below the long moving average, it is a bearish signal to sell/ go short.

If no crosses occur 0.5 is output indicating no signal.

I used 10 Simple Moving Averages which leads to 45 inputs to the Machine learning algorithm at each step as $\binom{10}{2} = 45$.

Similarly, I used 10 Exponential Moving Averages leading to 45 inputs also.

Bollinger Bands:

Bollinger bands are calculated by first calculating the n-period moving average of your data, and then we calculate the following bands:

Upper_upper = (n-period SMA) + 2.4 × (Standard deviation of n-period SMA)

Upper_lower = (n-period SMA) + (Standard deviation of n-period SMA)

When In an uptrend the data usually falls between these two bands and will bounce off one band and back up to the other and so a long signal is generated if the data is within these bands.

If the data goes outside the Upper_upper band however we signal a sell signal as it is assumed the price will always tend towards the mean.

Lower_lower = (n-period SMA) - 1.8 × (Standard deviation of n-period SMA)

Lower_upper = (n-period SMA) - (Standard deviation of n-period SMA)

When In a downtrend the data usually falls between these two bands and will bounce off one band and back up to the other and so a short signal is generated if the data is within these bands.

If the data goes outside the Lower_lower band however we signal a buy signal as it is assumed the price will always tend towards the mean.

If the price lies between the Upper_lower & Lower_upper bands there is no obvious trend and so we output 0.5 as the price is within 1 standard deviation of the mean.

I used the Factors 2.4 & 1.8 above as cryptocurrency uptrends tend to be quite aggressive and wanted to avoid signalling a down signal for prices outside the Upper_upper band.

I used 10 different length Bollinger Bands in my Models, resulting in 10 inputs at each step.

Relative Strength Index:

A measure of overbought/oversold instruments.

Measures the magnitude of recent price changes.

$$\text{RSI} = (100 - \frac{100}{1+\left(\frac{Average\ Up}{Average\ Down}\right)})$$

$$\text{Average Up} = \frac{Total\ up\ movement}{number\ of\ up\ movements}$$

$$\text{Average Down} = \frac{Total\ down\ movement}{number\ of\ down\ movements}$$

RSI above 80 signals overbought so we signal to sell/go short.

RSI below 20 signals oversold so we signal to buy/go long.

I made use of 3 different period RSI indicators in my models.

Moving Average Convergence Divergence:

A trend following momentum indicator.

mLine = (short EMA) − (longer EMA)

sLine = EMA (mLine, period length)

Signals are generated from the mLine crossing over the sLine much like with the Moving Averages above.

mLine crossing over the sLine indicates bullish trend.

mLine crossing under the sLine indicates bearish trend.

I made use of 10 different length MACD indicators.

Accumulation/Distribution Line:

$$\text{CMFV} = \frac{(Close-Low)-(High-Close)}{(High-Low)} \times Volume$$

This is calculated at each period, if it is greater than the previous period this is bullish as Money flow volume is increasing, if it is less than the previous period then it is a bearish signal.

This generates 1 signal for our machine learning algorithm.

A/D Line = (old CMFV) + (new CMFV)

The A/D Line is an array of these CMFV values, Allowing for a look at wider time periods & use in other indicators such as the Chaikin oscillator.

Money Flow Index:

A measure of overbought/oversold instruments.

$$\text{Money Flow Index} = 100 - \frac{100}{1 + Money\ Flow\ Ratio}$$

Where:

$$\text{Money Flow Ratio} = \frac{14\ period\ Positive\ Money\ Flow}{14\ period\ Negative\ Money\ Flow}$$

$$\text{Positive Money Flow} = \begin{cases} 0, & if\ price\ decreased\ over\ period \\ (Raw\ money\ flow), & if\ the\ price\ increased\ over\ the\ period \end{cases}$$

$$\text{Negative Money Flow} = \begin{cases} 0, & if\ price\ increased\ over\ period \\ (Raw\ money\ flow), & if\ the\ price\ decreased\ over\ the\ period \end{cases}$$

$$\text{Raw Money Flow} = \frac{(High + low + Close)}{3}$$

MFI above 80 signals overbought so we signal to sell/go short.

MFI below 20 signals oversold so we signal to buy/go long.

This generates 1 signal for our machine learning algorithm.

Average Directional Movement Index:

The ADX is an indicator which helps determine trend strength.

If the ADX is above 25 this indicates a strong trend.

If the ADX is below 20 this indicates a weak trend.

If -DI crosses +DI & ADX>20 signal buy

If +DI crosses -DI & ADX>20 signal sell

I generate 10 signals from 10 different periods from the ADX.

The Aroon Oscillator:

A trend following indicator used to gauge the strength of a current trend.

Aroon Oscillator = Aroon Up − Aroon Down

$$\text{Aroon Up} = 100 \times \frac{(n - (Periods\ Since\ n\text{−}period\ high)}{n}$$

$$\text{Aroon Down} = 100 \times \frac{(n - (Periods\ Since\ n\text{−}period\ Low)}{n}$$

If the Aroon Up > Aroon Down, we signal a buy/ go long.

If the Aroon Up < Aroon Down, we signal a sell/ go short.

Otherwise, we signal 0.5.

We generate 10 signals from 10 different periods from the Aroon Oscillator.

On Balance Volume:

On-balance volume is a momentum indicator that uses volume flow to predict

changes in stock price.

OBV shows crowd sentiment that can predict a bullish or bearish outcome.

$$\text{OBV} = OBV_{prev} + \begin{cases} volume, if\ Close > Close_{prev} \\ 0, \quad if\ Close = Close_{prev} \\ -volume, if\ Close < Close_{prev} \end{cases}$$

We signal buy if OBV > OBV_{prev} .

We signal sell if OBV < OBV_{prev}.

We generate 7 signals from 7 different periods from the OBV.

The Stochastic Oscillator:

A stochastic oscillator is a momentum indicator.

It is used to generate overbought and oversold trading signals.

$$\%K = \frac{C - L\_n}{H\_n - L\_n} \times 100$$

Where:

C = The most recent Closing price

L_n = The lowest price traded over the last n-days

H_n = The Highest price traded over the last n-days

%k = current value of the stochastic oscillator

%D is the 3-period moving average of %K.

If %D & %K are both above 80 this is a signal for overbought and so we sell/ go short.

If %D & %K are both below 20 this is a signal for oversold and so we buy/ go long.

Otherwise, we signal 0.5 as the indicator is not saying anything definitive.

We generate 10 signals from 10 different period Stochastic oscillators.

The Chaikin Oscillator:

The Chaikin Oscillator is calculated by taking the 3-period EMA of the Accumulation/distribution line and subtracting the 10-period EMA of the Accumulation/distribution line.

Chaikin Oscillator = EMA (ADgraph,3) − EMA (ADgraph,10)

We generate 9 signals from the Chaikin oscillator.

Each coming from comparing the current Chaikin value to a value from n-periods ago, if the value now is higher, we signal a buy/go long, if its lower we signal a sell/go short.

Otherwise, we signal 0.5 as the indicator is not signalling anything.

4. The Problem:

-A large dataset of prices and volume was needed to undergo this project, I recorded between 2 & 3 months of data from the 30th of November 2020 to the 15th of February 2021, from the Bitmex exchange through a WebSocket client which I ran on an AWS server.

-Bitcoin is such a volatile asset that it is quite difficult to predict its movements accurately.

-To do this I made use of the technical indicators specified above:

- Simple Moving Averages
- Exponential Moving Averages
- Bollinger Bands
- Relative Strength Index
- Moving Average Convergence Divergence
- Accumulation/Distribution Line
- Money Flow Index
- Average Directional Movement Index
- The Aroon Oscillator
- On Balance Volume
- The Stochastic Oscillator
- The Chaikin Oscillator

-In order to train the machine learning algorithms, I need to generate signals for every minute of data I've collected & generate a csv file of these signals for use in training and evaluating.

-The Machine learning algorithms are prone to overfitting on the training data & so I made sure to only train on 30% or less of the dataset for classification but later experimented with this for profit.

-Rigorous testing on Live data is needed to confirm that the models will continue to perform well into the future.

5. The Solution:

5.1 Training the Machine Learning Algorithms:

A csv file is created from these 162 input signals along with the actual output we want from the Machine learning Algorithms & the Closing price at that moment.

Here is a sample input vector to the ML algorithms:

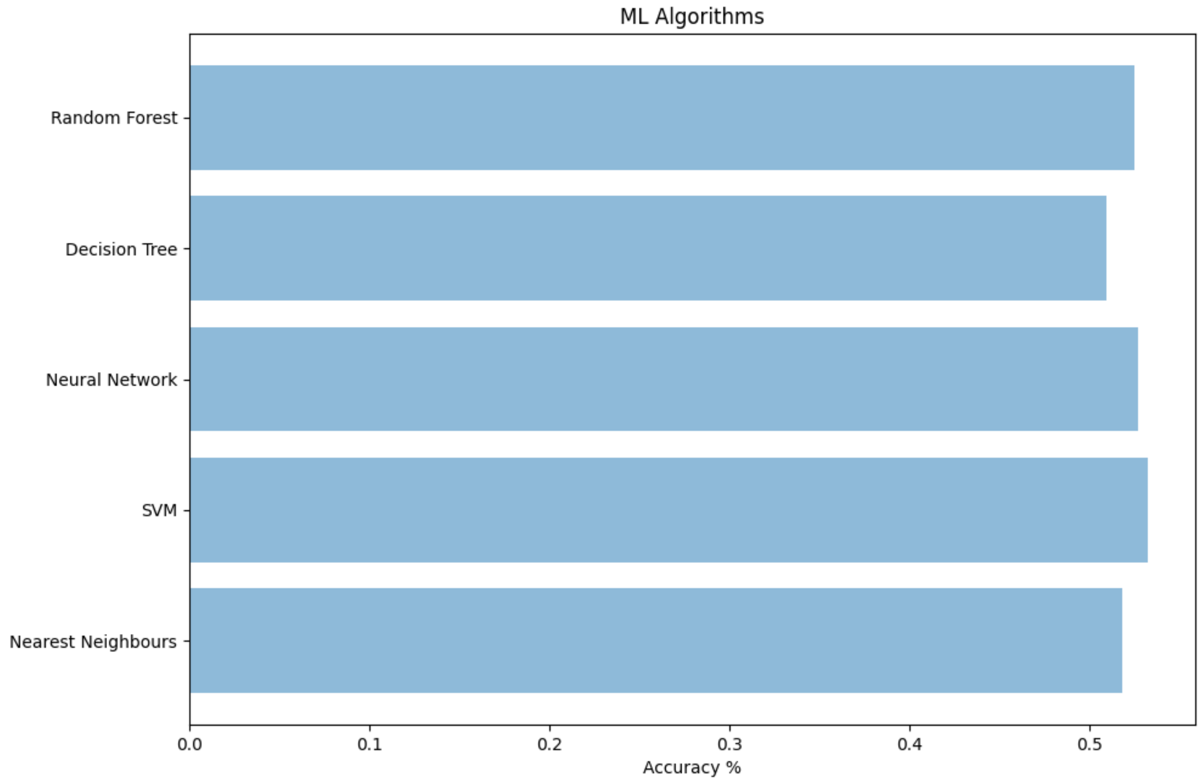
[1,1,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,0,0,.5,1,1,.5,.5,.5,.5,.5,1,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,1,1,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,1,1,1,1,1,1,0,1,0,.5,.5,.5,0,0,0,.5,.5,.5,.5,.5,.5,.5,0,1,1,0,0,0,.5,1,0,0,1,1,1,1,1,1,0,.5,.5,.5,.5,.5,.5,.5,.5,.5,.5,1,1,0,1,1,1,1,1,1,1,56238.10551]

I made use of Scikit-learn in order to train the machine learning algorithms. Using the train test split in the scikit learn package allowed me to keep the expected values separate to the input vectors avoiding data leakage. The price was also parsed off before feeding into the ML algorithms, this is only used in a later stage after the models have been trained to evaluate the profit.

To compare the different methods, I tested them on the same Train and Test sets, with 30% of the dataset allocated to training and the other 70% allocated to testing.

Classification Test Results:

Originally I tried predicting 1-minute ahead but the accuracy on classification was quite poor, here are the results:



traintype1results

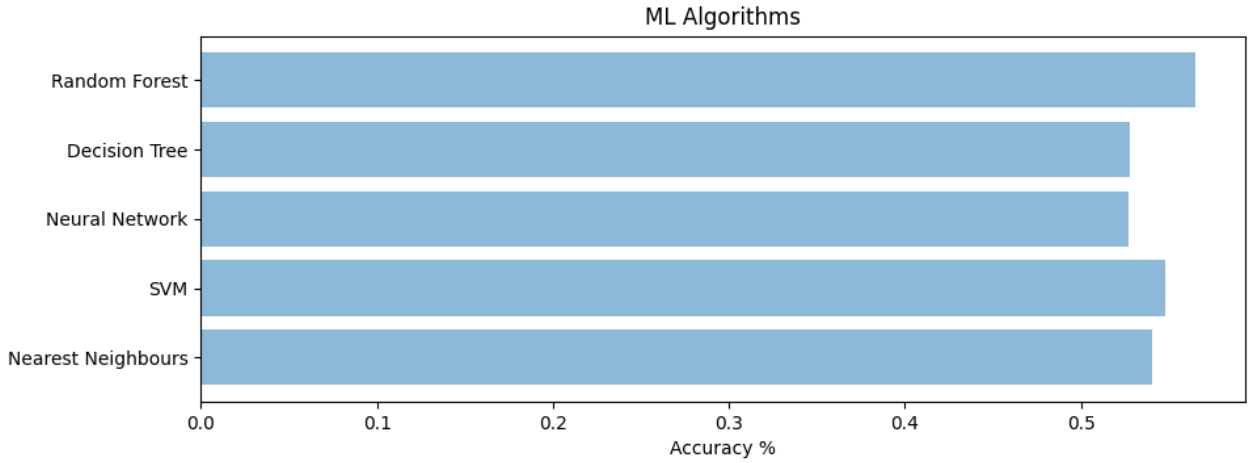
Confusion Matrices:

Nearest Neighbour:	Neural Network:	SVM:	Decision Tree:	Random Forest:
[[13596 18232]	[[13043 18785]	[[10513 21315]	[[15506 16322]	[[12947 18881]
[14387 21471]]	[13237 22621]]	[10343 25515]]	[16898 18960]]	[13263 22595]]
Accuracy: 51.8%	Accuracy: 52.69%	Accuracy: 53.22%	Accuracy: 50.92%	Accuracy: 52.51%

Predicting 5-minutes ahead:

The models performed much better on classification 5 minutes ahead.

The indicators seem to be suited to longer time periods.



train5minutestype2 results

Confusion Matrices:

Nearest Neighbour:	Neural Network:	SVM:	Decision Tree:	Random Forest:
[[18430 15937]	[[15191 19176]	[[17317 17050]	[[17964 16403]	[[19330 15037]
[16561 19750]]	[14277 22034]]	[14905 21406]]	[16969 19342]]	[15692 20619]]
Accuracy: 54.01%	Accuracy: 52.66%	Accuracy: 54.78%	Accuracy: 52.78%	Accuracy: 56.52%

5.2 Evaluating Machine Learning algorithms Profit & PV scores on the training Set:

To evaluate the profit & PV scores of the algorithms & for the results to have any chance of being consistant with live trading I needed to avoid overfitting, I trained the algorithms on 0.1% of the dataset with labels for 1 minute ahead to see if results were significant & not just a case over overfitting. This equates to 97 sample vectors out of 96695.

Then I simulated the flow of data from the collected data without shuffling, into the models and followed their classifications as signals for buying or selling. I only allowed one position to be open at a time, either a short or a long, so consecutive 1's or 0's are ignored. If the decision changed from short to long or vice versa, I calculated the profit and iterated the tradeNO counter, until the model had made it through the dataset.

Then I calculated the PV value as $\frac{\frac{total\ profit}{tradeNO} \times 100}{Price\ of\ a\ Bitcoin}$.

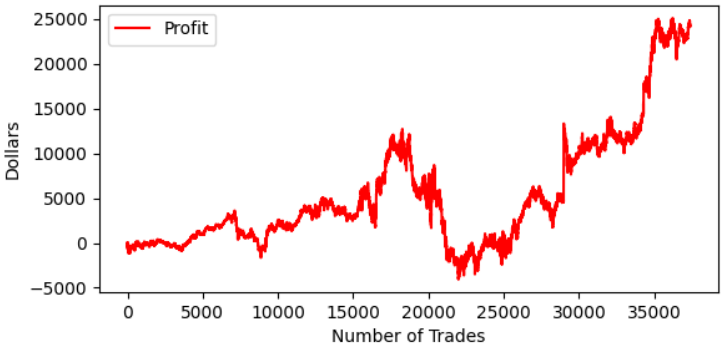
Results:

Here are the results of training on 0.1% then testing profits on whole dataset. (traintype1 was used for these)

Decision Tree:

Profit: 24159.0

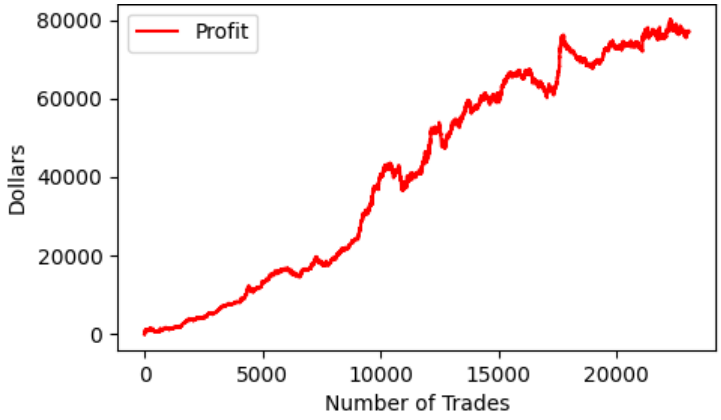
PV: 0.001076834617030381



SVM:

Profit: 77068.0

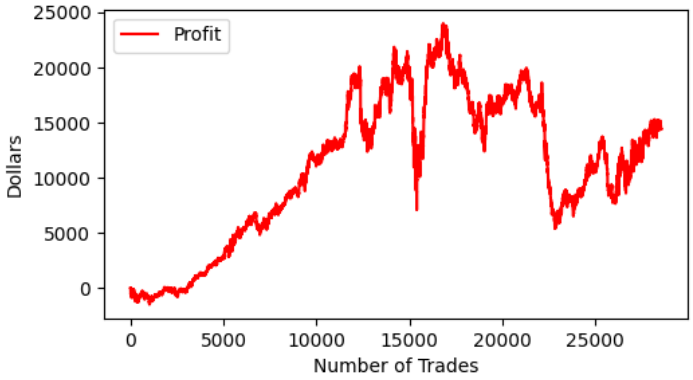
PV: 0.005567692529981216



15-Nearest Neighbour:

Profit: 14450.0

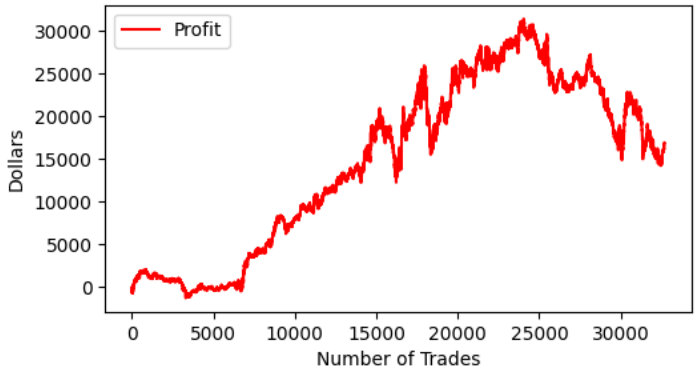
PV: 0.000843549328663164



Neural Network:

Profit: 16914.0

PV: 0.0008626334955169987



Random Forest:

Profit: 69265.0

PV: 0.003777665063211056