Machine Learning Advanced Nanodegree
Capstone Project Report

Crypto-Currency Price Forecasting

Swapnil Jadhav
18th Feb 2018

A. Definition

Project Overview

Cryptocurrency is an electronic money created with technology controlling its creation and protection, while hiding the identities of its users. Crypto is short for "cryptography", and cryptography is computer technology used for security, hiding information, identities and more. Currency simply means "money currently in use".

Cryptocurrencies are a digital cash designed to be quicker, cheaper and more reliable than our regular government issued money. Instead of trusting a government to create your money and banks to store, send and receive it, users transact directly with each other and store their money themselves. Because people can send money directly without a middleman, transactions are usually very affordable and fast.

In recent months, Cryptocurrency has become a hot topic and many big companies like Google and Microsoft and many banks started exploring solutions within this domain. Due to sudden increase in interest and also due to various government policies across countries, cryptocurrency market saw highly turbulent ups and down in the prices. Many experts suggested not to invest in non-mature system. But In my opinion, as a new technology this situation of distrust is bound to happen and some refined solution will come out in coming months.

Problem Statement

Mining with cryptocurrency has become difficult and return on investment is very less. Depending on usability/utility of cryptocurrency it can also be traded like any other commodity.

Recently there has been surge of currency exchanges which allows to buy and sell of cryptocurrencies (or simply coins). Though looks very similar to stock market, there are very few studies done on price variations and forecasting of the same.

Some of the famous cryptocurrency exchanges are -
http://coinbase.com/
http://cex.io/

In this capstone project following problem statement will be targeted to solve using Deep Learning methods.

Forecast the closing price of a particular crypto-currency for the next N days/hours given history of price variations for the same.

Few other mini problem statements can also be solved -

- Forecast price variation/volatility for each day
- Forecast price by hour/minutes instead of day-wise

In this project, only BitCoin(BTC) is taken for experimentation as data is not sufficient for many other crypto-coins.

Metrics

Performance of the algorithm can be checked with various matrices, but choosing following two for the convenience.

- Root Mean Squared Error (RMSE)
- If we convert problem to find only increase and decrease in next day as Classification ... then accuracy (Precision/Recall)

B. Analysis

Data Exploration

Collecting the useful data has been a problem for this project. There are three types of data that can be used in this project.

1. Market price data
2. Cryptocurrency related data
3. Social Media Analysis

And following is the brief explaining various issues and data used in this project.

Market price data

This data is very easy to get if we are considering only daily price data. Currently using "pandas_datareader" library to get cryptocurrency data from yahoo. Method is very easy and explained in "Data Generation" ipython file. The biggest problem is only 3-4 years of data that is ~1300 days of data is available and it is certainly not sufficient for any kind of deep learning project. Getting hourly or minute price data is not possible through almost any free API which tried.

So, data for this project is downloaded from kaggle.com where user "Zielak" has generously made it available. Complete details about the data and download links can be found at https://www.kaggle.com/mczielinski/bitcoin-historical-data/data.
This data consists of typical stock-market style data like opening-price, closing-price, high/low prices. Additionally, volume of bitcoin and volume of currency is also given at the minute level.

Cryptocurrency related data
Blockchain.com provides free APIs to get cryptocurrency specific data which could be very useful in forecasting.
Examples of such data points are as follows and names would suggest their intended meanings -

- n-transactions
- transactions-per-second
- hash-rate
- output-volume
- mempool-size
- market-cap
- cost-per-transaction-percent
- utxo-count
- bitcoin-unlimited-share
- n-transactions-excluding-popular
- median-confirmation-time
- n-orphaned-blocks
- mempool-growth
- cost-per-transaction
- blocks-size
- transaction-fees
- estimated-transaction-volume-usd
- avg-block-size
- estimated-transaction-volume
- mempool-count
- nya-support
- bip-9-segwit
- miners-revenue
- difficulty
- trade-volume
- total-bitcoins
- n-unique-addresses
- n-transactions-excluding-chains-longer-than-100
- my-wallet-n-users
- n-transactions-total
- transaction-fees-usd
- n-transactions-per-block

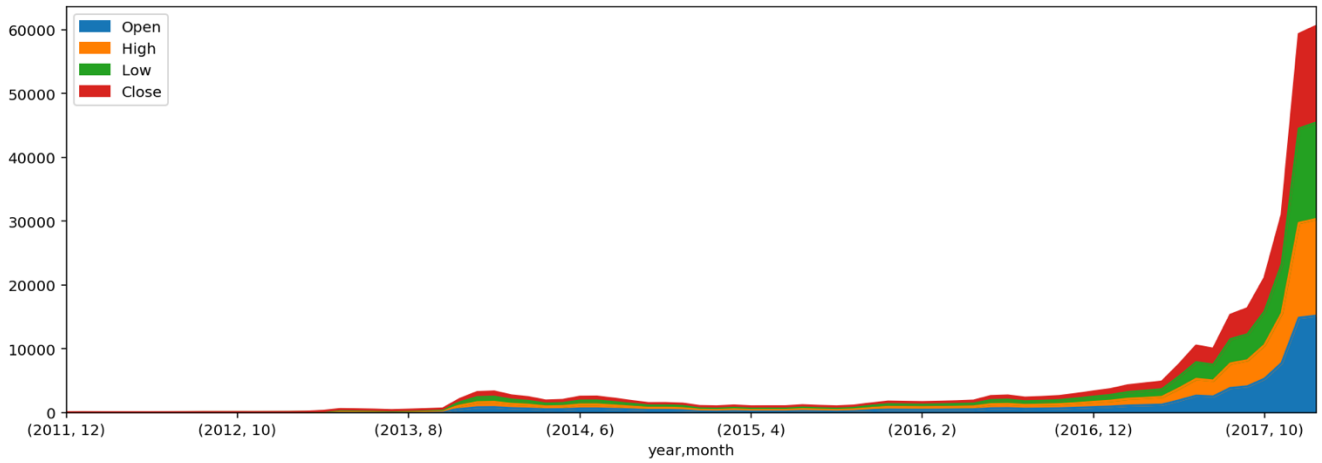
But the problem with the data is again it's at Day level instead of minute or hour level. Hence, could not use this data in main experimentation.

Social Media Analysis
Social interactions on twitter, Facebook and reddit can drive prices of the cryptocurrencies and stock market. In recent months, any news like government ban on cryptocurrency or heavy investment in certain crypto-coins have seen drastic effects in short span of time.
If properly accumulated and analysed this social media data could give a good trigger point to such trending events. In this project, due to time constraints and lack of free APIs, this data is not used.

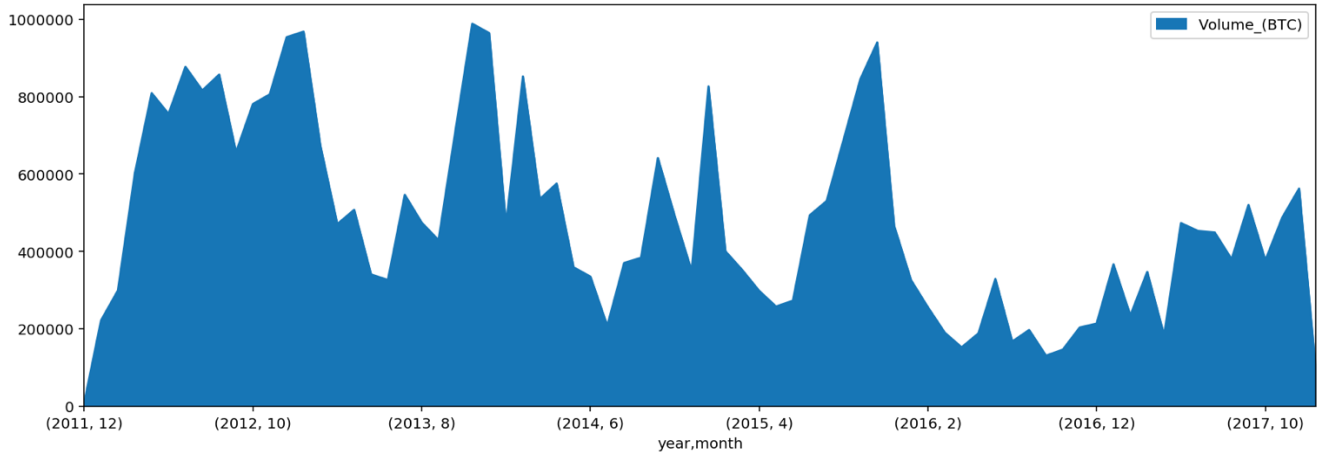
Exploratory Visualization

Data analysis is done in "Data Analysis" ipython file which can be referred for detailed analysis and visualization. Following are few graphs showing some trend analysis, seasonality checks.

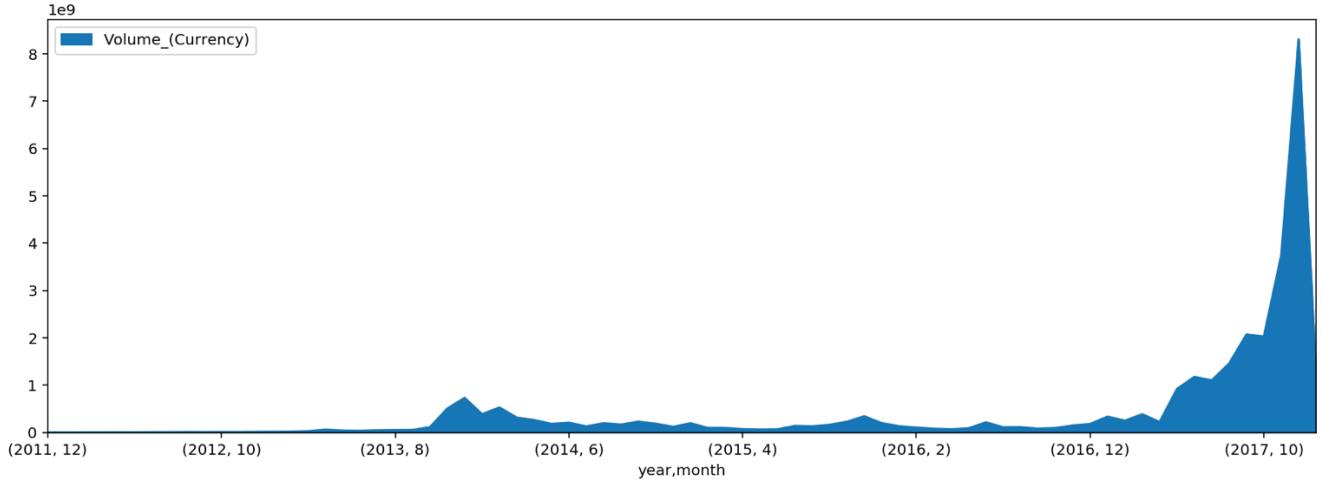
Price Trends



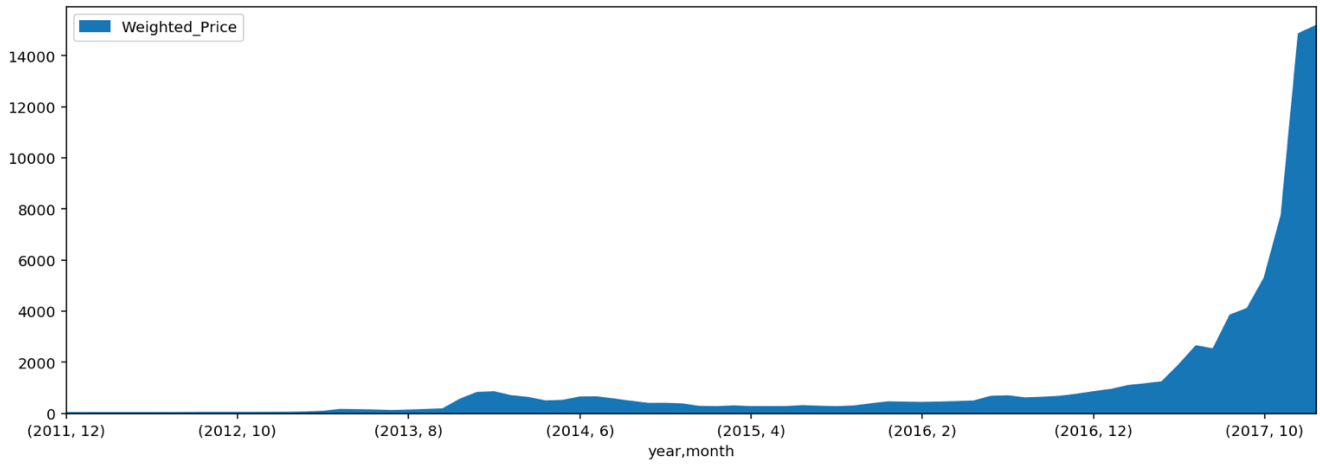
BitCoin Volume Trend



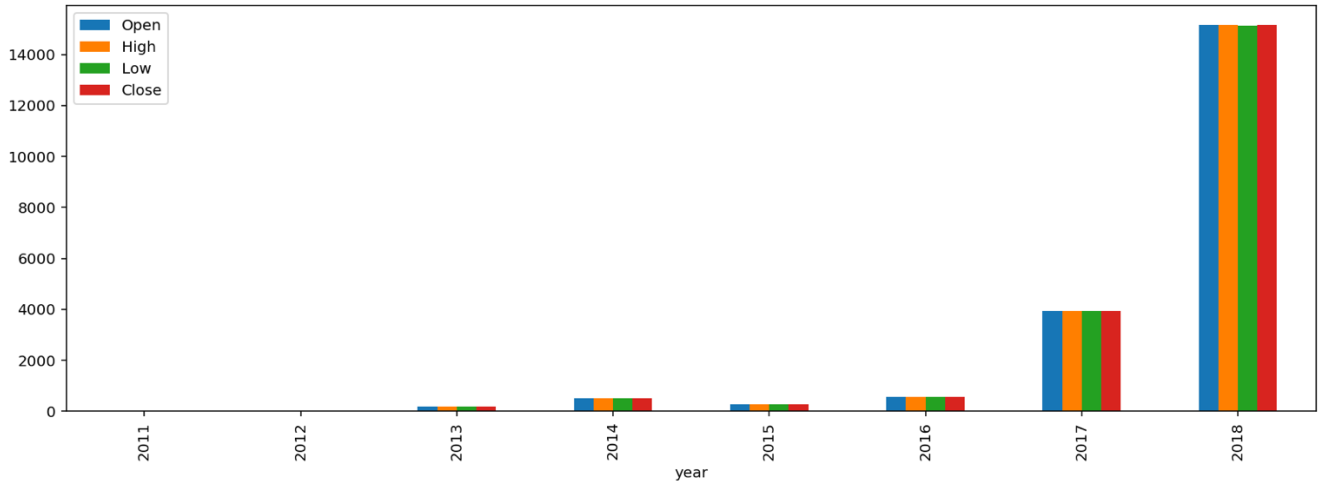
Volume of money invested on BitCoin



Average weighted price cumulated Trend



Average price yearly



Algorithms and Techniques

Creating the features for unknown behaviour in crypto-currency is very difficult and using Neural Network would be ideal to capture the same through hidden layers.

As it is time series model, ARIMA or moving average models could have been used as the base models. But instead XGBoost regressor is used as it is known for better overall performance than statistical time-series models.

Following deep-learning techniques are tried for this project -

- Shallow neural network
- Multi-layer neural network
- Recurrent neural networks (RNN)
- Long Short Term Memory networks (LSTM)
- Gated recurrent Unit (GRU)
- Bidirectional RNN
- Convolution Neural Network (CNN)

Benchmark

Cryptocurrency price forecasting is a very unique project and yet not extensively researched. Not even surface scratch compared to stock market prediction. There are no baseline methods or standards or benchmark or public dataset available to verify performance of the models generated for this capstone project.

Hence, XGBoost model output is used as a benchmark and certain passing criterion is set accordingly and as explained in "Metrics" section above.
RMSE less than 400, i.e. price error less than $400
>50% of precision, recall OR f-score

Assumption here is if we have model which gives less error in forecasting and more than 50% of the times able to predict next high/lows correctly ... profit can be incurred.

C. Methodology

Data Pre-processing

- Data Granularity: Running day-by-day price prediction had good data but less for the model generation and minute-by-minute model on GPU was taking >3 hours to generate. Both of these problems lead me to use hourly data which was kind of middle-ground in terms of quality and run & experimentation times. Hourly data is generated from minute data aggregation. High, Lows, Open and Close values are calculated accordingly for each hour of the day. Total data-point count was >51000.
- Date Features: Various date related features are also used like day_of_month, month, year, weekday, week.
- Past Time Features: For models to better understand the sequence and learn the pattern, last 3 days of data is given as separate features. It can be argued that it was not necessary as RNN itself would be able to understand the sequence patterns. But telling model explicitly about some past important features makes the model more stable and robust to the sudden changes.
- Scaling of Data: For XGBoost no scaling or normalization was required as it is based on decision tree and scaling will not make any effective changes. But for Deep Learning models, to converge all values should be small and in the range-bounded value region. Hence, MinMaxScalar is used to scale data within the limit of 0 to 1. This procedure can be found on almost all ipython file codes.

Implementation

System Setup

Following setup is used for the experimentation and development. And can be replicated easily with.

System: p2.xlarge
Operating System: Ubuntu 16.04 LTS
Software: Cuda 8.0, CuNN, tensorflow-gpu, Anaconda Python 2.7 and used Jupyter/Ipython for development purposes.
Libraries: Pandas, sklearn, keras, xgboost, numpy and matplotlib

Forecasting Models

There is no single model which was best or unique. As there are literally infinite solutions and model architectures that can be used to solve the same problem, I tried to use as many different networks and their combinations. Along with capstone project it was a good exercise to learn practical aspect of Deep Learning model design and implementation.

Algorithms Explored

1. Deep and Shallow networks
2. LSTM/GRU
3. CNN
4. Bi-directional RNN

Difficulties and solutions

- I created multiple ipython notebooks, so many times in-memory (RAM) became bottleneck to handle data.
- Cuda 9 installation was throwing error so instead of debugging moved to Cuda 8 which worked without hitch.
- With complex models on minute data 11gb of GPU memory was not enough and models could not train. Hence, had to design model architecture to fit in the GPU memory.

Refinement

- As explained in previous section, models are evolved step by step. But few extra steps were taken to get the better results.
- BatchNormalization is used to reduce overfitting and it significantly improved initial results so made it de facto for all next models.
- Dropout layer was used, again to reduce overfitting and to make model robust of significant input changes.
- BiDirectional wrapper is used to learn time-series sequence from both the sides. It improved the accuracy in some cases.
- In CNN network instead of default activation function PReLU is used which improved CNN performance over ReLU.
- Changing the batch size <=256 was resulting in overforecast and >=512 was resulting in underforecast. Hence, tuned it to 384, which performed well overall.
- Best model was saved with checkpointer callback api in keras

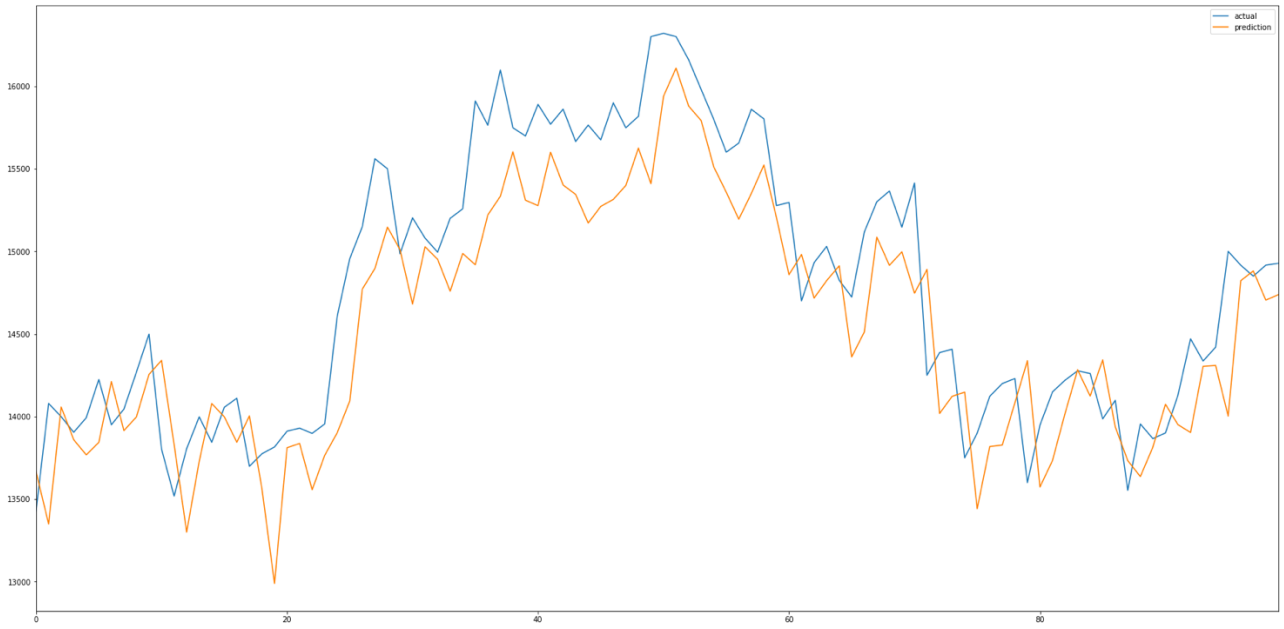
- Early stopping was implemented with callback api in keras, which helped run models faster.

D. Results

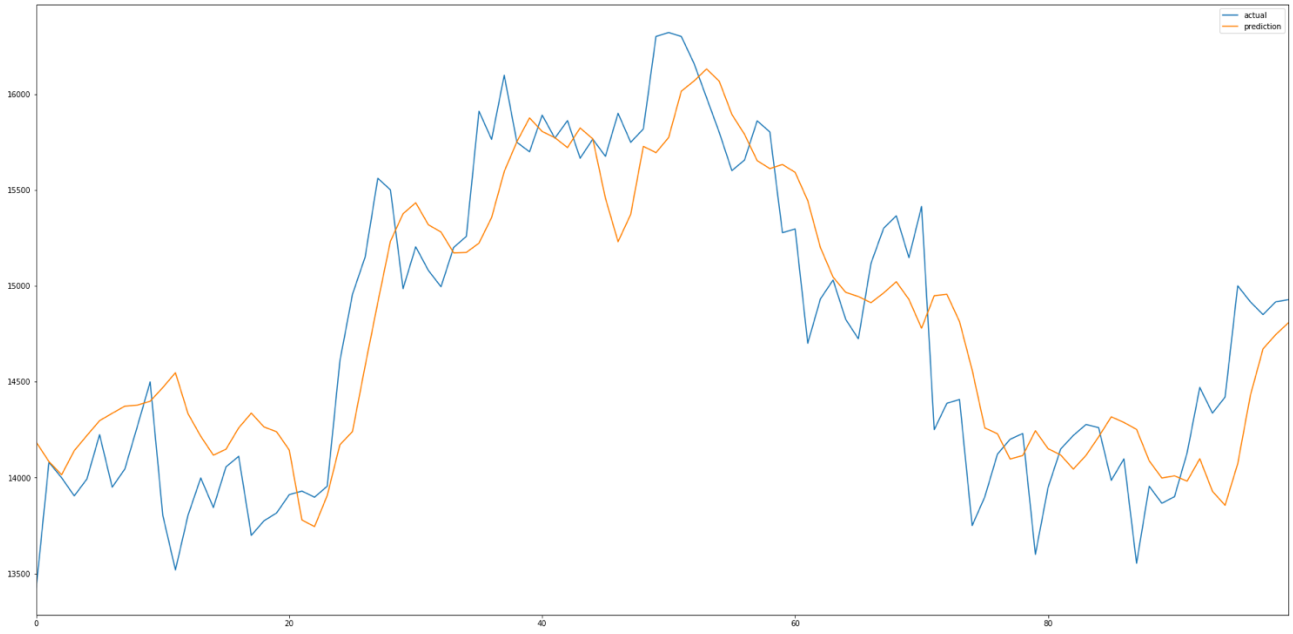
Model Evaluation and Validation

- Cryptocurrency Daily Forecast: As the data was very less simple layer LSTM network was used. As expected results were not good and hence decided to move to hourly price prediction model. Complete model run can be found in "Cryptocurrency Forecast Daily" ipython code.

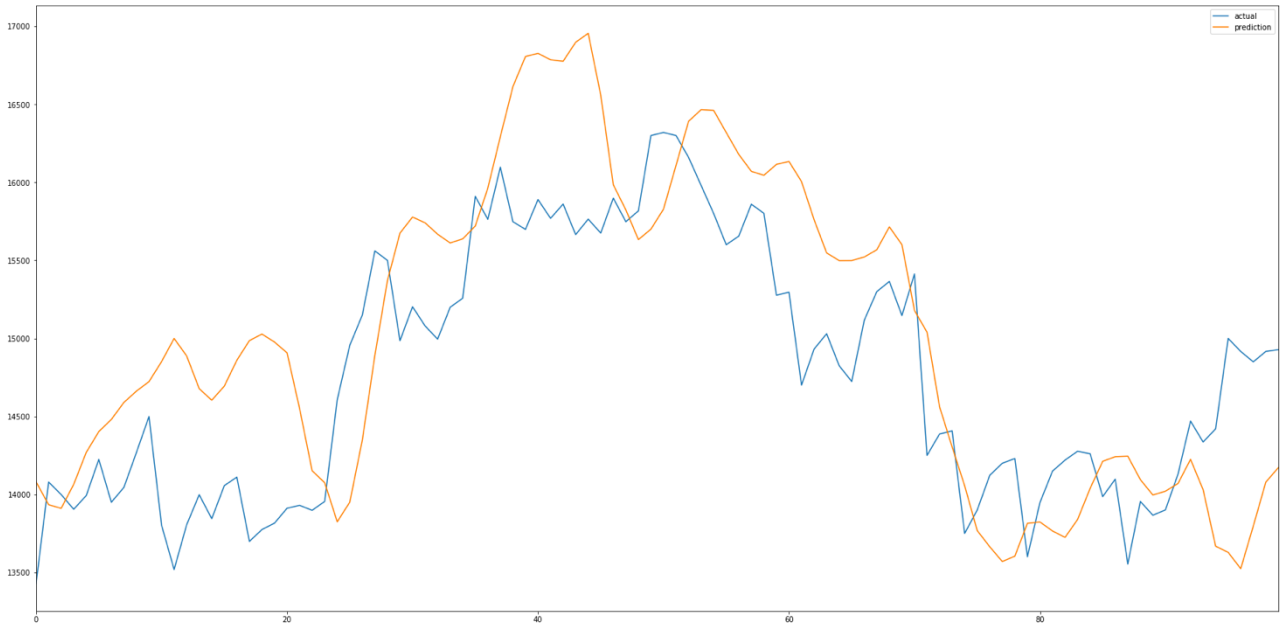
- XGBoost Regressor Model Hourly Forecast: Simple baseline XGB model was generated. Basic parameter tweaks are applied to get the better model. RMSE of 405 and f1-score of 0.46 was achieved, along with good trend pattern match as shown in graph below.



- Cryptocurrency Hourly Forecast Sequential Models: Multiple sequential models are tried. LSTM, GRU single and multiple layers of networks are used. Single layer Bi-Directional GRU model was found better than XGBoost with 0.47 of f1-score and 384 RMSE, along with good time series trend.



- Cryptocurrency Hourly Forecast Graph Based Models: To improve the f1-score further, tried graph based complex models. They performed worse than the sequential models. Though 3 layer CNN and hybrid model of CNN and RNN performed better in terms of f1-score, RMSE was not better than previous best mode. For CNN-RNN hybrid, RMSE was 672, f1-score of 0.52 and it captured trend nicely. For single layer CNN, RMSE was 771 and f1-score of 0.52. CNN was surprisingly better than LSTM and GRU models.



Cryptocurrency Hourly Forecast to Binary Regression Model:

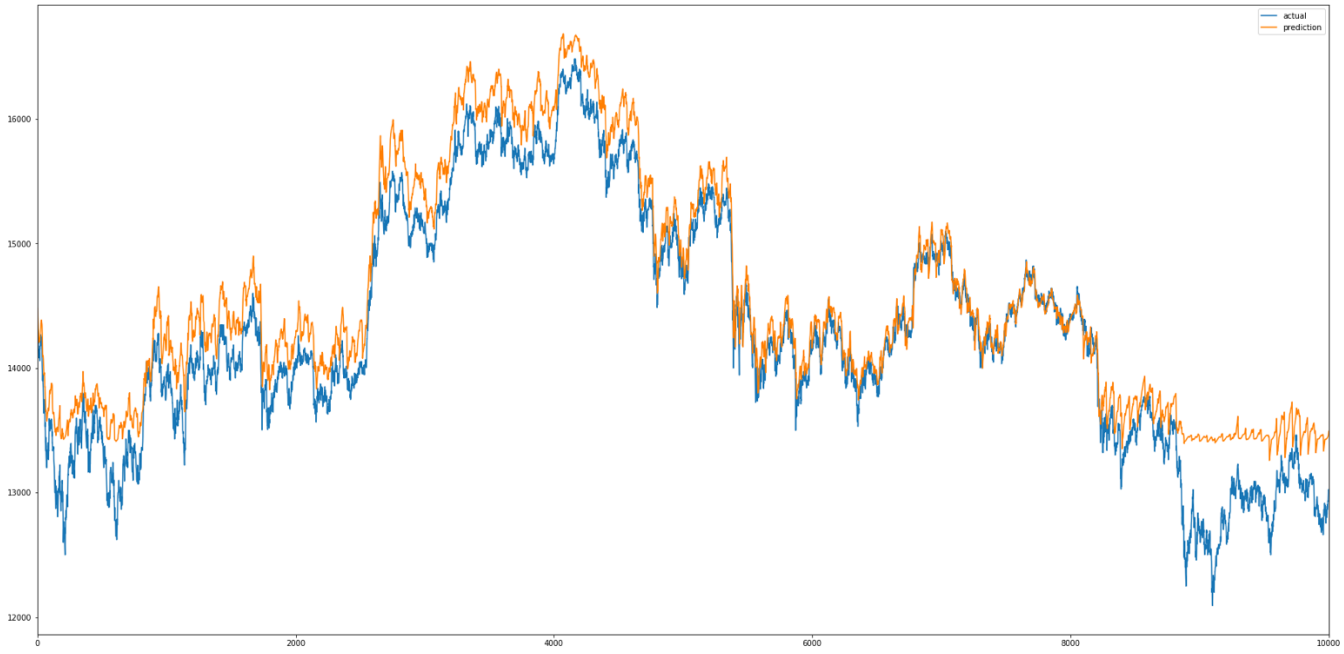
- Getting ~52% f1-score i.e. accuracy from above model might be a good performance. But human experts can beat this performance easily. It may be difficult to say by how much forecast would be

wrong i.e. RMSE but directional indications should be much more accurate. We should be able to predict tomorrow's price with good confidence if price is going up or down from today's price. So, created another model with output transformed to binary 1 and 0 showing uptrend and downtrend.

- Three different models are used, and they all gave >0.60 of f1-score. Here RMSE can't be compared. Point to be noted here is decision boundary of 0.5 is not used compulsorily. Decision boundary here means, if value is between 0 and X then it will be marked 0 and from X to 1, it will be marked as 1.
- For 3 layer CNN model 0.5 was the boundary and gave 0.59 f1-score. LSTM/GRU hybrid model gave 0.60 f1-score with 0.4 boundary. For LSTM/GRU and Bi-Directional hybrid graph f1-score of 0.64 was achieved with 0.2 boundary. In all above experiments, boundary was decided from validation set and applied on test set.

Cryptocurrency Minute Forecast

In ideal world where we can work full-time on such projects and there are no monetary bounds, I would have focused on minute data instead of hourly data. But it takes >3 hours to run even simple model and cost more on ec2 by hour. So only few runs were made for this approach but good results are found. f1-score of 0.50 and RMSE of 322 was achieved with two layers of LSTM and GRU sequential model. More accuracy could have been achieved with more experimentation time.



Justification

- As per the set benchmark, many models pass through the test and can be used in real scenarios.
- To capture the uptrend and downtrend and stable period, December 2017 was chosen as test period where prices were low initially, then increased and remained high for some time and then declined sharply.
- All the models with less than 400 RMSE and >0.5 f1-score followed the price trend.
- One model may not be completely enough for forecasting which involves monetary transactions and as shown above for trend and actual values two different models could be used.
- Even without cryptocurrency block-chain data, without social platform sentiment analysis models were able to perform better.
- Point to be noted that >50% accuracy in financial situations is often considered very good. Can be referred here for more details https://www.forbes.com/sites/rickferri/2013/01/10/ts-official-gurus-cant-accurately-predict-markets/ - 793f04227c00.

E. Conclusion

Personal Experience

- I decide to do the Cryptocurrency forecasting as capstone project as I lost some money in cryptocurrency trading and as newbie didn't know market well.
- Though, model was not the complete automation, it was a good enabler and influencer on my decisions. Following are my transaction details -

DEPOSIT HISTORY export csv Refresh

Coin	Amount	Confirmation	Status	Created
INR	4980.00000000	1 / 1	Confirmed	25-01-2018 11:26:58
XRP	42.15000000	1 / 1	Confirmed	20-01-2018 01:21:39
XRP	24.98000000	1 / 1	Confirmed	20-01-2018 00:45:44
XRP	124.98000000	1 / 1	Confirmed	20-01-2018 00:17:54
BCH	0.03540000	3 / 3	Confirmed	16-01-2018 19:07:57
BCH	0.03400000	3 / 3	Confirmed	16-01-2018 17:36:39
XRP	51.47000000	1 / 1	Confirmed	11-01-2018 19:46:46
XRP	50.00000000	1 / 1	Confirmed	11-01-2018 19:33:14
BCH	0.05100000	3 / 3	Confirmed	16-12-2017 20:08:46
BCH	0.04956818	3 / 3	Confirmed	15-12-2017 13:12:22

- I created the simple 1 layer GRU model for BitCoin(BTC) forecasting hourly and used it in period between 15 Dec 2017 to 25 Jan 2018. Though, I trusted that if BTC prices are going up/down ... other

coins will behave in a same way. This was the false assumption for some time in mid Dec 2017, but later proved to be correct.
- In totality, I used my own judgement and model output as influencer with only month of exposure to Crypto-Currency concepts ... I was able to earn 2.5 times the money I invested

Reflection

- Cryptocurrency forecasting is not being done effectively at least in public domain and it has good reasons for the same. It is still not a stable market and with government involvement uncertainty looms over adaptation of cryptocurrency.
- Main motivation was to work on problem on which there is little or no help, no research papers available and not much focus yet ... still very important problem to solve.
- Experiment started with collecting data which was itself a challenge in itself but kaggle.com's free data is used finally.
- Initial thought was that multi-layered RNN models will work better, but simple shallow networks of 1 or 2 layers work much better than complex models.
- CNN works on time series which was surprising for me and would be experimenting on the same for some time next.
- Results were as expected better than conventional regression model but running model takes more time than XGBoost regressor.
- Project was started as Personal benefit became the point of focus later as in my current role in company I am constantly working on time series forecasting problem and this project helped me think in many different ways.
- Working on a project continuously was a challenge along with the company work and hence the delay in submission and could have done more experiments to improve model further.

Improvement

- With more data, more accurate models can be created.
- Block-Chain data if found minute-wise or hour-wise would definitely boost the accuracy
- Social media analysis would give sudden price trend changes and would react proactively.
- As we have seen, CNN works well on time series, attention mechanism in CNN is worth a try and Pooling could also be used.
- Hyper-Parameter tuning is not done properly. Need to figure out best way to do tuning in a smart way. Genetic evolution of networks to be explored to evolve and self-learn the network architecture.
- Reinforcement learning could be tried to solve time-series problem.
- Concept of Genetic evolution of network architecture is to go through next.