Machine Learning Engineer Nanodegree

Capstone Proposal

Praveen Origanti January 29th, 2018

Proposal

Cryptocurrency Price Indicator (Investment and Trading)

Domain Background

Cryptocurrency (or digital currency) is a revolutionary concept enabling peer-to-peer currency exchange without relying on a central trusted authority (like banks). Most cryptocurrencies use a decentralized/distributed ledger system, operating on the principles of cryptography, crowdsourcing and game theory. Satoshi Nakamoto first proposed these principles and defined the first blockchain database whose unit of currency is defined as bitcoin [1].

Bitcoin was created in 2009 and has mostly been limited to exploration by technological enthusiasts. The U.S. Treasury classified bitcoin as a convertible decentralized virtual currency in 2013 and the Commodity Futures Trading Commission, CFTC, classified bitcoin as a commodity in September 2015. Since then, it has garnered greater attention from general public and media. Mostly driven by speculative investments, it has seen an exponential rise in value in the last couple of years and several scholars expect the trend to continue, while others consider it to a bubble. Also, numerous other cryptocurrencies have been created as either variants of bitcoin or new platforms to support other applications. Ethereum, Litecoin, Ripple are a few among the popular ones, also referred to as alt-coins.

Personally, I've been impressed with the technology for a while, but haven't made any investments. I would like to take this opportunity to understand the market (and technology) better and decide if it is a good time to invest in cryptocurrencies.

Problem Statement

The objective is to predict the closing price of bitcoin and ethereum over a 7-day period using historic data of these cryptocurrencies. This is a regression problem, as the goal is to predict the closing price.

Given the high volatility in these markets, it is hard to arrive at useful models just based on price data [2]. In this light, I would also like to explore the possibility of including twitter/reddit sentiment over this period of time.

Datasets and Inputs

The historic price data of cryptocurrencies (along with other alt-coins) is available on

coinmarketcap.com [3]. For each day from Aug 07 2015 to Feb 26 2018, I plan to retrieve the data from [3] for bitcoin and ethereum that includes open/close/low/high/ prices, volume and market cap. This would be 934 rows of data for bitcoin and ethereum .

I couldn't find freely accessible twitter data on cryptocurrencies. I plan to use twitter API to access relevant bitcoin/ethereum tweets and perform sentiment analysis using NLTK package. One of the goals of this project is to understand if there is a correlation between tweets and cryptocurrency prices.

If I cannot easily obtain twitter data for at least a six-month time window, I plan to look into Reddit (r/CryptoCurrency, r/ethtrader, r/BitcoinMarkets) and perform sentiment analysis on headlines.

Solution Statement

I'd begin with applying linear regression (using scikit-learn) on the available data. As this data has temporal information (time series), recurrent neural networks would naturally apply. I plan to train a Long-Short Term Memory (LSTM) using Keras package with Tensorflow backend. I plan to vary the window size in this range [4-days, 20-days]

Benchmark Model

The benchmark model for this project is a simple model relying ony on last day's price available. The predicted price is going to be the last available price point (persistence model)

Evaluation Metrics

R^2 metric, Mean Squared Error (MSE) and Mean Absolute Error (MAE) are potential metrics for this problem. Since this is framed as a regression problem and the target is continuous, MSE is a sensible evaluation metrics.

The model results would be considered satisfactory if the predicted results are better than the benchmark model.

Project Design

I plan to divide the problem into these steps:

Data preparation and exploratory analysis

Retrieve raw data from [3] and relevant tweets using twitter API, and load them into Pandas dataframes. Analyze the data to gain a deeper understanding of the attributes and their correlation. Produce plots as a visual aid.

Review if any data is missing and either impute the data or ignore such records, as appropriate.

Feature engineering

Translate tweets into sentiments using Vader algorithm (from NLTK package's

SentimentIntensityAnalyzer). The mean 'compound' for all tweets in a day would be used as a feature. This feature's value would be in the range [-1, +1], where -1 and +1 represent the most negative and the most positive sentiments respectively.

from nltk.sentiment.vader import SentimentIntensityAnalyzer as SIA

I'd include other engineered features such as percent change in price per day; volatility etc. Select relevant features and drop any, if no correlation is observed.

Feature normalization

Normalize the features to have zero mean and unit variance. Since the number of features is small, PCA is unnecessary.

Train-test split

Split dataset into train and test sets into 80-20 ratio. Since this is a time series, the latest 20% of data will be treated as test set (No randomization).

Model exploration

Split training set into train and validation sets and learn these models:

1. Persistence model (baseline)
2. Linear Regression
3. LSTM

Model optimization and evaluation

Evaluate the models on validation set(s). Regularize or fine tune hyper-parameters to find a good bias-variance balance. Exploring the hyper-parameter space could be done using GridSearch in scikit, need to find relevant methods for LSTM.

For cross-validation, I'd use TimeSeriesSplit() feature with GridSearch, where successive training sets would be supersets of the previous ones.

Publish results with appropriate visualization

Evaluate the final optimized models on test set and compare with benchmark model. Visualize the predicted and actual bitcoin price using seaborn/matplotlib.

References

1. https://bitcoin.org/bitcoin.pdf
2. https://dashee87.github.io/deep%20learning/python/predicting-cryptocurrency-prices-with-deep-learning/
3. https://coinmarketcap.com/
4. https://www.kaggle.com/ara0303/forecasting-of-bitcoin-prices