ID2221 Data Intensive - Project Report

John Magnusson and Lucas Q. Gongora

October 2020

1 Introduction

This project aims to tackle the problem of value forecasting on cryptocurrencies by analyzing historical data with machine learning tools to extract knowledge. In this project, we chose to focus on the cryptocurrency Bitcoin. However, the created code is adaptable to work on any cryptocurrency supported by the data source [1].

2 Dataset

In this project, we operate on two different datasets which we generated from the open API[1]. The first dataset uses available data, from 2014 until 2020, for the Bitcoin per hour, which includes the variables of time, volume and market price. The second dataset extends the first dataset with additional information related to social media on Bitcoin. This additional data consists of the number of comments, followers and number of people talking about Bitcoin on social media such as Reddit, Facebook and Twitter. Unfortunately, the API does not support to the same extent the social data as the trading one. Resulting in the social dataset getting reduced to be in the time frame of 2019 and 2020.

3 Method

The methodology of the project can be split up into three different phases: *Data-gathering*, *Pre-processing* and *Knowledge extraction*. Each phase will be presented in the following sections.

Data-gathering

To create the data set mentioned in section 2 we had to build a client that could communicate with the chosen API. The client communicates with REST over HTTP and can retrieve historical OHLCV pairs with the time prefix minute, hour and day. It can also retrieve social-historical data on time prefix hour and day. After choosing the data to fetch and the time frame, the client retrieves and stores the HTTP response content in a text file in JSON format. The code for the client can be found under the folder "Data retriever" on GitHub [2].

Pre-processing

For pre-processing we chose spark and sparkSQL as they are fast and provide a rich programming model. We used Scala together with spark because we wanted to get experience with the language. The framework helped us read the data from the previous step, execute transformations and actions needed for the machine learning phase. Those processes were:

- cleaning data: removing special characters, splitting strings and mapping the objects;
- splitting data: creating training and validation set from the dataset;
- computing statistics: calculating mean, min, max, standard deviation for the variables;

- feature creation and normalization: creating new variables and normalizing;
- merging datasets: joining the bitcoin and the social media dataset;
- persist data: saving the data to be read in the next step.

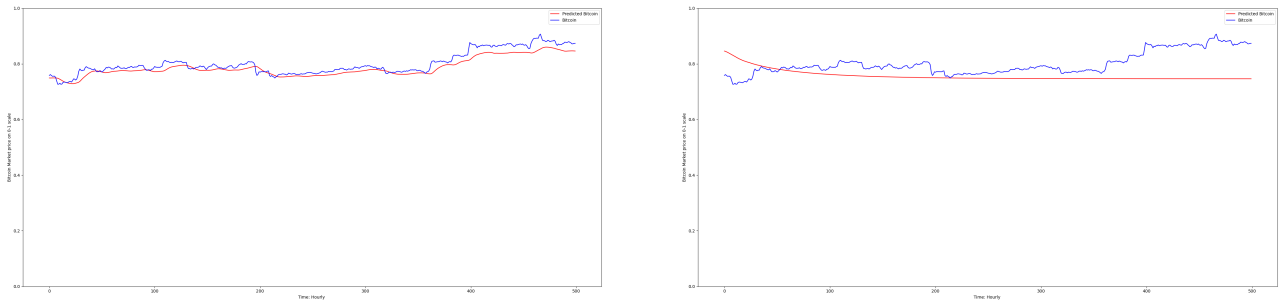
Knowledge extraction

To extract knowledge from the preprocessed data we implemented an LSTM model. LSTM models have been previously used to forecast stock price and other time series [3]. The model was trained on the two datasets and used to predict future values.

4 Results

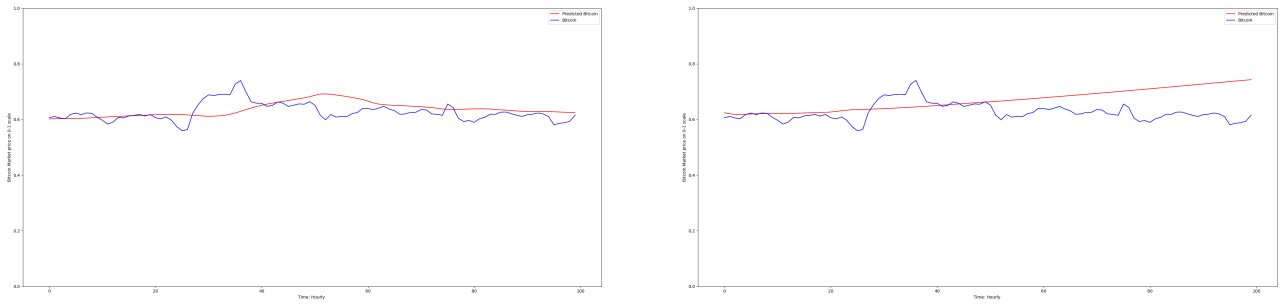
The goal of the project was to create an LSTM model that could predict the stock market value of Bitcoin in two different approaches: (1) estimate the model with an hourly update of the real value; (2) estimate hours ahead without updating the data.

Those two approaches were implemented in two different datasets (as explained in 2). The prediction was performed on the median price of the hour. For the dataset with only Cryptocurrency data, we added the volume as a parameter, and for the Social media were added volume and four more variables: Total page Views, Facebook Talking, Reddit Posts, Reddit Comments.



Updating model every new hourly event | Forecasting data without hourly update

Figure 3: Stock market prediction with 500 testing samples



updating model every new hourly event | forecasting data without hourly update

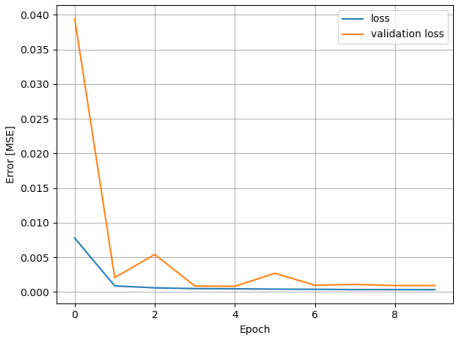
Figure 6: Social Media with 100 testing samples

As expected, the prediction with real-time data update performed better. Initially, one can see that models prediction are quite similar, but as time goes on, models that are allowed to update can parry changes meanwhile the models that forecast longer continue in the same trend. Seen both in figure 6 and 3.

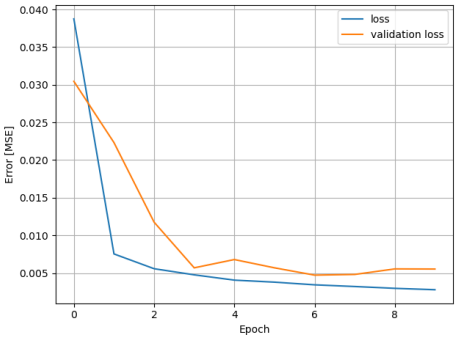
A comparison between two datasets is though to perform, mostly because they have different sized datasets when training and different parameters. However, it is possible to compare the performance of

the Social Media estimation by comparing it to the ground truth and once more using both strategies of hourly update, as seen in figure 6.

The loss function (figure 9) is very similar for both of them with the loss decreasing and getting close to convergence in the final epochs.



Stock market



Social Media

Figure 9: Loss plot for 10 epochs

5 How To Run

In this section, we will briefly explain how to run each part of the project. This project is built with Python 3, Jupyter notebook, Spark, and uses several libraries such as JSON, Tqdm, TensorFlow 2, NumPy, pandas and Matplotlib. Both python files can run from a terminal, but if you want to change hyperparameters, etc you need to modify the files. The Jupyter files you need to run in the notebook. Link to the GitHub repository: [2]

Data retriever

To run the data retriever simply run the main.py inside the "Data retriever" folder inside the project. Here you can choose if you want to fetch social data and trading data, between specif time frames, which cryptocurrency, etc. Once data is fetched it will be saved to the dataset folder inside the project with the filename you entered.

Pre-processing

Inside the "PreProcessing" folder you navigate to the Jupyter notebook script. Here simply step through each cell. The code will generate new files based on the input data, you mustn't have a folder with the same path, otherwise, you will get an error. Also, remind that you should not try to compile the Social Media part for the dataset with data from 2014 till 2020.

Knowledge extraction

Locate yourself to the "MachineLearning" folder. Here you run the main function to train the model that makes a prediction, once training is done, of the value of the cryptocurrency. There are several hyperparameters that you can change to your choosing.

References

[1] Cryptocompare. `https://min-api.cryptocompare.com/`. Accessed: 2020-09-22.

[2] Project github. `https://github.com/JohnMagnusson/ID2221_Data_Intensive-Project`.

[3] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman. Stock price prediction using lstm, rnn and cnn-sliding window model. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1643–1647, 2017.