Project Report Data Mining: M1

Topic: Bitcoin Price Prediction

Contents

Introduction:

Since Cryptocurrencies get more popular and mainstream, everybody is getting slowly in touch with it, I want to answer this question:

“Is an accurate Price Prediction with Machine Learning Methods possible?”

The Dataset that is used is from Kaggle.net (https://www.kaggle.com/mczielinski/bitcoin-historical-data/discussion) and contains the Bitcoin history from January 2012 until March 2021. Its size is about 300MB and it consists of 8 columns (a more detailed Feature Analysis is in the StepA-notebook):

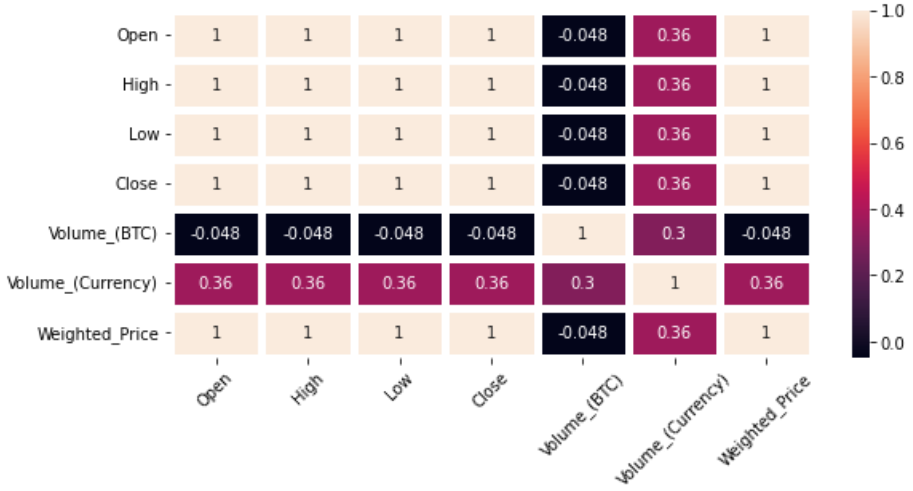
Id	Timestamp	High	Low	Open	Close	Volume_(BTC)	Volume_(Currency)	Weighted_Price
Int	Int	float	float	float	float	float	float	float

The Dataset has multiple NaN values which have been normalized in StepA-Notebook for Data Preparation. The enormous size of the Dataset has caused long running times of all used Models. Since one of the used Models is a Recurrent neural network (LSTM), which is computing intensive, I decided to use a Jupyterhub environment with GPU support, to be more specific a “NVIDIA Tesla K80” GPU with about 12 GB Graphics Memory.

There were more than 4.8 million entries (minute price of Bitcoin for about 10 Years). First tests have shown that it is not comfortable to use such a big amount of data, neither necessary. I decided to sum it up to days. After the transformation the length was only about 3300 entries. Furthermore, the Timestamp has been converted from “unix-timestamp” to days, to make handling the values more simple.

The Dataset contained about 1,2 million NaN values. This was fixed by first the interpolate function of pandas (method=”nearest”) and by summing up the single values to days.

As seen in the StepB and StepC for the explicit Training and Prediction only the “Weighted_Price” Feature has been used, because it fully correlates with the other OHLC(Open, High, Low, Close) values. Which can be seen in the following Graphic:



Results

Classification Metrics

There is no Classification needed since it is a Regression Problem, the “Weighted_Price” Price of the Dataset is the base. In future work other Features like the traded Volume should be considered to take into account.

The Algorithms/Models that have been used were:

- LSTM(Long Short-Term Memory)
- SVM/R(Support Vector Machine/Regression)
- Linear Regression (LR)

As expected LSTM(Long Short-Term Memory) has the most potential, since different Layers with different activation functions, Dropout, Units, … .Therefore it is well suitable to for the selected Project.

The following Regression metrics have been used to evaluate the models:

- Mean Absolute Error (MAE)
- Root Mean Square Error (RMSE)
- Explained Variance Score (EVS)
- R^2 Score (R2)
- Max Error Score (MES)

Comparison

The following tables show a comparison between multiple runs (with different Parameters, if possible) of the different Models. To explain my Evaluation the “best” value for every metric is shown in this table:

Metric:	**Best possible Value:**
MAE (Mean Absolute Error)	0
RMSE (Root Mean Square Error)	0
EVS (Explained Variance Score)	1
R^2 (R squared)	1
ME (Max Error)	0

LSTM(Long Short-term memory):

Firstly, the LSTM Model is evaluated with different Parameters like Epochs and Batch Size. The following tables show the different runs, the Epoch, Batch size, metric values, and the running time.

Name:	LSTM		
Run	1	2	3
Epochs	5	5	5
Batch Size	15	5	30
MAE	12410,95	5608,98	9812,32
RMSE	12852,17	5982,42	10063,96
EVS	0,41	0,65	0,60
R^2	-7,78	-1,86	-7,10
MES	18035,18	8783,10	12797,66
Running time	105 s	318 s	58s

For the first set of tests the size of Epochs, has not been changed (Epochs = 5), only the Batch size has been modified. The Result was that a smaller Batch size led to higher running time and vice versa. For Set 1 it can be concluded, that the best Parameter for Batch size is 5, so lower is better. MAE and RMSE are the smallest for all three runs, EVS and R^2 are the nearest to 1. MES is also the smallest of all values for run 2.

For Set 2, after altering the Batch size, the Batch size has been fixed to 15 and the number of Epochs has been modified. Four runs have been made and the “best” results have been achieved with the number of Epochs equal 100. The metrics of run number 5 have the most near to best values of all metrics.

Name:	LSTM			
Run	4	5	6	7
Epochs	50	100	200	500
Batch Size	15	15	15	15
MAE	8558,05	2685,13	6423,24	4207,50
RMSE	8753,61	3069,81	6722,89	4659,76
EVS	0,73	0,75	0,68	0,68
R^2	-5,13	0,25	-2,62	-0,74
MES	11583,26	5321,53	9884,57	7922,29
Running time	1044s	2036s	4086s	10258s

SVM(Support Vector Machine) and LR(Linear Regression):

The third Notebook (StepC) covers Linear Regression (LR) and SVM (Support Vector Machine). For the SVM multiple values for Gamma and C have been tested, with ambiguous results. None of the different runs have been clearly shown to be the best in all metrics. For clarity the cells with the most “to perfect” value have been marked green. For the first three runs of the SVM, where only Parameter C has been changed, it can be said that for a better MES and EVS a lower value (1e6) resulted in an improvement. Whereas for an improvement of R^2 and RMSE a higher value (1e10) showed better results.

Name:	LR	SVM				
Kernel	-	rbf	rbf	rbf	rbf	rbf
Gamma	-	1,00E-10	1,00E-10	1,00E-10	1,00E-08	1,00E-12
C	-	1,00E+08	1,00E+06	1,00E+10	1,00E+08	1,00E+08
MAE	9910,18	4906,41	8381,58	5018,35	4087,60	12811,80
RMSE	9931,85	5937,88	8423,71	5556,36	5582,87	12919,19
EVS	0,97	-1,78	0,94	0,54	-0,90	0,78
R^2	-6,89	-1,82	-4,68	-1,47	-1,49	-12,35
MES	10946,08	11375,19	9589,90	10110,94	17160,95	15573,40
	1s	1s	1s	14s	21s	18s

Conclusion:

It can be concluded, that the LSTM Model has the most potential for an accurate Prediction. That is mainly because LSTM has the most Parameters and Features which can be optimized. For example the Model consists of multiple Layers, with different activation functions(e.g. tanh, sigmoid, relu) and optimizer functions(e.g. adam, nadam, AdaDelta), etc… The Linear Regression Model has no Parameters which could be optimized. Support Vector Machine has some Parameters that can be optimized, but it seems that summed up LSTM is the best Model of this three to Predict the Bitcoin Price.

For Future improvements of the Project the LSTM Model could be optimized more and it should be considered to apply a specific Trading Strategy like Bollinger Bands, which gives a specific entry and exit Point to the market (when to buy/sell). Furthermore after applying the Strategy it should also be back tested, which means to test the Trading Strategy with Historical data an estimate the returns.