

Bitcoin Prediction with RNN

Shengjie Lin
Data Scientist



Problem Statement:

Make a model that predict bitcoin price.
See if it's usable in real world
application.



What is Bitcoin?

- Created, distributed, traded, and stored with the use of a decentralized ledger system, known as a blockchain

What gives Bitcoin value?

- Useful as a way to store and exchange value
- Because a group of people agree it has value

Key characteristic

- It has limited supply
- Easily divisible
- Durable
- Easy to send bitcoin
- Easy to verify the authenticity



My Approach

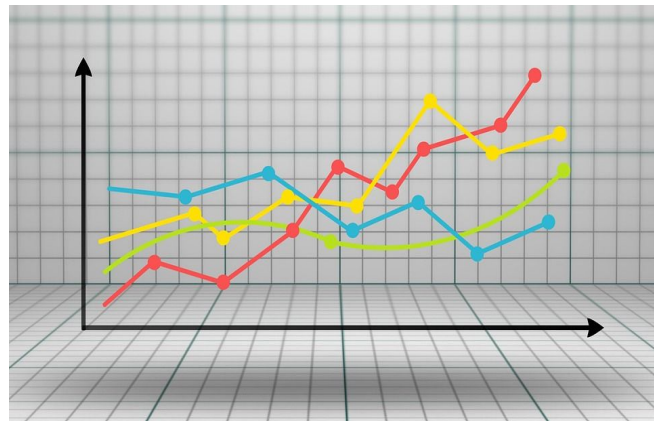
Multivariate time series forecasting with LSTM

Pros:

- Add features easily
- Datas don't need to be stationary

Cons:

- Hard to set up initially
- Computationally expensive
- Some features might weaken the model

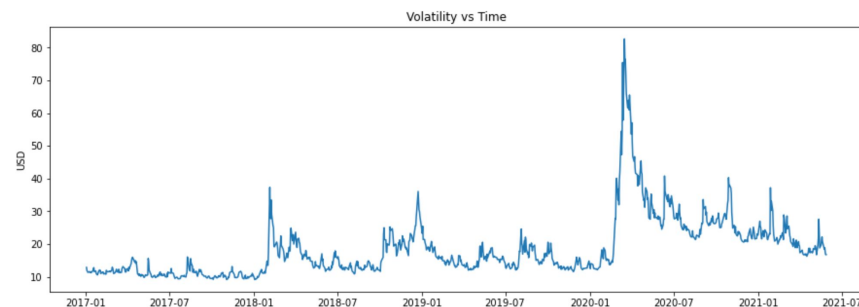


Features - Countries economic situation

- Gold price



- Market Volatility



- 10 year Minus 2 Year Treasury

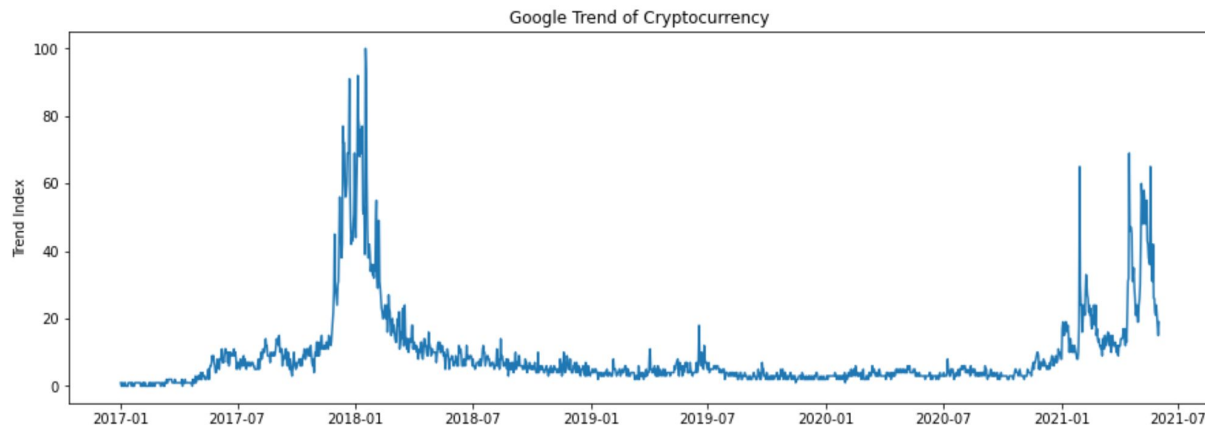


- Overall Market Performance



Features - Power of media

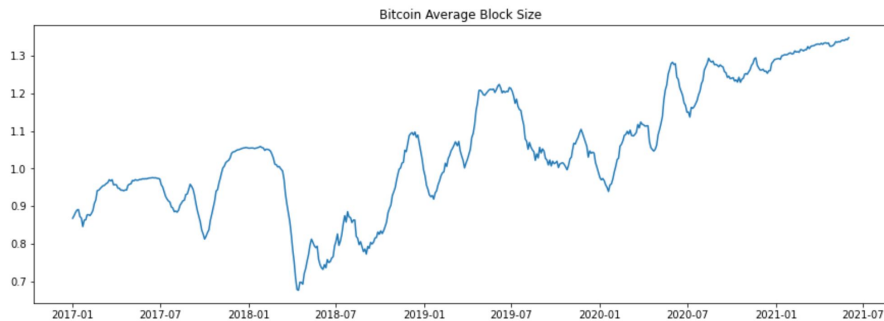
- Google Daily Trend



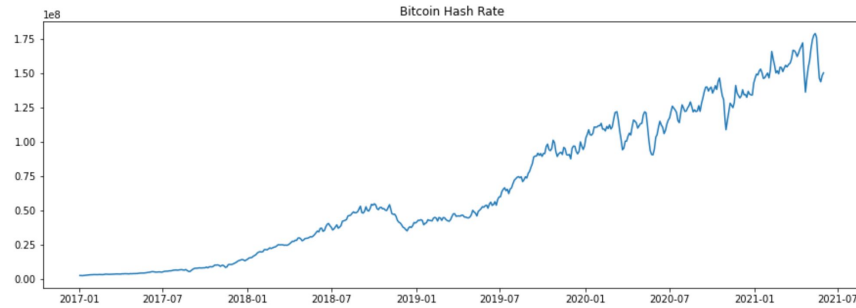
- To be added
 - Reddit
 - Twitter

Features - Blockchain information

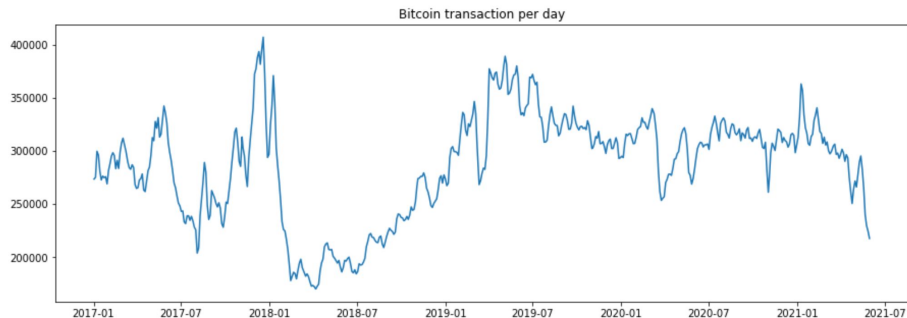
- Average Block Size



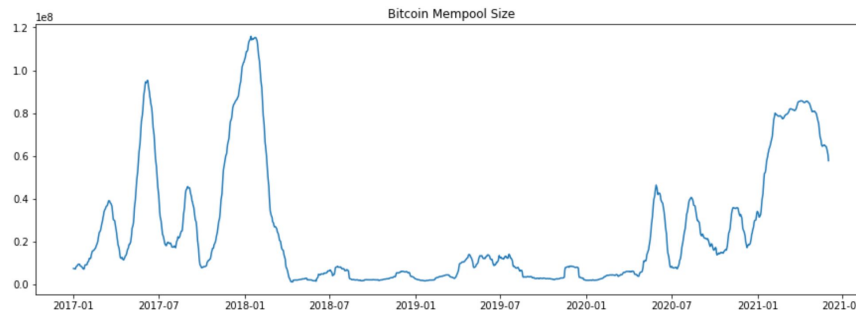
- Total Hash Rate (TH/s)



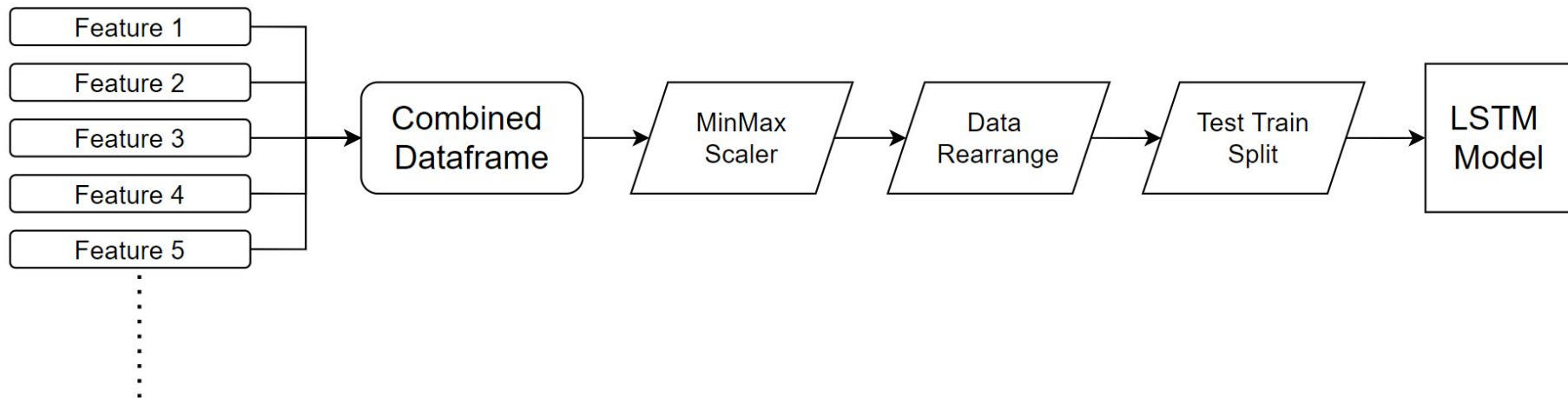
- Transaction Per Day



- Mempool Size (Bytes)



Data preprocessing



Data Rearrangement

Red: X (input)

Blue: Y (prediction)

Date	BTC_price	gold_price	treasury_10y2y	volatility	sp500	goog_trend	block_size	hash_rate	transaction	mempool
2017-01-03	1043.839966	110.470001	1.23	12.85	2257.830078	0.0	0.872457	2.479403e+06	274168.047619	7.422547e+06
2017-01-04	1154.729980	110.860001	1.22	11.85	2270.750000	1.0	0.874868	2.467371e+06	274751.952381	7.396877e+06
2017-01-05	1013.380005	112.580002	1.20	11.67	2269.000000	0.0	0.877742	2.455338e+06	275335.857143	7.326382e+06
2017-01-06	902.200989	111.750000	1.20	11.32	2276.979980	0.0	0.880615	2.443306e+06	283477.428571	8.111156e+06
2017-01-07	908.585022	112.056666	1.19	11.40	2274.286621	1.0	0.883489	2.426762e+06	291619.000000	8.401917e+06
2017-01-08	911.198975	112.363332	1.18	11.48	2271.593262	1.0	0.885492	2.410217e+06	299760.571429	8.692677e+06
2017-01-09	902.828003	112.669998	1.17	11.56	2268.899902	0.0	0.887496	2.393673e+06	298600.380952	8.815441e+06
2017-01-10	907.679016	113.150002	1.19	11.49	2268.899902	0.0	0.889499	2.436831e+06	297440.190476	9.202789e+06
2017-01-11	777.757019	113.500000	1.18	11.26	2275.320068	0.0	0.889973	2.479989e+06	296280.000000	9.484986e+06
2017-01-12	804.833984	113.910004	1.18	11.54	2270.439941	0.0	0.890447	2.523148e+06	291193.095238	9.543421e+06

Model Structure

```
model = Sequential()
model.add(LSTM(128,activation = 'tanh',input_shape=(X_train.shape[1],X_train.shape[2]),return_sequences=True))
model.add(Dropout(0.3))
model.add(LSTM(64,activation = 'tanh',return_sequences=False))
model.add(Dropout(0.3))
model.add(Dense(30,activation = 'tanh'))

model.add(Dense(y_train.shape[1]))

model.compile(optimizer='adam',loss='mse')
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
lstm (LSTM)	(None, 60, 128)	71168

dropout (Dropout)	(None, 60, 128)	0

lstm_1 (LSTM)	(None, 64)	49408

dropout_1 (Dropout)	(None, 64)	0

dense (Dense)	(None, 30)	1950

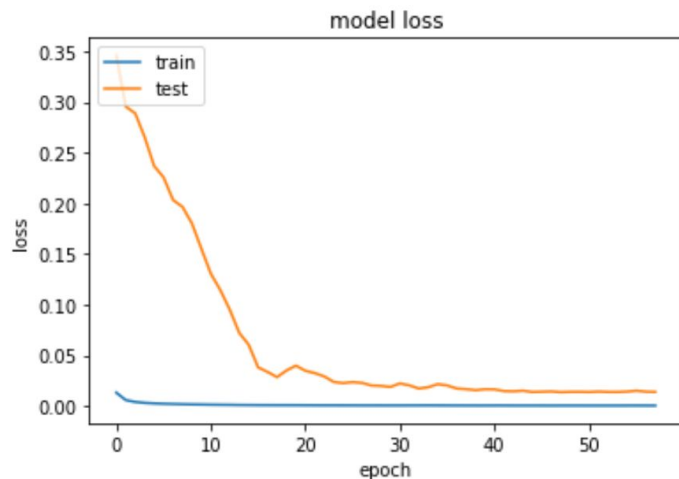
dense_1 (Dense)	(None, 14)	434
=====		

Total params: 122,960

Trainable params: 122,960

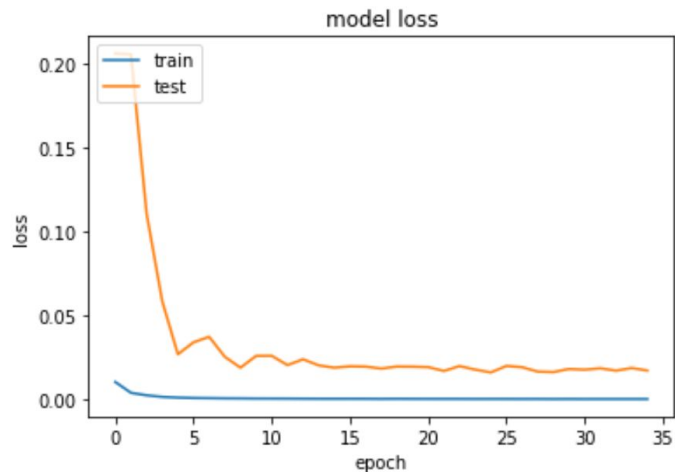
Non-trainable params: 0

Results



The final val loss is: 0.013585258275270462

With Features



The final val loss is: 0.01631840132176876

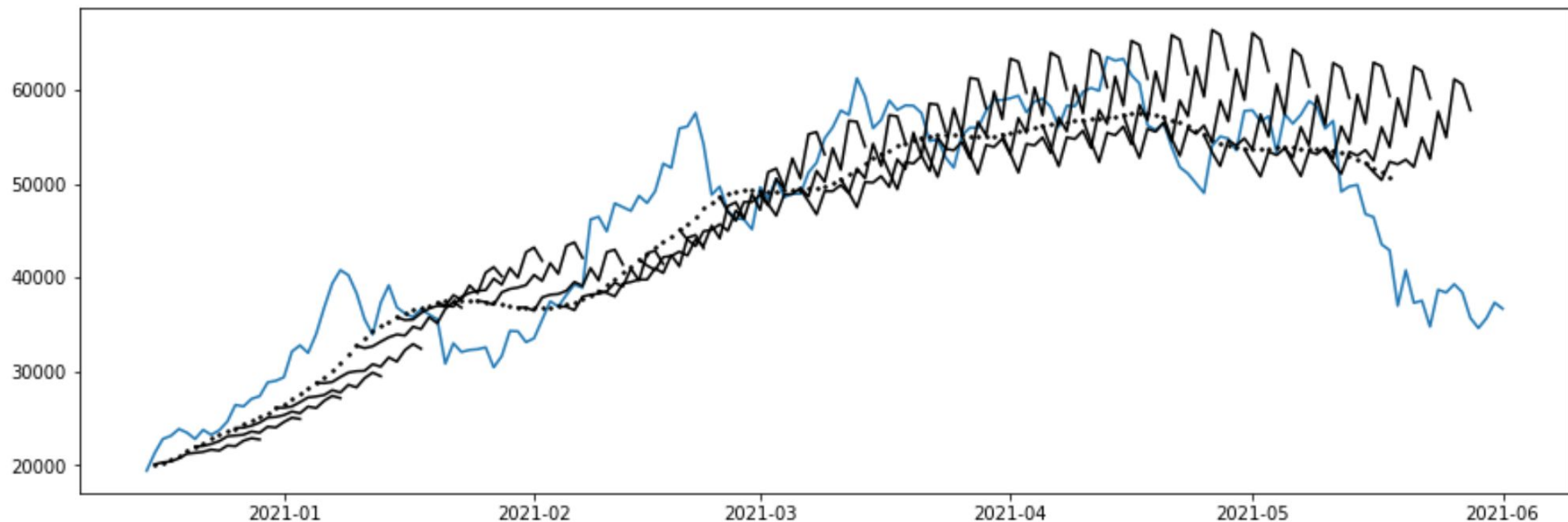
Without Features

Results

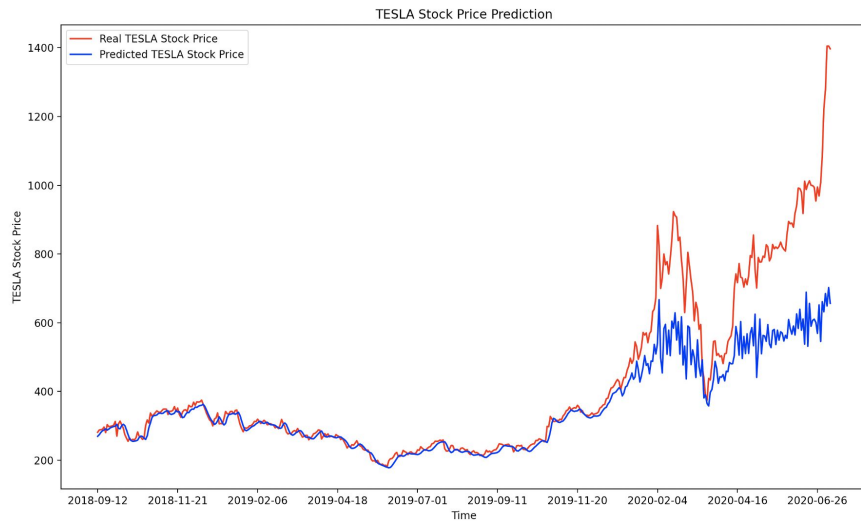
Blue Line: BTC Price

Black Line: 14 days BTC prediction

Black Dot: Day 1 price prediction



Caveat



One day prediction graph are pretty, but misleading.

Conclusion

Not suitable for real life application

Why didn't it work?

- Bitcoin's history as a store of value has been turbulent; it has gone through several cycles of boom and bust over its relatively short lifespan
- Lack of key information



Future Work

- Add Reddit speculation
- Add Twitter speculation
- Add Whale account trading history
- Experiment and reduce number of features