CRYPTOCURRENCY PREDICTION SYSTEM FOR 50 CRYPTOCOINS

Student Name : Sathya Priya Mahendran
Student ID : Q15817059
Course Title : Applied AI in Business
Course Leader : Dr Olufemi Isiaq
Date : 12/08/2022

Contents

Introduction

A cryptocurrency computer-generated coin that is protected by cryptography, which makes it practically difficult to forged or double-spend. An unidentified being called Satoshi Nakamoto created it and familiarized it to the world via a white paper in 2008. There are thousands and thousands of cryptocurrencies existing in the market now. Each cryptocurrency privileges to have a diverse purpose and condition.

Cryptocurrency price prediction can offer a lending hand to cryptocurrency depositors for making appropriate venture verdicts in order to obtain greater revenues while it can also support policy decision-making and economic investigators for learning cryptocurrency arcades performance. Cryptocurrency price prediction can be deliberated as a common type of time series problems, like the stock price prediction. The benefits of cryptocurrencies include inexpensive and quicker money transferals and distributed systems that do not collapse at a sole point of failure. The drawbacks of cryptocurrencies include their rate instability, high energy consumption for mining activities, and used in criminal actions.

Problem Identification

Cryptocurrency price prediction is considered to be a very challenging task, due to its confused and multifarious nature. In fine terms, the problematic account is to make a cryptocurrency predictive system that detects the cryptocurrency price form on their historical information. The pain fact of this project is predicting the value of cryptocurrency whether its value is high or low. This study evaluates some of the most successful and widely used deep learning algorithm for the purpose of forecasting cryptocurrency prices.

Value Proposition

- People who want to make more money with cryptocurrencies can benefit from machine learning.
- Maximize the profit and minimize the loss of the cryptocurrency by giving the best predicted price.
- Machine learning is creating input to practice in the crypto sphere is as a way to identify fraudulent trades.
- The most operative way of defining the future presentation of a given asset is through the study of the flow of its capitals.
- Recurrent neural networks are being utilized to superior learn and guess the trading forms of detailed investors.

Success Metrics

The accomplishment metric for the project is a performance quantity that is utilized to observe, analyze, optimize, and predict the cryptocurrency prices with sensibly high truthfulness by finding complex designs in highly intricate data which will be useful for both traders and investors alike.

Methodology

This segment shows the dataset and proposed machine learning algorithm in detail for predictive analysis of cryptocurrency prices.

Data Collection

Cryptocurrency historical data has been collected from **Yahoo Finance**, which is a broadcasting platform that is part of the Yahoo! system. The historical data consists of 7 features:

Date	Open	High	Low	Close	Adj Close	Volume
2019-08-09	11953.469727	11970.458008	11709.745117	11862.936523	11862.936523	18339989960
2019-08-10	11861.556641	11915.655273	11323.898438	11354.024414	11354.024414	18125355447
2019-08-11	11349.740234	11523.579102	11248.294922	11523.579102	11523.579102	15774371518
2019-08-12	11528.189453	11528.189453	11320.951172	11382.616211	11382.616211	13647198229
2019-08-13	11385.052734	11420.049805	10830.327148	10895.830078	10895.830078	16681503537

Figure 1: Historical Data of Cryptocurrency

I have collected data using **Yahoo Finance API**, which has a collection of libraries/APIs/methods to acquire historical and real time data for a variety of financial markets and products. The merits of Yahoo Finance API are completely free, impressive range of data, quick and easy to set yourself up and simple. I have used persistent data because it is deliberated long-lasting at rest with the approaching and leaving of software and devices. The free Yahoo Finance API has long been a consistent data source for many depositors.

Data Analysis

Data analysis denotes to the method of handling raw data to expose valuable insights and present decisions. During this practice, a data analyst or data scientist will organize, transform, and model a dataset. Data analysis confirms that this data is improved and ready to use. Below table shows the complete description of features of dataset:

Table 1: Feature Description

Date	The time at which each instance or entry has been composed from crypto-currency stock marketplace.
Open	Open value for every day according to individually timestamp.
High	Highest value on that day in which statistics has been composed.
Low	Lowest value on that day in which statistics has been composed.
Close	Final value of bitcoin on that day in which statistics has been composed.
Adj Close	The adjusted final value amends a stock's final value to mirror that stock's rate after accounting for any business activities.
Volume	Turnovers values in crypto-coins.

A univariate analysis contains of predicting cryptocurrencies made by comments fitting to a particular variable logged over time, in our case the opening value of the crypto-coins are measured. A multivariate analysis is a predictive analysis in which the dataset contains of the comments of numerous variables.

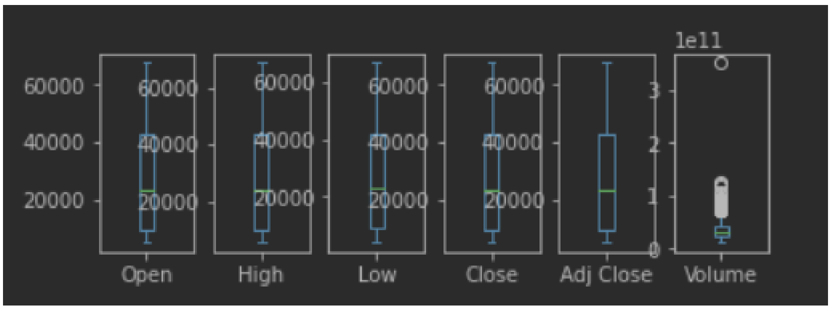

Figure 2: Univariate Analysis

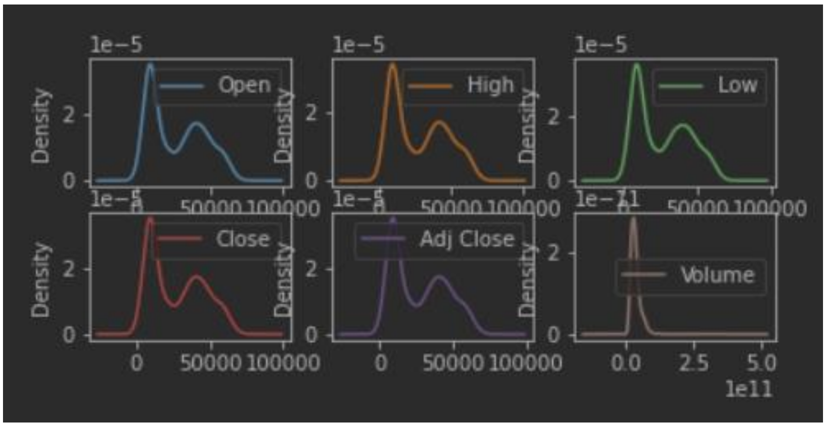

Figure 3: Multivariate Analysis

	Open	High	Low	Close	Adj Close	Volume
Open	1.000000	0.999066	0.998231	0.997605	0.997605	0.359114
High	0.999066	1.000000	0.998115	0.998942	0.998942	0.368866
Low	0.998231	0.998115	1.000000	0.998772	0.998772	0.338066
Close	0.997605	0.998942	0.998772	1.000000	1.000000	0.355559
Adj Close	0.997605	0.998942	0.998772	1.000000	1.000000	0.355559
Volume	0.359114	0.368866	0.338066	0.355559	0.355559	1.000000

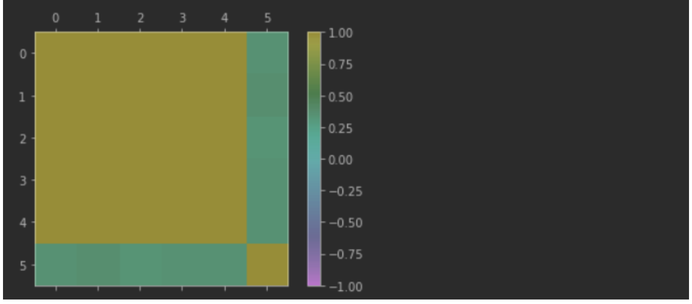
Figure 4: Correlation Analysis

Data Pre-processing

Data preprocessing is the method of altering fresh information into a comprehensible setup. It is too a significant phase in data mining as we can't work through fresh data. The superiority of the data would be tested before relating machine learning or data mining procedures.

1) To build a machine learning project, the primary object essential is a dataset as a machine learning project absolutely works on input. The composed data for a specific problematic in an appropriate setup is recognized as the dataset.

2) Data cleaning is the method to eliminate improper data, unfinished data and inexact data from the input source, and it also substitutes the missing values. There are some practices in data cleaning. Missing values can also be completed but it is not suggested when that dataset is large.

The feature's mean rate can be used to switch the missing value when the data is usually spread wherein in the case of non-normal scattering median rate of the feature can be used. **In my dataset, missing values do not exist; I checked and proceeded with splitting the data into the training and test set.**

```
#Checking the data for missing values
missing_data = dataset[dataset.isna().any(axis=1)]
print('\n The missing data are \n', dataset)
len(missing_data)
```

Figure 5: Checking for missing values in the dataset

3) This is one of the vital steps of data preprocessing as by doing this, we can boost the performance of our machine learning model.

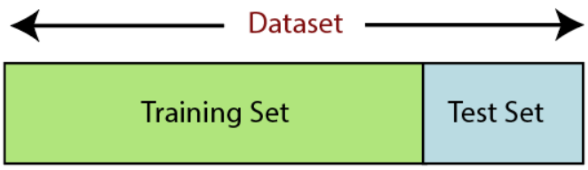

Figure 6: Splitting of Training and Test Data

```python
#train and test data sizes
train_size = int(len(data_scaled)*0.70)
test_size = len(data_scaled) - train_size
train_size, test_size
```

```python
#splitting into training ang testing sets
data_train, data_test = data_scaled[0:train_size,:], data_scaled[train_size:len(data_scaled),:1]
len(data_train), len(data_test)
```

Figure 7: Training and Test Data

I have divided my dataset into 70% training data and 30% test data.

Predicting future cryptocurrency using LSTM neural network

1) The primary stage we do to our data is to normalize its prices. The objective of normalization is to alter the standards of numerical records in the input to a communal measure, without twisting changes in the arrays of standards.

```python
#Using MinMaxScaler for normalising data between 0 & 1
from sklearn.preprocessing import MinMaxScaler

normalizer = MinMaxScaler(feature_range=(0,1))
data_scaled = normalizer.fit_transform(np.array(data).reshape(-1, 1))
len(data_scaled), len(data)
```

Figure 8: Normalization using MinMaxScaler

2) About the Algorithm:

Recurrent neural networks are a strong and influential type of neural network and are considered one of the supreme professional algorithms because they are the only ones with internal memory. The algorithm performs very well for sequential data such as time series, speech, text, financial data, audio, video, weather, and more. RNNs are able to form a much deeper considerate of a sequence and its context compared to other algorithms. In an RNN, the information goes through a cycle. When making a decision, it considers the current input and also what it has learned from the inputs it has received previously.

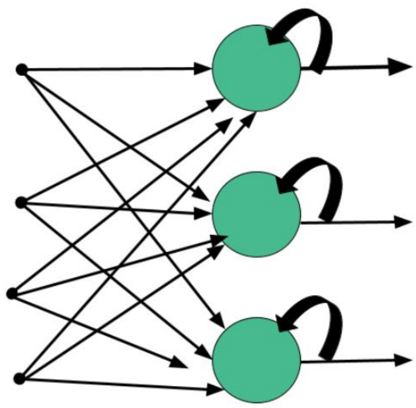

Figure 9: Recurrent neural networks

Long short-term memory networks are an extension of recurrent neural networks, which fundamentally extend the memory. Therefore it is well suitable to study from significant practices that have very long time lags in between. LSTMs permit RNNs to recollect inputs over a long period of time. This is because LSTMs contain information in a memory, much like the memory of a computer. The LSTM can read, write and delete information from its memory.
In an LSTM you have three gates: input, forget and output gate. These gates determine whether or not to let new input in (input gate), delete the information because it isn't important (forget gate), or let it impact the output at the current time step (output gate). Below is an illustration of an RNN with its three gates:

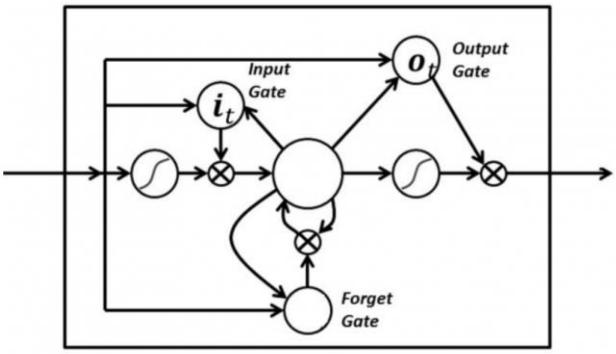

Figure 10: Long Short-term Memory Networks

The gates in an LSTM are analog in the form of sigmoid, denotation they range from zero to one. The statement that they are analog empowers them to do backpropagation.

3) Specifications about the process.ipynb file:

- Used Pycharm 3.9 for this project which is a dedicated Python Integrated Development Environment (IDE) providing an extensive sort of necessary tools for Python developers which are firmly incorporated to generate a convenient environment for productive Python, web, and data science development.
- Collected historical data from Yahoo Finance of last 3 years with interval of 1 day with the help of Yahoo Finance API.
- Then executed data cleaning process and fortunately no missing values found.
- Performed univariate and multivariate analysis for the dataset. We observed that addition of features to the dataset did not result in better forecasts, but performance and sometimes also results worsened. For this purpose, we obviously use the univariate analysis and only open feature for finest results.

```
#Performing uni-variate analysis
dataset.plot(kind='box', subplots= True, layout = (2,6), sharex = False)

#Density plot and skewness
dataset.plot(kind='density', subplots= True, layout =(3,3), sharex= False)
plt.show()
print(dataset.skew())
```

Figure 11: Univariate Analysis

```
#Performing multivariate analysis
correlations = dataset.corr(method='pearson')
print(correlations)

#Correlation matrix plot
corr_fig = plt.figure()
axis = corr_fig.add_subplot(111)
corr = axis.matshow(correlations, vmin=-1, vmax=1)

corr_fig.colorbar(corr)
ticks = np.arange(0,6,1)

axis.set_xticks(ticks)
axis.set_yticks(ticks)

plt.show()
```

Figure 12: Multivariate Analysis

- Using the library Matplotlib, we have graphed the Open values from the dataset.

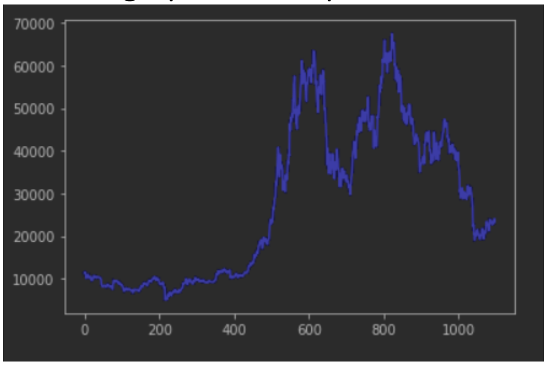

Figure 13: Open Values of the Dataset

- The moving average (MA) is a calculation of the average price that a crypto is tradeoff at over a fixed period. I have executed moving average for 100 and 200 days.

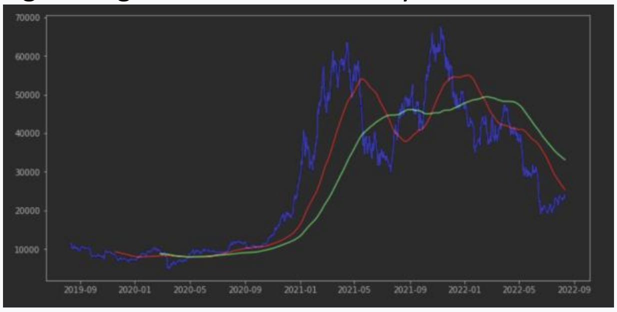

Figure 14: Moving Average for 100 vs 200 days

- Transforming features using MinMaxScaler by scaling each feature to a certain sort. This estimator measures and interprets each feature independently such that it is in the given sort on the training set, i.e. between zero and one.
- Then assigning the sizes of the training and testing data, splitting them into 70% training data and 30% testing data.
- Then creating dataset in time series for LSTM model and taking 100 days price as one record for training.

```
#creating dataset in time series for LSTM model
def create_data(record, step):
    x, y = [], []
    for i in range(len(record)-step-1):
        d = record[i:(i+step), 0]
        x.append(d)
        y.append(record[i + step, 0])
    return np.array(x), np.array(y)

#Taking 100 days price as one record for training
training_days = 100
x_train, y_train = create_data(data_train, training_days)
x_test, y_test = create_data(data_test, training_days)
x_train.shape, y_train.shape
```

Figure 15: Time Series for LSTM model and 100 days price for training

➢ Reshaping the data to fit into the LSTM model.

```
x_train = x_train.reshape(x_train.shape[0], x_train.shape[1], 1)
x_test = x_test.reshape(x_test.shape[0], x_test.shape[1], 1)
```

Figure 16: Reshaping data into LSTM model

➢ So as to construct the LSTM, we need to import a pair of modules from Keras:

Sequential for setting the neural network.

Dense for totaling a tightly connected neural network layer.

LSTM for totaling the Long Short-Term Memory layer.

```
#Creating LSTM model using keras
from keras.models import Sequential
from keras.layers import Dense, LSTM

algorithm = Sequential()
algorithm.add(LSTM(units = 50, return_sequences = True, input_shape = (x_train.shape[1], 1)))
algorithm.add(LSTM(units = 50, return_sequences = True))
algorithm.add(LSTM(units = 50))
algorithm.add(Dense(units = 1, activation= 'linear'))
algorithm.summary()
```

Figure 17: Construction of LSTM model from Keras

We improve the LSTM layer with the subsequent opinions:

1) 50 units for the dimensionality of the production plane.
2) return_sequences=True which decides whether to reoccurrence the preceding output in the output, or full sequence.
3) input_shape as the form of our training set.

When essential the Dense layer that stipulates the output of 1 unit. After this, we collect our model using the adam optimizer and set loss as the mean_squarred_error. Next, we fit the model to run on 100 epochs with a batch size of 64 depending on the specifications of your system, this might take a little minutes to finish successively.

```
#training model with adam optimizer and mean squared error loss function
algorithm.compile(loss='mean_squared_error', optimizer='adam')
algorithm.fit(x_train, y_train, validation_data=(x_test, y_test), epochs=100, batch_size=64)
```

Figure 18: Training model using Adam optimizer and Mean Squared Error

➢ Plotting loss, it shows that the loss has decreased significantly and model trained well.

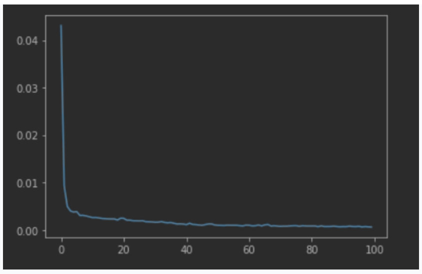

Figure 19: Loss

- Predicting training and testing data and using inverse transform, we achieve the actual value.

```
#Predicting train and test data
predicting_train = algorithm.predict(x_train)
predicting_test = algorithm.predict(x_test)
#Inverse transform to get actual value
predicting_train = normalizer.inverse_transform(predicting_train)
predicting_test = normalizer.inverse_transform(predicting_test)
```

Figure 20: Prediction and Actual Value using Inverse Transform

- Calculating MSE performance metrics for training data and testing data.

```
#Calculating MSE performance metrics for training data
import math
from sklearn.metrics import mean_squared_error
math.sqrt(mean_squared_error(y_train, predicting_train))
```

```
#Calculating MSE performance metrics for testing data
math.sqrt(mean_squared_error(y_test, predicting_test))
```

Figure 21: MSE Performance Metrics

- Comparing using visuals.

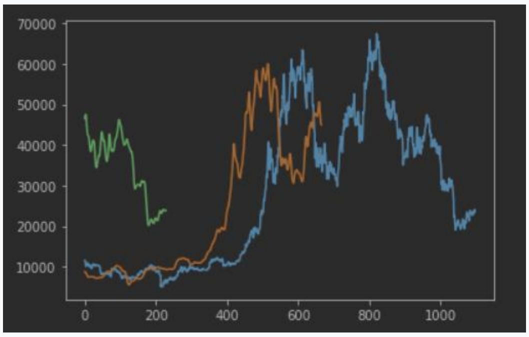

Figure 22: Data Visualization

- Comparing the predicted data to create uniform data visualization.

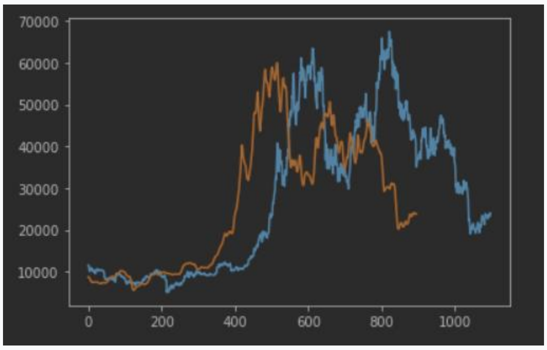

Figure 23: Predicted Data Visualization

- Creating and listing list of last 100 data.

- Predicting next 30 days price using the current data and it will predict in sliding window manner.
- Creating a dummy space to plot graph one after other.

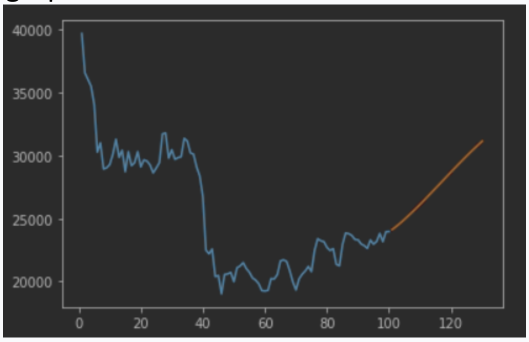

Figure 24: Dummy graph

- Extends fill the missing value with approximate value.

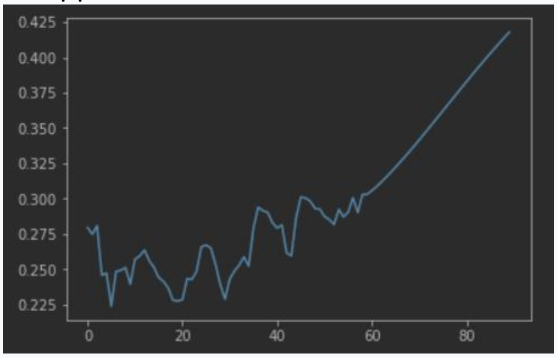

Figure 25: Fills the Missing Value

- Creating and plotting final results with predicted value for next 30 days.

Figure 26: Predicted value for next 30 days

4) Specifications about the main.py file:
   - Importing libraries.

```python
# Importing libraries
import numpy as np
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt
import streamlit as st
import yfinance as yf
from keras.models import Sequential
```

Figure 27

   - Title in the web page.

Cryptocurrency Prediction System for 50 Crypto-coins

Figure 28

- Displaying 50 crypto-coins in the form of drop-down menu and the selection of coin.

```
# Displaying 50 crypto-coins in the form of drop-down menu
crypto_symbol = st.selectbox('Select a Cryptocurrency:', ('BTC-USD', 'ETH-USD', 'USDT-USD', 'USDC-USD', 'BNB-USD',
                                                          'XRP-USD', 'BUSD-USD', 'ADA-USD', 'SOL-USD', 'DOT-USD',
                                                          'DOGE-USD', 'HEX-USD', 'DAI-USD', 'MATIC-USD', 'WTRX-USD',
                                                          'AVAX-USD', 'UNI1--USD', 'SHIB-USD', 'TRX-USD', 'STETH-USD',
                                                          'WBTC-USD', 'ETC-USD', 'LEO-USD', 'LTC-USD', 'YOUC-USD',
                                                          'FTT-USD', 'NEAR-USD', 'LINK-USD', 'CRO-USD', 'ATOM-USD',
                                                          'XLM-USD', 'FLOW-USD', 'XMR-USD', 'BCH-USD', 'ALGO-USD',
                                                          'BTCB-USD', 'VET-USD', 'APE3-USD', 'FIL-USD', 'ICP-USD',
                                                          'MANA-USD', 'XCN1-USD', 'SAND-USD', 'BIT1-USD', 'XTZ-USD',
                                                          'HBAR-USD', 'THETA-USD', 'NT-USD', 'TONCOIN-USD', 'AXS-USD'))
```

Figure 29

Figure 30

- Displaying the raw data of last 3 years.

Raw data of last 3 years

	Open	High	Low	Close	Adj Close	Volume
count	1,097.0000	1,097.0000	1,097.0000	1,097.0000	1,097.0000	1,097.0000
mean	27,561.2917	28,258.0096	26,774.4772	27,568.6452	27,568.6452	35,136,725,748.6527
std	18,196.0699	18,673.7236	17,635.7135	18,184.4106	18,184.4106	19,071,739,267.9503
min	5,002.5781	5,331.8340	4,106.9810	4,970.7881	4,970.7881	11,445,355,859.0000
25%	9,644.0762	9,782.3066	9,471.8467	9,648.7178	9,648.7178	23,160,469,766.0000
50%	23,291.4238	24,024.4902	22,802.6465	23,314.1992	23,314.1992	31,646,080,921.0000
75%	42,944.9766	43,903.0195	41,982.6172	42,892.9570	42,892.9570	42,213,940,994.0000
max	67,549.7344	68,789.6250	66,382.0625	67,566.8281	67,566.8281	350,967,941,479.0000

Figure 31

- Displaying Opening Price vs Time Chart.

Figure 32

- Displaying Opening Price vs Time Chart with 100MA.

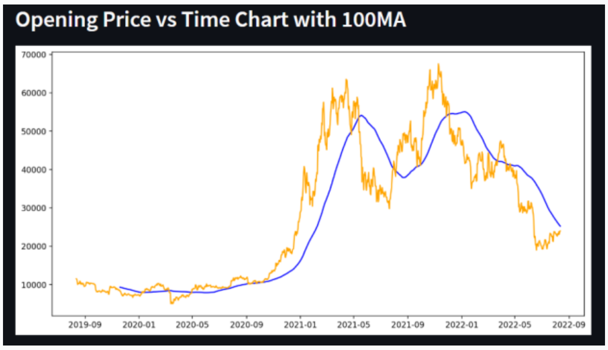

Figure 33

- Displaying Opening Price vs Time Chart with 100MA & 200MA.

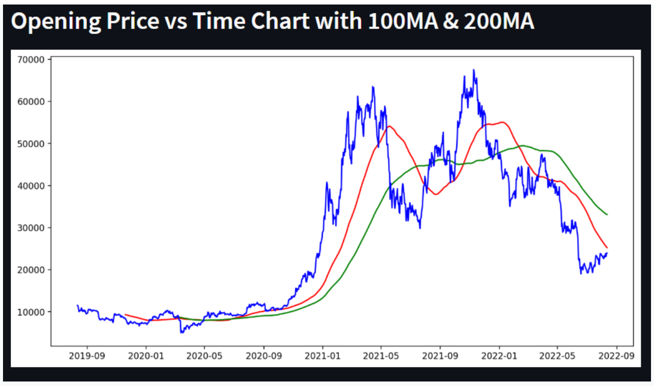

Figure 34

➢ Displaying prediction chart and approximate value for next 30 days.

Figure 35

Challenges

The challenges we faced are feature extraction, choosing suitable model, datasets in crypto are not predominantly precise and knowledge reusability.

Conclusion

RNNs and LSTM are outstanding knowledges and has boundless designs that can be used to examine and foresee time-series evidence. The motivation of the article was to include a simple project, try dissimilar stuffs and want to play with hyper parameters and layers.

References

1) Analytics Vidhya. (2021). *Bitcoin Price Prediction Using Recurrent Neural Networks and LSTM*. [online] Available at: https://www.analyticsvidhya.com/blog/2021/05/bitcoin-price-prediction-using-recurrent-neural-networks-and-lstm/.
2) Analytics Vidhya. (2022). *DogeCoin Prediction Using Time Series Analysis*. [online] Available at: https://www.analyticsvidhya.com/blog/2022/03/dogecoin-prediction-using-time-series-analysis/.
3) Anunaya, S. (2021). *Data Preprocessing in Data Mining -A Hands On Guide*. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2021/08/data-preprocessing-in-data-mining-a-hands-on-guide/.
4) Brownlee, J. (2016). *Sequence Classification with LSTM Recurrent Neural Networks in Python with Keras*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/sequence-classification-lstm-recurrent-neural-networks-python-keras/.
5) coderzcolumn.com. (n.d.). *Keras: LSTM Networks For Text Classification Tasks by Sunny Solanki*. [online] Available at: https://coderzcolumn.com/tutorials/artificial-intelligence/keras-lstm-networks-for-text-classification-tasks [Accessed 12 Aug. 2022].

6) Curiously. (n.d.). *Cryptocurrency price prediction using LSTMs | TensorFlow for Hackers (Part III)*. [online] Available at: https://curiously.com/posts/cryptocurrency-price-prediction-in-tensorflow-2/ [Accessed 12 Aug. 2022].
7) Guven, B. (2021). *Step-by-Step Guide — Building a Prediction Model in Python*. [online] Medium. Available at: https://towardsdatascience.com/step-by-step-guide-building-a-prediction-model-in-python-ac441e8b9e8b.
8) Hansun, S., Wicaksana, A. and Khaliq, A.Q.M. (2022). Multivariate cryptocurrency prediction: comparative analysis of three recurrent neural networks approaches. *Journal of Big Data*, 9(1). doi:10.1186/s40537-022-00601-7.
9) Josselyn, A. (2022). *Utilizing Long Short-Term Memory Networks to Forecast the Price of Bitcoin in R*. [online] Medium. Available at: https://medium.datadriveninvestor.com/utilizing-long-short-term-memory-networks-to-forecast-the-price-of-bitcoin-in-r-8a77a6512ddc [Accessed 12 Aug. 2022].
10) KDnuggets. (n.d.). *Using a Keras Long Short-Term Memory (LSTM) Model to Predict Stock Prices*. [online] Available at: https://www.kdnuggets.com/2018/11/keras-long-short-term-memory-lstm-model-predict-stock-prices.html.
11) MonkeyLearn Blog. (2021). *What Is Data Preprocessing & What Are The Steps Involved?* [online] Available at: https://monkeylearn.com/blog/data-preprocessing/.
12) Phemex. (n.d.). *Moving Averages: How to trade Crypto with the 20 MA? - Phemex Academy*. [online] Available at: https://phemex.com/academy/moving-averages#:~:text=This%20indicator%20is%20used%20to.
13) Python Engineer. (n.d.). *Build A Stock Prediction Web App In Python*. [online] Available at: https://www.python-engineer.com/posts/stockprediction-app/ [Accessed 12 Aug. 2022].
14) Sagar, A. (2019). *Cryptocurrency Price Prediction Using Deep Learning*. [online] Medium. Available at: https://towardsdatascience.com/cryptocurrency-price-prediction-using-deep-learning-70cfca50dd3a.
15) Tanwar, S., Patel, N.P., Patel, S.N., Patel, J.R., Sharma, G. and Davidson, I.E. (2021). Deep Learning-Based Cryptocurrency Price Prediction Scheme With Inter-Dependent Relations. *IEEE Access*, [online] 9, p.138633. Available at: https://www.academia.edu/61459903/Deep_Learning_based_Cryptocurrency_Price_Prediction_Scheme_with_Inter_dependent_Relations [Accessed 12 Aug. 2022].
16) TensorFlow. (n.d.). *Text classification with an RNN*. [online] Available at: https://www.tensorflow.org/text/tutorials/text_classification_rnn.
17) Uras, N., Marchesi, L., Marchesi, M. and Tonelli, R. (2020). Forecasting Bitcoin closing price series using linear regression and neural networks models. *PeerJ Computer Science*, 6, p.e279. doi:10.7717/peerj-cs.279.
18) www.javatpoint.com. (n.d.). *Data Preprocessing in Machine learning - Javatpoint*. [online] Available at: https://www.javatpoint.com/data-preprocessing-machine-learning.
19) www.youtube.com. (n.d.). *Live Stock Market Forecasting for next 'n' days using LSTM*. [online] Available at: https://www.youtube.com/watch?v=5Gm3bWNBoWQ [Accessed 12 Aug. 2022].
20) Zhang, S., Li, M. and Yan, C. (2022). The Empirical Analysis of Bitcoin Price Prediction Based on Deep Learning Integration Method. *Computational Intelligence and Neuroscience*, 2022, pp.1–9. doi:10.1155/2022/1265837.