

MACHINE LEARNING PROJECT

PROJECT REPORT

Cryptocurrency price prediction



PREPARED AND PRESENTED BY:

NAME: NITASH CHUMBER
ROLL NO.: 20103102

NAME: PRIYANKA SAINI
ROLL NO.: 20103111

WHAT IS CRYPTOCURRENCY?



A cryptocurrency or crypto is a virtual currency secured by cryptography. It is designed to work as a medium of exchange, where individual ownership records are stored in a computerized database. The defining trait of a cryptocurrency is that they are not issued by the government agency of any country making them immune against any interference and manipulation from them. Some of the well-known cryptocurrencies are Bitcoin, Dogecoin, Binance coin, Ave coin, Ethereum, Litecoin, Tron coin.

CHALLENGE

Our challenge is to forecast cryptocurrency prices using all the trading features like price, volume, open, high, low values present in the dataset.

The cryptocurrencies used in this project are:

1. Aave coin
2. Binance coin
3. Bitcoin
4. Cardano
5. Dogecoin
6. Ethereum
7. Litecoin
8. Solana

METHODOLOGY:

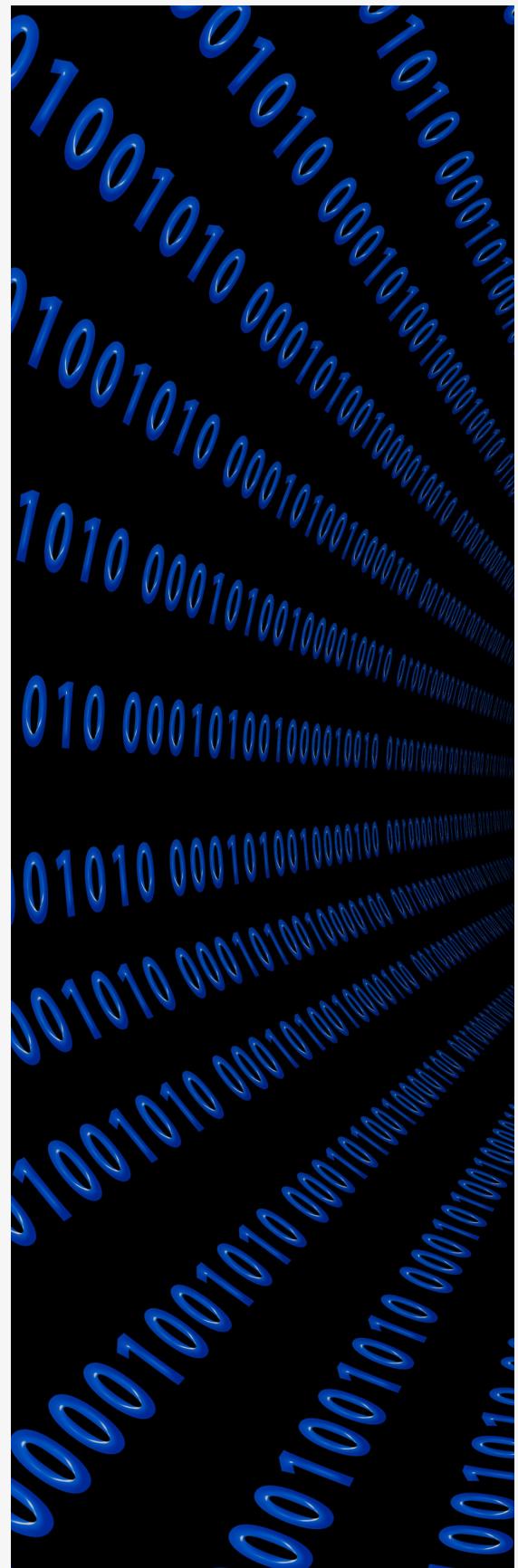
We will be going through the following steps to predict cryptocurrency prices:

1. Import the required libraries.
2. Get real-time cryptocurrency data.
3. Predict the price of cryptocurrency using the AutoTS model.
4. Repeat the same for all the cryptocurrencies.
5. Get the results.

DATASET

We have taken the datasets for some well-known cryptocurrencies, the datasets contain a total of five features, i.e,

- **Close Price:** It is the market close price for currency for that particular day.
- **High Price:** It is the highest price for currency for the day.
- **Low price:** It is the lowest price for currency for that day.
- **Open price:** It is the market open price for currency for that day.
- **Volume:** The volume of currency that is being in the trade for that day.
- **MarketCap:** Market capitalization (or market cap) is the total value of all the coins that have been mined.



CODE:

LET'S GET STARTED WITH THE CODE:

1. **Import modules:** We started with loading all the libraries and dependencies required.

```
pip install autots  
import numpy as np  
import pandas as pd
```

- **Autots:** It is an automatic machine learning library in Python which is developed for the task of automatic time series forecasting. We can use this library for any task of time series forecasting such as predicting the stock prices for the next n number of days.
- **Numpy:** It is used for working with arrays. It provides tools for integrating C, C++, etc. It is also useful in linear algebra, random number capability, etc. NumPy array can also be used as an efficient multi-dimensional container for generic data.
- **Pandas:** It is a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive.

2. Exploring data:

The datasets can be downloaded from here:

<https://www.kaggle.com/datasets/sudalairajkumar/cryptocurrencypricehistory/download>

Now that we have imported the modules, we will be loading the datasets and finding their shape and print the tail.

1. For aave coin:

```
aave = pd.read_csv("coin_Aave.csv")
print("History of AAve coin USD:",aave.shape)
print(aave.tail())
```

```
History of AAve coin USD: (275, 10)
   Sno    Name Symbol        Date      High       Low     Open \
270  271   Aave   AAVE  02-07-2021 23:59  234.483957  215.951539  230.814119
271  272   Aave   AAVE  03-07-2021 23:59  265.107342  228.485206  234.208478
272  273   Aave   AAVE  04-07-2021 23:59  289.001124  248.285491  259.399426
273  274   Aave   AAVE  05-07-2021 23:59  317.387234  263.433881  277.110533
274  275   Aave   AAVE  06-07-2021 23:59  346.714780  307.997525  307.997525

               Close      Volume  Marketcap
270  233.948437  256639522.8  3.001542e+09
271  259.655225  385377799.5  3.331546e+09
272  277.038792  427571943.3  3.555054e+09
273  307.829079  793140860.1  3.950269e+09
274  316.898507  988705452.8  4.066776e+09
```

2. For Binance coin:

```

bn = pd.read_csv("coin_BinanceCoin.csv")
print("History of Binance coin coin USd:",bn.shape)
print(bn.tail())

```

History of Binance coin coin USd: (1442, 10)						
	SNo	Name	Symbol	Date	High	Low
1437	1438	Binance Coin	BNB	2021-07-02 23:59:59	290.621683	277.350302
1438	1439	Binance Coin	BNB	2021-07-03 23:59:59	302.605867	283.434010
1439	1440	Binance Coin	BNB	2021-07-04 23:59:59	314.713019	292.787370
1440	1441	Binance Coin	BNB	2021-07-05 23:59:59	307.684855	293.523465
1441	1442	Binance Coin	BNB	2021-07-06 23:59:59	321.520965	302.195584
		Open	Close	Volume	Marketcap	
1437	287.754456	287.423094	1.133633e+09	4.410016e+10		
1438	287.215592	298.237117	1.113777e+09	4.575938e+10		
1439	298.113557	307.732096	1.387396e+09	4.721623e+10		
1440	307.684855	302.377980	1.504870e+09	4.639473e+10		
1441	302.195584	320.934802	2.203265e+09	4.924196e+10		

3. For Bitcoin:

```

btc = pd.read_csv("coin_Bitcoin.csv")
print("History of Bitcoin coin USD:",btc.shape)
print(btc.tail())

```

History of Bitcoin coin USD: (2991, 10)							
	SNo	Name	Symbol	Date	High	Low	
.	2986	2987	Bitcoin	BTC	2021-07-02 23:59:59	33939.588699	32770.680780
.	2987	2988	Bitcoin	BTC	2021-07-03 23:59:59	34909.259899	33402.696536
.	2988	2989	Bitcoin	BTC	2021-07-04 23:59:59	35937.567147	34396.477458
.	2989	2990	Bitcoin	BTC	2021-07-05 23:59:59	35284.344430	33213.661034
.	2990	2991	Bitcoin	BTC	2021-07-06 23:59:59	35038.536363	33599.916169
		Open	Close	Volume	Marketcap		
2986	33549.600177	33897.048590	3.872897e+10	6.354508e+11			
2987	33854.421362	34668.548402	2.438396e+10	6.499397e+11			
2988	34665.564866	35287.779766	2.492431e+10	6.615748e+11			
2989	35284.344430	33746.002456	2.672155e+10	6.326962e+11			
2990	33723.509655	34235.193451	2.650126e+10	6.418992e+11			

4. For Cardano coin:

```
car = pd.read_csv("coin_Cardano.csv")
print("History of Cardano coin Usd:",car.shape)
print(car.tail())
```

History of Cardano coin Usd:

SNo	Name	Symbol	Date	High	Low	Open	\	
1369	1370	Cardano	ADA	2021-07-02 23:59:59	1.394397	1.286607	1.332942	
1370	1371	Cardano	ADA	2021-07-03 23:59:59	1.441714	1.359664	1.394152	
1371	1372	Cardano	ADA	2021-07-04 23:59:59	1.493717	1.382153	1.404008	
1372	1373	Cardano	ADA	2021-07-05 23:59:59	1.461221	1.379284	1.461221	
1373	1374	Cardano	ADA	2021-07-06 23:59:59	1.456887	1.393282	1.404712	

	Close	Volume	Marketcap
1369	1.394397	2.159410e+09	4.454587e+10
1370	1.406836	2.028094e+09	4.494324e+10
1371	1.458184	1.806362e+09	4.658364e+10
1372	1.404898	1.759461e+09	4.488134e+10
1373	1.418053	1.477700e+09	4.530158e+10

5. For Dogecoin:

```
dc = pd.read_csv("coin_Dogecoin.csv")
print("History of Doge coin:",dc.shape)
print(dc.tail())
```

History of Doge coin: (2760, 10)

SNo	Name	Symbol	Date	High	Low	\	
2755	2756	Dogecoin	DOGE	2021-07-02 23:59:59	0.247997	0.238848	
2756	2757	Dogecoin	DOGE	2021-07-03 23:59:59	0.250214	0.242454	
2757	2758	Dogecoin	DOGE	2021-07-04 23:59:59	0.252567	0.243425	
2758	2759	Dogecoin	DOGE	2021-07-05 23:59:59	0.246419	0.227838	
2759	2760	Dogecoin	DOGE	2021-07-06 23:59:59	0.241910	0.229842	

	Open	close	Volume	Marketcap
2755	0.243982	0.245264	1.321471e+09	3.194925e+10
2756	0.245106	0.246411	9.170158e+08	3.210491e+10
2757	0.246425	0.246483	9.735115e+08	3.211767e+10
2758	0.246419	0.231614	1.267949e+09	3.018344e+10
2759	0.231216	0.234422	1.265920e+09	3.055252e+10

6. For Ethereum coin:

```
eth = pd.read_csv("coin_Ethereum.csv")
print("History of Ethereum coin:",aave.shape)
print(eth.tail())
```

```
History of Ethereum coin: (275, 10)
   SNo      Name Symbol          Date      High      Low \
2155  2156  Ethereum    ETH  2021-07-02 23:59:59  2155.596496  2021.824808
2156  2157  Ethereum    ETH  2021-07-03 23:59:59  2237.567155  2117.590013
2157  2158  Ethereum    ETH  2021-07-04 23:59:59  2384.286857  2190.837703
2158  2159  Ethereum    ETH  2021-07-05 23:59:59  2321.922836  2163.041394
2159  2160  Ethereum    ETH  2021-07-06 23:59:59  2346.294874  2197.919385

      Open      Close      Volume      Marketcap
2155  2109.892677  2150.040364  3.179621e+10  2.505527e+11
2156  2150.835025  2226.114282  1.743336e+10  2.594475e+11
2157  2226.550382  2321.724112  1.878711e+10  2.706217e+11
2158  2321.922836  2198.582464  2.010379e+10  2.562978e+11
2159  2197.919385  2324.679449  2.089186e+10  2.710286e+11
```

7. For Litecoin:

```
ltc = pd.read_csv("coin_Litecoin.csv")
print("History of Litecoin coin:",ltc.shape)
print(ltc.tail())
```

```
History of Litecoin coin: (2991, 10)
   SNo      Name Symbol          Date      High      Low \
2986  2987  Litecoin    LTC  2021-07-02 23:59:59  138.787700  130.935471
2987  2988  Litecoin    LTC  2021-07-03 23:59:59  141.356011  134.945288
2988  2989  Litecoin    LTC  2021-07-04 23:59:59  147.836059  137.096427
2989  2990  Litecoin    LTC  2021-07-05 23:59:59  144.849333  134.960263
2990  2991  Litecoin    LTC  2021-07-06 23:59:59  142.703568  135.924837

      Open      Close      Volume      Marketcap
2986  137.299274  136.943696  1.418981e+09  9.141322e+09
2987  136.930584  140.279688  1.236494e+09  9.364008e+09
2988  140.317998  144.905849  1.431657e+09  9.672815e+09
2989  144.849333  138.073246  1.338246e+09  9.216723e+09
2990  137.951668  138.985636  1.504907e+09  9.277627e+09
```

8. For Solana coin:

```
esol = pd.read_csv("coin_Solana.csv")
print("History of Solana coin:",sol.shape)
print(sol.tail())
```

```
History of Solana coin: (452, 10)
   SNo      Name Symbol        Date      High      Low     Open \
447  448    Solana    SOL  2021-07-02 23:59:59  34.031786  31.479924  33.306310
448  449    Solana    SOL  2021-07-03 23:59:59  35.404770  33.298475  34.015575
449  450    Solana    SOL  2021-07-04 23:59:59  35.502372  33.555737  34.495117
450  451    Solana    SOL  2021-07-05 23:59:59  34.461824  32.482692  34.282550
451  452    Solana    SOL  2021-07-06 23:59:59  34.978319  32.930307  32.930307

      close      volume   Marketcap
447  34.020482  4.402988e+08  9.275257e+09
448  34.478816  3.270200e+08  9.400216e+09
449  34.310601  3.034205e+08  9.354354e+09
450  32.984588  3.138393e+08  8.992833e+09
451  34.269140  3.653360e+08  9.343050e+09
```

All the Datasets have been loaded. Now, we can use them for predicting future prices.

3. APPLYING MACHINE LEARNING MODEL:

a. Creating the Model:

Here, we will be using the AutoTS model and import the model into the program.

For calling the model with our dataset we first need to define the model and pass the parameters according to our requirements. While creating the model we will give below parameters:

- forecast_length
- frequency
- ensemble
- Drop data

```
from autots import AutoTS  
model = AutoTS(forecast_length=10, frequency='infer',  
ensemble='simple', drop_data_older_than_periods=200)
```

b. Fitting the Model:

Then, create an AutoTS model object in order to fit the data points into the model using the fit function and then predict the prices for all data points using the predict function.

This step is different for all the cryptocurrencies.

Example: for Aave coin:

```
model = model.fit(aave, date_col='Date',  
value_col='Close', id_col=None)
```

This step will take some time as it will run our data through different models and check the best model for our data. By using the above code, we will predict the future price for all the cryptocurrencies.

c. Predicting and Forecasting:

We can use AutoTS forecast and predict function for this step.

```
prediction = model.predict()  
forecastaave = prediction.forecast
```

The complete code for applying model on all cryptocurrencies is below:

1. For Aave coin:

```
from autots import AutoTS  
  
model = AutoTS(forecast_length=10, frequency='infer',  
ensemble='simple', drop_data_older_than_periods=200)  
model = model.fit(aave, date_col='Date', value_col='Close',  
id_col=None)  
prediction = model.predict()  
forecastaave = prediction.forecast
```

2. For Binance coin:

```
from autots import AutoTS  
model = AutoTS(forecast_length=10, frequency='infer',  
ensemble='simple', drop_data_older_than_periods=200)  
model = model.fit(bn, date_col='Date', value_col='Close',  
id_col=None)  
prediction = model.predict()  
forecastbn = prediction.forecast
```

3. For Bitcoin:

```
from autots import AutoTS  
model = AutoTS(forecast_length=10, frequency='infer',  
ensemble='simple', drop_data_older_than_periods=200)  
model = model.fit(btc, date_col='Date', value_col='Close',  
id_col=None)  
prediction = model.predict()  
forecastbtc = prediction.forecast
```

4. For Cardano:

```
from autots import AutoTS  
model = AutoTS(forecast_length=10, frequency='infer',  
ensemble='simple', drop_data_older_than_periods=200)  
model = model.fit(car, date_col='Date', value_col='Close',  
id_col=None)  
prediction = model.predict()  
forecastcar = prediction.forecast
```

5. For Dogecoin:

```
from autots import AutoTS  
model = AutoTS(forecast_length=10, frequency='infer',  
ensemble='simple', drop_data_older_than_periods=200)  
model = model.fit(dc, date_col='Date', value_col='Close',  
id_col=None)  
prediction = model.predict()  
forecastdc = prediction.forecast
```

6. For Ethereum:

```
from autots import AutoTS  
model = AutoTS(forecast_length=10, frequency='infer',  
ensemble='simple', drop_data_older_than_periods=200)  
model = model.fit(eth, date_col='Date', value_col='Close',  
id_col=None)  
prediction = model.predict()  
forecasteth = prediction.forecast
```

7. For Litecoin:

```
from autots import AutoTS  
model = AutoTS(forecast_length=10, frequency='infer',  
ensemble='simple', drop_data_older_than_periods=200)  
model = model.fit(ltc, date_col='Date', value_col='Close',  
id_col=None)  
prediction = model.predict()  
forecastltc = prediction.forecast
```

8. For Solana coin:

```
from autots import AutoTS  
model = AutoTS(forecast_length=10, frequency='infer',  
ensemble='simple', drop_data_older_than_periods=200)  
model = model.fit(sol, date_col='Date', value_col='Close',  
id_col=None)  
prediction = model.predict()  
forecastsol = prediction.forecast
```

4. Final results: By using while loop and if, elif conditions, the code is completed for all the datasets.

```
while True:
```

```
    p = str(input("Enter the crypto currency: "))
```

```
    if (p == 'aave' or p == 'Aave coin prediction' or p == 'Aave coin'):
```

```
        print("Predicted price for Aave is: ",forecastaaave ,'\n')
```

```
    elif (p == 'binance coin' or p == 'Binance Coin prediction' or p == 'binance' or p == 'Binance'):
```

```
        print("Predicted price for Binance is: ",forecastbn , '\n')
```

```
    elif (p == 'bitcoin' or p == 'btc prediction' or p == 'Bitcoin' or p == 'btc' or p == 'Btc'):
```

```
        print("Predicted price for Bitcoin is: ",forecastbtc , '\n')
```

```
    elif (p == 'Dexitogecoin' or p == 'doge' or p == 'dogecoin'):
```

```
        print("Predicted price for Dogecoin is: ",forecastdc ,'\n')
```

```
    elif (p == 'Ethereum coin' or p == 'Eth' or p == 'eth' or p == 'ethereum'):
```

```
        print("Predicted price for Ethereum is: ",forecasteth ,'\n')
```

```
    elif (p == 'Litecoin' or p == 'ltc' or p == 'Ltc' or p == 'litecoin'):
```

```
        print("Predicted price for Litecoin is: ",forecastltc , '\n')
```

```
    elif (p == 'tro' or p == 'solana' or p == 'sol' or p == 'solana prediction' or p == 'Solana'):
```

```
        print("Predicted price for Solana is: ",forecastsol,' \n')
```

```
    elif (p == 'Cardano price' or p == 'cardano price' or p == 'cardano' or p == 'Cardano'):
```

```
        print("Predicted price for Cardano is: ",forecastcar,' \n')
```

```
    elif p ==('exit'):
```

```
        break
```

```
    else:
```

```
        print("Sorry this coin is not yet listed!\n")
```

Output:

```
Enter the crypto currency: binance
Predicted price:           Close
2021-07-07  306.851735
2021-07-08  307.464537
2021-07-09  309.219940
2021-07-10  309.582243
2021-07-11  312.578499
2021-07-12  313.778142
2021-07-13  315.362740
2021-07-14  317.734013
2021-07-15  322.259915
2021-07-16  325.880939

Enter the crypto currency: eth
Predicted price:           Close
2021-07-07  2185.357798
2021-07-08  2172.287736
2021-07-09  2157.789651
2021-07-10  2128.248390
2021-07-11  2114.040070
2021-07-12  2093.496925
2021-07-13  2077.638230
2021-07-14  2050.726425
2021-07-15  2000.756550
2021-07-16  1946.646167

Enter the crypto currency: exit
```

5. Conclusion: The program to predict the price of cryptocurrency using machine learning is implemented successfully.