BITCOIN PREDICTION USING ML

Linear Regression

Introduction

This project focuses on the prediction of the prices of Bitcoin, the most in-demand crypto-currency of today's world.
We predict the prices accurately by gathering data available at Binance while taking various hyper-parameters into consideration which have affected the bitcoin prices until now.

Overview

Since Bitcoin's first appearance in 2009, it has changed the world's financial landscape substantially. The decentralized cryptocurrency has established itself as an asset class recognized by many asset managers, large investment banks and hedge funds. As the speed of mainstream adoption continues to soar, it is also leading investors to explore new ventures, such as crypto options and futures.

Overview

Bitcoin has been historically known to be more volatile than regulated stocks and commodities. Its most recent surge in late December 2020, early January 2021 has brought about a lot of questions and uncertainties about the future financial landscape. Bitcoin is traded at slightly below USD 50,000, which is no small feat considering it entered 2020 at around USD 7,200.

About our project

Importing libraries

At first import the required libraries

```
import pandas as pd
import requests
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

Loading dataset

Loading the Dataset from the binance api

```
[ ]  url = 'https://api.binance.com/api/v3/klines'


     r = requests.get(url,params={'symbol':'BTCUSDT','interval': '1d','limit':2000})
     r
```

```
<Response [200]>
```

formating the Dataset

format columns name

Out[3]:

```
['open_time',
 'open',
 'high',
 'low',
 'close',
 'volume',
 'close_time',
 'quote_asset_volume',
 'number_of_trades',
 'taker_buy_base_asset_volume',
 'taker_buy_quote_asset_volume',
 'ignore']
```

formating the Dataset

Create our dataframe and drop the useless columns

Date	open	high	low	close	volume
2019-04-19	5258.44000000	5320.00000000	5175.00000000	5258.68000000	24611.23632300
2019-04-20	5258.68000000	5333.42000000	5230.10000000	5291.73000000	19168.90827400
2019-04-21	5292.91000000	5314.35000000	5165.00000000	5256.14000000	25549.57093900
2019-04-22	5257.41000000	5400.00000000	5208.35000000	5357.14000000	29563.85230900
2019-04-23	5357.14000000	5600.00000000	5332.41000000	5493.31000000	41262.10391700

Preprocessing the Data

Creating a new column for better representing day-wise values and cleaning dataframe

```
df["open"] = pd.to_numeric(df["open"])
df["close"] = pd.to_numeric(df["close"])
df['Mean'] = (df['open'] + df['close'])/2
df=df.dropna()
df.head(5)
```

Date	open	high	low	close	volume	Mean
2019-04-19	5258.44	5320.00000000	5175.00000000	5258.68	24611.23632300	5258.560
2019-04-20	5258.68	5333.42000000	5230.10000000	5291.73	19168.90827400	5275.205
2019-04-21	5292.91	5314.35000000	5165.00000000	5256.14	25549.57093900	5274.525
2019-04-22	5257.41	5400.00000000	5208.35000000	5357.14	29563.85230900	5307.275
2019-04-23	5357.14	5600.00000000	5332.41000000	5493.31	41262.10391700	5425.225

Preprocessing the Data

create a new column *'Date'* storing the converted values of date index

```
df['Date']=df.index
df.head(5)
```

Date	open	high	low	close	volume	Mean	Date
2019-04-19	5258.44	5320.00000000	5175.00000000	5258.68	24611.23632300	5258.560	2019-04-19
2019-04-20	5258.68	5333.42000000	5230.10000000	5291.73	19168.90827400	5275.205	2019-04-20
2019-04-21	5292.91	5314.35000000	5165.00000000	5256.14	25549.57093900	5274.525	2019-04-21
2019-04-22	5257.41	5400.00000000	5208.35000000	5357.14	29563.85230900	5307.275	2019-04-22
2019-04-23	5357.14	5600.00000000	5332.41000000	5493.31	41262.10391700	5425.225	2019-04-23

Modeling – Linear regression

Linear regression is a statistical method for modeling relationships between a dependent variable with a given set of independent variables.

Modeling

select the columns which we will use to fit our model. We will also select the target variable ‘Close’.

```
required_colmn = ['low', 'high', 'open',  'volume','Mean']
output_label = 'close'
```

Modeling

divide the our dataset into train and test parts.

```
X_train, X_test, Y_train, Y_test = train_test_split(
df[required_colmn],
df[output_label],
test_size = 0.3
)
```

Modeling

Creating the Model

```
model = LinearRegression()
model.fit(X_train, Y_train)
```

```
LinearRegression()
```

Predicting the Prices

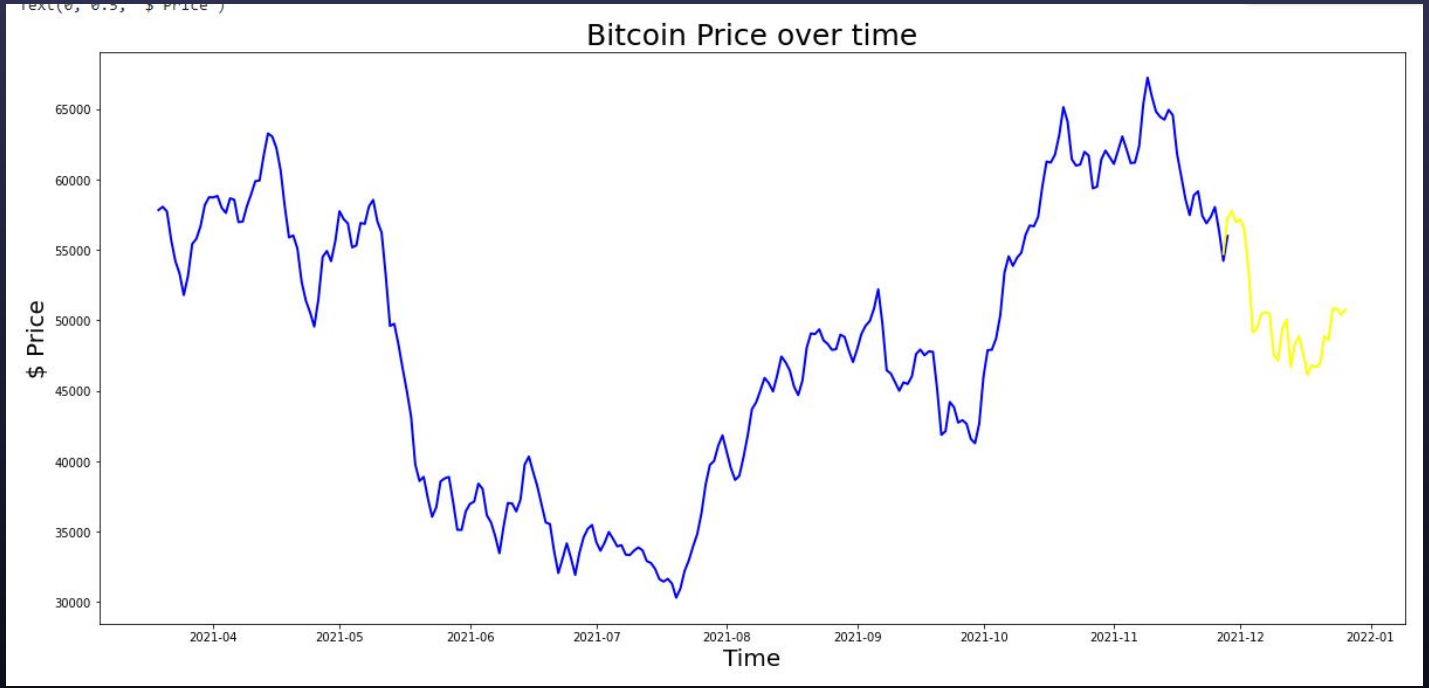
create a future dataset by shifting the original data by 20 days.

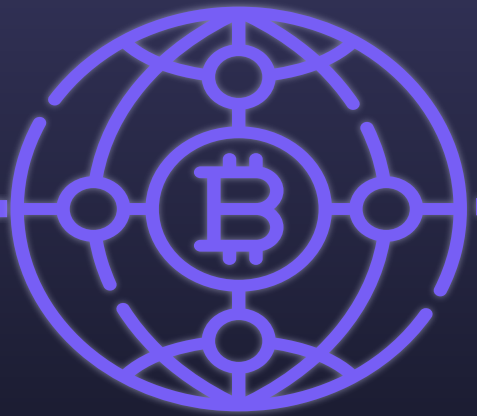
```
future_set = df.shift(20).tail(30)
```

```
prediction = model.predict(future_set[required_colmn])
```

Predicting the Prices

plot a graph showing the previous BTC prices in blue prices and our predicted prices in yellow

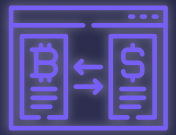




Did we achieve our Goals ?

We were successful to a good extent in predicting Bitcoin prices. But, in general, this will not work well in the real world. This is because history might not repeat itself. Also, new factors affecting Bitcoin's price can come into play with time.

Before we get to the end.



We'd like to thank everyone.

It was an honor to get to know All.

Thank you!

Any question ?

Group 2

Areej Thamir Kholoud Turki Amal