Import Crypto

April 12, 2022

[5]:
```
import yfinance as yf
import pandas as pd
import numpy as np
import seaborn as sns
from datetime import datetime
from datetime import timedelta
import plotly.graph_objects as go
from fbprophet import Prophet
from fbprophet.plot import plot_plotly, plot_components_plotly
import warnings
warnings.filterwarnings('ignore')
pd.options.display.float_format = '${:,.2f}'.format
```

[8]:
```
#downloading ETH historical data from yahoo finance with yfinance library
today = datetime.today().strftime('%Y-%m-%d')
start_date = '2016-01-01'

eth_df = yf.download('ETH-USD', start_date, today)
eth_df.tail()
```

```
[**********************100%**********************]  1 of 1 completed
```

[8]:

Date	Open	High	Low	Close	Adj Close	Volume
2022-04-08	$3,233.27	$3,301.61	$3,179.14	$3,192.07	$3,192.07	17557050669
2022-04-09	$3,191.98	$3,261.96	$3,187.47	$3,261.92	$3,261.92	9908112156
2022-04-10	$3,261.29	$3,303.00	$3,211.87	$3,211.87	$3,211.87	10427054790
2022-04-11	$3,209.58	$3,214.46	$2,962.76	$2,981.05	$2,981.05	21891804831
2022-04-12	$2,987.16	$3,076.92	$2,960.05	$3,023.27	$3,023.27	20259698688

1 Exploratory Data Analysis

[9]:
```
eth_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1616 entries, 2017-11-09 to 2022-04-12
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype
```

```
---  ------      --------------  -----
 0   Open        1616 non-null   float64
 1   High        1616 non-null   float64
 2   Low         1616 non-null   float64
 3   Close       1616 non-null   float64
 4   Adj Close   1616 non-null   float64
 5   Volume      1616 non-null   int64
dtypes: float64(5), int64(1)
memory usage: 88.4 KB
```

[10]:
```
#check if we have any NAs
eth_df.isnull().sum()
```

[10]:
```
Open         0
High         0
Low          0
Close        0
Adj Close    0
Volume       0
dtype: int64
```

[11]:
```
#check columns in a dataframe
eth_df.columns
```

[11]:
```
Index(['Open', 'High', 'Low', 'Close', 'Adj Close', 'Volume'], dtype='object')
```

[12]:
```
#adding date column to the df/reindexing
eth_df.reset_index(inplace=True)

eth_df.columns
```

[12]:
```
Index(['Date', 'Open', 'High', 'Low', 'Close', 'Adj Close', 'Volume'],
dtype='object')
```

[16]:
```
#we only need 2 columns for our forecasting prophet model
df = eth_df[['Date', 'Open']]

new_names = {
    'Date': 'ds',
    'Open': 'y',
}

df.rename(columns=new_names, inplace = True)
```

[17]:
```
#check if data is ready for prophet
df.tail()
```

[17]:
```
             ds         y
1611 2022-04-08 $3,233.27
1612 2022-04-09 $3,191.98
1613 2022-04-10 $3,261.29
1614 2022-04-11 $3,209.58
1615 2022-04-12 $2,987.16
```

[18]:
```
#plot the open price
x = df['ds']
y = df['y']

fig = go.Figure()

fig.add_trace(go.Scatter(x=x, y=y))

#Set title
fig.update_layout(
title_text = 'Time series plot of Ethereum Open Price')
```

2 Prophet model

[20]:
```
#fitting data into the Prophet model

m = Prophet(
seasonality_mode='multiplicative'
)

m.fit(df)
```

```
INFO:fbprophet:Disabling daily seasonality. Run prophet with
daily_seasonality=True to override this.
```

[20]:
```
<fbprophet.forecaster.Prophet at 0x16a66261c10>
```

[21]:
```
#now we can create an entire years worth of date data for our prophet model to␣
 ↪make predictions
future = m.make_future_dataframe(periods=365)
future.tail()
```

[21]:
```
             ds
1976 2023-04-08
1977 2023-04-09
1978 2023-04-10
1979 2023-04-11
1980 2023-04-12
```

3 Model Predictions

[22]:
```
forecast = m.predict(future)
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
#one year in advance
```

[22]:

	ds	yhat	yhat_lower	yhat_upper
1976	2023-04-08	$3,795.84	$2,235.47	$5,274.51
1977	2023-04-09	$3,793.83	$2,271.73	$5,215.67
1978	2023-04-10	$3,784.79	$2,274.64	$5,246.04
1979	2023-04-11	$3,769.76	$2,263.95	$5,204.02
1980	2023-04-12	$3,775.60	$2,237.83	$5,215.04

[23]:
```
#prediction for the next day
next_day = (datetime.today() + timedelta(days=1)).strftime('%Y-%m-%d')

#next day is 4/13/2022
forecast[forecast['ds'] == next_day]['yhat'].item()
```

[23]:
```
3055.09070239782
```

4 Forecast Plots

[25]:
```
plot_plotly(m, forecast)
```

5 Forecast Components

[26]:
```
#other components of our forecast model include trend, yearly, and weekly␣
 ↪visualiation diagrams
plot_components_plotly(m,forecast)
```

[ ]:
```
Our model tells us that:
    1. There will be an upward trend for the price of Ethereum
    2. The price of ETH is lowest in July and April on a Saturday.
    3. ETH is most expensive around November and May on a Thursday.
```