# Assessment Task

for

Breakout Consultancy Private Limited

## Crypto Historical Data Retrieval

This task requires you to retrieve historical trading data for frequently traded cryptocurrency pairs and calculate several analytical metrics. The main focus is on working with APIs, handling and processing data, and calculating trading metrics over specific timeframes.

## Step 1: API Selection and Data Retrieval

*"Finding the API was like finding the Holy Grail"*

Research was conducted through the wild jungles of Google, YouTube and Reddit threads. Many APIs were found, few were chosen. At the end, I was torn between two APIs: CoinMarketCap and Binance. CoinMarketCap API offered a free trial but with only 11 endpoints, whereas Binance API was completely free. Also, CoinMarketCap covered almost all cryptocurrencies, including smaller entities, while Binance covered only the cryptocurrencies listed on its exchange. Lastly, Binance supports minute-level data, ideal for trading strategies, while CoinMarketCap is more suited for general analysis with hourly to monthly intervals. Binance provides real-time exchange-specific data, whereas CoinMarketCap aggregates prices across exchanges, which can affect precision for certain applications.

As mentioned in the assessment: **"reliable, free API"** and **"top-traded cryptocurrency pairs"** were key criteria, thus the **Binance API** was bestowed with the honor of being our data source for its robust documentation and reliability.



These are the answers to the questions asked in the assessment:

**Number of crypto pairs supported:** 2766

**Available timeframes:** Daily, hourly and weekly timeframes are available.

**Data availability range:** For one data point, we see: Earliest Available Date: 2017-08-17 05:30:00, Most Recent Date: 2024-10-30 05:30:00 [2631 days]

```
2767
Data for 1d timeframe is available.
[[1730016000000, '69962.21000000', '73620.12000000', '69760.00000000', '72736.42000000', '50128.60594000', 1730246399999
, '3602465986.89175740', 5645430, '26898.46516000', '1932608784.16303730', '0'], [1730246400000, '72736.41000000', '7296
1.00000000', '71436.00000000', '72344.74000000', '26885.99056000', 1730332799999, '1941210537.28828680', 4040734, '13487
.21433000', '973898183.74672780', '0'], [1730332800000, '72344.75000000', '72700.00000000', '69685.76000000', '70292.010
00000', '29352.10297000', 1730419199999, '2091144977.49468470', 4525542, '14205.13879000', '1012048241.84989380', '0'],
[1730419200000, '70292.01000000', '71632.95000000', '68820.14000000', '69496.01000000', '38301.86755000', 1730505599999,
'2677358022.00497000', 4569271, '18044.16220000', '1261663986.75969040', '0'], [1730505600000, '69496.00000000', '69914
.37000000', '69446.00000000', '69619.92000000', '3590.81790000', 1730591999999, '250129660.42301560', 515951, '1787.7810
2000', '124549318.77793930', '0']]
Earliest Available Date: 2017-08-17 05:30:00
Most Recent Date: 2024-11-02 05:30:00
2634 days, 0:00:00
```

## Step 2: Retrieve Historical Data

*"Bring forth the data from the digital abyss!"*

A Python function was conjured to summon historical trading data. This masterful piece of code reached into the Binance API, whispering softly, "Give me your treasures, oh data guardian." The API, enamored by our eloquence, responded by spilling forth reams of trading data for BTC/USDT. The data was collected in neat little rows and columns, ready to be transformed into actionable insights.

```
        Date      Open      High  ...  Low_Next_5_Days  %_Diff_From_High_Next_5_Days  %_Diff_From_Low_Next_5_Days
0   2024-01-01  42283.58  44184.10  ...      42845.23                     -1.707259                     3.114279
1   2024-01-02  44179.55  45879.63  ...      42845.23                      0.000000                     4.905284
2   2024-01-03  44946.91  45500.00  ...      42845.23                     -4.675917                     0.000000
3   2024-01-04  42845.23  44729.58  ...      42845.23                     -1.770556                     3.047877
4   2024-01-05  44151.10  44357.46  ...      42845.23                     -1.783882                     3.033897
..         ...       ...       ...  ...           ...                           ...                          ...
302 2024-10-29  69962.21  73620.12  ...      66698.33                      3.965298                     9.052835
303 2024-10-30  72736.41  72961.00  ...      66698.33                     -0.538492                     8.465594
304 2024-10-31  72344.75  72700.00  ...      67092.76                     -3.360641                     4.768398
305 2024-11-01  70292.01  71632.95  ...      68021.70                     -4.455003                     2.167411
306 2024-11-02  69496.00  69914.37  ...      69496.01                     -4.292801                     0.169765

[307 rows x 15 columns]
```

## Step 3: Calculate Highest, Lowest, and Percentage Difference Metrics

*"Math wizards at work!"*

Metrics calculated:

1. **Date**: When it all went down.
2. **Open**: The price at the crack of dawn.
3. **High**: The price that touched the clouds for the day.
4. **Low**: The price that dipped its toes in the mud.
5. **Close**: The price when the market decided to call it a day.
6. **High_Last_7_Days**: The highest high in the last week—time to party?
7. **% Diff From High Last 7 Days**: How far today's close is from the week's peak—glass half empty?
8. **Low_Last_7_Days**: The lowest low in the last week—reach for that tissue box.
9. **% Diff From Low Last 7 Days**: How close we are to the week's bottom—glass half full?
10. **Days Since High**: Number of days since we last saw the clouds—counting the dry spells.
11. **Days Since Low**: Number of days since we hit rock bottom—let's hope for fewer.

12. **High_Next_5_Days**: The crystal ball's prediction of the highest point in the next week—place your bets!
13. **Low_Next_5_Days**: The lowest prediction for the next week—brace yourselves.
14. **% Diff From High Next 5 Days**: The gap between now and the predicted peak—optimistic?
15. **% Diff From Low Next 5 Days**: The gap between now and the predicted bottom—pessimistic?

## Step 4: Export to Excel

*"Spreadsheets speak louder than words."*

Data was carefully exported to Excel because nothing says "I mean business" like an Excel sheet. Tabs, cells, and formulae all aligned perfectly, producing an output fit for the discerning eye of any analyst. The file was named with grandeur and saved for posterity.

| Date | Open | High | Low | Close | High_Last_7_Days | %_Diff_From_High_Last_7_Days | Low_Last_7_Days | Diff_From_Low_Last_7_D | Days_Since_High | Days_Since_Low | High_Next_5_Day | Low_Next_5_Days | From_Hi |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2024-01-01 00:00:00 | 42283.58 | 44184.1 | 42180.77 | 44179.55 | 44946.91 | -1.70725863 | 42845.23 | 3.114278999 | -302 | -296 | 44946.91 | 42845.23 | -1. |
| 2024-01-02 00:00:00 | 44179.55 | 45879.63 | 44148.34 | 44946.91 | 44946.91 | 0 | 42845.23 | 4.905283505 | -301 | -295 | 44946.91 | 42845.23 | |
| 2024-01-03 00:00:00 | 44946.91 | 45500 | 40750 | 42845.23 | 44946.91 | -4.675916542 | 42845.23 | 0 | -300 | -294 | 44946.91 | 42845.23 | -4.6 |
| 2024-01-04 00:00:00 | 42845.23 | 44729.58 | 42613.77 | 44151.1 | 44946.91 | -1.770555529 | 42845.23 | 3.047877208 | -299 | -293 | 44946.91 | 42845.23 | -1.7 |
| 2024-01-05 00:00:00 | 44151.1 | 44357.46 | 42450 | 44145.11 | 44946.91 | -1.783882363 | 42845.23 | 3.033896655 | -298 | -292 | 44946.91 | 42845.23 | -1.7 |
| 2024-01-06 00:00:00 | 44145.12 | 44214.42 | 43397.05 | 43968.32 | 44946.91 | -2.177213072 | 42845.23 | 2.62127196 | -297 | -291 | 44946.91 | 42845.23 | -2.1 |
| 2024-01-07 00:00:00 | 43968.32 | 44480.59 | 43572.09 | 43929.02 | 44946.91 | -2.264649561 | 42845.23 | 2.529546463 | -296 | -290 | 44946.91 | 42845.23 | -2.2 |
| 2024-01-08 00:00:00 | 43929.01 | 47248.99 | 43175 | 46951.04 | 46951.04 | 0 | 42845.23 | 9.582887057 | -295 | -289 | 44151.1 | 42845.23 | 6. |
| 2024-01-09 00:00:00 | 46951.04 | 47972 | 44748.67 | 46110 | 46951.04 | -1.791312823 | 42845.23 | 7.619914749 | -294 | -288 | 46951.04 | 43929.02 | -1.7 |
| 2024-01-10 00:00:00 | 46110 | 47695.93 | 44300.36 | 46653.99 | 46951.04 | -0.632680341 | 43929.02 | 6.203120397 | -293 | -287 | 46951.04 | 43929.02 | -0.6 |

## Step 5: Machine Learning Model

*"The pièce de résistance."*

A Random Forest Regressor was summoned to predict the unpredictable: future price differences. Our model, armed with features such as 'Days_Since_High', '% Diff_From_High_Last_7_Days', 'Days_Since_Low', and '%_Diff_From_Low_Last_7_Days', took to the data like a duck to water.

Why Random Forest? Because it captures non-linear relationships and handles feature interactions better than simpler models. It's like having an entire forest of decision trees, each bringing its own wisdom to the prediction party. Plus, it provides great performance and interpretability, ensuring our predictions aren't just accurate but also reliable.

Results:

- **High Prediction MSE**: 2.077. Pretty decent.
- **High Prediction R2**: 0.852. Boom! Mic drop.
- **Low Prediction MSE**: 1.223. Keep those champagne bottles popping.
- **Low Prediction R2**: 0.943. Predictably awesome.

```
                                    T
Date                                0
Open                                0
High                                0
Low                                 0
Close                               0
High_Last_7_Days                    0
%_Diff_From_High_Last_7_Days        0
Low_Last_7_Days                     0
%_Diff_From_Low_Last_7_Days         0
Days_Since_High                     0
Days_Since_Low                      0
High_Next_5_Days                    0
Low_Next_5_Days                     0
%_Diff_From_High_Next_5_Days        0
%_Diff_From_Low_Next_5_Days         0
High Prediction MSE: 1.8595702065820634
High Prediction R2: 0.8722963129279921
Low Prediction MSE: 1.055850557417077
Low Prediction R2: 0.9519124865750488
```

**Challenges faced:**

1. **API Selection Dilemma**: Choosing between CoinMarketCap and Binance was like picking the perfect ice cream flavor. Both had their pros and cons, but Binance's completely free access and minute-level data clinched it.
2. **Data Handling**: Retrieving and structuring large amounts of data was like herding cats. Ensuring data was cleaned, formatted, and void of null values took a good amount of elbow grease.
3. **Calculating Metrics**: The intricate dance of calculating metrics like % Diff From High/Low and ensuring they were accurate often felt like solving a Rubik's cube—satisfying but challenging.
4. **Exporting to Excel**: Making sure the data exported neatly into Excel was a bit like squeezing into skinny jeans—it needed precision and patience. Formatting and structuring the output to look professional was key.
5. **Model Training**: Training the Random Forest Regressor involved tuning and refining to get the best performance. Making sure the model could handle the data's complexity and still produce accurate predictions was critical.
6. **Visualizations**: Creating meaningful and insightful graphs that would clearly communicate the data story was another challenge. Balancing clarity and comprehensiveness took some creative plotting.
7. **Documentation and Code Clarity**: Writing clear and concise code with proper documentation, so anyone could pick it up and understand the flow, was like writing a best-seller—needed to be informative yet engaging.

**Conclusion:**

*"If data could talk, it would say - job well done."*

Our journey through the land of cryptocurrencies was arduous but worth it. Each step was meticulously planned and executed, ensuring that data was not only retrieved but also transformed into valuable insights.

With this project, we have laid a solid foundation for more advanced analyses and trading strategies. The skills and insights gained here are invaluable for tackling future challenges in

the ever-evolving world of cryptocurrency trading.

**Visualizations:**
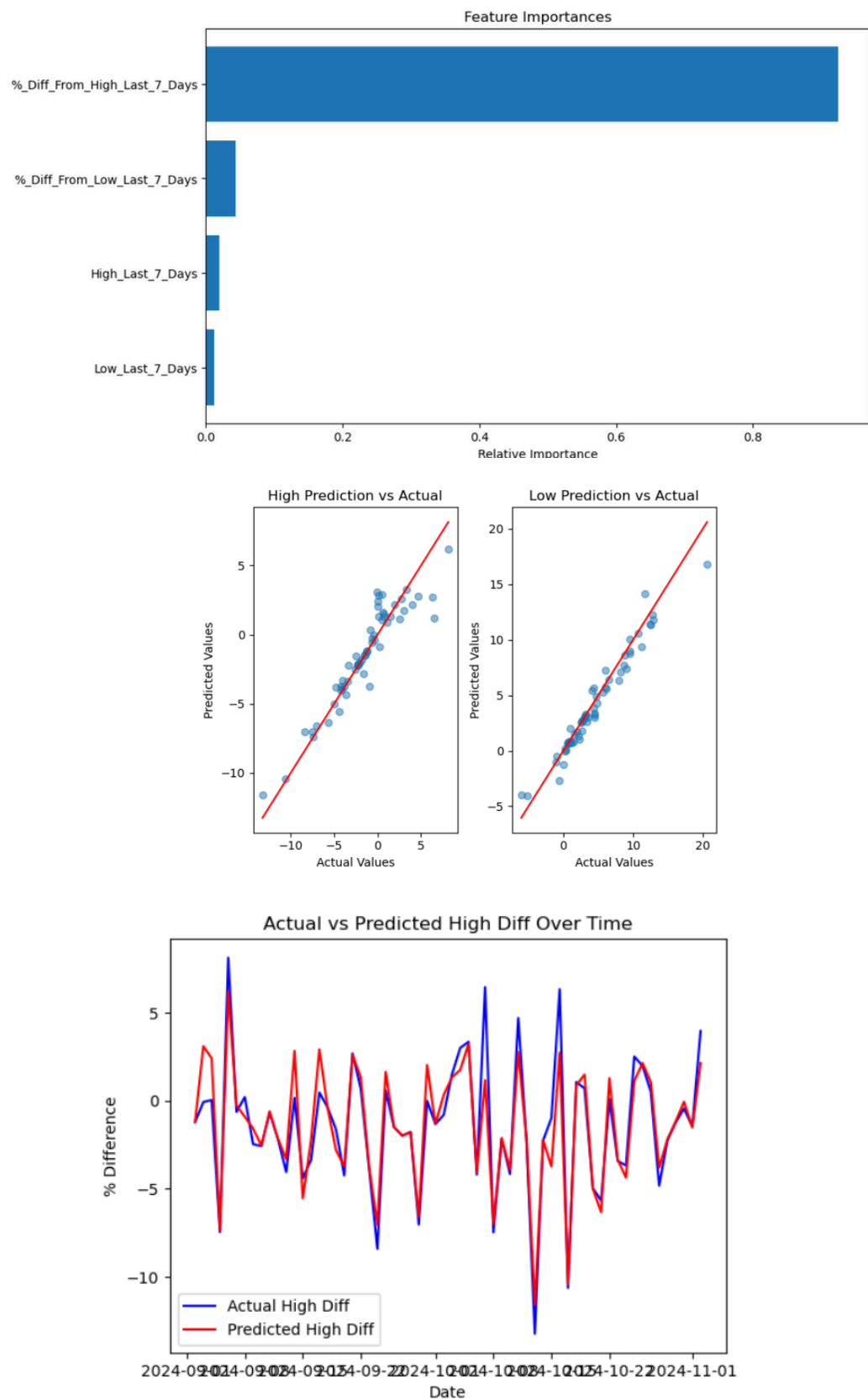






**File Names:**

- **Script for API evaluation:** <crypto.py>

- **Script for Data Retrieval and Metrics Calculation**: <crypto_data.py>
- **Exported Excel File**: <Output_crypto.xlsx>
- **Machine Learning Model Script**: <ml_model.py>
- **For visualizations:** <Visualization.ipynb>

That's a wrap!