



**Due Date:** November 3rd, 2024

**Submission Requirements:**

- **Python code file** (.py) containing all functions and logic.
- **Excel workbook** with crypto historical data.
- **Short project explanation** in a text document.
- **2-minute Loom video** explaining your approach.
- **Submission Email:** kapil@breakoutinvesting.in

**Internship Details:**

- **Stipend:** €200-400/month
- **Duration:** 3 months
- **Location:** Remote

**Assessment Task: 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.

**Task Breakdown and Detailed Requirements**

**Step 1: Identify a Reliable API for Crypto Data Retrieval**

**1. API Research and Selection**

- Find a **reliable, free API** that provides **historical data** for cryptocurrencies.
- Ensure the API can fetch daily historical data for **top-traded cryptocurrency pairs**.
- **Verify and document** the following API details:
  - **Number of crypto pairs supported:** List available pairs if possible.
  - **Available timeframes:** Confirm that daily, hourly, and/or weekly timeframes are supported.
  - **Data availability range:** Earliest available date and most recent date for data retrieval.

**Step 2: Retrieve Historical Data**

**1. Create a Python Function for Data Retrieval**

- Develop a function named `fetch_crypto_data` that:
  - Accepts two inputs: `crypto_pair` (e.g., "BTC/USD") and `start_date` (in YYYY-MM-DD format).
  - Retrieves **daily historical data** for the specified cryptocurrency pair starting from `start_date`.



- Returns a **DataFrame** with the following columns:
  - **Date**: The date of each record.
  - **Open**: Opening price of the cryptocurrency.
  - **High**: Highest price of the cryptocurrency.
  - **Low**: Lowest price of the cryptocurrency.
  - **Close**: Closing price of the cryptocurrency.

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

#### 1. Create a Python Function for Calculating Metrics

- Name this function `calculate_metrics`. This function should:
  - Accept the following inputs:
    - **data**: The DataFrame containing historical crypto data.
    - **variable1**: Integer representing a look-back period (e.g., 7 days) for historical high and low metrics.
    - **variable2**: Integer representing a look-forward period (e.g., 5 days) for future high and low metrics.

#### 2. Calculate and Add Columns as Below:

- **Historical High Price** (Column: `High_Last_{variable1}_Days`)
  - Calculate the highest price in the past `{variable1}` days.
- **Days Since High** (Column: `Days_Since_High_Last_{variable1}_Days`)
  - Calculate the difference in days between the current `Date` and the date of the `High_Last_{variable1}_Days`.
- **% Difference from Historical High** (Column: `%_Diff_From_High_Last_{variable1}_Days`)
  - Calculate the percentage difference between the current `Close` price and the high price found in the past `{variable1}` days.
- **Historical Low Price** (Column: `Low_Last_{variable1}_Days`)
  - Calculate the lowest price in the past `{variable1}` days.
- **Days Since Low** (Column: `Days_Since_Low_Last_{variable1}_Days`)
  - Calculate the difference in days between each date and the date of the calculated low.
- **% Difference from Historical Low** (Column: `%_Diff_From_Low_Last_{variable1}_Days`)
  - Calculate the percentage difference between the current `Close` price and the low price found in the past `{variable1}` days.
- **Future High Price** (Column: `High_Next_{variable2}_Days`)
  - Calculate the highest price within the next `{variable2}` days.
- **% Difference from Future High** (Column: `%_Diff_From_High_Next_{variable2}_Days`)



- Calculate the percentage difference between the current **Close** price and the value from **High\_Next\_{variable2}\_Days**.
- **Future Low Price** (Column: **Low\_Next\_{variable2}\_Days**)
  - Calculate the lowest price within the next {variable2} days.
- **% Difference from Future Low** (Column: **%\_Diff\_From\_Low\_Next\_{variable2}\_Days**)
  - Calculate the percentage difference between the current **Close** price and the value from **Low\_Next\_{variable2}\_Days**.

**Sample Data (Transposed to fit on the page)**

Date	2024-01-01	2024-01-02	2024-01-03	2024-01-04	2024-01-05	2024-01-06	2024-01-07	2024-01-08
Open	50000	50500	51500	52500	53000	53500	54000	54500
High	51000	52000	53000	53500	54000	54500	55000	55500
Low	49500	50000	50500	51000	51500	52000	52500	53000
Close	50500	51500	52500	53000	53500	54000	54500	55000
High_Last_7_Days	51000	52000	53000	53500	54000	54500	55000	55500
Days_Since_High_Last_7_Days	1	1	1	1	1	1	1	1
%_Diff_From_High_Last_7_Days	-1.96%	-0.96%	-0.94%	-0.93%	-0.93%	-0.92%	-0.91%	-0.90%
Low_Last_7_Days	49500	49500	50000	50500	51000	51500	52000	52500
Days_Since_Low_Last_7_Days	2	2	2	3	4	5	6	7
%_Diff_From_Low_Last_7_Days	2.02%	4.04%	5.00%	4.95%	4.90%	4.85%	4.81%	4.76%
High_Next_5_Days	52000	53000	53500	54000	54500	55000	55500	56000
%_Diff_From_High_Next_5_Days	2.97%	2.91%	1.90%	1.89%	1.87%	1.85%	1.80%	1.78%



Low_Next_5_Days	49000	50000	51000	51500	52000	52500	53000	53500
%_Diff_From_Low_Next_5_Days	-2.97%	-2.91%	-2.86%	-2.83%	-2.81%	-2.78%	-2.75%	-2.73%

### 3. Machine Learning Model

In this section, you will create a machine learning model that predicts future price differences based on recent historical data. This additional component will assess your ability to build and evaluate a machine learning model using calculated trading metrics.

#### 1. Objective:

- Train a machine learning model to predict two target variables:

- `%_Diff_From_High_Next_{variable2}_Days`
- `%_Diff_From_Low_Next_{variable2}_Days`

- Use the following four features as input variables:

- `Days_Since_High_Last_{variable1}_Days`
- `%Diff_From_High_Last{variable1}_Days`
- `Days_Since_Low_Last_{variable1}_Days`
- `%Diff_From_Low_Last{variable1}_Days`

#### 2. Requirements:

- Create a separate Python file, `ml_model.py`, that contains the code for training and testing this machine learning model.
- Implement two main functions in `ml_model.py`:
  - `train_model`: This function will train the machine learning model, evaluate its performance, and return the model's accuracy on a test dataset.
  - `predict_outcomes`: This function will use the trained model to make predictions based on new input values for the four features. The function should take the values of these four features as input and output predicted values for `%_Diff_From_High_Next_{variable2}_Days` and `%_Diff_From_Low_Next_{variable2}_Days`.

---

### Additional Requirements and Documentation

#### 1. Documentation

- **Code Comments**: Clearly document each function and major calculation step.
- **Short Project Explanation**: Provide a text document summarizing your approach, decisions, and challenges faced.

#### 2. Loom Video: Record a 2-minute Loom video explaining:



- Your approach to solving the task.
  - Any challenges faced and how they were overcome.
  - How you utilized any AI tools (if applicable).
- 

## Deliverables

- **Python Code File:**
    - Code for API data retrieval and metric calculations should be modular and well-documented.
  - **Excel Workbook:**
    - Include sheets for each cryptocurrency pair's historical data.
  - **Short Project Explanation:**
    - Provide insights into your approach, challenges, and rationale.
  - **Loom Video:**
    - Explain your process, approach, and any unique solutions you applied.
- 

## Evaluation Criteria

1. **API Selection and Data Retrieval:** Ability to research, choose, and use a suitable API.
2. **Data Handling and Excel Output:** Skill in structuring, formatting, and exporting data effectively.
3. **Functionality and Accuracy:** Ability to calculate metrics correctly and efficiently.
4. **Code Clarity and Documentation:** Clear organization and readable code.
5. **Communication in Loom Video:** Ability to clearly and confidently explain your approach.

## Next Steps

If we are impressed with your submission, we will invite you for an interview to discuss your approach, thought process, and coding decisions in more detail.

