Assessment Task: Fetching and Analyzing Top 50 Live Cryptocurrency Data

Objective:

The goal of this assessment is to fetch live cryptocurrency data for the top 50 cryptocurrencies, analyze it, and present the data in a live-updating Excel sheet. The Excel sheet should continuously update with the latest cryptocurrency prices.

Assessment steps-

1- Fetch Live Data

Use a public API (e.g., CoinGecko, CoinMarketCap, or Binance API) to fetch the top 50 cryptocurrencies by market capitalization.

The data should include at least the following fields:

- Cryptocurrency Name
- Symbol
- Current Price (in USD)
- Market Capitalization
- 24-hour Trading Volume
- Price Change (24-hour, percentage)

2-Data Analysis

Perform basic analysis on the live data fetched. Your analysis should include:

- Identifying the top 5 cryptocurrencies by market cap.
- Calculating the average price of the top 50 cryptocurrencies.
- Analyzing the highest and lowest 24-hour percentage price change among the top 50.

3-Live-Running Excel Sheet

Set Up Live Updating in Excel:

- fetch live data directly into Excel from the API.

Your sheet must:

- Continuously update the data (e.g., every 5 minutes).
- Show the **live prices** and other key metrics.

4-Submission Requirements:

I. Python Script
II. Excel sheet should show **live updating data**. Share link also to see live updated sheet in excel.
III. **Analysis Report**: A brief report (PDF or Word) summarizing the key insights and analysis from the data fetched.

Objective 1: Fetch Live Data

To accomplish this, the script retrieves real-time cryptocurrency data from an API (in this case, Binance using the ccxt library), specifically:

• Cryptocurrency Name: This is the whole name of the cryptocurrency, such as Ethereum or Bitcoin.

• Symbol: The cryptocurrency's abbreviation (e.g., BTC, ETH).

• Current Price (in USD): The cryptocurrency's current price expressed in USD.

• Market Capitalization: A cryptocurrency's whole market worth, which is determined by multiplying its price by its supply.

• 24-Hour Trading Volume: The sum of all trades during the previous 24 hours.

• Price Change (24-Hour, Percent): The price changes as a percentage over the previous day.

The script gathers information on the top 50 cryptocurrencies by market capitalization using the Binance API. For additional examinations, this data is saved and shown in an Excel file.

Fetching funtion:

```
tickers = binance_ex.fetch_tickers()
```

Using pandas to transform into a dataframe:

```
data = pd.DataFrame(tickers).transpose()
```

Getting top 50 crypto according to market cap:

```
data = data.sort_values("Market Cap Estimate", ascending=False)

top_50 = data.head(50)
```

Objective 2: Data Analysis

1. Market Cap Ranking of the Top 5 Cryptocurrencies:

   The 5 cryptocurrencies with the most market capitalization are determined by the script.

   ```
   top_5_by_market_cap = top_50_data.sort_values(by="Market Cap Estimate", ascending=False).head(5)
   ```

2. The Top 50 Cryptocurrencies' Average Price:

   The average price of the top 50 cryptocurrencies is determined by the script.

   ```
   top_50_data["Current Price (USD)"] = top_50_data["Current Price (USD)"].replace("[$,]", "",
   regex=True).astype(float)
   avg_price = top_50_data["Current Price (USD)"].mean()
   ```

3. 24-hour Percentage Price Change Highest and Lowest:

   Cryptocurrencies with the biggest and smallest 24-hour percentage price movements are identified by the script.

   ```
   top_50_data["24h Price Change (%)"] = top_50_data["24h Price Change (%)"].replace("[%]", "",
   regex=True).astype(float)
   highest_change = top_50_data.loc[top_50_data["24h Price Change (%)"].idxmax()]
   lowest_change = top_50_data.loc[top_50_data["24h Price Change (%)"].idxmin()]
   ```

All three analysis were complete and displayed in the excel sheet along side the live fetched data.

Objective 3: Live-Running Excel-sheet

The script uses xlwings to connect to an Excel worksheet to continually monitor and update the data. The features listed below are put into practice:

```
import xlwings as xl
xlsheet = xl.Book('binance-crypto.xlsx')
binance_sheet = xlsheet.sheets('Binance_data')
```

Constant Data Fetching:

To keep the data up to current, the script retrieves real-time data every five seconds (or at other predetermined times).

```
import time as tm
tm.sleep(5)
```

To make sure the data is reliable, it obtains current prices, trade volumes, and other pertinent indicators.

Show in Excel:

An Excel document displays the retrieved data, which includes the prices, market capitalizations, trade volumes, and the top 50 cryptocurrencies.

```
top_50_data = fetch_top_cryptos(binance_ex)
# Clear previous data from the Excel sheet
binance_sheet.clear()
# Write the new data into the sheet starting at cell A1
binance_sheet.range('A1').value = top_50_data
```

The continually updated sheet offers room for data analysis and displays the most recent values for every coin.

Excel analysis:

The Excel document provides a thorough analysis for convenient viewing and includes important insights (such as the top 5 cryptocurrencies, the average price, and the highest/lowest 24-hour price fluctuations).
In the bitcoin market, users may easily compare measurements and spot trends or patterns.

```
  binance_sheet.range(f"A{analysis_start_row + 2}").value = [
      "Average Price of Top 50 Cryptos:",
      f"${avg_price:,.2f}",
]
binance_sheet.range(f"A{analysis_start_row + 3}").value = [
      "Highest 24h Price Change:",
      f"{highest_change['Name']} ({highest_change['24h Price Change (%)']:.2f}%)",
]
binance_sheet.range(f"A{analysis_start_row + 4}").value = [
      "Lowest 24h Price Change:",
      f"{lowest_change['Name']} ({lowest_change['24h Price Change (%)']:.2f}%)",
]

# Write the top 5 table
top_5_start_row = analysis_start_row + 6  # Leave a gap before the top 5 table
binance_sheet.range(f"A{top_5_start_row}").value = "Top 5 Cryptos by Market Cap"
binance_sheet.range(f"A{top_5_start_row + 1}").value = list(top_5_by_market_cap.columns)
binance_sheet.range(f"A{top_5_start_row + 2}").value = top_5_by_market_cap.values
```

The ability to watch and assess crypto performance in real time, straight from Excel, is made possible by the live updates function, which guarantees that the data stays current.

Source Code

```
# %%
import ccxt
import pandas as pd
import xlwings as xl
import time as tm
xlsheet = xl.Book('binance-crypto.xlsx')
binance_sheet = xlsheet.sheets('Binance_data')


# %%
binance_ex = ccxt.binance()


# %%
# the fetched data didn’t had any name with it so I manually made a dictionary and mapped the names according to the symbol
symbol_to_name = {
    "BTC": "Bitcoin",
    "ETH": "Ethereum",
    "BNB": "Binance Coin",
    "XRP": "Ripple",
    "USDT": "Tether",
    "USDC": "USD Coin",
    "ADA": "Cardano",
    "ETHFI": "Ethereum Fair",
    "ARB": "Arbitrum",
    "OP": "Optimism",
    "SOL": "Solana",
    "DOGE": "Dogecoin",
    "DOT": "Polkadot",
    "FLOKI": "Floki Inu",
    "SHIB": "Shiba Inu",
    "AVAX": "Avalanche",
    "MATIC": "Polygon",
```

"LTC": "Litecoin",

"ATOM": "Cosmos",

"LINK": "Chainlink",

"XLM": "Stellar",

"TRX": "Tron",

"ETC": "Ethereum Classic",

"XMR": "Monero",

"ALGO": "Algorand",

"BCH": "Bitcoin Cash",

"VET": "VeChain",

"ICP": "Internet Computer",

"FIL": "Filecoin",

"HBAR": "Hedera",

"EGLD": "MultiversX (Elrond)",

"QNT": "Quant",

"FLOW": "Flow",

"CHZ": "Chiliz",

"APT": "Aptos",

"NEAR": "Near Protocol",

"GRT": "The Graph",

"AAVE": "Aave",

"KSM": "Kusama",

"CRV": "Curve DAO Token",

"SAND": "The Sandbox",

"MANA": "Decentraland",

"AXS": "Axie Infinity",

"FTM": "Fantom",

"RUNE": "THORChain",

"ZEC": "Zcash",

"SNX": "Synthetix",

"ENJ": "Enjin Coin",

"DYDX": "dYdX",

"BAT": "Basic Attention Token",

"CAKE": "PancakeSwap",

"STX": "Stacks",

"YFI": "yearn.finance",

"UNI": "Uniswap",

"1INCH": "1inch",

"LDO": "Lido DAO",

"WAVES": "Waves",

"CELR": "Celer Network",

"IMX": "Immutable X",

"ANC": "Anchor Protocol",

"RAY": "Raydium",

"SRM": "Serum",

"OMG": "OMG Network",

"ZIL": "Zilliqa",

"HNT": "Helium",

"CELO": "Celo",

"GALA": "Gala",

"ENS": "Ethereum Name Service",

"BNT": "Bancor",

"HOT": "Holo",

"KAVA": "Kava",

"OCEAN": "Ocean Protocol",

"COMP": "Compound",

"MKR": "Maker",

"BAL": "Balancer",

"UMA": "UMA",

"REN": "Ren",

"SKL": "SKALE",

"ANKR": "Ankr",

"CTSI": "Cartesi",

"AR": "Arweave",

"LRC": "Loopring",

"KLAY": "Klaytn",

"IOST": "IOST",

"RVN": "Ravencoin",

"MTL": "Metal",

"TWT": "Trust Wallet Token",

```
  "ALICE": "My Neighbor Alice",
  "COTI": "COTI",
  "CVC": "Civic",
  "XNO": "Nano",
  "REQ": "Request",
  "SC": "Siacoin",
  "ONT": "Ontology",
  "NKN": "NKN",
  "STMX": "StormX",
  "DENT": "Dent",
  "WIN": "WINkLink",
  "TFUEL": "Theta Fuel",
  "ZRX": "0x",
  "RSR": "Reserve Rights",
  "ICX": "ICON",
  "CHR": "Chromia",
  "PHA": "Phala Network",
  "REEF": "Reef",
  "BAND": "Band Protocol",
  "NEIRO": "NEIRO",
  "ACT": "ACT",
  "FLOKI": "Floki Inu",
  "PEPE": "PEPE",
  "BONK": "BONK",
  "FDUSD": "FDUSD",
  "LUMIA": "LUMIA",
  "TROY": "TROY",
  "ACA": "ACA",
  "PNUT": "PNUT",
  "USDC": "USD Coin",
  "SUI": "SUI",
  "ARKM": "ARKM",
}
```

```
# %%
def fetch_top_cryptos(binance_ex):
    #Fetch and process the top 50 cryptocurrencies by market cap from Binance API.
    tickers = binance_ex.fetch_tickers()

    # Convert the tickers dictionary into a DataFrame
    data = pd.DataFrame(tickers).transpose()
    data = data.dropna(axis=1, how='all')

    # Extract and process required fields
    data = data[["symbol", "last", "quoteVolume", "percentage", "baseVolume"]]
    data.columns = [
        "Symbol",
        "Current Price (USD)",
        "24h Volume (USD)",
        "24h Price Change (%)",
        "Market Cap Estimate",
    ]
    # Add cryptocurrency names using the `symbol_to_name` mapping
    data['Base Currency'] = data['Symbol'].apply(lambda x: x.split('/')[0])
    data['Name'] = data['Base Currency'].map(symbol_to_name)

    # Calculate approximate market capitalization
    data["Market Cap Estimate"] = data["Market Cap Estimate"] * data["Current Price (USD)"]

    # Sort by market capitalization
    data = data.sort_values("Market Cap Estimate", ascending=False)

    # Keep the top 50 only
    top_50 = data.head(50)

    # Add dollar signs to currency fields
    currency_fields = ["Current Price (USD)", "24h Volume (USD)", "Market Cap Estimate"]
    for field in currency_fields:
        top_50[field] = top_50[field].apply(lambda x: f"${x:,.2f}" if pd.notnull(x) else "-")
```

```
    # Format percentage change
    top_50["24h Price Change (%)"] = top_50["24h Price Change (%)"].apply(lambda x: f"{x:.2f}%" if pd.notnull(x) else "-")

    # Reorder columns
    top_50 = top_50[["Name", "Symbol", "Current Price (USD)", "Market Cap Estimate", "24h Volume (USD)", "24h Price Change (%)"]]

    return top_50

# %%
while True:
        top_50_data = fetch_top_cryptos(binance_ex)

        # Clear previous data from the Excel sheet
        binance_sheet.clear()

        # Write the new data into the sheet starting at cell A1
        binance_sheet.range('A1').value = top_50_data
        # Analysis
        # 1. Top 5 cryptocurrencies by market cap
        top_5_by_market_cap = top_50_data.sort_values(by="Market Cap Estimate", ascending=False).head(5)

        # 2. Average price of the top 50 cryptocurrencies
        top_50_data["Current Price (USD)"] = top_50_data["Current Price (USD)"].replace("[$,]", "", regex=True).astype(float)
        avg_price = top_50_data["Current Price (USD)"].mean()

        # 3. Highest and lowest 24-hour percentage price change
        top_50_data["24h Price Change (%)"] = top_50_data["24h Price Change (%)"].replace("[%]", "", regex=True).astype(float)
        highest_change = top_50_data.loc[top_50_data["24h Price Change (%)"].idxmax()]
        lowest_change = top_50_data.loc[top_50_data["24h Price Change (%)"].idxmin()]

        # Write analysis to Excel
        analysis_start_row = len(top_50_data) + 3  # Start a few rows below the data
```

```
# Add headers
binance_sheet.range(f"A{analysis_start_row}").value = "Analysis"
binance_sheet.range(f"A{analysis_start_row + 1}").value = [
    "Metric", "Details"
]

# Write individual metrics
binance_sheet.range(f"A{analysis_start_row + 2}").value = [
    "Average Price of Top 50 Cryptos:",
    f"${avg_price:,.2f}",
]
binance_sheet.range(f"A{analysis_start_row + 3}").value = [
    "Highest 24h Price Change:",
    f"{highest_change['Name']} ({highest_change['24h Price Change (%)']:.2f}%)",
]
binance_sheet.range(f"A{analysis_start_row + 4}").value = [
    "Lowest 24h Price Change:",
    f"{lowest_change['Name']} ({lowest_change['24h Price Change (%)']:.2f}%)",
]

# Write the top 5 table
top_5_start_row = analysis_start_row + 6  # Leave a gap before the top 5 table
binance_sheet.range(f"A{top_5_start_row}").value = "Top 5 Cryptos by Market Cap"
binance_sheet.range(f"A{top_5_start_row + 1}").value = list(top_5_by_market_cap.columns)
binance_sheet.range(f"A{top_5_start_row + 2}").value = top_5_by_market_cap.values

# Delay
print("\nData and analysis updated successfully in Excel.\n")
tm.sleep(5)
```

GitHub Link: https://github.com/Viswa02code/Primetrade.ai_assessment

Excel Link: binance-crypto.xlsx

As a result, I have presented the Assessment as clearly as possible.

Suggestions: Just click the GitHub link and download the file in zip. Run the Python application on your PC. The Excel will automatically generate and run live, and all the analyses will also be updated every 5 seconds. The Internet speed should be moderate to fetch the data.

Thank you for considering my application and giving me this opportunity to show my skills.