

Cryptocurrency Data Analysis Report

1. Introduction

This project aims to analyze cryptocurrency data using the CoinMarketCap API. The steps include fetching data, processing it, and visualizing the results to understand trends and changes in the cryptocurrency market.

2. Setup and Data Fetching

2.1 API Registration

- **Action:** Registered for a CoinMarketCap API key.
- **Key Used:** 802c100a-8ccc-4ddd-8bb9-514a7881943e

2.2 Development Environment

- **Libraries Imported:**
 - `requests` for API interactions
 - `json` for parsing data
 - `pandas` for data manipulation
 - `seaborn` and `plotly` for visualization

2.3 Fetching Data from API

Code:

```
python
Copy code
from requests import Session
import json

url = 'https://pro-api.coinmarketcap.com/v1/cryptocurrency/listings/latest'
parameters = {'start': '1', 'limit': '1000', 'convert': 'USD'}
headers = {'Accepts': 'application/json', 'X-CMC_PRO_API_KEY': '802c100a-8ccc-4ddd-8bb9-514a7881943e'}

session = Session()
session.headers.update(headers)

try:
    response = session.get(url, params=parameters)
    data = json.loads(response.text)
    print(data)
except (ConnectionError, Timeout, TooManyRedirects) as e:
    print(e)
```

Outcome: Successfully fetched data from the CoinMarketCap API.

3. Data Parsing and Feature Engineering

3.1 Parsing JSON Data

Code:

```
python
Copy code
import pandas as pd
pd.set_option('display.max_columns', None)
df = pd.json_normalize(data['data'])
```

Outcome: Data parsed into a Pandas DataFrame.

3.2 Feature Engineering

Code:

```
python
Copy code
df['timestamp'] = pd.to_datetime('now')

def api_runner():
    global df
    url = 'https://pro-api.coinmarketcap.com/v1/cryptocurrency/listings/latest'
    parameters = {'start': '1', 'limit': '333', 'convert': 'USD'}
    headers = {'Accepts': 'application/json', 'X-CMC_PRO_API_KEY': '802c100a-8ccc-4ddd-8bb9-514a7881943e'}

    session = Session()
    session.headers.update(headers)

    try:
        response = session.get(url, params=parameters)
        data = json.loads(response.text)
    except (ConnectionError, Timeout, TooManyRedirects) as e:
        print(e)
        return

    df2 = pd.json_normalize(data['data'])
    df2['timestamp'] = pd.to_datetime('now')
    df = pd.concat([df, df2], ignore_index=True)

    file_path = r"C:\Users\PC\OneDrive\Dokumente\Python Scripts\API1.csv"
    if not os.path.isfile(file_path):
        df.to_csv(file_path, header='column_names')
    else:
        df.to_csv(file_path, mode='a', header=False)
```

Automatic Update: Data is updated every minute and appended to a CSV file.

4. Data Transformation

4.1 Data Aggregation

Code:

```
python
Copy code
pd.set_option('display.float_format', lambda x: '%.5f' % x)
df3 = df.groupby('name', sort=False)[['quote.USD.percent_change_1h',
'quote.USD.percent_change_24h', 'quote.USD.percent_change_7d',
'quote.USD.percent_change_30d', 'quote.USD.percent_change_60d',
'quote.USD.percent_change_90d']].mean()
```

Outcome: Aggregated percentage changes over various periods.

4.2 Data Reshaping

Code:

```
python
Copy code
df4 = df3.stack()
df5 = df4.to_frame(name='values')
df6 = df5.reset_index()
df7 = df6.rename(columns={'level_1': 'percentage_change'})
df7['percentage_change'] =
df7['percentage_change'].replace(['quote.USD.percent_change_1h',
'quote.USD.percent_change_24h', 'quote.USD.percent_change_7d',
'quote.USD.percent_change_30d', 'quote.USD.percent_change_60d',
'quote.USD.percent_change_90d'], ['1h', '24h', '7d', '30d', '60d', '90d'])
df8 = df7.head(25)
```

Outcome: Prepared data for visualization, focusing on the top 25 cryptocurrencies.

5. Data Visualization

5.1 Visualization Using Seaborn

Code:

```
python
Copy code
import seaborn as sns
sns.catplot(x='percentage_change', y='values', hue='name', data=df8,
kind='point')
```

Outcome: Created a point plot to visualize percentage changes over various periods.

5.2 Visualization Using Plotly

Code:

```
python
Copy code
import plotly.express as px

# Scatter Plot
fig = px.scatter(df8, x='percentage_change', y='values', color='name',
symbol='name', title='Cryptocurrency Percentage Change')
fig.show()

# Histogram
fig = px.histogram(df8, x='percentage_change', y='values', color='name',
title='Histogram')
fig.show()
```

Outcome: Generated interactive scatter and histogram plots to analyze cryptocurrency trends.

6. Conclusion

- **Data Fetching:** Successfully automated the process of fetching and updating cryptocurrency data.
- **Data Transformation:** Aggregated and reshaped data to analyze percentage changes over multiple periods.
- **Visualization:** Utilized Seaborn and Plotly for insightful visualizations, revealing trends and changes in the cryptocurrency market.

Next Steps:

- **Data Enrichment:** Integrate additional data sources for more comprehensive analysis.
 - **Performance Optimization:** Improve data fetching and processing efficiency.
-