

GrandMaster Grapher

Github Repository Link

Step 1: Project Overview

- Objective: Create a data science project that scrapes chess player ratings from a website, saves the data in a CSV file, and visualizes the ratings using Matplotlib.- Tools and Libraries: Python, Requests, BeautifulSoup, CSV,pandas and Matplotli

Making a Virtual Environment

1. python -m venv gmaster - Makes a Virtual Environment
2. .\gmaster\Scripts\activate - Activates the Virtual Environment
3. pip install ipykernel - Installs ipykernel
4. python -m ipykernel install --name=gmaster - Links the Kernel to the Virtual Environment

Kernel - gmaster

Step 2: Import Libraries

Import the necessary Python libraries:

```
pip install requests
Collecting requests
  Downloading requests-2.31.0-py3-none-any.whl (62 kB)
----- 62.6/62.6 kB 1.1 MB/s
eta 0:00:00
Collecting charset-normalizer<4,>=2
  Downloading charset_normalizer-3.3.2-cp311-cp311-win_amd64.whl (99
kB)
----- 99.9/99.9 kB 640.2 kB/s
eta 0:00:00
Collecting idna<4,>=2.5
  Downloading idna-3.6-py3-none-any.whl (61 kB)
----- 61.6/61.6 kB 1.1 MB/s
eta 0:00:00
Collecting urllib3<3,>=1.21.1
  Downloading urllib3-2.1.0-py3-none-any.whl (104 kB)
----- 104.6/104.6 kB 754.7 kB/s
eta 0:00:00
Collecting certifi>=2017.4.17
  Downloading certifi-2023.11.17-py3-none-any.whl (162 kB)
----- 162.5/162.5 kB 810.1 kB/s
```

```

eta 0:00:00
Installing collected packages: urllib3, idna, charset-normalizer,
certifi, requests
Successfully installed certifi-2023.11.17 charset-normalizer-3.3.2
idna-3.6 requests-2.31.0 urllib3-2.1.0
Note: you may need to restart the kernel to use updated packages.

[notice] A new release of pip available: 22.3 -> 23.3.2
[notice] To update, run: python.exe -m pip install --upgrade pip

pip install beautifulsoup4

Collecting beautifulsoup4
  Downloading beautifulsoup4-4.12.3-py3-none-any.whl (147 kB)
----- 147.9/147.9 kB 1.3 MB/s
eta 0:00:00
Collecting soupsieve>1.2
  Downloading soupsieve-2.5-py3-none-any.whl (36 kB)
Installing collected packages: soupsieve, beautifulsoup4
Successfully installed beautifulsoup4-4.12.3 soupsieve-2.5
Note: you may need to restart the kernel to use updated packages.

[notice] A new release of pip available: 22.3 -> 23.3.2
[notice] To update, run: python.exe -m pip install --upgrade pip

pip install matplotlib

Collecting matplotlib
  Downloading matplotlib-3.8.2-cp311-cp311-win_amd64.whl (7.6 MB)
----- 7.6/7.6 MB 983.3 kB/s
eta 0:00:00
Collecting contourpy>=1.0.1
  Downloading contourpy-1.2.0-cp311-cp311-win_amd64.whl (187 kB)
----- 187.6/187.6 kB 1.0 MB/s
eta 0:00:00
Collecting cycler>=0.10
  Downloading cycler-0.12.1-py3-none-any.whl (8.3 kB)
Collecting fonttools>=4.22.0
  Downloading fonttools-4.47.2-cp311-cp311-win_amd64.whl (2.2 MB)
----- 2.2/2.2 MB 1.2 MB/s eta
0:00:00
Collecting kiwisolver>=1.3.1
  Downloading kiwisolver-1.4.5-cp311-cp311-win_amd64.whl (56 kB)
----- 56.1/56.1 kB 1.5 MB/s
eta 0:00:00
Collecting numpy<2,>=1.21
  Downloading numpy-1.26.3-cp311-cp311-win_amd64.whl (15.8 MB)
----- 15.8/15.8 MB 652.8 kB/s
eta 0:00:00

```

```
Requirement already satisfied: packaging>=20.0 in e:\grandmaster
grapher\gmaster\lib\site-packages (from matplotlib) (23.2)
Collecting pillow>=8
  Downloading pillow-10.2.0-cp311-cp311-win_amd64.whl (2.6 MB)
  ----- 2.6/2.6 MB 711.1 kB/s
eta 0:00:00
Collecting pyparsing>=2.3.1
  Downloading pyparsing-3.1.1-py3-none-any.whl (103 kB)
  ----- 103.1/103.1 kB 743.2 kB/s
eta 0:00:00
Requirement already satisfied: python-dateutil>=2.7 in e:\grandmaster
grapher\gmaster\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in e:\grandmaster grapher\
gmaster\lib\site-packages (from python-dateutil>=2.7->matplotlib)
(1.16.0)
Installing collected packages: pyparsing, pillow, numpy, kiwisolver,
fonttools, cycler, contourpy, matplotlib
Successfully installed contourpy-1.2.0 cycler-0.12.1 fonttools-4.47.2
kiwisolver-1.4.5 matplotlib-3.8.2 numpy-1.26.3 pillow-10.2.0
pyparsing-3.1.1
Note: you may need to restart the kernel to use updated packages.
```

```
[notice] A new release of pip available: 22.3 -> 23.3.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
pip install pandas
```

```
Collecting pandas
  Downloading pandas-2.2.0-cp311-cp311-win_amd64.whl (11.6 MB)
  ----- 11.6/11.6 MB 698.9 kB/s
eta 0:00:00
Requirement already satisfied: numpy<2,>=1.23.2 in e:\grandmaster
grapher\gmaster\lib\site-packages (from pandas) (1.26.3)
Requirement already satisfied: python-dateutil>=2.8.2 in e:\
grandmaster grapher\gmaster\lib\site-packages (from pandas) (2.8.2)
Collecting pytz>=2020.1
  Using cached pytz-2023.3.post1-py2.py3-none-any.whl (502 kB)
Collecting tzdata>=2022.7
  Downloading tzdata-2023.4-py2.py3-none-any.whl (346 kB)
  ----- 346.6/346.6 kB 430.6 kB/s
eta 0:00:00
Requirement already satisfied: six>=1.5 in e:\grandmaster grapher\
gmaster\lib\site-packages (from python-dateutil>=2.8.2->pandas)
(1.16.0)
Installing collected packages: pytz, tzdata, pandas
Successfully installed pandas-2.2.0 pytz-2023.3.post1 tzdata-2023.4
Note: you may need to restart the kernel to use updated packages.
```

```
[notice] A new release of pip available: 22.3 -> 23.3.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
import requests

from bs4 import BeautifulSoup

import csv

import matplotlib.pyplot as plt

import pandas as pd
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_23528\4080736814.py:1:
DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major
release of pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type,
and better interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at
https://github.com/pandas-dev/pandas/issues/54466
```

```
import pandas as pd
```

Step 3: Scraping Data

Scrape live world rankings and performance metrics of the top 10 chess players from 2700chess.com

[Link to 2700chess](#)

```
from IPython.display import Image, display
# Specify the path to the image file or use the filename if it's in
the same directory.
image_path = 'E:\Grandmaster Grapher/ratings.png'
display(Image(filename=image_path))
```

←

↻

🔒

https://2700chess.com

🏠

🔍

🌐

🔗

🔖

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

🔒

Step 4: Extracting Data

```
import requests
from bs4 import BeautifulSoup

def scrape_chess_ratings():
    url = "https://2700chess.com/"

    response = requests.get(url)

    if response.status_code == 200:
        soup = BeautifulSoup(response.content, "html.parser")

        player_rows = soup.find_all("tr", class_=["leaders",
"top_ten", "all_lines"])[0:10]

        data = []

        for row in player_rows:
            ranking = row.find("td", class_="pos").text.strip()
            wc_status = "Yes" if "top_ten" in row["class"] else "No"
            name = row.find("td", class_="name").a.text.strip()
            classical_rating = int(row.find("td",
class_="rating").strong.text.strip())
            rapid_rating = int(row.find("td", class_="rapid-
rating").text.strip())
            blitz_rating = int(row.find("td", class_="blitz-
rating").text.strip())
```

```

        birth_year = int(row.find("td",
class_="birth").span.text.strip())

        # Calculate age based on the current year (2024)
        current_year = 2024
        age = current_year - birth_year

        player_data = {
            "Ranking": ranking,
            "WC": wc_status,
            "Name": name,
            "Classical Rating": classical_rating,
            "Rapid Rating": rapid_rating,
            "Blitz Rating": blitz_rating,
            "Age": age
        }

        data.append(player_data)

    return data

else:
    print("Failed to retrieve data. Status code:",
response.status_code)
    return None

```

Step 5: Saving to CSV File

- Saves data in CSV File format
- Headers
 - a. Ranking
 - b. WC {World Champion}
 - c. Name
 - d. Classical
 - e. Rapid
 - f. Blitz
 - g. Age
- Ratings are subject to change. {Rating as respect to 21 January 2024}

```

def save_to_csv(data, filename="chess_ratings.csv"):
    with open(filename, mode="w", newline="", encoding="utf-8") as
file:
        writer = csv.writer(file)
        writer.writerow(["Ranking", "WC", "Name", "Classical Rating",
"Rapid Rating", "Blitz Rating", "Age"])
        writer.writerows(data)

```

```

def read_and_display_csv(filename="chess_ratings.csv"):
    with open(filename, mode="r", encoding="utf-8") as file:
        reader = csv.reader(file)
        for row in reader:
            print(row)
read_and_display_csv()

['Ranking', 'WC', 'Name', 'Classic', 'Rapid', 'Blitz', 'Age']
['1', '', 'Carlsen, Magnus', '2830', '2823', '2886', '33']
['2', '', 'Caruana, Fabiano', '2804', '2729', '2790', '31']
['3', '', 'Nakamura, Hikaru', '2788', '2746', '2874', '36']
['4', 'Champion', 'Ding, Liren', '2774.5', '2817.6', '2774.2', '31']
['5', '', 'Nepomniachtchi, Ian', '2760.8', '2754', '2800', '33']
['6', '', 'Firouzja, Alireza', '2757.2', '2724', '2863', '20']
['7', '', 'So, Wesley', '2757', '2742', '2767', '30']
['8', '', 'Giri, Anish', '2754.1', '2687', '2715', '29']
['9', '', 'Dominguez Perez, Leinier', '2752', '2706', '2645', '40']
['10', '', 'Karjakin, Sergey', '2750', '2724', '2646', '34']

from IPython.display import Image, display
image_path = 'E:\Grandmaster Grapher\data.png'
display(Image(filename=image_path))

```

chess_ratings.csv - Excel

File Home Insert Page Layout Formulas Data Review View Tell me what you want to do... Share

Clipboard Font Alignment Number Styles Cells Editing

J15

	A	B	C	D	E	F	G	H	I
1	Ranking	WC	Name	Classic	Rapid	Blitz	Age		
2	1		Carlsen, Magnus	2830	2823	2886	33		
3	2		Caruana, Fabiano	2804	2729	2790	31		
4	3		Nakamura, Hikaru	2788	2746	2874	36		
5	4	Champion	Ding, Liren	2774.5	2817.6	2774.2	31		
6	5		Nepomniachtchi, Ian	2760.8	2754	2800	33		
7	6		Firouzja, Alireza	2757.2	2724	2863	20		
8	7		So, Wesley	2757	2742	2767	30		
9	8		Giri, Anish	2754.1	2687	2715	29		
10	9		Dominguez Perez, Leinier	2752	2706	2645	40		
11	10		Karjakin, Sergey	2750	2724	2646	34		
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									
25									
26									
27									
28									
29									

chess_ratings

Ready 100%

Step 6: Visualizing Data

The Graph shows the following data

1. Ranking: Current World Ranking
2. Name: Player's Name
3. Classical Rating: Player's rating in Classical Time Control
4. Rapid Rating: Player's rating in Rapid Time Control

5. Blitz Rating: Player's rating in Blitz Time Control

```
# Load data from CSV file
df = pd.read_csv("chess_ratings.csv")

# Extracting data
names = df["Name"]
ages = df["Age"]
classical_ratings = df["Classic"]
rapid_ratings = df["Rapid"]
blitz_ratings = df["Blitz"]

# Plotting
plt.figure(figsize=(10, 6))

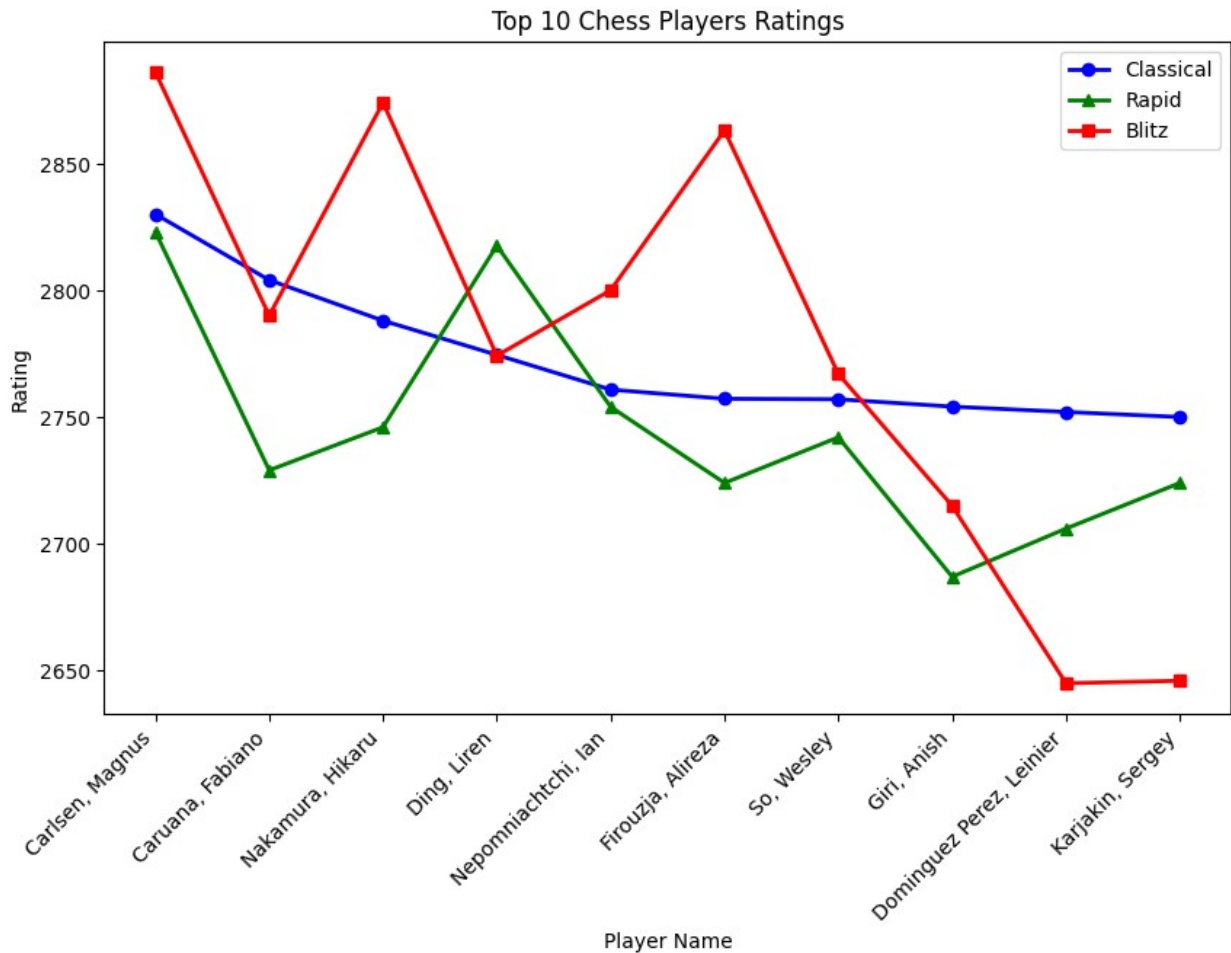
# Line chart for Classical Ratings
plt.plot(names, classical_ratings, label='Classical', marker='o',
color='blue', linestyle='--', linewidth=2)

# Line chart for Rapid Ratings
plt.plot(names, rapid_ratings, label='Rapid', marker='^',
color='green', linestyle='--', linewidth=2)

# Line chart for Blitz Ratings
plt.plot(names, blitz_ratings, label='Blitz', marker='s', color='red',
linestyle='--', linewidth=2)

# Adding labels and title
plt.xlabel('Player Name')
plt.ylabel('Rating')
plt.title('Top 10 Chess Players Ratings')
plt.legend()
plt.xticks(rotation=45, ha='right')

# Display the chart
plt.show()
```



Step 7: Conclusion

Grandmaster Grapher, powered by web scraping and key Python libraries such as Pandas, Matplotlib, and CSV, elegantly visualizes real-time data from 2700chess.com. With a focus on the top 10 chess players, this project not only offers a dynamic snapshot of rankings and ratings but also provides a solid foundation for further data exploration and analysis in the world of elite chess.

Step 8: Insights

Performance Trends

The visualization provided by Grandmaster Grapher reveals insightful performance trends among the top 10 chess players in classical, rapid, and blitz time controls. Understanding these trends offers valuable insights into player strengths and preferences across different game formats.

Player Comparisons

The tool facilitates easy comparisons between players, enabling enthusiasts and analysts to identify disparities and similarities in their ratings and rankings. This comparative analysis contributes to a better understanding of individual player dynamics.

Age vs. Rating Correlation

Connecting player age with their respective ratings unveils potential correlations between age and performance. This insight could spark further analysis into the dynamics of chess players' careers and the impact of age on their play in the world of elite chess.

s.

Step 9: Future Prospects

Historical Analysis

Extend the project to include historical data, enabling a longitudinal analysis of players' progress over time. This could unveil patterns and contribute to discussions on the evolution of chess strategies.

Interactive Features

Incorporate interactive elements into the visualization, allowing users to hover over data points for additional details or dynamically filter and explore specific players or time controls. This enhances user engagement and exploration.

Machine Learning Integration

Explore the integration of machine learning models to predict future player ratings based on historical data. This could provide intriguing forecasts and contribute to discussions on the future of chess and player performance.

Expanded Data Sources

Consider expanding data sources to include additional chess platforms or tournament results, providing a more comprehensive view of players' performances and rankings.

User Customization

Implement features that allow users to customize the visualization based on their preferences. This could include selecting specific players, time periods, or rating categories, enhancing the user experience.

By pursuing these future prospects, Grandmaster Grapher can evolve into a versatile tool for chess enthusiasts, researchers, and analysts, offering deeper insights and a richer understanding of the world of elite chess.

