PORTOFOLIO

Scraping Data

PRESENTED BY HANSEN VERNANDEZ

Project Introduction

Anime has become a globally recognized form of entertainment with millions of fans around the world. MyAnimeList (MAL) is one of the **most popular platforms** that provides detailed information about anime, including ratings, number of episodes, release type, and user popularity.

In this project, I performed **web scraping** on the **Top 250 Anime page on MyAnimeList** to extract key information that can be used for further data analysis. The main goals of this project are to:

- **Automatically** collect anime data from the MyAnimeList website.
- **Clean** and **explore** the collected dataset.
- Generate **insights** from the Top 250 anime, such as trends by release year, dominant release types (TV, Movie, etc.), and popularity based on user engagement.

About me

Name: Hansen Vernandez

I am an **Information Systems** student with a keen interest in **data engineering**. I have experience in managing and processing data through various projects involving data engineering, data analysis, machine learning, and system design. I am used to building data pipelines, managing data infrastructure, and ensuring data quality and integrity to support better decision making. My **career goal** is to become a **Data Engineer** who can create efficient, scalable, and impactful data solutions that drive business success.

🔍 *It is a capital mistake to theorize before one has data.*

Data Preparation

- **requests** used to send HTTP requests to the web (like visiting a webpage).
- **pandas** used for data manipulation and analysis.
- **matplotlib** and **seaborn** used for data visualization
- **duckdb** is an in-process SQL OLAP database management system.
- **BeautifulSoup** used for parsing HTML and XML documents.

On the right, is the list of URLs that point to the Top Anime by Popularity pages on **MyAnimeList**. Each URL corresponds to a page that lists anime ranked by popularity in batches of 50.

```python
import requests as rs
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import duckdb as ddb # Using
from bs4 import BeautifulSoup as bs

con = ddb.connect()
```

```python
urls = [
    'https://myanimelist.net/topanime.php?type=bypopularity',
    'https://myanimelist.net/topanime.php?type=bypopularity&limit=50',
    'https://myanimelist.net/topanime.php?type=bypopularity&limit=100',
    'https://myanimelist.net/topanime.php?type=bypopularity&limit=150',
    'https://myanimelist.net/topanime.php?type=bypopularity&limit=200',
]
```

```
data = []
header = []

for url in urls:
    response = rs.get(url)
    soup = bs(response.content, 'html.parser')


    table = soup.find('table', class_='top-ranking-table')
    header_row = soup.find('tr', class_='table-header')
    rows = table.find_all('tr', class_='ranking-list')

    if header_row and not header:
        header_columns = header_row.find_all('td')
        for td in header_columns[:3]:
            header.append(td.text.strip())
        header.append('episodes')
        header.append('date_aired')
        header.append('members')

    for tr in rows:
        row_data = []

        # Rank
        rank = tr.find('td', class_='rank')
        row_data.append(rank.text.strip() if rank else '')

        # Title
        title_td = tr.find('td', class_='title')
        title_div = title_td.find('div', class_='di-ib clearfix') if title_td else None
        row_data.append(title_div.text.strip() if title_div else '')

        # Score
        score = tr.find('td', class_='score')
        row_data.append(score.text.strip() if score else '')

        # Episodes
        info_div = tr.find('div', class_='information di-ib mt4')
        total_eps = ''
        if info_div:
            lines = list(info_div.stripped_strings)
            total_eps = lines[0]
        row_data.append(total_eps)

        # Date
        total_date = ''
        if info_div:
            lines = list(info_div.stripped_strings)
            total_date = lines[1]
        row_data.append(total_date)

        # Member
        total_members = ''
        if info_div:
            lines = list(info_div.stripped_strings)
            total_members = lines[2]
        row_data.append(total_members)

        data.append(row_data)
```

Web Scraping

The summary of this code:

1. Scrapes anime data across 5 pages (**Top 250**)
2. Collects: **Rank, Title, Score, Episodes, Aired Date, Members**
3. Organizes them into a structured data list and builds a header.

Change into Dataframe

After **scraping data** from MyAnimeList's Top 250 Anime by Popularity, convert the data into a structured **Pandas DataFrame**

```
df = pd.DataFrame(data, columns=header)
df
```

	Rank	Title	Score	episodes	date_aired	members
0	1	Shingeki no Kyojin	8.56	TV (25 eps)	Apr 2013 - Sep 2013	4,158,024 members
1	2	Death Note	8.62	TV (37 eps)	Oct 2006 - Jun 2007	4,102,384 members
2	3	Fullmetal Alchemist: Brotherhood	9.10	TV (64 eps)	Apr 2009 - Jul 2010	3,519,653 members
3	4	One Punch Man	8.49	TV (12 eps)	Oct 2015 - Dec 2015	3,380,575 members
4	5	Kimetsu no Yaiba	8.43	TV (26 eps)	Apr 2019 - Sep 2019	3,248,495 members
...	...	...	...	...	...	...
245	246	Higurashi no Naku Koro ni	7.87	TV (26 eps)	Apr 2006 - Sep 2006	834,183 members
246	247	InuYasha	7.87	TV (167 eps)	Oct 2000 - Sep 2004	832,555 members
247	248	High School DxD BorN	7.41	TV (12 eps)	Apr 2015 - Jun 2015	829,886 members
248	249	Owari no Seraph: Nagoya Kessen-hen	7.61	TV (12 eps)	Oct 2015 - Dec 2015	827,338 members
249	250	Yamada-kun to 7-nin no Majo	7.53	TV (12 eps)	Apr 2015 - Jun 2015	826,533 members

```
df[['release_type', 'episodes']] = df['episodes'].str.extract(r'^(.*?)\s*\(([^)]+)\s+eps\)')
```

Regex Breakdown: `r'^(.*?)\s*\(([^)]+)\s+eps\)'`

```
^          # Start of the string
(.*?)      # 1st Capture Group: Lazily matches any characters (release type like "TV" or "Movie")
\s*        # Matches optional whitespace (space/tab) after the release type
\(         # Escaped parenthesis: Matches literal '(' character
([^)]+)    # 2nd Capture Group: Matches 1+ characters that are NOT ')'
\s+eps     # Matches whitespace followed by "eps"
\)         # Escaped parenthesis: Matches literal ')' character
```

Changing date_aired into 2 dates which is the date_aired and date_ended

```
df[['date_aired', 'date_ended']] = df['date_aired'].str.split(' - ', expand=True).apply(lambda x: x.str.strip())
```

Changing both date_aired and date_ended into month and year in their respective parts

```
df[['month_aired', 'year_aired']] = df['date_aired'].str.strip().str.split(n=1, expand=True)
```

```
df[['month_ended', 'year_ended']] = df['date_ended'].str.strip().str.split(n=1, expand=True)
```

```
df['date_aired'] = df['date_aired'].str.replace('-', '', regex=False).str.strip()
```

```
df['year_aired'] = df['year_aired'].str.replace('-', '', regex=False).str.strip()
```

```
df['year_aired'] = pd.to_numeric(df['year_aired'], errors='coerce')
```

```
df['year_ended'] = pd.to_numeric(df['year_ended'], errors='coerce')
df['year_ended'] = df['year_ended'].fillna(9999).astype(int)
```

Remove the string 'members' from the columns, so that it only containes the numbers

```
df['members'] = df['members'].str.replace(' members', '', regex=False)
```

```
df['members'] = df['members'].str.replace('[^0-9]', '', regex=True).astype(int)
```

Data Cleaning

What had been done in data cleaning:

1. Split the **episodes** column into two new columns: **release_type** and **episodes**
2. Splitting **date_aired** into **date_aired** and **date_ended**
3. Split both **date_aired** and **date_ended** into: **month_aired, year_aired, month_ended, year_ended**
4. Removes any stray dashes (-) or white spaces from the year or date fields
5. Converts years into numeric values
6. Removes the word ' **members**' and non-numeric characters like commas.
7. Converts the cleaned values to integers (e.g., '**3,248,495 members**' → **3248495**).

Change into Dataframe

After **scraping data** from MyAnimeList's Top 250 Anime by Popularity, convert the data into a structured **Pandas DataFrame**

```
df = pd.DataFrame(data, columns=header)
df
```

	Rank	Title	Score	episodes	date_aired	members
0	1	Shingeki no Kyojin	8.56	TV (25 eps)	Apr 2013 - Sep 2013	4,158,024 members
1	2	Death Note	8.62	TV (37 eps)	Oct 2006 - Jun 2007	4,102,384 members
2	3	Fullmetal Alchemist: Brotherhood	9.10	TV (64 eps)	Apr 2009 - Jul 2010	3,519,653 members
3	4	One Punch Man	8.49	TV (12 eps)	Oct 2015 - Dec 2015	3,380,575 members
4	5	Kimetsu no Yaiba	8.43	TV (26 eps)	Apr 2019 - Sep 2019	3,248,495 members
...	...	...	...	...	...	...
245	246	Higurashi no Naku Koro ni	7.87	TV (26 eps)	Apr 2006 - Sep 2006	834,183 members
246	247	InuYasha	7.87	TV (167 eps)	Oct 2000 - Sep 2004	832,555 members
247	248	High School DxD BorN	7.41	TV (12 eps)	Apr 2015 - Jun 2015	829,886 members
248	249	Owari no Seraph: Nagoya Kessen-hen	7.61	TV (12 eps)	Oct 2015 - Dec 2015	827,338 members
249	250	Yamada-kun to 7-nin no Majo	7.53	TV (12 eps)	Apr 2015 - Jun 2015	826,533 members

Top 250 Anime List That Released per Year

```python
# Clean year_aired
df['year_aired'] = pd.to_numeric(df['year_aired'], errors='coerce')
df_clean = df.dropna(subset=['year_aired'])  # remove NaNs
df_clean['year_aired'] = df_clean['year_aired'].astype(int)

# Count and plot
top_anime_per_year = df['year_aired'].value_counts().sort_index()

df_plot = pd.DataFrame({
    'year': top_anime_per_year.index,
    'count': top_anime_per_year.values
})

plt.figure(figsize=(13, 6))
sns.barplot(data=df_plot, x='year', y='count', hue='year', palette='pastel', dodge=False, legend=False)

plt.title('Number of Top 250 Anime Released per Year', fontsize=14)
plt.xlabel('Year Aired')
plt.ylabel('Number of Anime in Top 250')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

Data Visualization

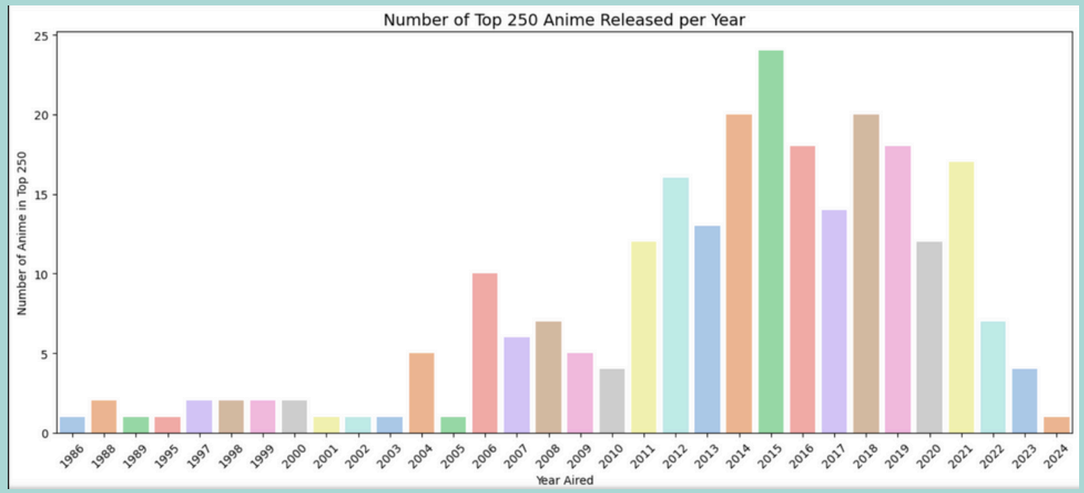
The summary of this code:

1. Cleaning the year_aired column
2. Counting anime per year
3. Preparing the data for plotting
4. Plotting the bar chart
5. Adding labels and formatting

This bar chart reveals how many of the Top 250 most popular anime were released each year. It helps identify trends, such as which years had the most influential or highly rated anime productions.

Visualization

This bar chart reveals how many of the **Top 250 most popular anime were released each year**. It helps identify trends, such as which years had the most influential or highly rated anime productions.



Thanks!

Do you have any questions?

 hansen.vernandez@gmail.com

 linkedin.com/in/hansenvernandez