Final Project MyAnimeList

Veer Mangat

Introduction

Determine relationships between
- Genre and Score
- Studio and Score
- Episode Count and Score
- Premiere Season and Score

Important to
- studios planning out shows
- viewers choosing shows to watch

Methodology

Dataset was a cleaned CSV file
    episode count, rating, rank, premiere season, studio, genre

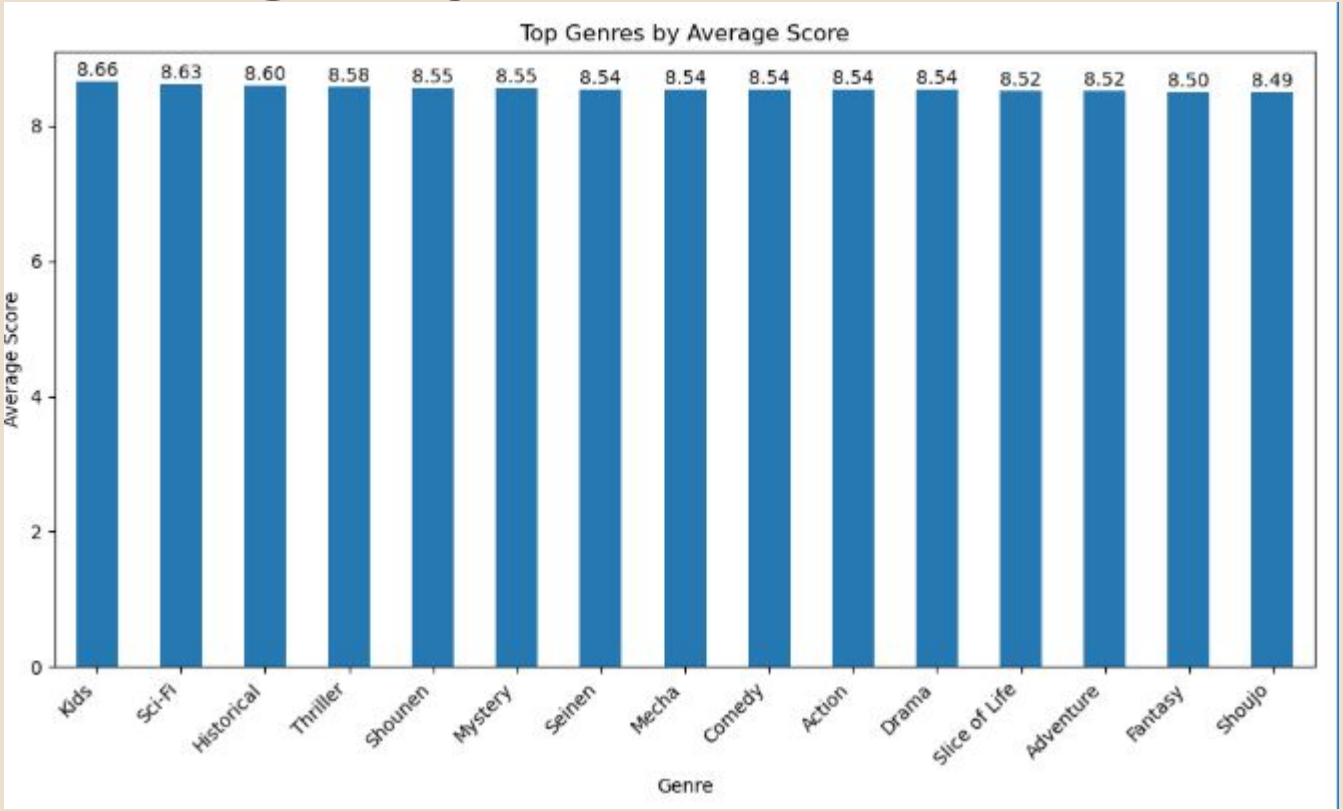
One unit of analysis is an individual Anime

Used matplotlib and pandas in a Jupyter Notebook

1

Results

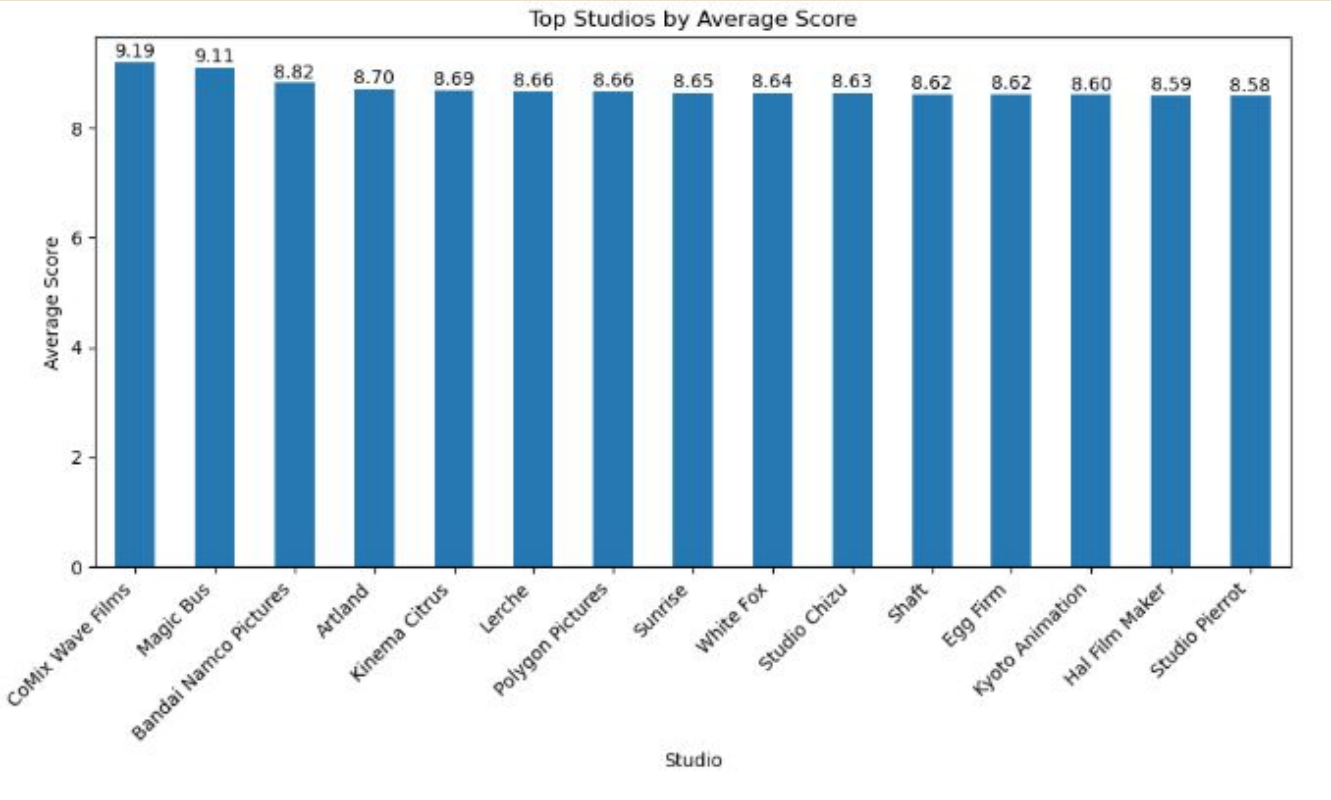
Ratings by Genre



1. Kids
2. Sci-Fi
3. Historical

Very similar average ratings
No clear genre domination

Ratings by Studio

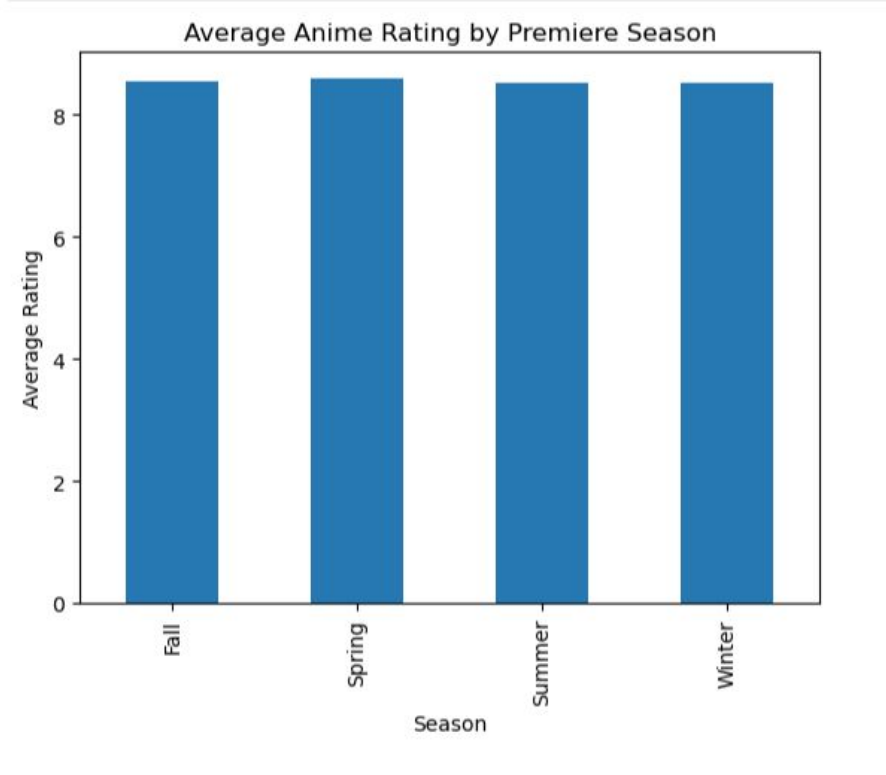


Results list notable studios

Includes dissolved or investor-only studios

Lacks studios such as Ghibli, Madhouse, MAPPA, Wit

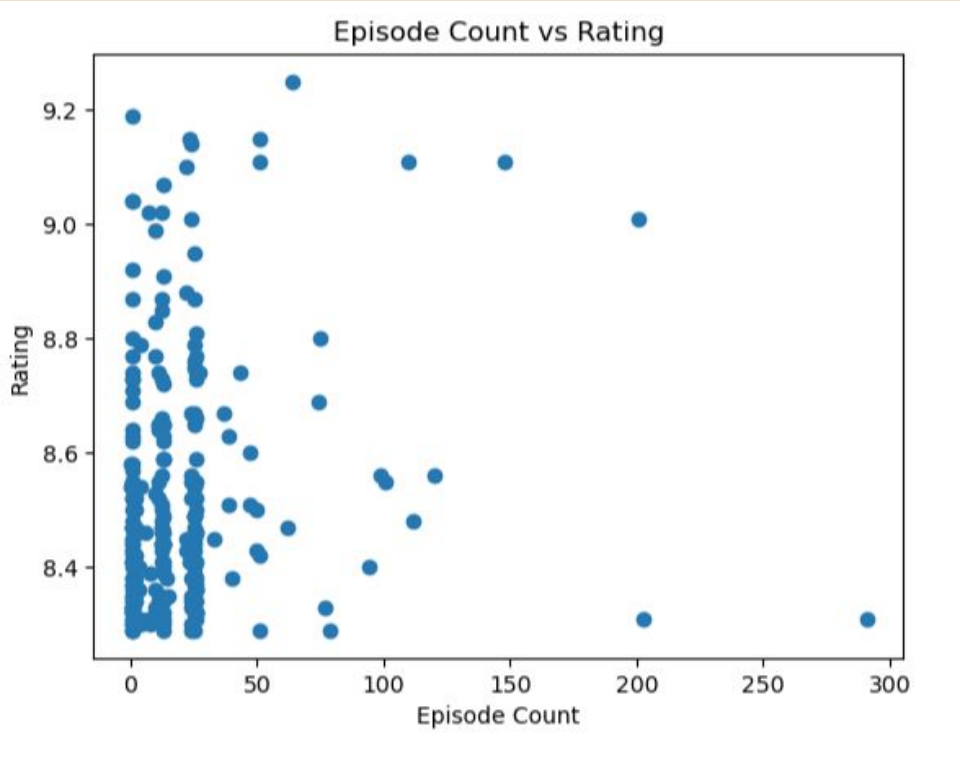
Ratings by Season



Remains identical across all seasons

Release season has minimal to no impact on Overall ratings

Ratings by Episodes



Anime episode count shows no clear Correlation with ratings

Both short and long series have widely varying scores

Future

Does popularity correlate more strongly with the ratings?

Do certain studios consistently produce high-rated anime in specific genres?

How did “niche” genres or animes polarize the overall ratings?

Appendix

```
plt.scatter(df["episodes"], df["score"])
plt.xlabel("Episode Count")
plt.ylabel("Rating")
plt.title("Episode Count vs Rating")
plt.show()
```

```
df["season"] = df["premiered"].str.split().str[0]

df.groupby("season")["score"].mean().plot(kind="bar")
plt.xlabel("Season")
plt.ylabel("Average Rating")
plt.title("Average Anime Rating by Premiere Season")
plt.show()
```

```
#split multi-genre strings into lists
df['genre_list'] = df['genre'].str.split(', ')

genre_expanded = df.explode('genre_list')

genre_avg = genre_expanded.groupby('genre_list')['score'].mean().sort_values(ascending=False)

plt.figure(figsize=(10,6))
#take top 15
ax = genre_avg.head(15).plot(kind='bar')



plt.title('Top Genres by Average Score')
plt.xlabel('Genre')
plt.ylabel('Average Score')
plt.xticks(rotation=45, ha='right')

#add numbers on top of bars
for i, v in enumerate(genre_avg.head(15)):
    ax.text(i, v + 0.02, f"{v:.2f}", ha='center', va='bottom')

plt.tight_layout()
plt.show()
```

```
df['studio_list'] = df['studio'].str.split(', ')

#expan so each studio gets its own row
studio_expanded = df.explode('studio_list')

#gr by studio and find average score
studio_avg = studio_expanded.groupby('studio_list')['score'].mean().sort_values(ascending=False)

#take top 15
top_studios = studio_avg.head(15)



plt.figure(figsize=(10,6))
ax = top_studios.plot(kind='bar')

plt.title('Top Studios by Average Score')
plt.xlabel('Studio')
plt.ylabel('Average Score')
plt.xticks(rotation=45, ha='right')

#add numbers on top of bars
for i, v in enumerate(top_studios):
    ax.text(i, v + 0.02, f"{v:.2f}", ha='center', va='bottom')

plt.tight_layout()
plt.show()
```