Declaration of A.I Usage

I acknowledge the use of [1] **GitHub Copilot** **(https://github.com/features/copilot)** to [2] **To help with the writing of code and reduce time taken to write similar lines of code repeatedly**. The prompts (03/05/2024) used include:

As with the way GitHub Copilot works you don’t really need to feed it prompts so for somethings I got off Copilot I did not use a prompt to obtain it, But I will say I did use generative A.I in sections where I was basically writing the same code but only changing a few words such as in my scatter plots, coefficients of determination and OLS, so while I wrote all of the first versions of it I had Copilot do the rest of them

The prompt outputs were as follows / are included in an appendix (depending on volume) [4] Output 1[View Appendix], Output 2[View Appendix], Output 3[View Appendix]. The output from the generative artificial intelligence was used to [5] help me write my code fasters as it can just reproduce the same code, I wrote with the slight changes I need in it for my code to work. So while I could have written it on my own I deemed it faster and a better use of time to use Copilot to do it for me.

**Appendix**

Output 1 =

plt.title('Studio vs Rating')

plt.xlabel('Studio')

plt.ylabel('Rating')

plt.scatter(data['Studio\_ID'], data['Rating'])

plt.plot(np.unique(data['Studio\_ID']), np.poly1d(np.polyfit(data['Studio\_ID'], data['Rating'], 1))(np.unique(data['Studio\_ID'])), color='red')

plt.show()

plt.title('Release Year vs Rating')

plt.xlabel('Release Year')

plt.ylabel('Rating')

plt.scatter(data['Release\_year'], data['Rating'])

plt.plot(np.unique(data['Release\_year']), np.poly1d(np.polyfit(data['Release\_year'], data['Rating'], 1))(np.unique(data['Release\_year'])), color='red')

plt.show()

plt.title('Studio vs Episodes')

plt.xlabel('Studio')

plt.ylabel('Episodes')

plt.scatter(data['Studio\_ID'], data['Episodes'])

plt.plot(np.unique(data['Studio\_ID']), np.poly1d(np.polyfit(data['Studio\_ID'], data['Episodes'], 1))(np.unique(data['Studio\_ID'])), color='red')

plt.show()

plt.title('Release Year vs Episodes')

plt.xlabel('Release Year')

plt.ylabel('Episodes')

plt.scatter(data['Release\_year'], data['Episodes'])

plt.plot(np.unique(data['Release\_year']), np.poly1d(np.polyfit(data['Release\_year'], data['Episodes'], 1))(np.unique(data['Release\_year'])), color='red')

plt.show()

plt.title('Studio vs Release Year')

plt.xlabel('Studio')

plt.ylabel('Release Year')

plt.scatter(data['Studio\_ID'], data['Release\_year'])

plt.plot(np.unique(data['Studio\_ID']), np.poly1d(np.polyfit(data['Studio\_ID'], data['Release\_year'], 1))(np.unique(data['Studio\_ID'])), color='red')

plt.show()

Output 2 =

model = LinearRegression().fit(data[['Studio\_ID']], data['Rating'])

Rating = np.array([Ratings for Ratings in range(0, 100)]).reshape(-1, 1)

model.fit(data[['Studio\_ID']], data['Rating'])

r\_sq = model.score(data[['Studio\_ID']], data['Rating'])

print('coefficient of determination:', r\_sq)

model = LinearRegression().fit(data[['Release\_year']], data['Rating'])

Rating = np.array([Ratings for Ratings in range(0, 100)]).reshape(-1, 1)

model.fit(data[['Release\_year']], data['Rating'])

r\_sq = model.score(data[['Release\_year']], data['Rating'])

print('coefficient of determination:', r\_sq)

Output 3 =

studio\_id = data['Studio\_ID']

studio\_id = sm.add\_constant(studio\_id)

model = sm.OLS(data['Rating'], studio\_id).fit()

print(model.summary())

print(model.scale\*\*0.5)

release\_year = data['Release\_year']

release\_year = sm.add\_constant(release\_year)

model = sm.OLS(data['Rating'], release\_year).fit()

print(model.summary())

print(model.scale\*\*0.5)