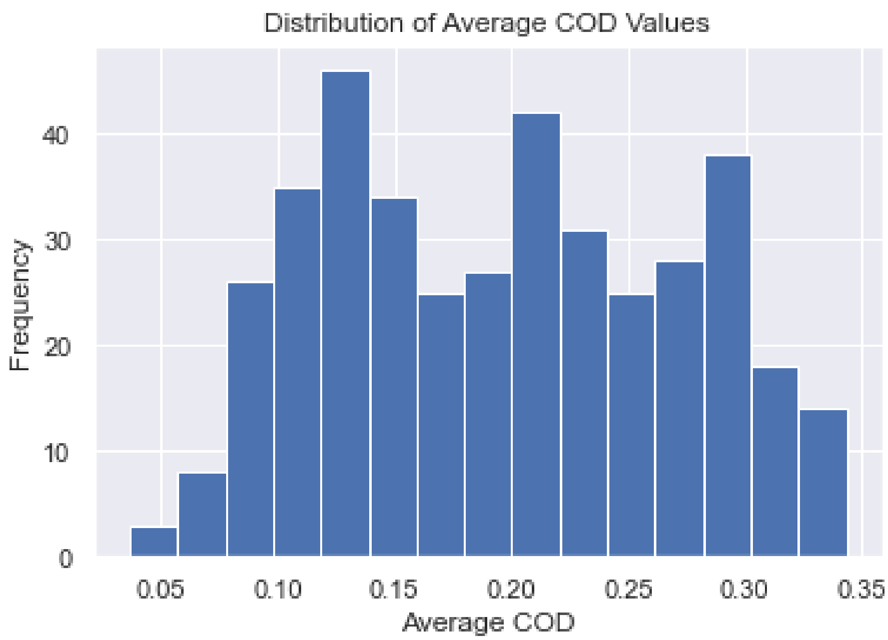
Introduction to Data Science
Data Analysis Project Report #2
Group28: Ceci Chen (zc1634), Lia Wang (rw2618), Maggie Xu (jx1206)

Question (1):

For each of the 400 movies, we first use a simple linear regression model to predict its ratings based on the ratings of the other 399 movies. And for each movie, we identify which other movie's ratings predict its ratings the best. We then calculate the Coefficient of Determination (COD) for each of these models and report the average COD of these 400 models.



Fig(a)

Fig(a) is the histogram of the 400 average COD values. From the distribution, we can observe that most of the values cluster around the 0.15 to 0.25 range. There are fewer observations with very low (near 0.05) or higher (near 0.35) average COD values Based on the table Fig(b), we measure how well the performance or characteristics of the "Best Predictor" movie predicts the corresponding aspects of the listed movie. The "10 Easiest" section lists the movies with the highest COD values, implying that their outcomes are the easiest to predict based on their "Best Predictor". For example, "Escape from LA (1996)" has a COD of 0.713554 with "Patton (1970)" as its best predictor. Conversely, the "10 Hardest" section has movies with the lowest COD values, suggesting that predicting their outcomes is more difficult based on their "Best Predictor". For instance, "Avatar (2009)" has the lowest COD value of 0.079485, with "Bad Boys (1995)" as its best predictor.

		Movie	Best Predictor	COD
10 Easiest	116	Escape from LA (1996)	Sexy Beast (2000)	0.649610
	109	Sexy Beast (2000)	The Silencers (1966)	0.659436
	377	The Lookout (2007)	Patton (1970)	0.713554
	203	Erik the Viking (1989)	I.Q. (1994)	0.731507
	298	Crimson Tide (1995)	The Straight Story (1999)	0.678454
	240	The Bandit (1996)	Best Laid Plans (1999)	0.711222
	395	Patton (1970)	The Lookout (2007)	0.713554
	287	The Straight Story (1999)	Congo (1995)	0.700569
	363	Miller's Crossing (1990)	The Lookout (2007)	0.656781
	309	Heavy Traffic (1973)	Ran (1985)	0.692734
10 Hardest	87	Shrek (2001)	Shrek 2 (2004)	0.451027
	75	Pirates of the Caribbean: Dead Man's Chest (2006)	Pirates of the Caribbean: At World's End (2007)	0.367212
	55	Clueless (1995)	Escape from LA (1996)	0.141426
	186	The Avengers (2012)	Captain America: Civil War (2016)	0.272223
	57	Shrek 2 (2004)	Shrek (2001)	0.451027
	190	The Cabin in the Woods (2012)	The Evil Dead (1981)	0.143887
	9	Black Swan (2010)	Sorority Boys (2002)	0.117080
	95	Interstellar (2014)	Torque (2004)	0.111343
	84	The Conjuring (2013)	The Exorcist (1973)	0.198474
	80	Avatar (2009)	Bad Boys (1995)	0.079485

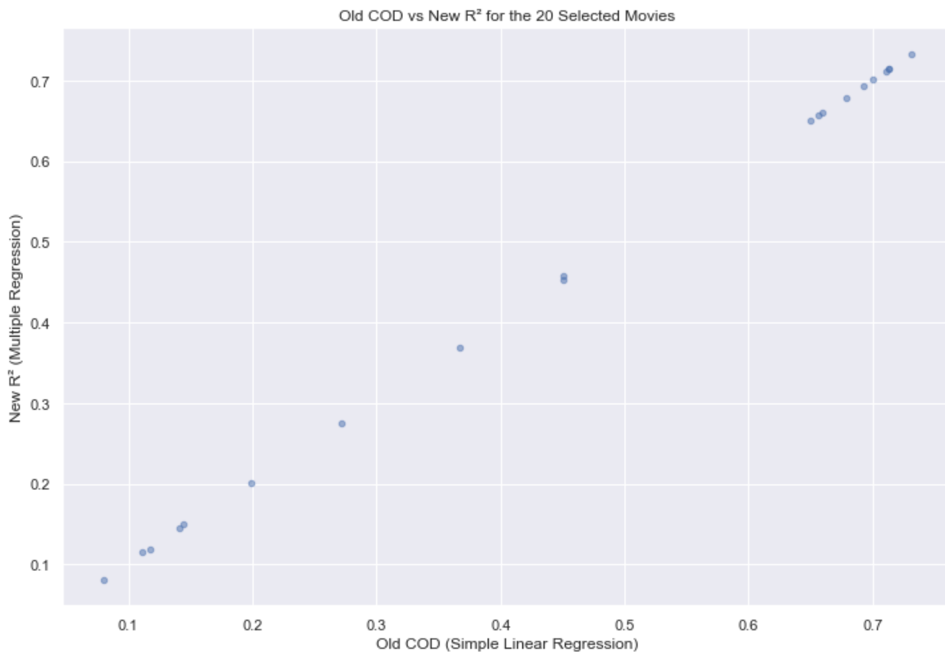
Fig(b)

Question (2):

We first extract data for gender identity (column 475), sibship status (column 476), social viewing preferences (column 477), and the best predicting movie's ratings. Fill the nans values in gender identity with value -1. For each of the 20 movies, we then construct a multiple regression model using the ratings of the best predicting movie and the three additional predictors. We then compare the old COD values in Q1 to the R2 values from the new multiple regression models. We eventually create a scatterplot with the old COD values on the x-axis and the new R2 values on the y-axis to visualize the relationship between the predictive power of the simple linear models and the multiple regression models.

	old r2	new r2
0	0.649610	0.650248
1	0.659436	0.661056
2	0.713554	0.715080
3	0.731507	0.732332
4	0.678454	0.678762
5	0.711222	0.711735
6	0.713554	0.714680
7	0.700569	0.700932
8	0.656781	0.657228
9	0.692734	0.692935
10	0.451027	0.452851
11	0.367212	0.368486
12	0.141426	0.144948
13	0.272223	0.275614
14	0.451027	0.458518
15	0.143887	0.150299
16	0.117080	0.118175
17	0.111343	0.115860
18	0.198474	0.200380
19	0.079485	0.081787

An increase in R2 would indicate that the additional predictors have contributed to a better fit of the model, whereas a decrease would suggest that they do not have a significant predictive value. Based on the plot Fig(c), we can see that most points are above where the diagonal line would be, suggesting that for most of the 20 movies, the inclusion of additional predictors has improved the predictive power of the model (higher R2). Moreover, there is a positive trend visible, where movies with a higher COD from the simple linear regression models tend to also have higher R2 values in the multiple regression models. This indicates that movies which were easier to predict with just one predictor continue to be predictable when more predictors are added, and possibly even more so.



Fig(c)

Question (3) & Question (4):

For Question (3) and (4), We pick the 30 movies from the sorted COD identified by question 1, from index 185 to 215, so that we are sure that the movies were not used in question 2. We picked 10 other movies randomly as the predictor input, using the function df.columns.difference(middle_range_movie_names).tolist()[:10]. The movies we selected are: ['10 Things I Hate About You (1999)', '10000 BC (2008)', '13 Going on 30 (2004)', '21 Grams (2003)', '25th Hour (2002)', '28 Days Later (2002)', '3000 Miles to Graceland (2001)', '8 Mile (2002)', 'A Beautiful Mind (2001)', "A Bug's Life (1998)"]

(3): For each of the 30 movies, we fit a ridge regression with the 10 randomly chosen movies. Since we only have alpha as the hyperparameter, we used Grid Search to find the best alpha by minimizing RMSE. In the tuning process, we fixed the range of alphas to make the selected alpha value not too high or too low to avoid underfitting and overfitting. After several iterations, we thought alphas in the range of 5-10 are optimal. We did an 80/20 train/test split for the model fitting. The betas are small in magnitude for the ridge regressions, which means all features have some small influence (due to shrinkage). This table below shows the results we got, including the RMSE, alphas, and betas of each regression.

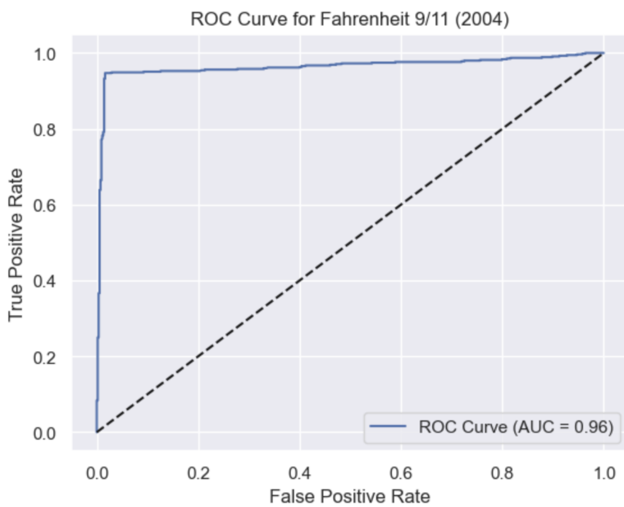
	Movie	RMSE	Alpha	Weights
0	Crossroads (2002)	0.286837	10.000000	[0.0, -0.0030763304163559614, 0.15701682415614...
1	The Green Mile (1999)	0.294768	10.000000	[0.0, 0.11262143143373228, 0.10726147364756657...
2	You're Next (2011)	0.327881	10.000000	[0.0, 0.11362057765631538, 0.21998654253825886...
3	Man on Fire (2004)	0.334144	10.000000	[0.0, 0.05076837328714767, 0.01328292524465111...
4	Aliens (1986)	0.341551	10.000000	[0.0, 0.10648715387749506, 0.17729552307535276...
5	Gone in Sixty Seconds (2000)	0.371411	10.000000	[0.0, -0.03757677271748244, -0.104401852496877...
6	Big Daddy (1999)	0.373071	10.000000	[0.0, 0.009739445675856674, 0.0758916134711297...
7	Child's Play (1988)	0.381841	10.000000	[0.0, 0.15800126465555137, 0.0855346197751011,...
8	Full Metal Jacket (1987)	0.385251	10.000000	[0.0, 0.1198217470263509, 0.060524478303064566...
9	The Thing (1982)	0.386915	10.000000	[0.0, 0.08100736122941841, -0.0613085350609957...
10	Knight and Day (2010)	0.395318	10.000000	[0.0, 0.14573642260607186, 0.22169879291993214...
11	The Others (2001)	0.395662	10.000000	[0.0, 0.02731038183068342, -0.0510164426966135...
12	12 Monkeys (1995)	0.398316	10.000000	[0.0, -0.08324338358754502, -0.010522358470396...
13	Blues Brothers 2000 (1998)	0.399286	10.000000	[0.0, 0.07965489573264324, -0.0216137949312956...
14	The Poseidon Adventure (1972)	0.402020	10.000000	[0.0, -0.04309244968495744, 0.0551652603811999...
15	Braveheart (1995)	0.406394	10.000000	[0.0, 0.14222731849589168, 0.02871717507188043...
16	Halloween (1978)	0.409804	10.000000	[0.0, 0.07605065569701676, 0.10201295214793066...
17	The Mist (2007)	0.415698	10.000000	[0.0, -0.009894505096162227, 0.184210291793396...
18	The Transporter (2002)	0.423934	8.697490	[0.0, 0.054622027119999784, 0.1956952163009663...
19	Baby Geniuses (1999)	0.425127	10.000000	[0.0, 0.0634125367073402, 0.15792855228999642,...
20	The Intouchables (2011)	0.442557	10.000000	[0.0, 0.11912197840748104, 0.18428061417173006...
21	Honey (2003)	0.444671	10.000000	[0.0, -0.02306587812311467, 0.0922347896148125...
22	Bad Boys (1995)	0.446550	7.564633	[0.0, 0.13008873500864634, 0.07547637890917878...
23	One Flew Over the Cuckoo's Nest (1975)	0.446647	10.000000	[0.0, 0.1139867980724061, 0.05683462572943384,...
24	Angels in the Outfield (1994)	0.450396	10.000000	[0.0, 0.1628216911637669, 0.1609083336743855, ...
25	Armageddon (1998)	0.458000	10.000000	[0.0, 0.08498743949718891, 0.11450061245012651...
26	Bad Boys 2 (2003)	0.463821	10.000000	[0.0, 0.06053487575992422, 0.03913442570095343...
27	Memento (2000)	0.482213	10.000000	[0.0, 0.1884058412034122, 0.07925301474850328,...
28	Rocky (1976)	0.527067	10.000000	[0.0, 0.12909622943331414, 0.05226414608488689...
29	The Truman Show (1998)	0.552330	10.000000	[0.0, 0.10100013680205899, 0.14709021719469925...

<u>(4):</u> For each of the 30 movies, we fit a LASSO regression with the 10 randomly chosen movies. We again used Grid Search to find the best alpha by minimizing RMSE. We did the same for the alpha range to avoid underfitting and overfitting. The betas are zero for non-influential features, and we found that many features were non-influential for the outcome. This table below shows the results we got, including the RMSE, alphas, and betas of each regression.

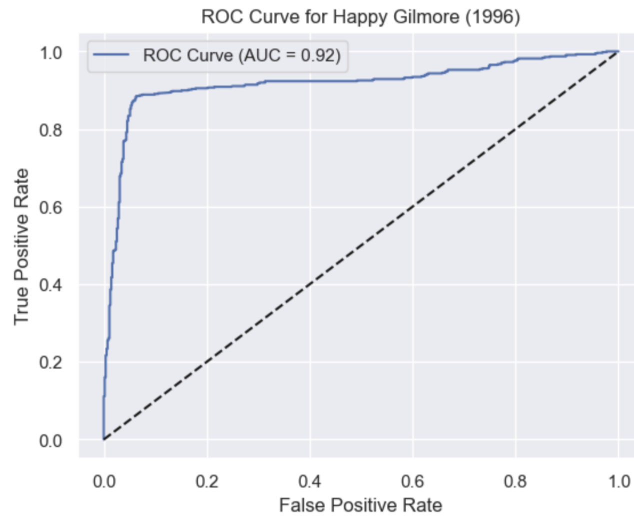
	Movie	RMSE	Alpha	Weights
0	The Green Mile (1999)	0.302870	0.013219	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
1	Man on Fire (2004)	0.303695	0.070548	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
2	Crossroads (2002)	0.314025	0.030539	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
3	Gone in Sixty Seconds (2000)	0.317967	0.030539	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
4	Blues Brothers 2000 (1998)	0.338477	0.010000	[0.0, 0.0, -0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...
5	Aliens (1986)	0.339159	0.005722	[0.0, 0.0, 0.1968913988380116, 0.0144960205193...
6	You're Next (2011)	0.357770	0.017475	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
7	Big Daddy (1999)	0.358436	0.023101	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -0.0,...
8	The Mist (2007)	0.371687	0.013219	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
9	Child's Play (1988)	0.374458	0.023101	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
10	The Thing (1982)	0.386404	0.040370	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
11	Bad Boys 2 (2003)	0.390358	0.030539	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
12	The Poseidon Adventure (1972)	0.392469	0.002477	[0.0, -0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.1487369...
13	Braveheart (1995)	0.395946	0.017475	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
14	12 Monkeys (1995)	0.396022	0.007565	[0.0, -0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -0.0...
15	Honey (2003)	0.400907	0.030539	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -0.0,...
16	Knight and Day (2010)	0.403956	0.023101	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
17	Full Metal Jacket (1987)	0.404736	0.017475	[0.0, 0.0, -0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...
18	Angels in the Outfield (1994)	0.409301	0.030539	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
19	Armageddon (1998)	0.409808	0.040370	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
20	The Transporter (2002)	0.412349	0.023101	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
21	Halloween (1978)	0.419810	0.023101	[0.0, 0.0, 0.0, -0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...
22	One Flew Over the Cuckoo's Nest (1975)	0.422321	0.017475	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
23	Baby Geniuses (1999)	0.427427	0.040370	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
24	The Others (2001)	0.432841	0.013219	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -0.0,...
25	The Intouchables (2011)	0.455198	0.040370	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
26	Memento (2000)	0.455213	0.023101	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
27	Bad Boys (1995)	0.464541	0.000811	[0.0, 0.0842106156068392, 0.0534893871485942, ...
28	The Truman Show (1998)	0.513628	0.023101	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
29	Rocky (1976)	0.526520	0.030539	[0.0, 0.0, 0.0, -0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...

Question (5):

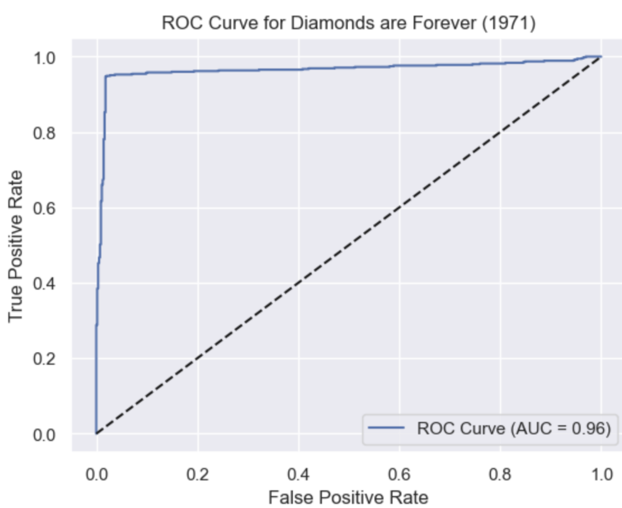
We extracted the first 400 movie columns and computed the average movie enjoyment for each user, which is the mean value of each row, from the non-imputed data. We replaced the missing row mean values with the mean values of average movie enjoyment and used this imputed data as ‘X’ (*X = avg_enjoyment.fillna(avg_enjoyment.mean()).values.reshape(-1, 1)*). We then computed the average movie ratings for each movie, which is the mean value of each column, from the non-imputed data and sorted the movies by their mean values in an ascending order. We picked the middle 4 movies based on the sorted movie list from index 198 to 201, which are *'Fahrenheit 9/11 (2004)', 'Happy Gilmore (1996)', 'Diamonds are Forever (1971)'* and *'Scream (1996)'*. For each of the movies, we first convert the imputed data of the movie into a binary dataset with its median rating value that label 1 when ratings above median and label 0 when ratings below median. With a 5-folds cross-validation, we are able to avoid overfitting and get AUC scores for each movie. Then, we fitted a logistic regression model for each movie and X respectively. After predicting each movie with X, we are able to plot out ROC curve for each movie as well as get an average AUC value and a model coefficient (beta) as shown below:



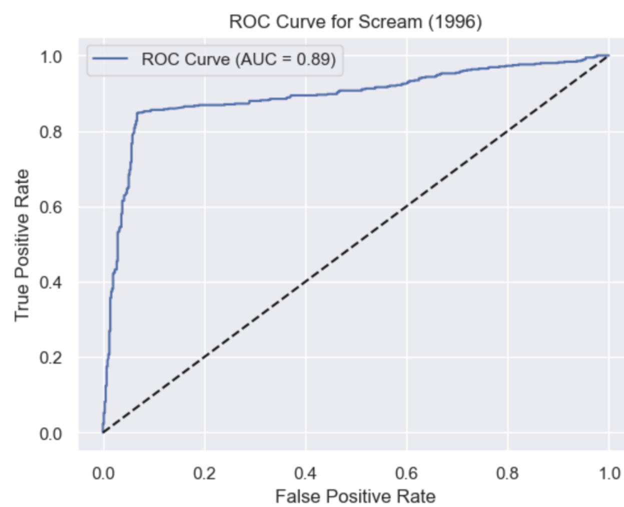

```
Movie: Fahrenheit 9/11 (2004)
Average AUC: 0.9636157403897186
Model Coefficients: [7.39636383]
```




```
Movie: Happy Gilmore (1996)
Average AUC: 0.9169490484494656
Model Coefficients: [5.201532]
```




```
Movie: Diamonds are Forever (1971)
Average AUC: 0.9647942982788689
Model Coefficients: [7.32554404]
```



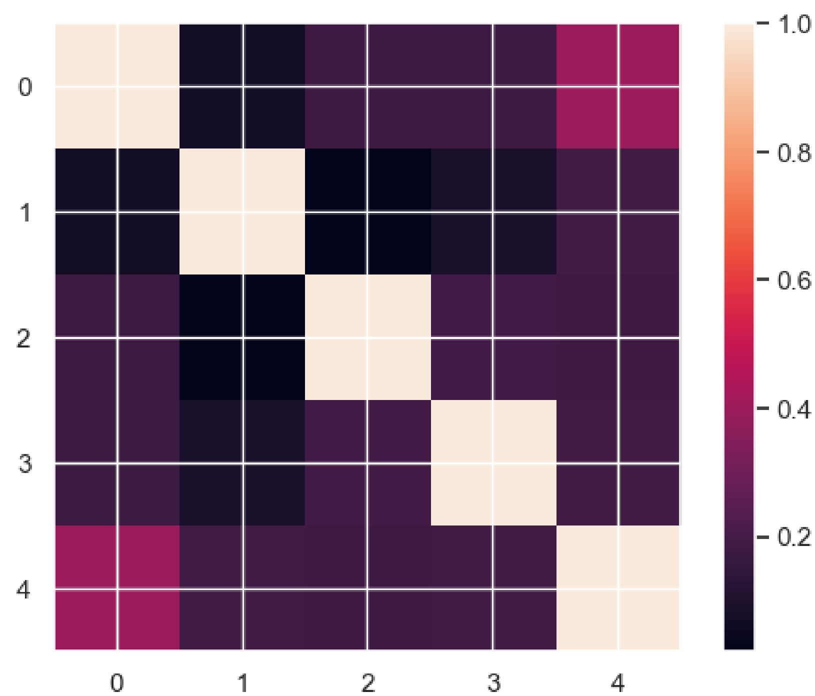

```
Movie: Scream (1996)
Average AUC: 0.8919377511562667
Model Coefficients: [4.41567957]
```

	Movie	AUC	Model_Coef (Beta)
0	Fahrenheit 9/11 (2004)	0.963616	7.396364
1	Happy Gilmore (1996)	0.916949	5.201532
2	Diamonds are Forever (1971)	0.964794	7.325544
3	Scream (1996)	0.891938	4.415680

From the above table, we can see that all of the four models have a relatively high AUC score and their ROC curves also maintain a high true positive rate while keeping false positive rate low across various thresholds, which means that our four models are having a good performance in predicting. As a result, we can conclude that the quality of all our 4 models are quite good.

<u>Extra Credit:</u> Compute the feature importance of 'Avatar (2009)' and 'Titanic (1997)' with *Random Forest Regression Model* to find out which personality traits are more influential in predicting the ratings for these two movies.

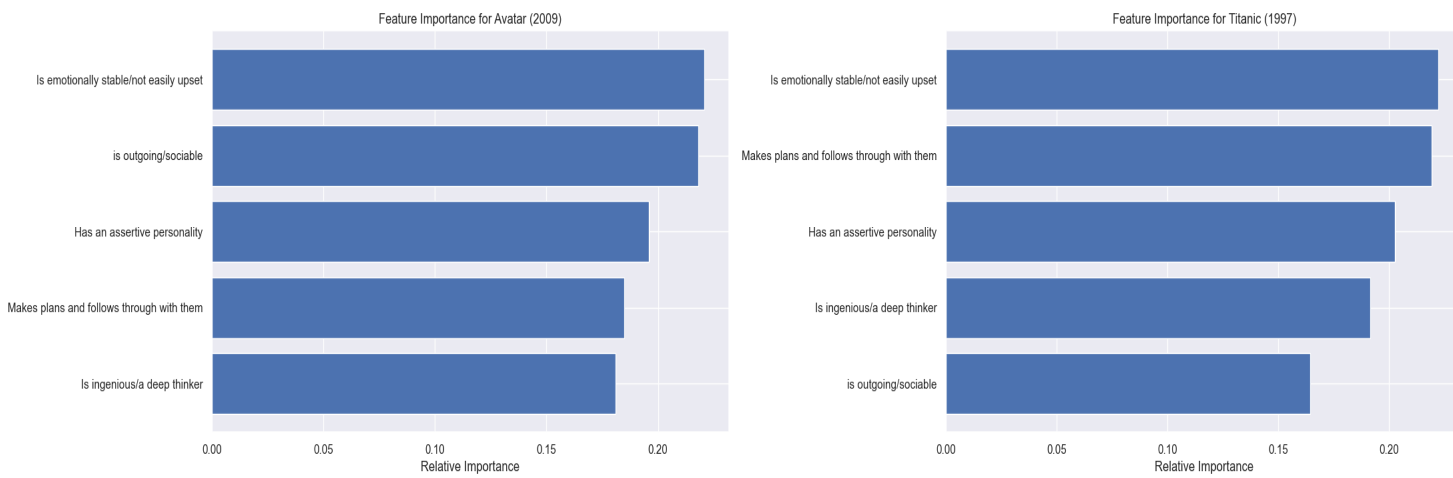
We first extracted the data of two movies from the non-imputed dataset and handled the missing values by replacing them with 0 for further computation. We then selected 5 personality related questions *('is outgoing/sociable', 'Is ingenious/a deep thinker', 'Is emotionally stable/not easily upset', 'Makes plans and follows through with them'* and *'Has an assertive personality')* from the non-imputed dataset and also handled the missing values by replacing them with 0, which used as our 5 predictors in the following model fitting step. To avoid any strong correlation or any confounds among these personality questions, we visualized the correlation matrix on 5 personality questions as the figure below. From this figure, we confirmed that there will be little confounds in this personality dataset and the datasets are ready for the model fitting step.



After doing an 80/20 train/test split on personality dataset (X) and two movies (y_avatar & y_titanic) respectively, we conducted a Random Forest Regressor model (rf_avatar & rf_titanic) for each of the two movies with the function RandomForestRegressor(n_estimators=100, random_state=42). By fitting on training data with two models, we got the following feature importance for two movies:

	Avatar (2009)	Titanic (1997)
is outgoing/sociable	0.218015	0.164354
Is ingenious/a deep thinker	0.180843	0.191425
Is emotionally stable/not easily upset	0.220690	0.222262
Makes plans and follows through with them	0.184702	0.219169
Has an assertive personality	0.195751	0.202790

We visualized the importance in a descending order in the figures below for each movie for a better understanding:



By analyzing the above two figures, we can have the following conclusion: For Avatar (2009), *'Is emotionally stable/not easily upset'* and *'is outgoing/sociable'* are more influential personality traits than the others in predicting the movie ratings; For Titanic (1997), *'Is emotionally stable/not easily upset'* and *'Makes plans and follows through with them'* are more influential personality traits than the others in predicting the movie ratings.

project2_finalcode

December 4, 2023

[1]:
```python
import numpy as np
import numpy.ma as ma
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import r2_score
from sklearn.preprocessing import PolynomialFeatures
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import KFold, StratifiedKFold, LeaveOneOut,␣
↪LeavePOut, validation_curve, learning_curve, GridSearchCV, RandomizedSearchCV
from sklearn.linear_model import LinearRegression, Lasso, Ridge, ElasticNet,␣
↪LogisticRegression
from sklearn.metrics import roc_auc_score, roc_curve
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import mean_squared_error, accuracy_score
from sklearn.model_selection import cross_val_score
from sklearn.pipeline import make_pipeline
import warnings
sns.set()
warnings.filterwarnings("ignore")
```

[2]:
```python
data0 = pd.read_csv('movieReplicationSet.csv')
```

[3]:
```python
data0.head()
```

[3]:
```
   The Life of David Gale (2003)  Wing Commander (1999)  \
0                            NaN                    NaN
1                            NaN                    NaN
2                            NaN                    NaN
3                            NaN                    NaN
4                            NaN                    NaN

   Django Unchained (2012)  Alien (1979)  \
0                      4.0           NaN
1                      1.5           NaN
2                      NaN           NaN
```

```
3                      2.0           NaN
4                      3.5           NaN

   Indiana Jones and the Last Crusade (1989)  Snatch (2000)  \
0                                        3.0            NaN
1                                        NaN            NaN
2                                        NaN            NaN
3                                        3.0            NaN
4                                        0.5            NaN

   Rambo: First Blood Part II (1985)  Fargo (1996)  \
0                                NaN           NaN
1                                NaN           NaN
2                                NaN           NaN
3                                NaN           NaN
4                                0.5           1.0

   Let the Right One In (2008)  Black Swan (2010)  …  \
0                          NaN                NaN  …
1                          NaN                NaN  …
2                          NaN                NaN  …
3                          NaN                4.0  …
4                          NaN                0.0  …

   When watching a movie I cheer or shout or talk or curse at the screen  \
0                                                1.0
1                                                3.0
2                                                5.0
3                                                3.0
4                                                2.0

   When watching a movie I feel like the things on the screen are happening to
me  \
0                                                6.0
1                                                1.0
2                                                4.0
3                                                1.0
4                                                3.0

   As a movie unfolds I start to have problems keeping track of events that
happened earlier  \
0                                                2.0
1                                                1.0
2                                                3.0
3                                                1.0
4                                                2.0
```

```
   The emotions on the screen "rub off" on me - for instance if something sad is
happening I get sad or if something frightening is happening I get scared  \
0                                                5.0
1                                                6.0
2                                                5.0
3                                                4.0
4                                                5.0

   When watching a movie I get completely immersed in the alternative reality of
the film  \
0                                                5.0
1                                                5.0
2                                                5.0
3                                                5.0
4                                                6.0

   Movies change my position on social economic or political issues  \
0                                                5.0
1                                                3.0
2                                                4.0
3                                                3.0
4                                                4.0

   When watching movies things get so intense that I have to stop watching  \
0                                                1.0
1                                                2.0
2                                                4.0
3                                                1.0
4                                                4.0

   Gender identity (1 = female; 2 = male; 3 = self-described)  \
0                                                1.0
1                                                1.0
2                                                1.0
3                                                1.0
4                                                1.0

   Are you an only child? (1: Yes; 0: No; -1: Did not respond)  \
0                                                  0
1                                                  0
2                                                  1
3                                                  0
4                                                  1

   Movies are best enjoyed alone (1: Yes; 0: No; -1: Did not respond)
0                                                  1
1                                                  0
```

```
2                                                    0
3                                                    1
4                                                    1

[5 rows x 477 columns]
```

[4]:
```
movie_columns = data0.iloc[:, :400]
remaining_columns = data0.iloc[:, 400:]

# Applying the logic separately to each part
movie_filled = movie_columns.apply(lambda x: x.fillna((x.mean() + movie_columns.
 ↪mean(axis=1)) / 2))
remaining_filled = remaining_columns.apply(lambda x: x.fillna((x.mean() +␣
 ↪remaining_columns.mean(axis=1)) / 2))

# Concatenating the two parts back together
data = pd.concat([movie_filled, remaining_filled], axis=1)
```

Data handling suggestions: To answer the questions properly, you'll have to do some kind of imputation of missing ratings (nans). Given the scope of this class, replacing them with a blend (50/50 is ok) of the arithmetic mean of each column and each row might be most suitable. Don't get used to this – there are many problems with this approach. But for now, this is ok - you'll learn more sophisticated methods later. But let's say that the rating of user 350 for movie 200 is missing and that the average rating of this user for other movies is 4 and the average rating (by other users) for this movie is 3, the to-be-imputed rating would be 3.5, using this method.

[32]:
```
data.iloc[:, :400].isnull().sum(axis=1)
```

[32]:
```
0       0
1       0
2       0
3       0
4       0
       ..
1092    0
1093    0
1094    0
1095    0
1096    0
Length: 1097, dtype: int64
```

[33]:
```
nan_counts_per_row2 = data.iloc[:, 400:].isnull().sum(axis=1)
```

[34]:
```
data.iloc[896, :400] = data.iloc[896, :400].fillna(data.mean())
data.iloc[896, 400:] = data.iloc[896, 400:].fillna(data.mean())
```

[35]:
```
df = data.iloc[:, :400]
df
```

[35]:
```
      The Life of David Gale (2003)  Wing Commander (1999)  \
0                          2.447086               2.381992
1                          2.439294               2.374200
2                          2.733065               2.667971
3                          2.282975               2.217880
4                          2.209132               2.144038
…                               …                     …
1092                       2.675658               2.610563
1093                       3.000000               4.000000
1094                       2.641923               2.576828
1095                       2.770970               2.705876
1096                       2.512595               2.447500

      Django Unchained (2012)  Alien (1979)  \
0                    4.000000      2.725235
1                    1.500000      2.717443
2                    3.234118      3.011214
3                    2.000000      2.561123
4                    3.500000      2.487281
…                         …            …
1092                 3.176711      2.953806
1093                 3.413546      3.190641
1094                 3.142976      2.920071
1095                 3.272023      3.049119
1096                 4.000000      2.790743

      Indiana Jones and the Last Crusade (1989)  Snatch (2000)  \
0                                      3.000000       2.670257
1                                      2.752945       2.662464
2                                      3.046716       2.956236
3                                      3.000000       2.506145
4                                      0.500000       2.432303
…                                           …             …
1092                                   3.500000       2.898828
1093                                   4.000000       4.000000
1094                                   2.955574       2.865093
1095                                   3.084621       2.994141
1096                                   2.500000       2.735765

      Rambo: First Blood Part II (1985)  Fargo (1996)  \
0                              2.554121      2.821232
1                              2.546329      2.813440
2                              2.840100      3.107211
3                              2.390009      2.657120
```

```
4                              0.500000      1.000000
…                                 …            …
1092                           2.782692      3.049803
1093                           2.500000      3.286638
1094                           2.748957      3.500000
1095                           2.878005      3.145116
1096                           2.619629      3.000000

      Let the Right One In (2008)  Black Swan (2010)  …  X-Men 2 (2003)  \
0                        2.619604           2.827211  …        2.828460
1                        2.611812           2.819419  …        2.820668
2                        2.905583           3.113190  …        3.114439
3                        2.455492           4.000000  …        2.664348
4                        2.381650           0.000000  …        2.500000
…                             …                …   …             …
1092                     2.848175           3.055782  …        3.057031
1093                     3.500000           3.500000  …        4.000000
1094                     2.814440           3.022047  …        3.023296
1095                     2.943488           3.151095  …        3.152344
1096                     2.685112           3.500000  …        2.893968

      The Usual Suspects (1995)  The Mask (1994)  Jaws (1975)  \
0                      2.921947         2.650951     4.000000
1                      2.914154         2.643159     2.673112
2                      3.207926         2.936930     2.966883
3                      3.000000         2.486840     2.516793
4                      2.683993         3.000000     2.442950
…                           …              …            …
1092                   3.150518         2.879523     2.909476
1093                   3.387353         4.000000     3.500000
1094                   3.116783         2.845788     2.875741
1095                   3.245831         2.974835     3.004788
1096                   2.987455         2.716460     3.500000

      Harry Potter and the Chamber of Secrets (2002)  Patton (1970)  \
0                                           0.500000       2.510773
1                                           4.000000       2.502981
2                                           3.500000       2.796752
3                                           2.500000       2.346661
4                                           2.769704       2.272819
…                                              …             …
1092                                        4.000000       2.739344
1093                                        3.500000       4.000000
1094                                        4.000000       2.705609
1095                                        2.500000       2.834657
1096                                        4.000000       2.576281
```

```
      Anaconda (1997)  Twister (1996)  MacArthur (1977)  \
0            2.519156        2.572578          2.428806
1            2.511364        2.564786          2.421013
2            2.805135        2.858557          2.714784
3            2.355044        2.408466          2.264694
4            2.281202        1.500000          2.190852
…                 …               …               …
1092         2.747727        2.801149          2.657377
1093         3.500000        4.000000          4.000000
1094         2.713992        2.767414          2.623642
1095         2.843040        2.896462          2.752690
1096         2.584664        2.638086          2.494314

      Look Who's Talking (1989)
0                      2.540410
1                      2.532618
2                      2.826389
3                      2.376299
4                      2.302456
…                         …
1092                   2.768981
1093                   4.000000
1094                   2.735247
1095                   2.864294
1096                   2.605918

[1097 rows x 400 columns]
```

0.1 Q1

For each of the 400 movies, use a simple linear regression model to predict the ratings. Use the ratings of the *other* 399 movies in the dataset to predict the ratings of each movie (that means you'll have to build 399 models for each of the 400 movies). For each of the 400 movies, find the movie that predicts ratings the best. Then report the average COD of those 400 simple linear regression models. Please include a histogram of these 400 COD values and a table with the 10 movies that are most easily predicted from the ratings of a single other movie and the 10 movies that are hardest to predict from the ratings of a single other movie (and their associated COD values, as well as which movie ratings are the best predictor, so this table should have 3 columns).

[9]:
```
best_predictors = {}
average_cod_values = {}

# Iterate over each movie to create models
for movie in df.columns:
    other_movies = df.drop(columns=[movie])
    cod_list = []
    best_cod = -float('inf')
```

```
    best_predictor = None

    for predictor in other_movies.columns:
        # Prepare the data
        X = other_movies[predictor].values.reshape(-1, 1)
        y = df[movie].values

        # Create and fit the model
        reg = LinearRegression().fit(X, y)
        y_hat = reg.predict(X)

        # Calculate COD
        r2 = r2_score(y, y_hat)
        cod_list.append(r2)

        # Check if this is the best predictor so far
        if r2 > best_cod:
            best_cod = r2
            best_predictor = predictor

    # Calculate the average COD for this movie
    average_cod = np.mean(cod_list)
    average_cod_values[movie] = average_cod

    # Store the best predictor and its COD
    best_predictors[movie] = (best_predictor, best_cod)
```

[10]:
```
best_predictors
```

[10]:
```
{'The Life of David Gale (2003)': ('The King of Marvin Gardens (1972)',
  0.5675329673680642),
 'Wing Commander (1999)': ('From Hell (2001)', 0.5606275642181675),
 'Django Unchained (2012)': ('The Life of David Gale (2003)',
  0.23233530678010406),
 'Alien (1979)': ('Aliens (1986)', 0.32954793641177993),
 'Indiana Jones and the Last Crusade (1989)': ('Indiana Jones and the Temple of
Doom (1984)',
  0.3744782737500618),
 'Snatch (2000)': ('Slackers (2002)', 0.45983684517772305),
 'Rambo: First Blood Part II (1985)': ('Pieces of April (2003)',
  0.28911668225139187),
 'Fargo (1996)': ('Brazil (1985)', 0.28672798290158985),
 'Let the Right One In (2008)': ('Slackers (2002)', 0.4406344097244561),
 'Black Swan (2010)': ('Sorority Boys (2002)', 0.11708033979272658),
 'King Kong (1976)': ('Unforgiven (1992)', 0.21748992290395797),
 'The Machinist (2004)': ('Escape from LA (1996)', 0.428765958035613),
 'A Nightmare on Elm Street (1984)': ('Tropic of Cancer (1970)',
```

```
  0.21174664733151294),
 'Brazil (1985)': ('Change of Habit (1969)', 0.525606321750943),
 'The Fast and the Furious (2001)': ('Terminator 3: Rise of the Machines
(2003)',
  0.1689914228239079),
 'Change of Habit (1969)': ('Cool Hand Luke (1967)', 0.5749972440338817),
 'American Beauty (1999)': ('Slackers (2002)', 0.22639811603072113),
 'Psycho (1960)': ('What Lies Beneath (2000)', 0.27765221706564924),
 'Terminator 3: Rise of the Machines (2003)': ('Terminator 2: Judgement Day
(1991)',
  0.30452524220042365),
 'Night of the Living Dead (1968)': ('Escape from LA (1996)',
  0.3935617517507587),
 'Man on Fire (2004)': ('Cool Hand Luke (1967)', 0.44722999341040504),
 'Star Wars: Episode IV - A New Hope (1977)': ('Star Wars: Episode V - The
Empire Strikes Back (1980)',
  0.4727745504679234),
 'The Silence of the Lambs (1991)': ('The Shining (1980)', 0.1723976576053532),
 'The Others (2001)': ('Barbarella (1968)', 0.389955443245133),
 'Minority Report (2002)': ('From Hell (2001)', 0.3357214607453456),
 'Sling Blade (1996)': ('The King of Marvin Gardens (1972)',
  0.5115934334763836),
 "Schindler's List (1993)": ('Leon (1994)', 0.22818369184942433),
 '3000 Miles to Graceland (2001)': ('The King of Marvin Gardens (1972)',
  0.558714358298324),
 'Magnolia (1999)': ('The King of Marvin Gardens (1972)', 0.46088245584732046),
 'The Karate Kid Part II (1986)': ('Chain Reaction (1996)',
  0.2529603837789236),
 'Planet of the Apes (2001)': ('Equilibrium (2002)', 0.18877948940600064),
 'The Godfather: Part II (1974)': ('The Godfather (1972)',
  0.39617822835795946),
 'Indiana Jones and the Temple of Doom (1984)': ('Indiana Jones and the Last
Crusade (1989)',
  0.3744782737500617),
 'Indiana Jones and the Raiders of the Lost Ark (1981)': ('Indiana Jones and the
Last Crusade (1989)',
  0.31366080451639533),
 'The Iron Giant (1999)': ('JFK (1991)', 0.265315865019783),
 'The Matrix Revolutions (2003)': ('The Matrix Reloaded (2003)',
  0.3134122306874445),
 'North (1994)': ('Chain Reaction (1996)', 0.5486437055810682),
 'The Lost World: Jurassic Park (1997)': ('Jurassic Park III (2001)',
  0.3093194144657909),
 'The Texas Chainsaw Massacre (1974)': ('Friday the 13th Part III (1982)',
  0.21775755829373267),
 'Taxi Driver (1976)': ('Diamonds are Forever (1971)', 0.3760593567253243),
 'Back to the Future (1985)': ('The 51st State (2001)', 0.1792274435983402),
```

```
 '13 Going on 30 (2004)': ("Can't Hardly Wait (1998)", 0.16016372820860814),
 'Sorority Boys (2002)': ('Pieces of April (2003)', 0.5093646907331285),
 'The Bridges of Madison County (1995)': ('Brazil (1985)', 0.4794519850173842),
 'Billy Madison (1995)': ('Happy Gilmore (1996)', 0.346955699822978),
 'Chain Reaction (1996)': ('North (1994)', 0.5486437055810682),
 'Batman & Robin (1997)': ('Broken Arrow (1996)', 0.2134026883872746),
 'Jurassic Park III (2001)': ('The Lost World: Jurassic Park (1997)',
  0.30931941446579103),
 'Platoon (1986)': ('The 51st State (2001)', 0.5140055325223654),
 'Signs (2002)': ('Barb Wire (1996)', 0.2984500538227536),
 'Terms of Endearment (1983)': ('Boomerang (1992)', 0.48539510629832994),
 'Mission: Impossible II (2000)': ('Boomerang (1992)', 0.18847616919961718),
 'Lost in Translation (2003)': ('Cool Hand Luke (1967)', 0.36734540692857864),
 'Star Trek: The Motion Picture (1979)': ('The Lookout (2007)',
  0.27761238679240374),
 'Inglorious Bastards (2009)': ('Django Unchained (2012)', 0.2307261350343075),
 'Clueless (1995)': ('Escape from LA (1996)', 0.141426437225317),
 'The Omen (1976)': ('Brazil (1985)', 0.3112575836120164),
 'Shrek 2 (2004)': ('Shrek (2001)', 0.4510268177555402),
 'Good Will Hunting (1997)': ('Brazil (1985)', 0.2520083892171434),
 'Just Like Heaven (2005)': ('Change of Habit (1969)', 0.45129545653278824),
 'Showgirls (1995)': ('Change of Habit (1969)', 0.45253811016249046),
 'Diamonds are Forever (1971)': ('Sexy Beast (2000)', 0.5593682670723853),
 'Crossroads (2002)': ('Pieces of April (2003)', 0.4125064980825913),
 'Pieces of April (2003)': ('Wing Commander (1999)', 0.5113219338879923),
 'Torque (2004)': ('Tropic of Cancer (1970)', 0.49813178680420256),
 'Poltergeist (1982)': ('Night of the Living Dead (1968)',
  0.23922736758249052),
 'Fear and Loathing in Las Vegas (1998)': ('Slackers (2002)',
  0.4767535689999072),
 'Barbarella (1968)': ('The King of Marvin Gardens (1972)',
  0.6065715562668179),
 'The King of Marvin Gardens (1972)': ('Barbarella (1968)',
  0.6065715562668179),
 'The Poseidon Adventure (1972)': ('The King of Marvin Gardens (1972)',
  0.4387589579009237),
 'The Rock (1996)': ('The King of Marvin Gardens (1972)', 0.43839183751048405),
 'Love Story (1970)': ('The King of Marvin Gardens (1972)',
  0.49057949084024044),
 'The Last Samurai (2003)': ('FeardotCom (2002)', 0.2944741300915371),
 'The Jungle Book (1967)': ('Tarzan (1999)', 0.18879929537034412),
 'The Exorcist (1973)': ('The Conjuring (2013)', 0.19847391483204524),
 "Pirates of the Caribbean: Dead Man's Chest (2006)": ("Pirates of the
Caribbean: At World's End (2007)",
  0.367212431026693),
 'Gone in Sixty Seconds (2000)': ('The 51st State (2001)',
  0.41102463304683523),
```

```
 'Funny Girl (1968)': ('The King of Marvin Gardens (1972)',
  0.5166055041789005),
 'Honey (2003)': ('Sexy Beast (2000)', 0.4029410853904397),
 'Blues Brothers 2000 (1998)': ('The 51st State (2001)', 0.4216782524022984),
 'Avatar (2009)': ('Bad Boys (1995)', 0.07948469093084631),
 'The Pianist (2002)': ('Torque (2004)', 0.2952376833317627),
 'Godzilla (1998)': ('The Final Conflict (1981)', 0.23642588067285408),
 'Fight Club (1999)': ('Snatch (2000)', 0.18631096026793525),
 'The Conjuring (2013)': ('The Exorcist (1973)', 0.19847391483204524),
 'Top Gun (1986)': ('The Lookout (2007)', 0.2876855563160826),
 'Slackers (2002)': ('Change of Habit (1969)', 0.5699468166410963),
 'Shrek (2001)': ('Shrek 2 (2004)', 0.4510268177555402),
 '12 Monkeys (1995)': ('Change of Habit (1969)', 0.40187112859351204),
 'From Hell (2001)': ('The King of Marvin Gardens (1972)', 0.5622853817697189),
 'Dead Poets Society (1989)': ('Sexy Beast (2000)', 0.23353635369814807),
 'Once Upon a Time in America (1984)': ('Pieces of April (2003)',
  0.501260155402371),
 'Equilibrium (2002)': ('Change of Habit (1969)', 0.4433325904344152),
 'Star Wars: Episode II - Attack of the Clones (2002)': ('Star Wars: Episode 1 -
The Phantom Menace (1999)',
  0.4010061938750953),
 'The Thing (1982)': ('Sexy Beast (2000)', 0.3765194385193833),
 'Interstellar (2014)': ('Torque (2004)', 0.11134259626426413),
 'Full Metal Jacket (1987)': ('Escape from LA (1996)', 0.409381411830151),
 'Big Fish (2003)': ('Chain Reaction (1996)', 0.35984601041296693),
 'Cool Hand Luke (1967)': ('Change of Habit (1969)', 0.5749972440338817),
 'A Beautiful Mind (2001)': ('The King of Marvin Gardens (1972)',
  0.2917601050542089),
 'Sholay (1978)': ('The 51st State (2001)', 0.579596595565732),
 'The 51st State (2001)': ('Sexy Beast (2000)', 0.6323439673995209),
 'Die Hard With a Vengeance (1995)': ('De-Lovely (2004)', 0.4881081951802324),
 'Elf (2003)': ('The Doom Generation (1995)', 0.19858879359094883),
 'The Blue Lagoon (1980)': ('Crimson Tide (1995)', 0.4289149370934736),
 'Hellraiser (1987)': ('MacArthur (1977)', 0.4555512074582013),
 'Moonraker (1979)': ('Unforgiven (1992)', 0.6190337625229674),
 'Leon (1994)': ('Once Upon a Time in the West (1968)', 0.49147734717177416),
 'Mystic River (2003)': ('Escape from LA (1996)', 0.5724541927325252),
 'Sexy Beast (2000)': ('The Silencers (1966)', 0.6594355043318669),
 'Beetle Juice (1988)': ('De-Lovely (2004)', 0.2644118282477149),
 'Andaz Apna Apna (1994)': ('The Doom Generation (1995)', 0.6186491631397999),
 'The Proposal (2009)': ('The Vow (2012)', 0.18804119443379363),
 'The Shining (1980)': ('Psycho (1960)', 0.23705154245096038),
 'The Land That Time Forgot (1974)': ('The Bandit (1996)', 0.5607136652452841),
 'The Perfect Storm (2000)': ('Stir Crazy (1980)', 0.5042670997545744),
 'Escape from LA (1996)': ('Sexy Beast (2000)', 0.6496095342338333),
 'Shutter Island (2010)': ("Miller's Crossing (1990)", 0.1822618592594336),
 'JFK (1991)': ('Unforgiven (1992)', 0.5566577680328803),
```

```
 'Barb Wire (1996)': ('The Firm (1993)', 0.6327462491912709),
 'Oldboy (2003)': ('Andaz Apna Apna (1994)', 0.45373787011193845),
 'Carrie (1976)': ('Sexy Beast (2000)', 0.27450454571671823),
 'The Good the Bad and the Ugly (1966)': ('The 51st State (2001)',
  0.43953187121124326),
 'Speed 2: Cruise Control (1997)': ('The Lookout (2007)', 0.5232381272294993),
 'The Lord of the Rings: The Fellowship of the Ring (2001)': ('The Lord of the
Rings: The Two Towers (2002)',
  0.6651097646606816),
 'The Talented Mr. Ripley (1999)': ('The Lookout (2007)', 0.44779347361133226),
 'Casino (1995)': ('Escape from LA (1996)', 0.5307042731618601),
 'A Time to Kill (1996)': ('Boomerang (1992)', 0.6110096761404729),
 'Blazing Saddles (1974)': ('FeardotCom (2002)', 0.477276207145264),
 'The Doom Generation (1995)': ('Andaz Apna Apna (1994)', 0.6186491631397999),
 'Armageddon (1998)': ('Billy Jack (1971)', 0.3656646463245087),
 'X-Men (2000)': ('X-Men 2 (2003)', 0.45307251800810433),
 'Arachnophobia (1990)': ('The Land That Time Forgot (1974)',
  0.39969578204849276),
 'Stir Crazy (1980)': ('The Silencers (1966)', 0.6819831258114571),
 'Billy Jack (1971)': ('Sexy Beast (2000)', 0.6263251337499547),
 'The Silencers (1966)': ('Stir Crazy (1980)', 0.681983125811457),
 'The Three Musketeers (1993)': ('The 51st State (2001)', 0.46545927543595256),
 'Girl Interrupted (1999)': ('Sexy Beast (2000)', 0.36462482484888026),
 'Finding Nemo (2003)': ('Monsters  Inc.(2001)', 0.32851214942146456),
 'Tropic of Cancer (1970)': ('Stir Crazy (1980)', 0.5846898600473227),
 'The Sixth Sense (1999)': ('Boomerang (1992)', 0.26938991773730425),
 'I Know What You Did Last Summer (1997)': ('A Time to Kill (1996)',
  0.3024000214198381),
 'Indiana Jones and the Kingdom of the Crystal Skull (2008)': ('Crimson Tide
(1995)',
  0.21165478575965402),
 'Divine Secrets of the Ya-Ya Sisterhood (2002)': ('Stir Crazy (1980)',
  0.6198002669878374),
 'Ace Ventura: When Nature Calls (1995)': ('FeardotCom (2002)',
  0.3230798212990107),
 'Dances with Wolves (1990)': ('The Deer Hunter (1978)', 0.4235507851172867),
 'Date and Switch (2014)': ('Boomerang (1992)', 0.5904189099662547),
 'The Intouchables (2011)': ('The Station Agent (2003)', 0.39633600477723063),
 'Mrs. Doubtfire (1993)': ('Analyze That (2002)', 0.26873173507467896),
 'Ghostbusters (2016)': ('The Doom Generation (1995)', 0.1920869574739884),
 'Almost Famous (2000)': ('Sexy Beast (2000)', 0.4004940928581887),
 'Blade Runner (1982)': ('The Final Conflict (1981)', 0.31562411638911303),
 'Unforgiven (1992)': ('Sexy Beast (2000)', 0.6333133697330241),
 "Rosemary's Baby (1968)": ('Close Encounters of the Third Kind (1977)',
  0.3940414297384146),
 'Cheaper by the Dozen (2003)': ('Moonraker (1979)', 0.2333794139978912),
 "Can't Hardly Wait (1998)": ('The Straight Story (1999)', 0.5593234185003938),
```

```
 'Die Another Day (2002)': ('Escape from LA (1996)', 0.48097298232755425),
 'Toy Story 2 (1999)': ('Toy Story 3 (2010)', 0.4649078494661363),
 'Transformers: Age of Extinction (2014)': ('Iron Man 3 (2013)',
  0.22095781054766084),
 'Like Stars on Earth (2007)': ("Father's Day (1997)", 0.6030690044338824),
 'Terminator 2: Judgement Day (1991)': ('Terminator 3: Rise of the Machines
(2003)',
  0.30452524220042365),
 '25th Hour (2002)': ('Sexy Beast (2000)', 0.5118116636201016),
 "Who's Afraid of Virginia Woolf (1966)": ('Date and Switch (2014)',
  0.4919069238844387),
 'Adaption (2002)': ('Boomerang (1992)', 0.5730054666414603),
 'Life is Beautiful (1997)': ('De-Lovely (2004)', 0.45807452090588796),
 'Room (2015)': ("Bram Stoker's Dracula (1992)", 0.24554983891668047),
 'Scream (1996)': ('Scream 3 (2000)', 0.2666806734044629),
 'The Evil Dead (1981)': ('Escape from LA (1996)', 0.42869529140270746),
 'Gangs of New York (2002)': ('Casino (1995)', 0.4787682578224548),
 'Stand By Me (1986)': ('Sexy Beast (2000)', 0.4516926929036629),
 'The Vow (2012)': ('The Proposal (2009)', 0.18804119443379375),
 'Toy Story 3 (2010)': ('Toy Story 2 (1999)', 0.4649078494661363),
 'The Matrix Reloaded (2003)': ('The Matrix Revolutions (2003)',
  0.3134122306874443),
 'Once Upon a Time in the West (1968)': ('FeardotCom (2002)',
  0.6210084837658294),
 'Star Wars: Episode V - The Empire Strikes Back (1980)': ('Star Wars: Episode
VI - The Return of the Jedi (1983)',
  0.500443762521317),
 'War Games (1983)': ('The 51st State (2001)', 0.5517758033735717),
 'Kill Bill: Vol. 2 (2004)': ('Kill Bill: Vol. 1 (2003)', 0.509623270383415),
 'Saving Private Ryan (1998)': ('Unforgiven (1992)', 0.22385033240338614),
 'Just Married (2003)': ('De-Lovely (2004)', 0.44366097725463416),
 'Being John Malkovich (1999)': ('Escape from LA (1996)', 0.5069031823128882),
 "Father's Day (1997)": ('The Lookout (2007)', 0.6225047855681649),
 'Batman (1989)': ('The Lookout (2007)', 0.3106768235976567),
 'Se7en (1995)': ('Escape from LA (1996)', 0.36009431204746156),
 'Happy Gilmore (1996)': ('Billy Madison (1995)', 0.34695569982297814),
 'My Big Fat Greek Wedding (2002)': ("Father's Day (1997)",
  0.2374130283145185),
 'Boomerang (1992)': ('Sexy Beast (2000)', 0.6292893132747557),
 'The Avengers (2012)': ('Captain America: Civil War (2016)',
  0.27222306742529523),
 'In America (2002)': ('Stir Crazy (1980)', 0.6358040642735169),
 'Tarzan (1999)': ('Best Laid Plans (1999)', 0.24123412792901122),
 'Scent of a Woman (1992)': ('Sexy Beast (2000)', 0.5349927079807026),
 'The Cabin in the Woods (2012)': ('The Evil Dead (1981)',
  0.14388686955485108),
 'Spider-Man (2002)': ('Batman (1989)', 0.17667963521735575),
```

'Broken Arrow (1996)': ('The Bandit (1996)', 0.6106470169683111),
'Baby Geniuses (1999)': ('Sexy Beast (2000)', 0.39495523216292006),
'Battlefield Earth (2000)': ('Sexy Beast (2000)', 0.6031072175254055),
'The Firm (1993)': ('Barb Wire (1996)', 0.6327462491912708),
'De-Lovely (2004)': ('The Firm (1993)', 0.6305766639060497),
'Die Hard (1988)': ('Escape from LA (1996)', 0.36732443127717784),
'The Lord of the Rings: The Two Towers (2002)': ('The Lord of the Rings: The Fellowship of the Ring (2001)',
 0.6651097646606814),
'The Blair Witch Project (1999)': ('FeardotCom (2002)', 0.23135619388263173),
'Judge Dredd (1995)': ('The Bandit (1996)', 0.6139590247785986),
'10 Things I Hate About You (1999)': ('The Insider (1999)',
 0.26824573614951697),
'The Insider (1999)': ('I.Q. (1994)', 0.6394983899860376),
'Erik the Viking (1989)': ('I.Q. (1994)', 0.731507476731657),
"Pirates of the Caribbean: At World's End (2007)": ('Pirates of the Caribbean: The Curse of the Black Pearl (2003)',
 0.39487603864032406),
'The Ring (2002)': ('Erik the Viking (1989)', 0.21037108215809563),
'The Truman Show (1998)': ('The Deer Hunter (1978)', 0.3923193454414887),
'Forrest Gump (1994)': ('Rain Man (1988)', 0.17548272387581054),
'I.Q. (1994)': ('Erik the Viking (1989)', 0.731507476731657),
'Goodfellas (1990)': ('Patton (1970)', 0.4161597725339885),
'Uptown Girls (2003)': ('The Firm (1993)', 0.4289831185912517),
'Beauty and the Beauty (1991)': ('War Games (1983)', 0.266712140492971),
'Black Hawk Down (2001)': ('Erik the Viking (1989)', 0.4600544396549501),
'Knight and Day (2010)': ('Point Break (1991)', 0.4268440575456022),
'The Shawshank Redemption (1994)': ('Crimson Tide (1995)',
 0.36090438569871985),
'Sleepy Hollow (1999)': ('Erik the Viking (1989)', 0.469013529620556),
'The Holiday (2006)': ('What Lies Beneath (2000)', 0.3447317675535898),
'Sixteen Candles (1984)': ('Best Laid Plans (1999)', 0.2919731409204682),
'10000 BC (2008)': ('The Bandit (1996)', 0.34657854995103166),
'Austin Powers: The Spy Who Shagged Me (1999)': ('Austin Powers in Goldmember (2002)',
 0.41411802815118837),
'The Lion King (1994)': ('Toy Story (1995)', 0.2864665585541085),
"Child's Play (1988)": ('The Lookout (2007)', 0.39804514260446466),
'Anger Management (2002)': ('The Bandit (1996)', 0.4044004250316604),
'Angels in the Outfield (1994)': ('Heavy Traffic (1973)', 0.4054095875331265),
'Wild Wild West (1999)': ('Crimson Tide (1995)', 0.4958107957317134),
'Split (2016)': ('The Lookout (2007)', 0.24445643399755868),
'Bad Boys (1995)': ('Bad Boys 2 (2003)', 0.48578461879700974),
'The Prestige (2006)': ('What Lies Beneath (2000)', 0.4477433358890974),
'American Graffiti (1973)': ('Sexy Beast (2000)', 0.584905283421949),
'Air Force One (1997)': ('The Final Conflict (1981)', 0.5692789141276591),
"Harry Potter and the Sorcerer's Stone (2001)": ('Harry Potter and the Chamber

```
of Secrets (2002)',
  0.5639080588803832),
 'Close Encounters of the Third Kind (1977)': ('I.Q. (1994)',
  0.6346072938424792),
 'Hollow Man (2000)': ('Crimson Tide (1995)', 0.6248934705853029),
 'Point Break (1991)': ('What Lies Beneath (2000)', 0.6020763412153762),
 'I Robot (2004)': ('Sexy Beast (2000)', 0.3058090246266103),
 'Batman: The Dark Knight (2008)': ('Crimson Tide (1995)',
  0.20690691530697125),
 'Ghost (1990)': ('Crimson Tide (1995)', 0.47548568057992247),
 "A Bug's Life (1998)": ('I.Q. (1994)', 0.23723801304914904),
 'American Pie (1999)': ('Crimson Tide (1995)', 0.24255736561609254),
 'Daredevil (2003)': ('The Bandit (1996)', 0.2771523965023349),
 'The Bandit (1996)': ('Best Laid Plans (1999)', 0.7112222468014325),
 'Grease (1978)': ('The Straight Story (1999)', 0.2764343445926314),
 'The Girl Next Door (2004)': ('Erik the Viking (1989)', 0.46421707030321835),
 'The Godfather (1972)': ('The Godfather: Part II (1974)',
  0.39617822835795946),
 'Cloverfield (2008)': ('Blow (2001)', 0.3229678994009517),
 'Rush Hour 2 (2001)': ('The Mummy Returns (2001)', 0.2875419018575336),
 'Bruce Almighty (2003)': ('Crimson Tide (1995)', 0.24973080421660543),
 'Girl With a Pearl Earring (2003)': ('What Lies Beneath (2000)',
  0.5330994935354925),
 'Grown Ups 2 (2013)': ('The Core (2003)', 0.17111918539600846),
 'Best Laid Plans (1999)': ('The Bandit (1996)', 0.7112222468014324),
 "Bram Stoker's Dracula (1992)": ('The Straight Story (1999)',
  0.6313284174648424),
 'Fahrenheit 9/11 (2004)': ('Erik the Viking (1989)', 0.5198458306313596),
 'Donnie Darko (2001)': ('Patton (1970)', 0.36067537079665835),
 'Bad Teacher (2011)': ('Hollow Man (2000)', 0.17815055825328208),
 'Cable Guy (1996)': ('Crimson Tide (1995)', 0.5798731998183908),
 'Ice Age (2002)': ('On Golden Pond (1981)', 0.18829625359866053),
 'Misery (1990)': ('What Lies Beneath (2000)', 0.538283805050229),
 '8 Mile (2002)': ('The Final Conflict (1981)', 0.27153119655474733),
 'Harry Potter and the Deathly Hallows: Part 2 (2011)': ("Harry Potter and the
Sorcerer's Stone (2001)",
  0.46524427603107166),
 'Ouija: Origin of Evil (2016)': ('The Lookout (2007)', 0.2664071247159604),
 'The Deer Hunter (1978)': ('A Perfect Murder (1998)', 0.5826765012002226),
 "There's Something About Mary (1998)": ('Patton (1970)', 0.4482141186433588),
 'Zoolander (2001)': ('Sexy Beast (2000)', 0.2213322335372352),
 'The Core (2003)': ('I.Q. (1994)', 0.6513410144105207),
 'Spirited Away (2001)': ('Erik the Viking (1989)', 0.23613333031054096),
 'Rocky (1976)': ('Best Laid Plans (1999)', 0.36912719959849993),
 'Traffic (2000)': ('The Bandit (1996)', 0.5383396035696302),
 'Monsters  Inc.(2001)': ('Toy Story (1995)', 0.3818434877352124),
 'Thoroughly Modern Millie (1967)': ('The Station Agent (2003)',
```

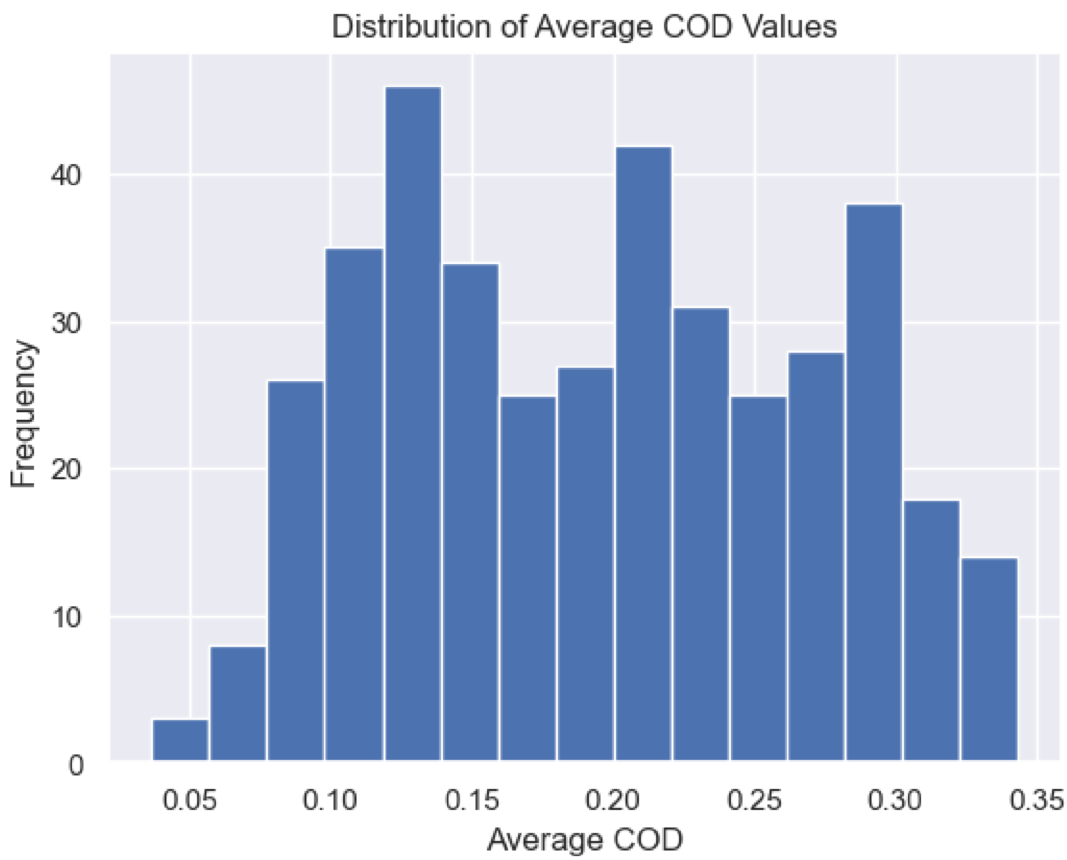
```
  0.5014864880671385),
 'Requiem for a Dream (2000)': ('Once Upon a Time in the West (1968)',
  0.468272574195977),
 'Downfall (2004)': ('The Lookout (2007)', 0.6432951942975043),
 'L.A. Confidential (1997)': ('The Deer Hunter (1978)', 0.5767612496270459),
 'Chicago (2002)': ('What Lies Beneath (2000)', 0.4965947103384032),
 'Star Wars: Episode 1 - The Phantom Menace (1999)': ('Star Wars: Episode II -
Attack of the Clones (2002)',
  0.4010061938750952),
 'Rain Man (1988)': ('Erik the Viking (1989)', 0.46186653126171995),
 'What Lies Beneath (2000)': ('Erik the Viking (1989)', 0.655060560957997),
 'Toy Story (1995)': ('Toy Story 2 (1999)', 0.4506862901251196),
 "Boy's Don't Cry (1999)": ('Erik the Viking (1989)', 0.5357808238386108),
 'Pearl Harbor (2001)': ('The Final Conflict (1981)', 0.3239298874561414),
 'A.I. Artificial Intelligence (2001)': ('I.Q. (1994)', 0.5081179085042079),
 'The Sting (1973)': ('I.Q. (1994)', 0.5750405394168969),
 'Scream 3 (2000)': ('Friday the 13th Part III (1982)', 0.329702765979356),
 'Congo (1995)': ('The Straight Story (1999)', 0.7005689836445022),
 'Bowling For Columbine (2002)': ('What Lies Beneath (2000)',
  0.541365966851986),
 'What Women Want (2000)': ('What Lies Beneath (2000)', 0.5096411595211338),
 'Home Alone (1990)': ('The Lookout (2007)', 0.2164818093848906),
 'How the Grinch Stole Christmas (2000)': ('Erik the Viking (1989)',
  0.2354168149199536),
 'The Straight Story (1999)': ('Congo (1995)', 0.7005689836445022),
 'The Hulk (2003)': ('The Bandit (1996)', 0.2735440714392723),
 'Gigli (2002)': ('What Lies Beneath (2000)', 0.5552540059374398),
 'Rocky V (1991)': ('The Bandit (1996)', 0.38946486297888794),
 'The Visit (2015)': ('Downfall (2004)', 0.30666286096448125),
 'Titanic (1997)': ('Cocktail (1988)', 0.15413567330482103),
 'A Clockwork Orange (1971)': ('Heavy Traffic (1973)', 0.3439020085320824),
 "Charlie's Angels (2000)": ('Crimson Tide (1995)', 0.29254733279393363),
 'Friday the 13th Part III (1982)': ('The Final Conflict (1981)',
  0.3556565722326055),
 'Hannibal (2001)': ('The Straight Story (1999)', 0.40715066243622655),
 'Pulp Fiction (1994)': ('The Deer Hunter (1978)', 0.2701543518011379),
 'Crimson Tide (1995)': ('The Straight Story (1999)', 0.6784535648314336),
 'Blow (2001)': ('The Straight Story (1999)', 0.6391034530171278),
 'Ran (1985)': ('Heavy Traffic (1973)', 0.6927335239652475),
 'Mulholland Dr. (2001)': ('FeardotCom (2002)', 0.52459992132808),
 'Apocalypse Now (1979)': ('Midnight Cowboy (1969)', 0.40651790736162496),
 'Cinema Paradiso (1988)': ('MacArthur (1977)', 0.564975452997534),
 'Double Jeopardy (1999)': ('The Lookout (2007)', 0.5872946161757617),
 'The Big Lebowski (1998)': ('Escape from LA (1996)', 0.41767373518258677),
 'The Matrix (1999)': ("Miller's Crossing (1990)", 0.27451243594544805),
 'The Lord of the Rings: The Return of the King (2003)': ('The Lord of the
Rings: The Fellowship of the Ring (2001)',
```

```
  0.5584103130598053),
 'Reservoir Dogs (1992)': ('Patton (1970)', 0.4464863510133942),
 'Heavy Traffic (1973)': ('Ran (1985)', 0.6927335239652475),
 'Memento (2000)': ("Miller's Crossing (1990)", 0.3867827621812314),
 'Dogville (2003)': ("Miller's Crossing (1990)", 0.517941991308019),
 'American Psycho (2000)': ('Tropic of Cancer (1970)', 0.2979851610111176),
 'Kill Bill: Vol. 1 (2003)': ('Kill Bill: Vol. 2 (2004)', 0.509623270383415),
 'The Fugitive (1993)': ("Miller's Crossing (1990)", 0.5359125672665054),
 'Bend it Like Beckham (2002)': ('Crimson Tide (1995)', 0.2612300033002951),
 'Austin Powers in Goldmember (2002)': ('Austin Powers: The Spy Who Shagged Me
(1999)',
  0.41411802815118837),
 'The Mummy Returns (2001)': ('The Mummy (1999)', 0.4436621311632055),
 'The Nightmare Before Christmas (1993)': ('The Insider (1999)',
  0.21068277590228535),
 'La La Land (2016)': ('The Lookout (2007)', 0.14851372649350147),
 'Flowers in the Attic (1987)': ('The Lookout (2007)', 0.4808673027729854),
 '28 Days Later (2002)': ("Miller's Crossing (1990)", 0.42384844923471443),
 'The Princess Bride (1987)': ('Cocktail (1988)', 0.22292893520504675),
 'The Green Mile (1999)': ('The Game (1997)', 0.4329819455909022),
 'Predator (1987)': ('The Final Conflict (1981)', 0.449550255392416),
 'A Night at the Roxbury (1998)': ('Patton (1970)', 0.5527256724178615),
 'Ed Wood (1994)': ('The Station Agent (2003)', 0.6713849837329366),
 'Aliens (1986)': ("Miller's Crossing (1990)", 0.4095733957811697),
 'Meet the Parents (2000)': ('FeardotCom (2002)', 0.3625043226721463),
 'Independence Day (1996)': ('Erik the Viking (1989)', 0.2695325366400816),
 'Who Framed Roger Rabbit (1988)': ('FeardotCom (2002)', 0.343222159157451),
 'As Good as it Gets (1997)': ('MacArthur (1977)', 0.5223511569192691),
 'Butch Cassidy and the Sundance Kid (1969)': ('Along Came a Spider (2002)',
  0.6080739505347184),
 "You're Next (2011)": ("Miller's Crossing (1990)", 0.4384356044459746),
 'The Final Conflict (1981)': ('The Lookout (2007)', 0.7001881161214467),
 'City of God (2002)': ('MacArthur (1977)', 0.5491603781881604),
 'Star Wars: Episode VII - The Force Awakens (2015)': ('Star Wars: Episode VI -
The Return of the Jedi (1983)',
  0.480667581702678),
 'The Transporter (2002)': ('I.Q. (1994)', 0.3933040084427799),
 'Cast Away (2000)': ('The Lookout (2007)', 0.3952430145330451),
 'Bad Boys 2 (2003)': ('Bad Boys (1995)', 0.48578461879700974),
 'The Babadook (2014)': ('The Lookout (2007)', 0.23173445972670914),
 'Saw (2004)': ('Halloween (1978)', 0.21212652812060406),
 'Star Wars: Episode VI - The Return of the Jedi (1983)': ('Star Wars: Episode V
- The Empire Strikes Back (1980)',
  0.500443762521317),
 'Scary Movie (2000)': ('Hollow Man (2000)', 0.21035169146899402),
 'E.T. The Extra-Terrestrial (1982)': ('The Final Conflict (1981)',
  0.2732911704269627),
```

```
 'American History X (1998)': ('Patton (1970)', 0.47231197991523044),
 'FeardotCom (2002)': ('The Final Conflict (1981)', 0.6910371117905774),
 'Halloween (1978)': ('FeardotCom (2002)', 0.369865233230816),
 'Along Came a Spider (2002)': ('The Lookout (2007)', 0.632651586784865),
 'The Mist (2007)': ('The Lookout (2007)', 0.3673933313865456),
 'Aladdin (1992)': ('Beauty and the Beauty (1991)', 0.2517915979848775),
 'Pirates of the Caribbean: The Curse of the Black Pearl (2003)': ("Pirates of
the Caribbean: At World's End (2007)",
  0.39487603864032406),
 'Men in Black (1997)': ('Men in Black II (2002)', 0.4313587529374374),
 'Eternal Sunshine of the Spotless Mind (2004)': ('Analyze That (2002)',
  0.32668553511792064),
 "Ocean's Eleven (2001)": ('Analyze That (2002)', 0.33678101795516846),
 'Men in Black II (2002)': ('Men in Black (1997)', 0.43135875293743753),
 'Cocktail (1988)': ('The Lookout (2007)', 0.6268457026256169),
 'The Wolf of Wall Street (2013)': ('As Good as it Gets (1997)',
  0.17154457601658468),
 'The Game (1997)': ('Erik the Viking (1989)', 0.5906790680750406),
 'Red Sonja (1985)': ('The Lookout (2007)', 0.6408947751704267),
 "One Flew Over the Cuckoo's Nest (1975)": ("Miller's Crossing (1990)",
  0.3638334713662861),
 'Freddy Got Fingered (2001)': ('Heavy Traffic (1973)', 0.5926799301108321),
 'The Village (2004)': ('The Lookout (2007)', 0.5834305873382578),
 "Miller's Crossing (1990)": ('The Lookout (2007)', 0.6567814482605832),
 "My Best Friend's Wedding (1997)": ('Erik the Viking (1989)',
  0.4521158644791258),
 'Iron Man 3 (2013)': ('Captain America: Civil War (2016)',
  0.2553244278306489),
 'Big Daddy (1999)': ('Crimson Tide (1995)', 0.40020471823187165),
 'Suspiria (1977)': ("Miller's Crossing (1990)", 0.6110562698338586),
 'The Passenger (1975)': ('Broken Arrow (1996)', 0.5549019048518792),
 'The Station Agent (2003)': ('Ed Wood (1994)', 0.6713849837329366),
 'Jurassic Park (1993)': ('The Lost World: Jurassic Park (1997)',
  0.2968088595606392),
 'Captain America: Civil War (2016)': ('The Avengers (2012)',
  0.2722230674252951),
 'A Perfect Murder (1998)': ('Sexy Beast (2000)', 0.6101281082791372),
 'Analyze That (2002)': ('The Lookout (2007)', 0.6334262606152418),
 'Braveheart (1995)': ('Blow (2001)', 0.39661256410208945),
 'Inception (2010)': ('The Game (1997)', 0.1751119967562147),
 'Groundhog Day (1993)': ('FeardotCom (2002)', 0.36787705611106325),
 'The Lookout (2007)': ('Patton (1970)', 0.7135542589926912),
 '21 Grams (2003)': ('The Lookout (2007)', 0.6609185178922179),
 'Gladiator (2000)': ('FeardotCom (2002)', 0.27815132044092306),
 'Midnight Cowboy (1969)': ('Patton (1970)', 0.5817704576293936),
 'Fatal Attraction (1987)': ('The Bandit (1996)', 0.49342329734890955),
 'House of Sand and Fog (2003)': ('The Straight Story (1999)',
```

```
 0.5658946686388151),
 'On Golden Pond (1981)': ('Red Sonja (1985)', 0.5888934170016711),
 'The Mummy (1999)': ('The Mummy Returns (2001)', 0.44366213116320563),
 'The Descent (2005)': ("Miller's Crossing (1990)", 0.5087548377843522),
 'Runaway Bride (1999)': ('Escape from LA (1996)', 0.485289709921102),
 'Harry Potter and the Goblet of Fire (2005)': ('Harry Potter and the Chamber of
Secrets (2002)',
  0.5473830364255856),
 'Gods and Generals (2003)': ('Sexy Beast (2000)', 0.5702224232432986),
 'My Father and My Son (2005)': ('The Lookout (2007)', 0.5210596299036837),
 'X-Men 2 (2003)': ('X-Men (2000)', 0.453072518008104330),
 'The Usual Suspects (1995)': ("Miller's Crossing (1990)",
  0.45875318301412005),
 'The Mask (1994)': ('The Lookout (2007)', 0.24130419456655916),
 'Jaws (1975)': ('JFK (1991)', 0.22379967035544512),
 'Harry Potter and the Chamber of Secrets (2002)': ("Harry Potter and the
Sorcerer's Stone (2001)",
  0.5639080588803831),
 'Patton (1970)': ('The Lookout (2007)', 0.7135542589926913),
 'Anaconda (1997)': ('The Lookout (2007)', 0.3468630914638312),
 'Twister (1996)': ('Sexy Beast (2000)', 0.41886873317188167),
 'MacArthur (1977)': ("Miller's Crossing (1990)", 0.6244019043081227),
 "Look Who's Talking (1989)": ('Ran (1985)', 0.4998956669137101)}
```

[11]:
```
average_cod_list = list(average_cod_values.values())

# Plotting the distribution of average COD values
plt.hist(average_cod_list, bins=15)
plt.title('Distribution of Average COD Values')
plt.xlabel('Average COD')
plt.ylabel('Frequency')
plt.show()
```



[12]:
```python
# Convert the results to a DataFrame for easier manipulation
results_df = pd.DataFrame({
    'Movie': best_predictors.keys(),
    'Best Predictor': [best[0] for best in best_predictors.values()],
    'COD': [best[1] for best in best_predictors.values()],
    'Average COD': average_cod_values.values()
})

# Sort the DataFrame by Average COD
sorted_df = results_df.sort_values(by='Average COD', ascending=False)

# Select the top 10 and bottom 10 movies
top_10 = sorted_df.head(10)
bottom_10 = sorted_df.tail(10)

predictmovies20 = pd.concat([top_10, bottom_10], keys=['10 Easiest', '10␣
↪Hardest']).drop(columns=['Average COD'])
justmovies = pd.concat([top_10, bottom_10])
```

[13]:
```python
predictmovies20
```

[13]:

		Movie
10 Easiest	116	Escape from LA (1996)
	109	Sexy Beast (2000)
	377	The Lookout (2007)
	203	Erik the Viking (1989)
	298	Crimson Tide (1995)
	240	The Bandit (1996)
	395	Patton (1970)
	287	The Straight Story (1999)
	363	Miller's Crossing (1990)
	309	Heavy Traffic (1973)
10 Hardest	87	Shrek (2001)
	75	Pirates of the Caribbean: Dead Man's Chest (2006)
	55	Clueless (1995)
	186	The Avengers (2012)
	57	Shrek 2 (2004)
	190	The Cabin in the Woods (2012)
	9	Black Swan (2010)
	95	Interstellar (2014)
	84	The Conjuring (2013)
	80	Avatar (2009)

		Best Predictor	COD
10 Easiest	116	Sexy Beast (2000)	0.649610
	109	The Silencers (1966)	0.659436
	377	Patton (1970)	0.713554
	203	I.Q. (1994)	0.731507
	298	The Straight Story (1999)	0.678454
	240	Best Laid Plans (1999)	0.711222
	395	The Lookout (2007)	0.713554
	287	Congo (1995)	0.700569
	363	The Lookout (2007)	0.656781
	309	Ran (1985)	0.692734
10 Hardest	87	Shrek 2 (2004)	0.451027
	75	Pirates of the Caribbean: At World's End (2007)	0.367212
	55	Escape from LA (1996)	0.141426
	186	Captain America: Civil War (2016)	0.272223
	57	Shrek (2001)	0.451027
	190	The Evil Dead (1981)	0.143887
	9	Sorority Boys (2002)	0.117080
	95	Torque (2004)	0.111343
	84	The Exorcist (1973)	0.198474
	80	Bad Boys (1995)	0.079485

0.2 Q2

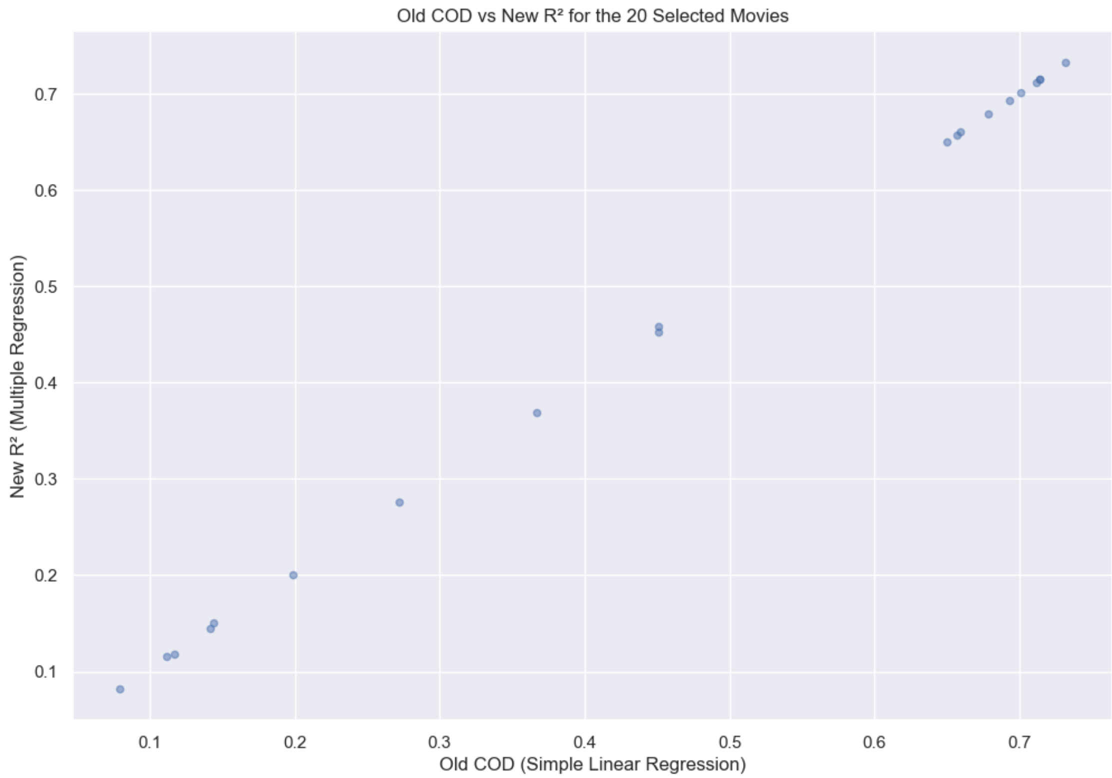
For the 10 movies that are best and least well predicted from the ratings of a single other movie (so 20 in total), build multiple regression models that include gender identity (column 475), sibship status (column 476) and social viewing preferences (column 477) as additional predictors (in addition to the best predicting movie from question 1). Comment on how R^2 has changed relative to the answers in question 1. Please include a figure with a scatterplot where the old COD (for the simple linear regression models from the previous question) is on the x-axis and the new R^2 (for the new multiple regression models) is on the y-axis.

[14]:
```
data0.iloc[:, 474:477].isnull().sum(axis = 0)
```

[14]:
```
Gender identity (1 = female; 2 = male; 3 = self-described)              24
Are you an only child? (1: Yes; 0: No; -1: Did not respond)              0
Movies are best enjoyed alone (1: Yes; 0: No; -1: Did not respond)       0
dtype: int64
```

[15]:
```
# Placeholder lists for the old COD and the new R² values
old_cod = predictmovies20['COD'].tolist()
new_r2 = []

# Iterate through the DataFrame rows to build the multiple regression models
for movie in justmovies['Movie']:
    complete_ratings = data[movie].values.reshape(-1,1)
    best_predictor = justmovies.loc[justmovies['Movie'] == movie, 'Best␣
↪Predictor']
    # Extract the ratings for the best predictor movie
    best_predictor_ratings = data[best_predictor].values.reshape(-1,1)
    gender_identity = data0.iloc[:, 474].fillna(-1).values.reshape(-1,1)
    sibship_status = data.iloc[:, 475].values.reshape(-1,1)
    social_viewing_preferences = data.iloc[:, 476].values.reshape(-1,1)

    X = np.concatenate((best_predictor_ratings, gender_identity,␣
↪sibship_status, social_viewing_preferences), axis=1)
    y = complete_ratings

    # Fit the multiple regression model
    model = LinearRegression().fit(X, y)

    # Predict the target movie ratings
    y_hat = model.predict(X)
    r2 = r2_score(y, y_hat)

    # Calculate the new R² value and add it to the list
    new_r2.append(r2)
```

[16]:
```
plt.figure(figsize=(12, 8))
plt.scatter(old_cod, new_r2, alpha = 0.5, s=20)
plt.xlabel('Old COD (Simple Linear Regression)')
plt.ylabel('New R² (Multiple Regression)')
plt.title('Old COD vs New R² for the 20 Selected Movies')
plt.grid(True)
plt.show()
```



[17]:
```
pd.DataFrame({'old r2': old_cod, 'new r2': new_r2})
```

[17]:

	old r2	new r2
0	0.649610	0.650248
1	0.659436	0.661056
2	0.713554	0.715080
3	0.731507	0.732332
4	0.678454	0.678762
5	0.711222	0.711735
6	0.713554	0.714680
7	0.700569	0.700932
8	0.656781	0.657228
9	0.692734	0.692935
10	0.451027	0.452851

```
11   0.367212  0.368486
12   0.141426  0.144948
13   0.272223  0.275614
14   0.451027  0.458518
15   0.143887  0.150299
16   0.117080  0.118175
17   0.111343  0.115860
18   0.198474  0.200380
19   0.079485  0.081787
```

0.3 Q3

3)Pick 30 movies in the middle of the COD range, as identified by question 1 (that were not used in question 2). Now build a regularized regression model with the ratings from 10 other movies (picked randomly, or deliberately by you) as an input. Please use ridge regression, and make sure to do suitable hyperparameter tuning. Also make sure to report the RMSE for each of these 30 movies in a table, after doing an 80/20 train/test split. Comment on the hyperparameters you use and betas you find by doing so

[18]:
```python
# Picked the middle 30 movies
middle_range_movies = sorted_df[200-15:200+15]

# Extract just the movie names from the middle range
middle_range_movie_names = middle_range_movies['Movie'].tolist()

# Select 10 other movies to use as predictors
predictor_movie_names = df.columns.difference(middle_range_movie_names).
 ↪tolist()[:10]
predictor_movie_names
```

[18]:
```
['10 Things I Hate About You (1999)',
 '10000 BC (2008)',
 '13 Going on 30 (2004)',
 '21 Grams (2003)',
 '25th Hour (2002)',
 '28 Days Later (2002)',
 '3000 Miles to Graceland (2001)',
 '8 Mile (2002)',
 'A Beautiful Mind (2001)',
 "A Bug's Life (1998)"]
```

[19]:
```python
# Define the Polynomial Ridge Regression function
def PolynomialRidgeRegression(degree=2, **kwargs):
    return make_pipeline(PolynomialFeatures(degree), Ridge(**kwargs))

# Define a range of alphas for Ridge
alphas = np.logspace(-2, 1, 100)
```

```python
# Create the grid
param_grid = {'ridge__alpha': alphas}

# Initialize a DataFrame to store RMSE values
rmse_results = []

# Loop over each of the 30 selected middle-range movies
for movie in middle_range_movie_names:
    # Prepare the feature matrix X and target vector y
    X = df[predictor_movie_names].values
    y = df[movie].values

    # Perform an 80/20 train/test split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

    # Grid Search for Hyperparameter Tuning
    grid_search = GridSearchCV(
        PolynomialRidgeRegression(),
        param_grid=param_grid,
        scoring='neg_mean_squared_error',
        cv=5
    )
    grid_search.fit(X_train, y_train)

    # Best model
    best_model = grid_search.best_estimator_
    y_pred = best_model.predict(X_test)
    rmse = np.sqrt(mean_squared_error(y_test, y_pred))

    # Store the results
    rmse_results.append({
        'Movie': movie,
        'RMSE': rmse,
        'Alpha': grid_search.best_params_['ridge__alpha'],
        'Weights': best_model.named_steps['ridge'].coef_
    })

# Create a DataFrame with the results
rmse_df = pd.DataFrame(rmse_results)

# Sort the DataFrame by RMSE and reset index
rmse_df_sorted = rmse_df.sort_values(by='RMSE').reset_index(drop=True)

# Display the DataFrame
rmse_df_sorted
```

[19]:

```
                                   Movie      RMSE      Alpha  \
0                      Crossroads (2002)  0.286837  10.000000
1                  The Green Mile (1999)  0.294768  10.000000
2                     You're Next (2011)  0.327881  10.000000
3                     Man on Fire (2004)  0.334144  10.000000
4                          Aliens (1986)  0.341551  10.000000
5             Gone in Sixty Seconds (2000)  0.371411  10.000000
6                       Big Daddy (1999)  0.373071  10.000000
7                    Child's Play (1988)  0.381841  10.000000
8               Full Metal Jacket (1987)  0.385251  10.000000
9                       The Thing (1982)  0.386915  10.000000
10                 Knight and Day (2010)  0.395318  10.000000
11                     The Others (2001)  0.395662  10.000000
12                     12 Monkeys (1995)  0.398316  10.000000
13            Blues Brothers 2000 (1998)  0.399286  10.000000
14        The Poseidon Adventure (1972)  0.402020  10.000000
15                     Braveheart (1995)  0.406394  10.000000
16                      Halloween (1978)  0.409804  10.000000
17                       The Mist (2007)  0.415698  10.000000
18                The Transporter (2002)  0.423934   8.697490
19                  Baby Geniuses (1999)  0.425127  10.000000
20               The Intouchables (2011)  0.442557  10.000000
21                          Honey (2003)  0.444671  10.000000
22                       Bad Boys (1995)  0.446550   7.564633
23 One Flew Over the Cuckoo's Nest (1975)  0.446647  10.000000
24        Angels in the Outfield (1994)  0.450396  10.000000
25                     Armageddon (1998)  0.458000  10.000000
26                     Bad Boys 2 (2003)  0.463821  10.000000
27                        Memento (2000)  0.482213  10.000000
28                          Rocky (1976)  0.527067  10.000000
29                The Truman Show (1998)  0.552330  10.000000

                                              Weights
0   [0.0, -0.0030763304163796113, 0.15701682415613…
1   [0.0, 0.11262143143371749, 0.10726147364757009…
2   [0.0, 0.11362057765632744, 0.21998654253825742…
3   [0.0, 0.05076837328717706, 0.01328292524463463…
4   [0.0, 0.10648715387751483, 0.17729552307535168…
5   [0.0, -0.03757677271748001, -0.104401852496886…
6   [0.0, 0.009739445675861744, 0.0758916134711343…
7   [0.0, 0.1580012646555516, 0.08553461977510944,…
8   [0.0, 0.11982174702632291, 0.06052447830305312…
9   [0.0, 0.08100736122940354, -0.0613085350609983…
10  [0.0, 0.14573642260609368, 0.22169879291994146…
11  [0.0, 0.027310381830690417, -0.051016442696634…
12  [0.0, -0.08324338358753722, -0.010522358470420…
13  [0.0, 0.07965489573263326, -0.0216137949313067…
```

```
14   [0.0, -0.04309244968495371, 0.0551652603811859…
15   [0.0, 0.1422273184959001, 0.02871717507188733,…
16   [0.0, 0.07605065569700549, 0.10201295214792722…
17   [0.0, -0.009894505096142717, 0.184210291793391…
18   [0.0, 0.054622027120014835, 0.1956952163009349…
19   [0.0, 0.06341253670736055, 0.1579285522900093,…
20   [0.0, 0.11912197840748644, 0.18428061417176117…
21   [0.0, -0.023065878123082938, 0.092234789614828…
22   [0.0, 0.1300887350086438, 0.07547637890916023,…
23   [0.0, 0.11398679807241358, 0.05683462572943145…
24   [0.0, 0.1628216911637786, 0.16090833367438834,…
25   [0.0, 0.08498743949723012, 0.11450061245013378…
26   [0.0, 0.06053487575992637, 0.03913442570097730…
27   [0.0, 0.18840584120338652, 0.0792530147485297,…
28   [0.0, 0.12909622943333268, 0.05226414608489949…
29   [0.0, 0.10100013680204745, 0.14709021719471202…
```

0.4 Q4

4)Repeat question 3) with LASSO regression. Again, make sure to comment on the hyperparameters you use and betas you find by doing so.

[20]:
```python
# Define the Polynomial Lasso Regression function
def PolynomialLassoRegression(degree=2, **kwargs):
    return make_pipeline(PolynomialFeatures(degree), Lasso(**kwargs))

# Define a range of alphas for Lasso
alphas = np.logspace(-10, 2, 100)

# Create the grid
param_grid = {'lasso__alpha': alphas}

# Initialize a DataFrame to store RMSE values
rmse_results = []

# Loop over each of the 30 selected middle-range movies
for movie in middle_range_movie_names:
    # Prepare the feature matrix X and target vector y
    X = df[predictor_movie_names].values
    y = df[movie].values

    # Perform an 80/20 train/test split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=42)

    # Grid Search for Hyperparameter Tuning
    lasso_grid = GridSearchCV(
```

```python
        PolynomialLassoRegression(),
        param_grid=param_grid,
        scoring='neg_mean_squared_error',
        cv=5
    )
    lasso_grid.fit(X_train, y_train)

    # Best model
    best_model = lasso_grid.best_estimator_
    y_pred = best_model.predict(X_test)
    rmse = np.sqrt(mean_squared_error(y_test, y_pred))

    # Store the results
    rmse_results.append({
        'Movie': movie,
        'RMSE': rmse,
        'Alpha': lasso_grid.best_params_['lasso__alpha'],
        'Weights': best_model.named_steps['lasso'].coef_
    })


# Create a DataFrame with the results
rmse_df = pd.DataFrame(rmse_results)
rmse_df_sort = rmse_df.sort_values(by = 'RMSE').reset_index(drop = True)
# Display the DataFrame
rmse_df_sort
```

[20]:

```
                              Movie      RMSE     Alpha  \
0              The Green Mile (1999)  0.302870  0.013219
1                 Man on Fire (2004)  0.303695  0.070548
2                  Crossroads (2002)  0.314025  0.030539
3         Gone in Sixty Seconds (2000)  0.317967  0.030539
4           Blues Brothers 2000 (1998)  0.338477  0.010000
5                      Aliens (1986)  0.339159  0.005722
6                  You're Next (2011)  0.357770  0.017475
7                   Big Daddy (1999)  0.358436  0.023101
8                    The Mist (2007)  0.371687  0.013219
9                Child's Play (1988)  0.374458  0.023101
10                  The Thing (1982)  0.386404  0.040370
11                   Bad Boys 2 (2003)  0.390358  0.030539
12      The Poseidon Adventure (1972)  0.392469  0.002477
13                 Braveheart (1995)  0.395946  0.017475
14                 12 Monkeys (1995)  0.396022  0.007565
15                      Honey (2003)  0.400907  0.030539
16             Knight and Day (2010)  0.403956  0.023101
17          Full Metal Jacket (1987)  0.404736  0.017475
18      Angels in the Outfield (1994)  0.409301  0.030539
```

19	Armageddon (1998)	0.409808	0.040370
20	The Transporter (2002)	0.412349	0.023101
21	Halloween (1978)	0.419810	0.023101
22	One Flew Over the Cuckoo's Nest (1975)	0.422321	0.017475
23	Baby Geniuses (1999)	0.427427	0.040370
24	The Others (2001)	0.432841	0.013219
25	The Intouchables (2011)	0.455198	0.040370
26	Memento (2000)	0.455213	0.023101
27	Bad Boys (1995)	0.464541	0.000811
28	The Truman Show (1998)	0.513628	0.023101
29	Rocky (1976)	0.526520	0.030539

	Weights
0	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
1	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
2	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
3	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
4	[0.0, 0.0, -0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,…
5	[0.0, 0.0, 0.19689139883801599, 0.014496020519…
6	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
7	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -0.0,…
8	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
9	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
10	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
11	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
12	[0.0, -0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.1487369…
13	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
14	[0.0, -0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -0.0…
15	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -0.0,…
16	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
17	[0.0, 0.0, -0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,…
18	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
19	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
20	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
21	[0.0, 0.0, 0.0, -0.0, 0.0, 0.0, 0.0, 0.0, 0.0,…
22	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
23	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
24	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -0.0,…
25	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
26	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
27	[0.0, 0.08421061560684191, 0.05348938714858694…
28	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
29	[0.0, 0.0, 0.0, -0.0, 0.0, 0.0, 0.0, 0.0, 0.0,…

0.5 Q5

5) Compute the average movie enjoyment for each user (using only real, non-imputed data). Use these averages as the predictor variable X in a logistic regression model. Sort the movies order of increasing rating (also using only real, non-imputed data). Now pick the 4 movies in the middle of the score range as your target movie. For each of them, do a media split (now using the imputed data) of ratings to code movies above the median rating with the Y label 1 (= enjoyed) and movies below the median with the label 0 (= not enjoyed). For each of these movies, build a logistic regression model (using X to predict Y), show figures with the outcomes and report the betas as well as the AUC values. Comment on the quality of your models. Make sure to use cross-validation methods to avoid overfitting.

[21]:
```
# Compute the average for each user using real, non-imputed data
avg_enjoyment = data0.iloc[:, :400].mean(axis=1)
X = avg_enjoyment.fillna(avg_enjoyment.mean()).values.reshape(-1, 1)  #␣
 ↪Predictor variable X
X
```

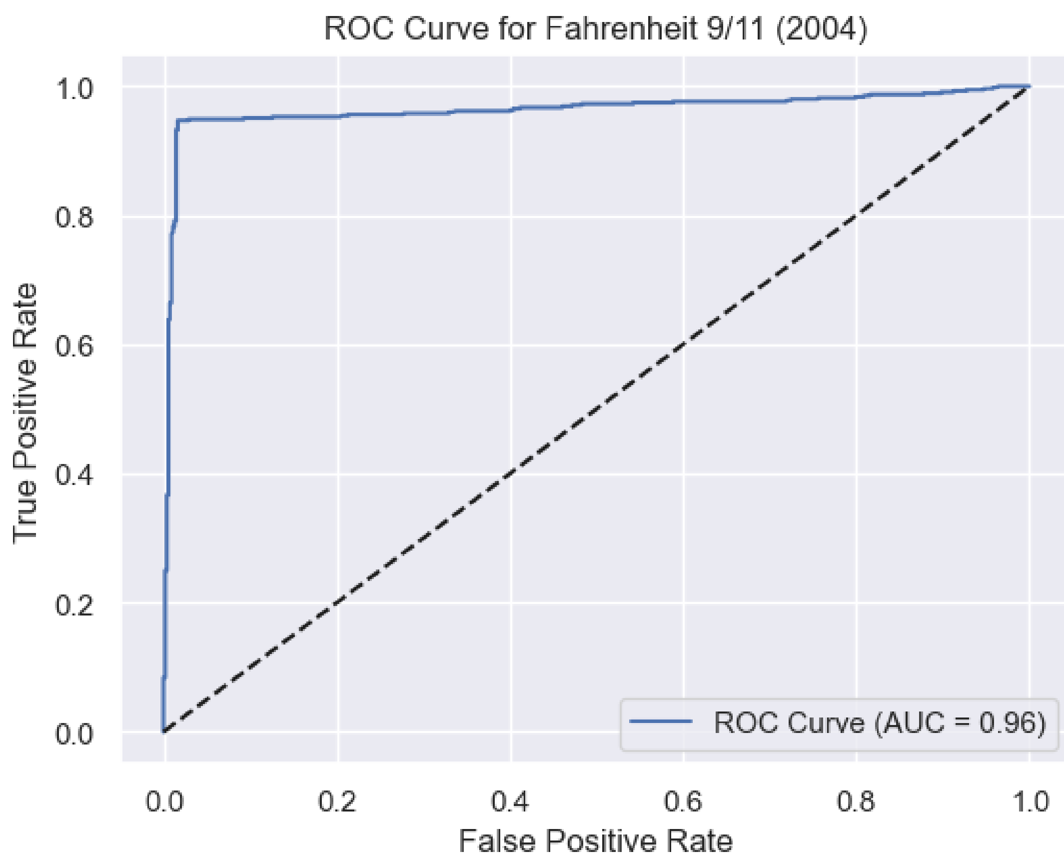
[21]:
```
array([[2.74285714],
       [2.72727273],
       [3.31481481],
       …,
       [3.13253012],
       [3.390625  ],
       [2.87387387]])
```

[22]:
```
# Sort movies based on their average rating using non-imputed data
movie_avg_ratings = data0.iloc[:, :400].mean().sort_values()
# Pick 4 middle movies
middle_movies = movie_avg_ratings.iloc[198:202].index.to_list()
middle_movies
```

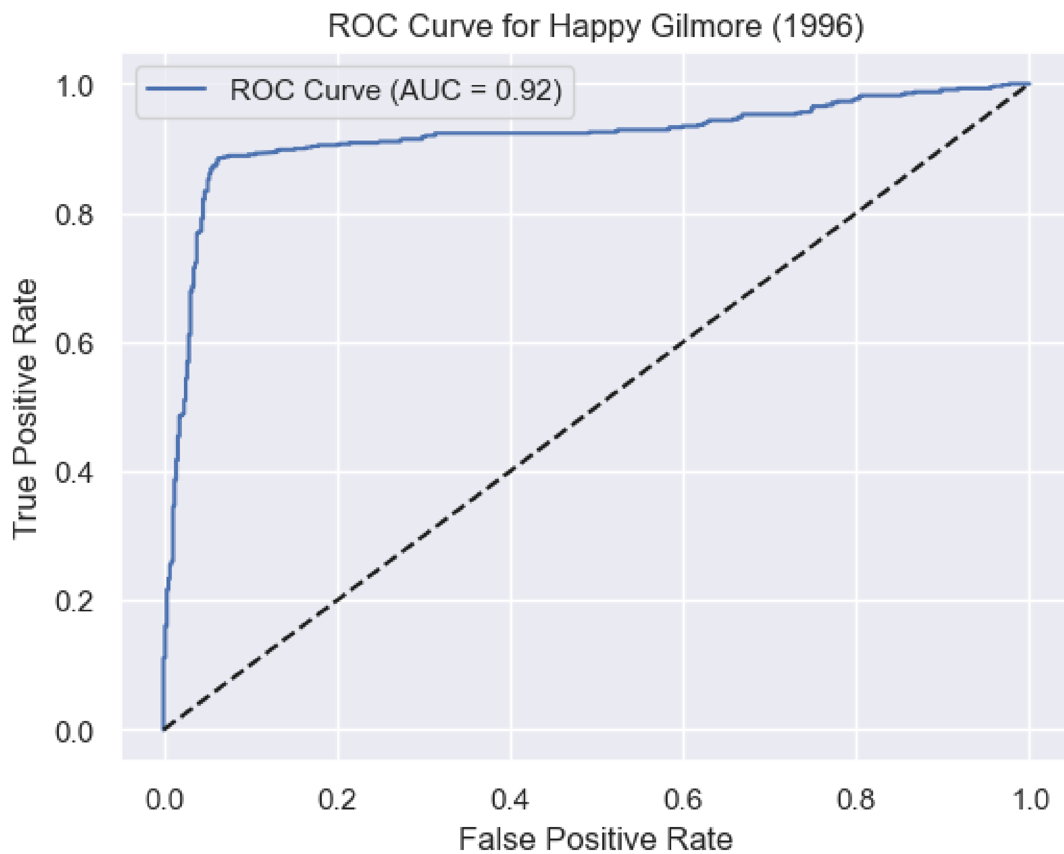
[22]:
```
['Fahrenheit 9/11 (2004)',
 'Happy Gilmore (1996)',
 'Diamonds are Forever (1971)',
 'Scream (1996)']
```

[23]:
```
# Loop over each of the 4 middle movies
for movie in middle_movies:
    avg_auc = []
    model_coef = []

    Y = df[movie].values
    median_rating = np.median(Y)
    Y_binary = (Y > median_rating).astype(int)  # Binary classification
    model = LogisticRegression()
    auc_scores = cross_val_score(model, X, Y_binary, cv=5, scoring='roc_auc')
```

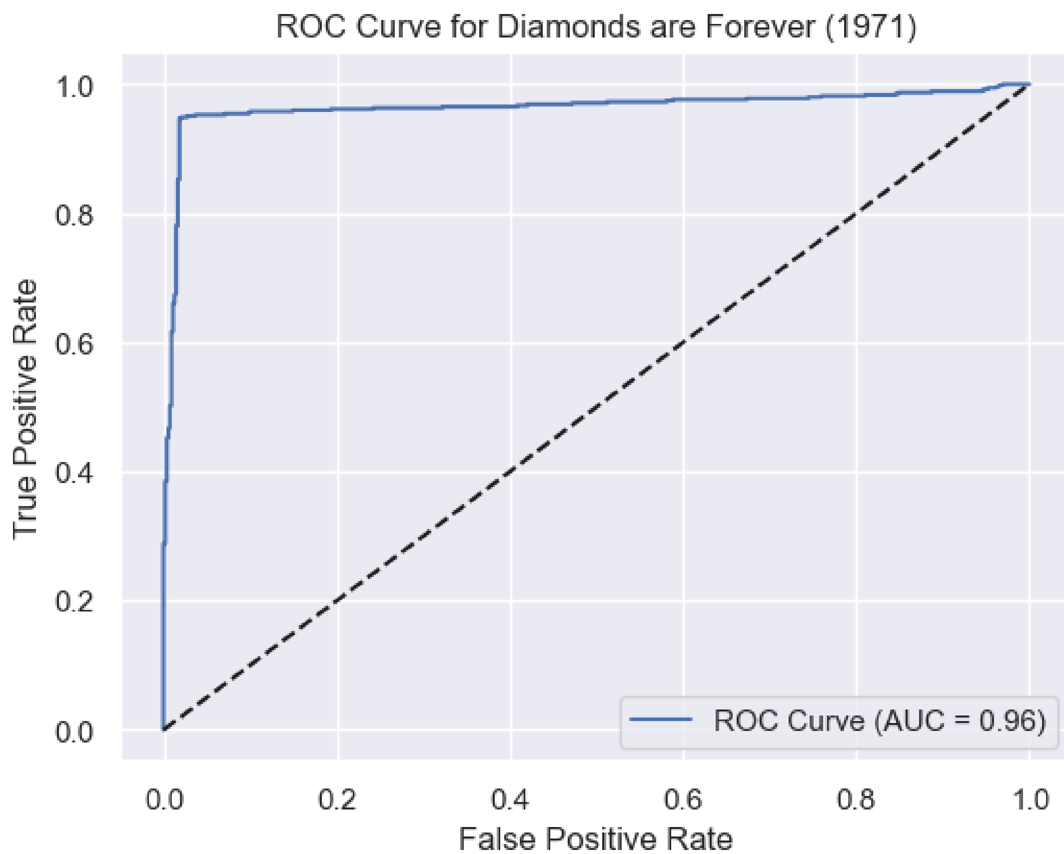
```python
# Fit the model on the entire dataset (for plotting ROC curve)
model.fit(X, Y_binary)

# ROC Curve
fpr, tpr, thresholds = roc_curve(Y_binary, model.predict_proba(X)[:, 1])
plt.figure()
plt.plot(fpr, tpr, label=f'ROC Curve (AUC = {np.mean(auc_scores):.2f})')
plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title(f'ROC Curve for {movie}')
plt.legend(loc='best')
plt.show()

# Print model information
print(f"Movie: {movie}")
print(f"Average AUC: {np.mean(auc_scores)}")
print(f"Model Coefficients: {model.coef_[0]}")
print("\n")
```



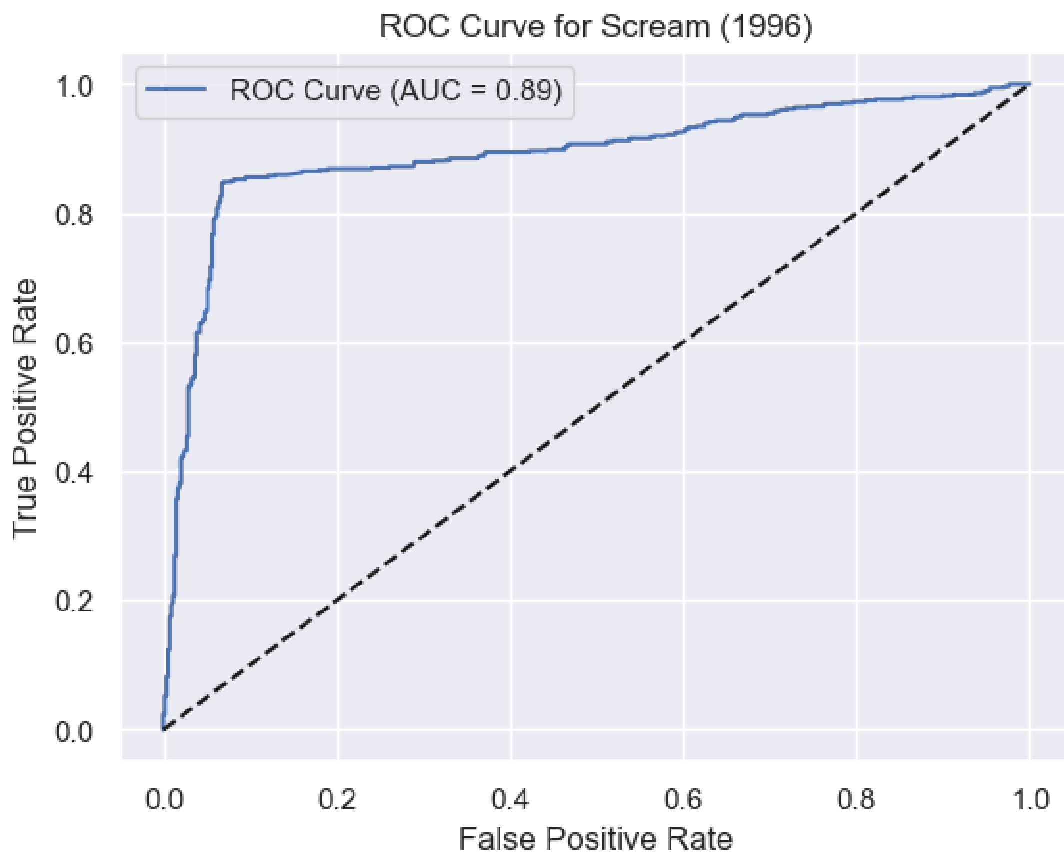
```
Movie: Fahrenheit 9/11 (2004)
Average AUC: 0.9636157403897186
Model Coefficients: [7.39636383]
```




```
Movie: Happy Gilmore (1996)
Average AUC: 0.9169490484494656
Model Coefficients: [5.201532]
```




```
Movie: Diamonds are Forever (1971)
Average AUC: 0.9647942982788689
Model Coefficients: [7.32554404]
```




```
Movie: Scream (1996)
Average AUC: 0.8919377511562667
Model Coefficients: [4.41567957]
```

[24]:
```
avg_auc = [0.9636157403897186, 0.9169490484494656, 0.9647942982788689, 0.
 ↪8919377511562667]
model_coef = [7.39636383, 5.201532, 7.32554404, 4.41567957]
pd.DataFrame({'Movie': middle_movies,
              'AUC': avg_auc,
              'Model_Coef (Beta)': model_coef})
```

[24]:

	Movie	AUC	Model_Coef (Beta)
0	Fahrenheit 9/11 (2004)	0.963616	7.396364
1	Happy Gilmore (1996)	0.916949	5.201532
2	Diamonds are Forever (1971)	0.964794	7.325544
3	Scream (1996)	0.891938	4.415680

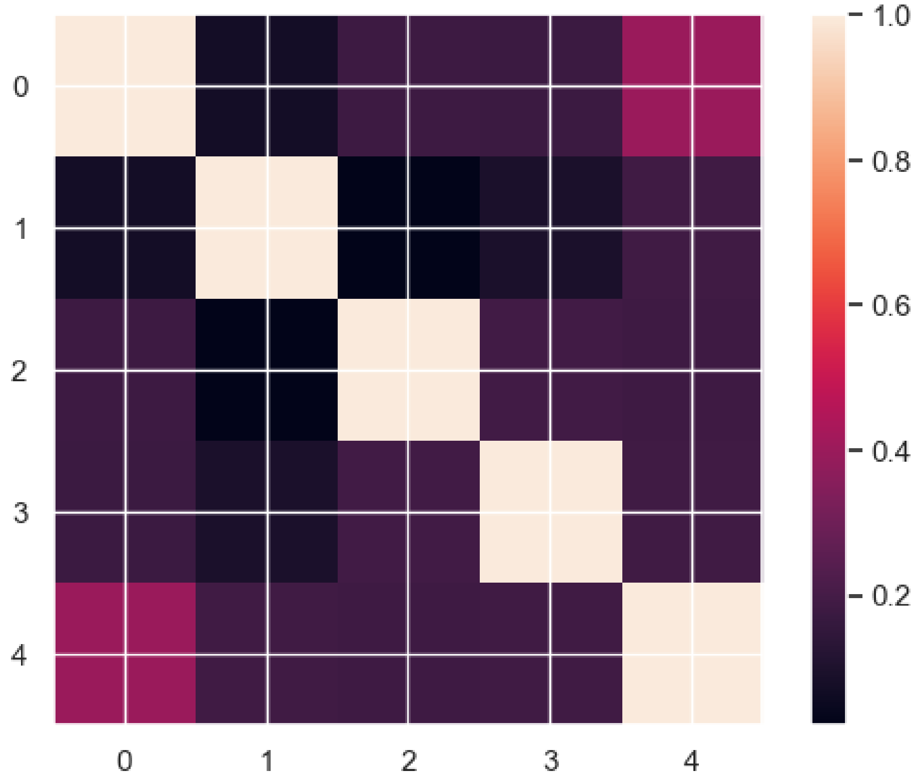
0.6 Extra Credit

Use machine learning methods of your choice to tell us something interesting and true about the movies in this dataset that is not already covered by the questions above [for an additional 5% of the grade score].

[25]:
```
from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor
y_avatar = data0.loc[:,['Avatar (2009)']].fillna(0)
y_titanic = data0.loc[:,['Titanic (1997)']].fillna(0)

y = data0.loc[:,['Avatar (2009)','Titanic (1997)' ]].fillna(0)
movie_name = ['Avatar (2009)','Titanic (1997)' ]
```

[26]:
```
X = data.loc[:, [
    'is outgoing/sociable',
    'Is ingenious/a deep thinker',
    'Is emotionally stable/not easily upset',
    'Makes plans and follows through with them',
    'Has an assertive personality'
]].fillna(0)

personality = [
    'is outgoing/sociable',
    'Is ingenious/a deep thinker',
    'Is emotionally stable/not easily upset',
    'Makes plans and follows through with them',
    'Has an assertive personality'
]
```

[27]:
```
r = np.corrcoef(X,rowvar=False)
plt.imshow(r)
plt.colorbar()
plt.show()
```



[28]:
```
Xa_train, Xa_test, ya_train, ya_test = train_test_split(X, y_avatar,␣
 ↪test_size=0.2, random_state = 42)

Xt_train, Xt_test, yt_train, yt_test = train_test_split(X, y_titanic,␣
 ↪test_size=0.2, random_state = 42)
```

[29]:
```
rf_titanic = RandomForestRegressor(n_estimators=100, random_state=42)
rf_avatar = RandomForestRegressor(n_estimators=100, random_state=42)

# Train the models
rf_titanic.fit(Xt_train, yt_train)
rf_avatar.fit(Xa_train, ya_train)


# Feature importance for Titanic movie
titanic_feature_importance = rf_titanic.feature_importances_

# Feature importance for Avatar movie
avatar_feature_importance = rf_avatar.feature_importances_
```

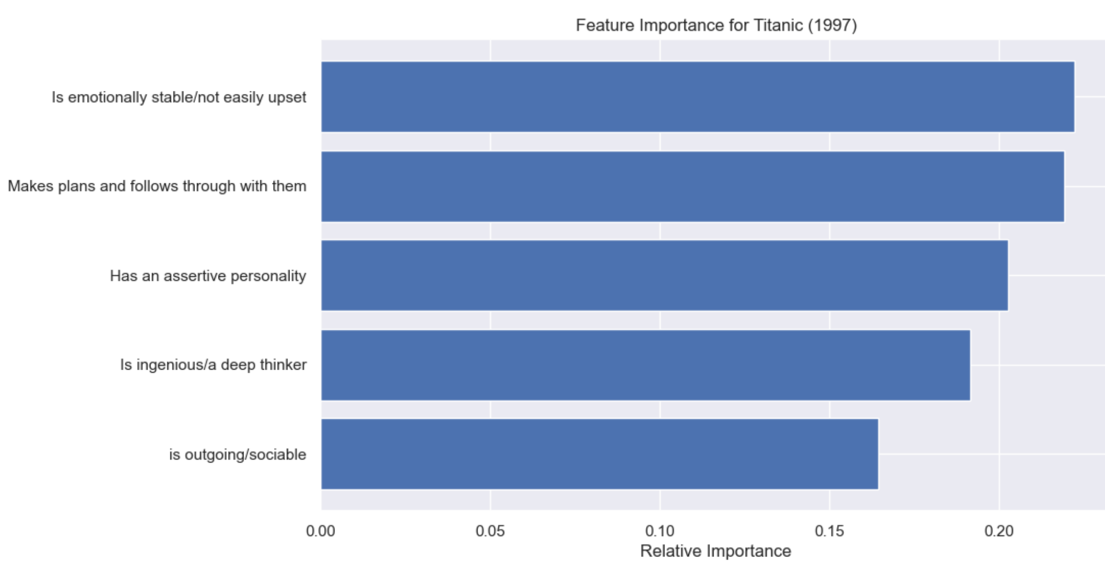
```python
avatar_fi = pd.DataFrame(avatar_feature_importance, index = personality,␣
    ↪columns = ['Avatar (2009)'])
titanic_fi = pd.DataFrame(titanic_feature_importance, index = personality,␣
    ↪columns = ['Titanic (1997)'])
pd.concat([avatar_fi, titanic_fi], axis = 1)
```

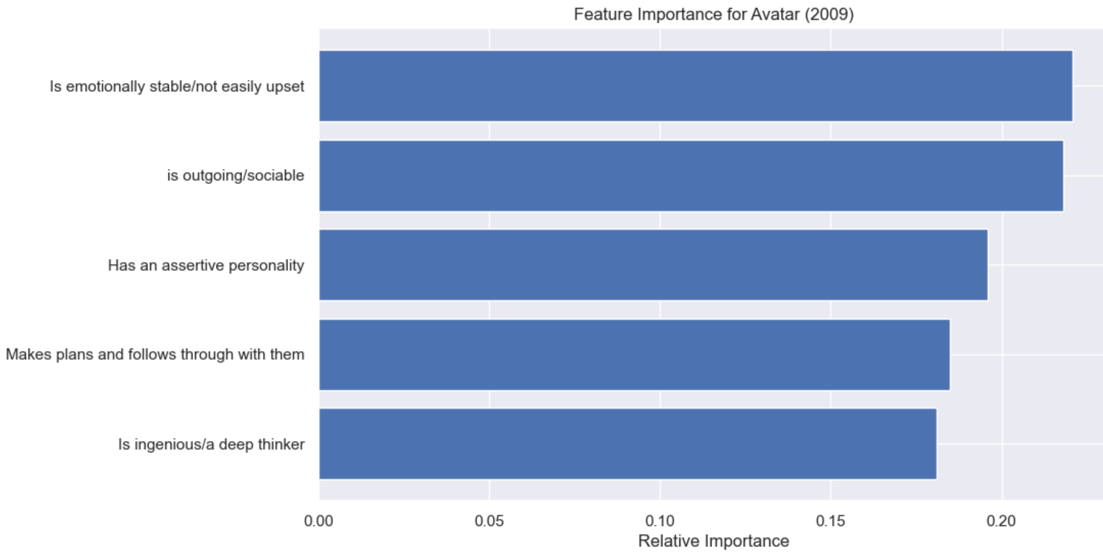
[29]:

	Avatar (2009)	Titanic (1997)
is outgoing/sociable	0.218015	0.164354
Is ingenious/a deep thinker	0.180843	0.191425
Is emotionally stable/not easily upset	0.220690	0.222262
Makes plans and follows through with them	0.184702	0.219169
Has an assertive personality	0.195751	0.202790

[30]:

```python
# Function to plot feature importance
def plot_rf_feature_importance(importances, features, title):
    indices = np.argsort(importances)

    plt.figure(figsize=(10, 6))
    plt.title(title)
    plt.barh(range(len(indices)), importances[indices], color='b',␣
    ↪align='center')
    plt.yticks(range(len(indices)), [features[i] for i in indices])
    plt.xlabel('Relative Importance')
    plt.show()

plot_rf_feature_importance(titanic_feature_importance, personality, 'Feature␣
    ↪Importance for Titanic (1997)')
plot_rf_feature_importance(avatar_feature_importance, personality, 'Feature␣
    ↪Importance for Avatar (2009)')
```




Feature Importance for Avatar (2009)
Is emotionally stable/not easily upset
is outgoing/sociable
Has an assertive personality
Makes plans and follows through with them
Is ingenious/a deep thinker
0.00
0.05
0.10
0.15
0.20
Relative Importance

```
[ ]:
```