Data Analysis Project 2
Haohai Pang, Jiasheng Ni
Dec 3rd, 2023

Note: In regression analysis, particularly when using the ratings of movie A to predict those of movie B, it's common to encounter scenarios where some audience members have only watched movie A, while others have exclusively watched movie B. This discrepancy in viewer overlap can lead to irregular dimensions in the input and output data. To address this, we propose two strategies: 1) Fill Empty Values with Average Movie Ratings **2) Capture Ratings from Mutual Reviewers Only**. We employ option 2 in this exercise as filling missing values with the average rating (option 1) may introduce bias, especially in the context of already closely rated movies within the 0-5 scale.

1. 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 (average) 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).

D: For this problem, we first fit a simple linear regression model parametrized by $y = \beta_0 + \beta_1 x$ where x is the other movie and y is the target movie that we want to predict. Instead of using mean imputation, we build our linear regression model with the users who have rated both movie x and movie y (dropna by rows). For each prediction result, we will get a COD value(R^2) indicating whether movie x is a good predictor for movie y. After that, we average these COD values(each movie has 399 such COD values) and get the movie x with the highest COD value as the best predictor. Then we sort the movie by their average COD values and pick out the top 10 and the bottom 10.

Y: The reason why we don't use a mean imputer is that filling missing values with the average rating (option 1) may introduce bias, especially in the context of already closely rated movies within the 0-5 scale.

F: The average COD Values are shown below, in terms of a single histogram, each histogram contains the statistical summary of the COD value for a single movie across all other 399 movies as the simple predictor.
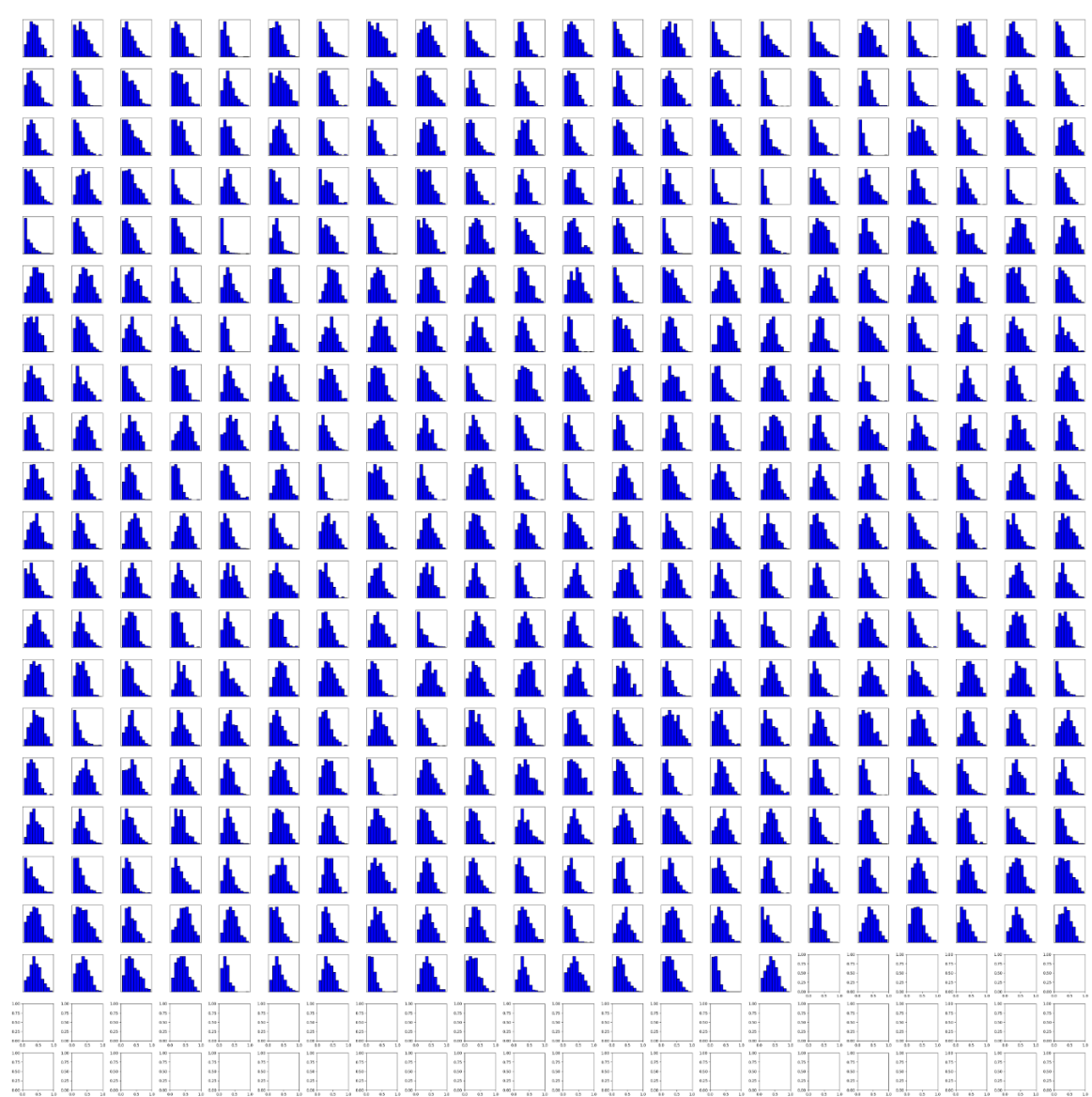
The Top 10 best predicted movies are shown below:

movie	average_COD	best_predictor
Escape from LA (1996)	0.344879	Runaway Bride (1999)
Moonraker (1979)	0.316437	Escape from LA (1996)
Billy Jack (1971)	0.313400	Sexy Beast (2000)
Sexy Beast (2000)	0.312442	Scent of a Woman (1992)

Erik the Viking (1989)	0.305948	I.Q. (1994)
The Lookout (2007)	0.302674	The Village (2004)
Crimson Tide (1995)	0.296956	Cable Guy (1996)
Andaz Apna Apna (1994)	0.294969	The Doom Generation (1995)
FeardotCom (2002)	0.286531	Ran (1985)
Patton (1970)	0.282983	The Village (2004)

The Top 10 least predicted movies are shown below:

movie	average_COD	best_predictor
Avatar (2009)	0.025545	Bad Boys (1995)
The Conjuring (2013)	0.030845	The Exorcist (1973)
Interstellar (2014)	0.039403	Torque (2004)
Spider-Man (2002)	0.047217	Batman (1989)
The Avengers (2012)	0.049374	Captain America: Civil War (2016)
Pirates of the Caribbean: Dead Man's Chest (2006)	0.053017	Pirates of the Caribbean: At World's End (2007)
The Fast and the Furious (2001)	0.056074	Gone in Sixty Seconds (2000)
The Matrix Revolutions (2003)	0.058122	The Matrix Reloaded (2003)
Black Swan (2010)	0.058234	Moonraker (1979)
Shrek 2 (2004)	0.061940	Shrek (2001)

A: We find that such simple linear regression models perform poorly since, for each movie pair, chances are that there are very few users who have watched both movies, which will result in underfitting problem. Moreover, we found that the movie that are least predicted from other movies are actually those very popular movies like Interstellar and the man series. The reason could be that popular movies often have a broad and diverse audience with varying tastes and preferences and this diversity makes it difficult to find a consistent pattern in the ratings. Moreover, hit movies often bring something unique to the table, whether it's an innovative story, groundbreaking special effects, or a standout performance by an actor.

These unique qualities can defy the trends established by previous movies.

2) 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.

D: We incorporate 5 additional features in each simple linear regression model, making it to be multiple linear regression model. Then we draw the scatterplot for each movie(top 10 predicted, bottom 10 predicted), trying to find if there is any correlation between the old average COD value and new average COD value.

Y: As stated before, we need more social-economic features (not just ratings data) to improve the interpretability of our regression model. As stated before, for highly popular or new movies, there may not be enough data from other movies to make accurate predictions, especially if they are pioneering a new genre or style, thus incorporating non-rating features are crucial to the performance of our models.
F: Below are the results that we get:

The result for the top 10 most predicted movies

movie	old_COD	new_COD
Escape from LA (1996)	0.344879	0.732205
Moonraker (1979)	0.316437	0.638274
Billy Jack (1971)	0.313400	0.638461
Sexy Beast (2000)	0.312442	0.677596
Erik the Viking (1989)	0.305948	0.706803
The Lookout (2007)	0.302674	0.696941
Crimson Tide (1995)	0.296956	0.728535
Andaz Apna Apna (1994)	0.294969	0.694848
FeardotCom (2002)	0.286531	0.751242
Patton (1970)	0.282983	0.768436
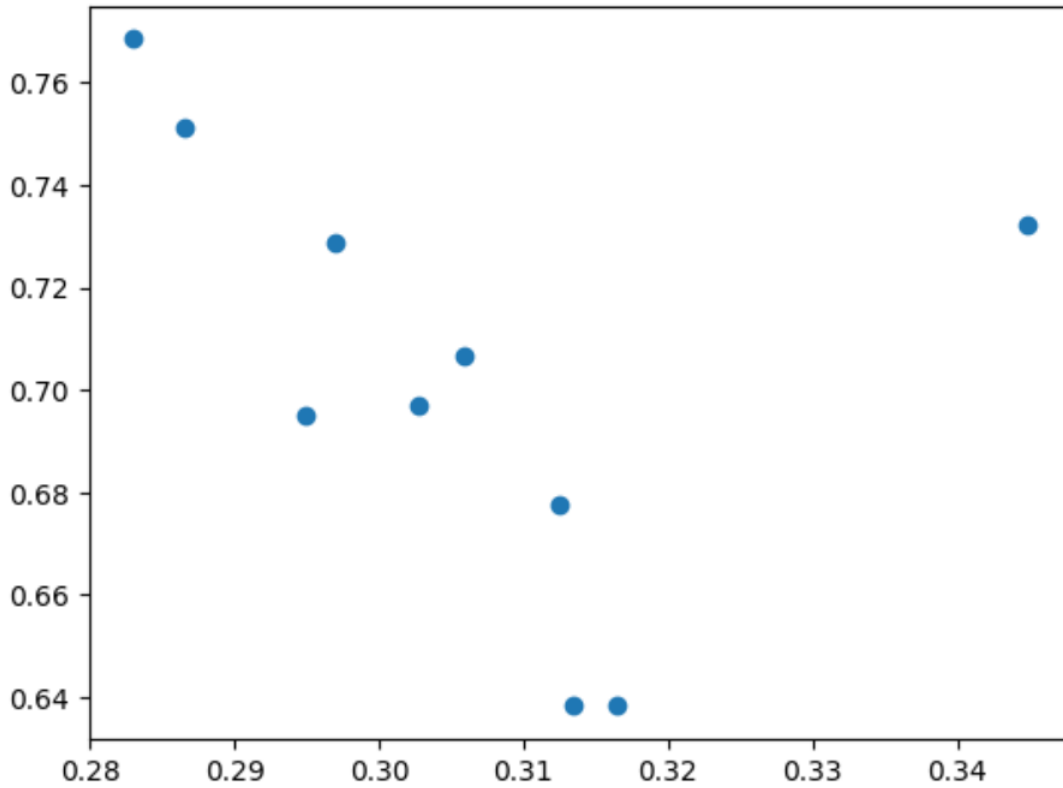
The result for the bottom 10 predicted movies

	movie	old_COD	new_COD
Avatar (2009)		0.025545	0.167469
The Conjuring (2013)		0.030845	0.369116
Interstellar (2014)		0.039403	0.260169
Spider-Man (2002)		0.047217	0.265738
The Avengers (2012)		0.049374	0.433000
Pirates of the Caribbean: Dead Man's Chest (2006)		0.053017	0.525276
The Fast and the Furious (2001)		0.056074	0.267020
The Matrix Revolutions (2003)		0.058122	0.396051

Black Swan (2010)	0.058234	0.237812
Shrek 2 (2004)	0.061940	0.527765
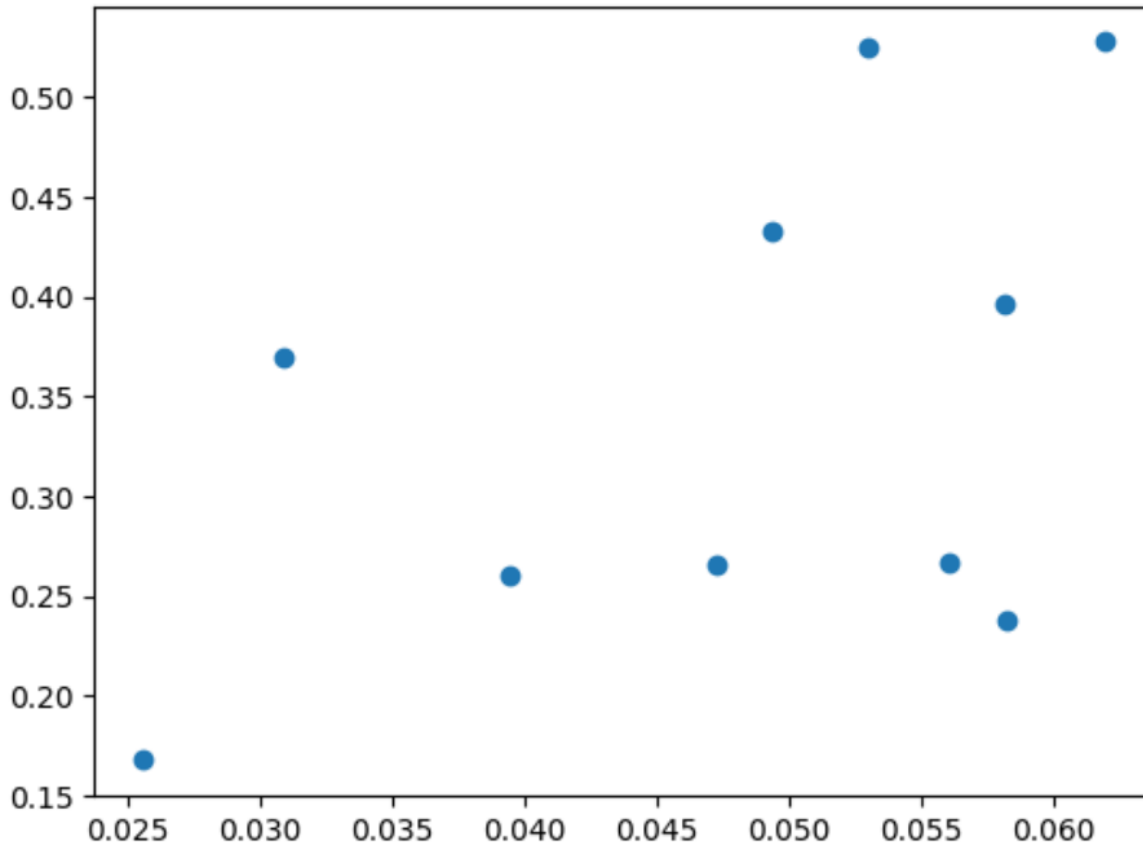
A: We find that as we add more predictors to our linear regression model, the R^2 metric increases, which makes sense since the more features we add into our model, the higher the R^2 is. Moreover, we notice that since for those added columns, there are very few missing values within, which means we have more data to create a robust regression model.

But based on the scatter plot, we observe that for the top 10 predicted movies from problem 1), the higher the old COD is, the lower the new COD is. The reason could be that incorporating social-economic features reduces the overfitting caused by just using ratings as predictors.

Additional features increase the variability in the dataset. This can help the model avoid focusing too narrowly on a limited set of features, which is often a cause of overfitting.

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.

D: We first pick 30 movies from the middle average COD range using random sampling. Then for each of these movies, we randomly sample 10 other movies as our predictor to fit a regularized multiple linear regression model based on an 80/20 split scheme. For each fitted model, we get the RMSE as the metric for model performance and the corresponding betas from the model. We first fix the alpha value to be 0.1 and then variate the alpha value to hypertune the model to find the best alpha. The lower the average RMSE of the 30 models, the better the alpha is.

Y: Since RMSE represents the mean squared error of the model, the lower it is, the better the model parameters are. Generally, we want our alpha to be large but not too large. One reason is to control overfitting, where a larger alpha imposes a greater penalty on the size of the coefficients, which can help control overfitting by keeping the coefficients small and reducing the model's complexity. However, if alpha is too large: it can introduce too much bias, causing the model to underfit. This means that the model becomes too simple and fails to capture the underlying trend in the data, resulting in poor predictive performance.

F: The findings are listed below, we variate the alpha to see the change in RMSE in RMSE and betas

	movie	predictor	RMSE	alpha	betas
0	Spirited Away (2001)	[Barb Wire (1996), King Kong (1976), Divine Se...	0.671223	0.01	[-0.02, -0.02, -0.42, -0.27, 0.3, 0.99, -0.08,...
1	Spirited Away (2001)	[Barb Wire (1996), King Kong (1976), Divine Se...	0.673769	0.1	[-0.01, -0.01, -0.41, -0.26, 0.3, 0.96, -0.07,...
2	Spirited Away (2001)	[Barb Wire (1996), King Kong (1976), Divine Se...	0.975231	10	[0.02, 0.02, -0.1, 0.02, 0.16, 0.24, 0.04, 0.3...
3	Spirited Away (2001)	[Barb Wire (1996), King Kong (1976), Divine Se...	1.124964	100	[0.03, 0.03, 0.01, 0.06, 0.06, 0.08, 0.05, 0.1...

4	Spirited Away (2001)	[Barb Wire (1996), King Kong (1976), Divine Se...	1.473300	1000	[0.01, 0.01, 0.0, 0.01, 0.01, 0.02, 0.01, 0.02...

The hypertuning results are as follows:

Grid Search Results
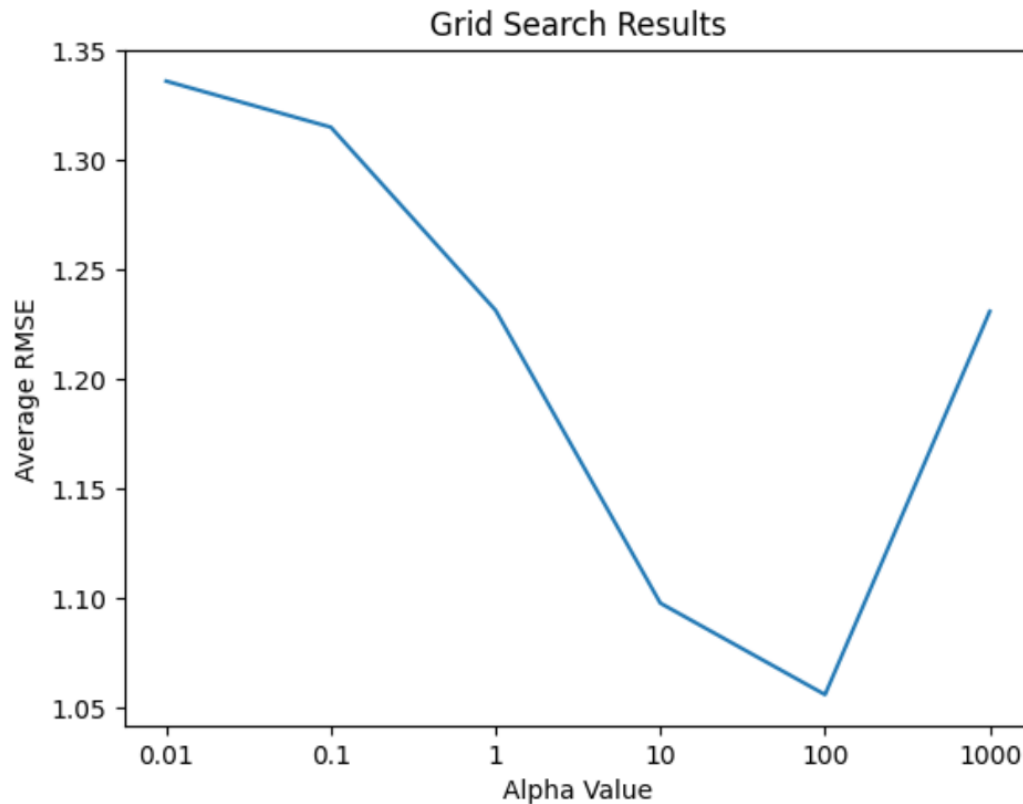
A: Based on the plot we see that when we choose the regularization coefficient to be 100, we get on average the least RMSE across all 30 movies. Also based on the plot, If alpha is too large, we suffer from underfitting. If the alpha is too small, the model is overfitting. Moreovwe, we see from the table that the larger the alpha value is, the smaller the betas are, which corresponds to the ability of ridge regression to decrease the magnitude of the coefficients.

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

D: We first pick 30 movies from the middle average COD range using random sampling. Then for each of these movies, we randomly sample 10 other movies as our predictor to fit a regularized multiple linear regression model based on an 80/20 split scheme. For each fitted model, we get the RMSE as the metric for model performance and the corresponding betas from the model. We first fix the alpha value to be 0.1 and then variate the alpha value to hypertune the model to find the best alpha. The lower the average RMSE of the 30 models, the better the alpha is.

Y: Since RMSE represents the mean squared error of the model, the lower it is, the better the model parameters are. Generally, we want our alpha to be large but not too large. One reason is to control overfitting, where a larger alpha imposes a greater penalty on the size of the coefficients, which can help control overfitting by keeping the coefficients small and reducing the model's complexity. However, if alpha is too large: it can introduce too much bias, causing the model to underfit. This means that the model becomes too simple and fails to capture the underlying trend in the data, resulting in poor predictive performance.

F: The findings are as follows, we variate the alpha to see the change in RMSE in RMSE and betas

	movie	predictor	RMSE	alpha	betas
0	Star Wars: Episode V - The Empire Strikes Back...	[Honey (2003), King Kong (1976), Divine Secret...	0.718700	0.01	[-0.04, -0.04, 0.08, 0.03, 0.47, 0.22, 0.34, -...
1	Star Wars: Episode V - The Empire Strikes Back...	[Honey (2003), King Kong (1976), Divine Secret...	0.714237	0.1	[-0.04, -0.04, 0.08, 0.04, 0.46, 0.22, 0.33, -...
2	Star Wars: Episode V - The Empire Strikes Back...	[Honey (2003), King Kong (1976), Divine Secret...	0.677541	1	[-0.03, -0.03, 0.09, 0.07, 0.41, 0.24, 0.29, -...

3	Star Wars: Episode V - The Empire Strikes Back...	[Honey (2003), King Kong (1976), Divine Secret...	0.596228	10	[-0.02, -0.02, 0.09, 0.12, 0.25, 0.24, 0.14, -...
4	Star Wars: Episode V - The Empire Strikes Back...	[Honey (2003), King Kong (1976), Divine Secret...	0.826624	100	[0.02, 0.02, 0.06, 0.07, 0.1, 0.11, 0.04, 0.03...
5	Star Wars: Episode V - The Empire Strikes Back...	[Honey (2003), King Kong (1976), Divine Secret...	1.312609	1000	[0.00, 0.01, 0.01, 0.01, 0.02, 0.00, 0.01, 0.0...

The hypertuning results are as follows:
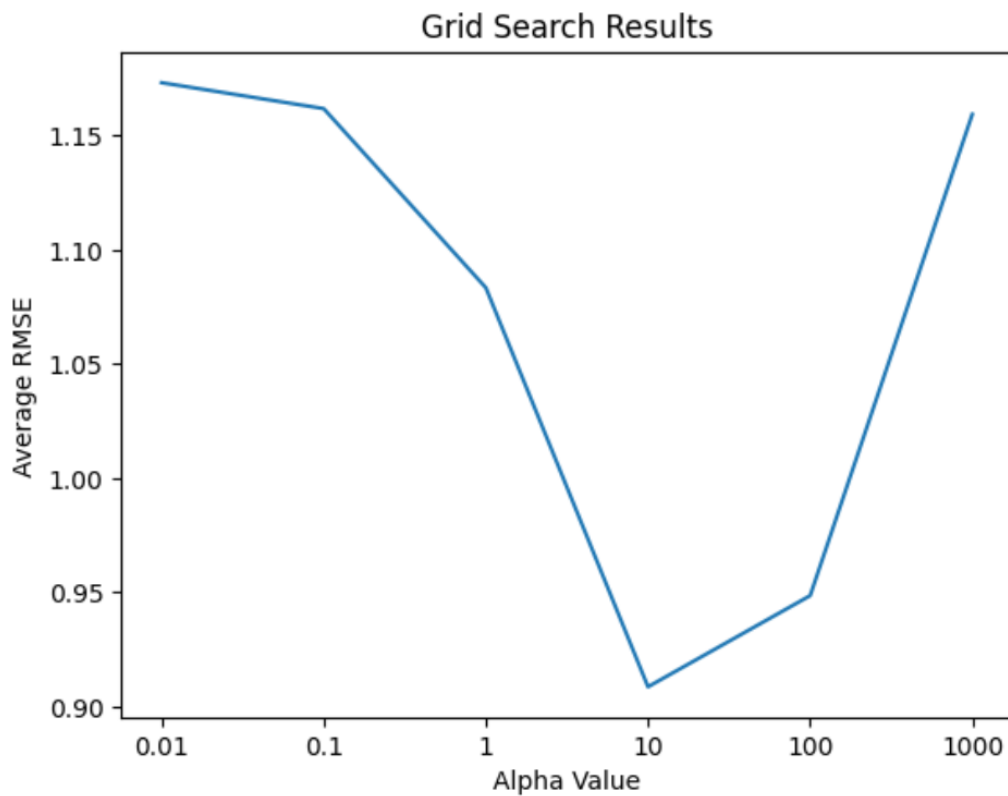
A: Based on the plot we see that when we choose the regularization coefficient to be 10, we get on average the least RMSE across all 30 movies. Also based on the plot, If alpha is too large, we suffer from underfitting. If the alpha is too small, the model is overfitting. Moreover, we see from the table that the larger the alpha value is, the smaller the betas are, and when we set alpha to be 1000, some of the

coefficients become zero, which corresponds to the ability of lasso regression to select a subset of features.
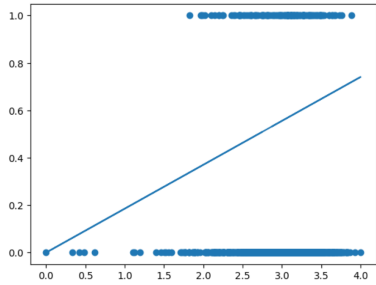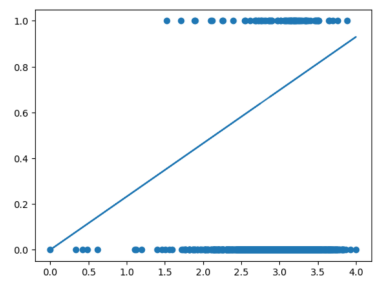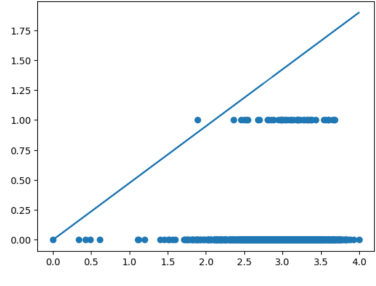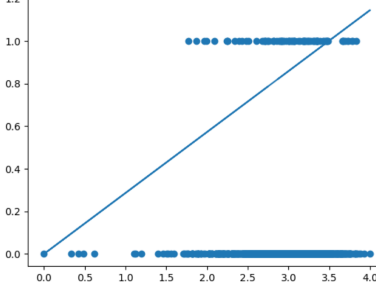
5) Compute the average movie enjoyment for each user (using only real, non-imputed data). Use these averages as the predictor variable X in a logistic regression model. Sort the movies order of increasing rating (also using only real, non-imputed data). Now pick the 4 movies in the middle of the score 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.

D: We first calculate each user's average movie enjoyment (using the non-nan data, all the nan values will be set to zero automatically) as feature X. We then sort the movies in ascending order of rating (again, using only real, non-imputed data). We then perform the media split (here we use median imputation) of ratings for each of 4 movies picked from the middle of the score range to create the target label Y. Then we fit a logistic regression model using cross validation method with X as predictor and Y as label and calculate the AUC values and betas. For each movie, we select the logistic regression model with the highest AUC score.

Y: Before median splitting, we decide to use median imputation instead of mean imputation, since using median imputation can be more appropriate than mean imputation. This method is robust to outliers and can be more representative of the central tendency, especially in skewed distributions or when there are many missing values, which is the case in our data set.

F: The results are as follows:

	movie	best betas	best_AUC _score	Decision Boundary

0	The Blair Witch Project (1999)	[-2.45, 0.18]	0.670387	
1	The Blue Lagoon (1980)	[-3.2, 0.23]	0.639851	
2	Who's Afraid of Virginia Woolf (1966)	[-4.60, 0.47]	0.766667	
3	28 Days Later (2002)	[-3.28, 0.29]	0.723872	

A: From the above modeling process, we observe that the AUC score is pretty high, showing that average movie enjoyment is a decent predictor to predict whether a movie is good or bad. The reason many be that average enjoyment scores typically reflect the collective sentiment of viewers. If most people enjoy a movie, it's often considered a good sign of its overall appeal. But the limitation of this is that the average enjoyment can be highly subjective and can vary widely among different audiences. Factors like genre preference, cultural background, and individual taste can significantly influence enjoyment scores. So if we perform the same modeling process on a different dataset, the AUC score may not be that high. Moreover, average scores can be skewed

by extreme ratings or by a non-representative sample of viewers. For instance, fans of a specific genre might rate a movie highly, while others might not find it enjoyable. So we should also question how the movie data is collected.

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].

We would like to **cluster all reviewers** in this system by exploring different clustering strategies and see what is the empirical difference. We notice that column 475 indicates Gender identity (1 = female, 2 = male, 3 = self-described) and column 476 indicates Only child (1 = yes, 0 = no, -1 = no response) are features of individuals. Based on these two features, theoretically there should be at most 3*3=9 groups. After empirical experiments, we found a **fact: K=4 is the most reasonable according to the elbow test.**

D: We use two features (gender, only child) to construct a **2-dimensional representation** of each reviewer (1097 in total). Then, we **get rid of** pairs that contain **any NaN** values. There are 1073 remaining. Then, we performed **K-means** to cluster them. We tried **K=3,4,.9**. Next, we performed the elbow test to determine the number of clusters in the optimal situation.

Y: We wanted to try a clustering algorithm on this toy example. We are curious about what will happen when there are less than 8 clusters because in that case some categories will have to be merged with another. This could potentially provide insights on which reviewers are more similar than others. We get rid of NaN because otherwise the data will not be 2-dimensional and thus could not be clustered.

F: We noticed that we have multiple points that are exactly the same, they are treated as a single point during the clustering process. A basic finding is that none of the reviewers report their gender as 3 = self-described. So there are in fact only 8 different points which means there could be up to 8 clusters. We visualized the data points of each cluster.

A: The elbow test results show that **K=4 is a clear elbow**, which we believe is the optimal number of clusters. We also noticed that when **K is small, only child=0,1 are merged** into one group. This shows that **Only child status is not a distinguishable characteristic** of individual reviewers.

K-means Clustering (K=3)   K-means Clustering (K=4)   K-means Clustering (K=5)

K-means Clustering (K=6)   K-means Clustering (K=7)   K-means Clustering (K=8)

Elbow Test for Optimal K