Introduction to Data Science - DS-GA 1001

Data Analysis Project 2

Shuyan Liu • Saakshi More • Antonio Vela Gartner

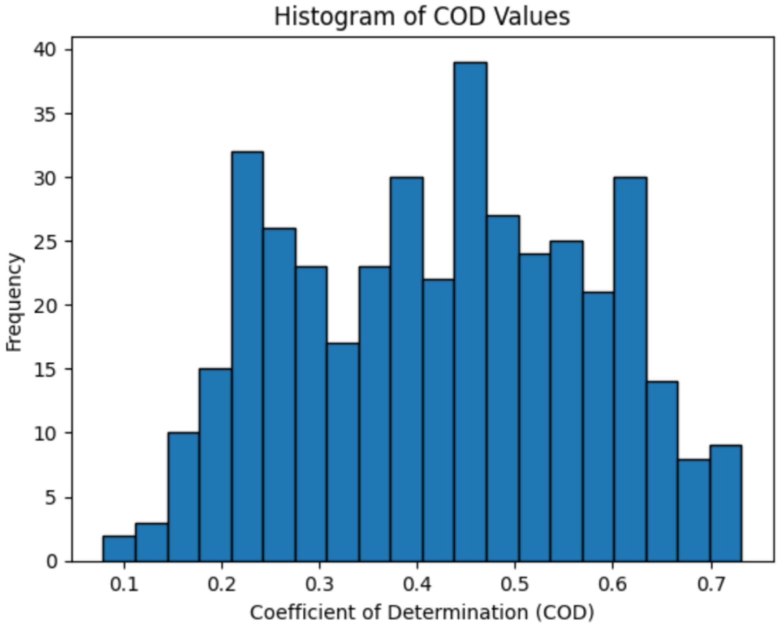
Data Handling:

Since we will be analyzing the data with regression, we need to handle missing values. The missing values in the movie rating data are labeled as NaN, these were filled with a 50/50 split of row means and column means. Moreover, user 896 did not rate any of the movies so we removed the entire row since it provides no information.

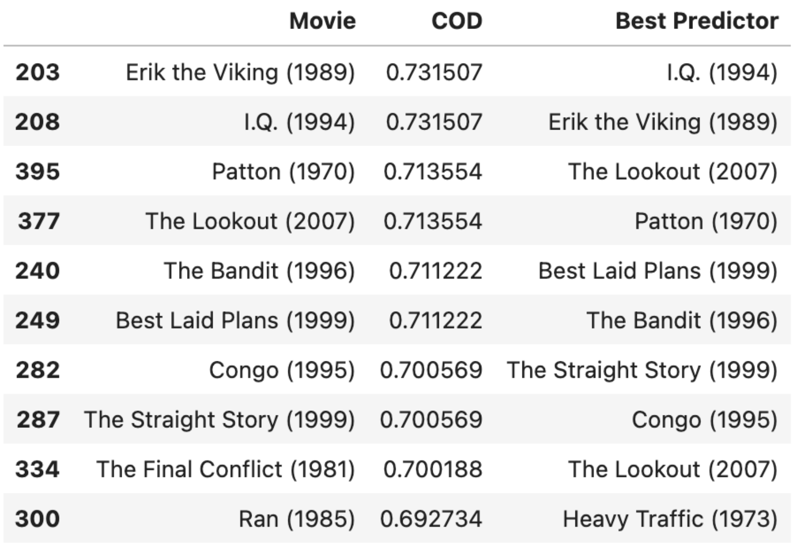
We decided not to apply any data transformation for this dataset since the data collected is on the same scale.

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 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).**

Here we set out to find the best single movie predictor for each movie, in a sense we are trying to find the most similar movie for each movie. To do so, we ran linear regression models on each movie using each individual remaining movie and picked the regression model with the highest COD. The average COD was 0.423 and the distribution of CODs can be seen below.



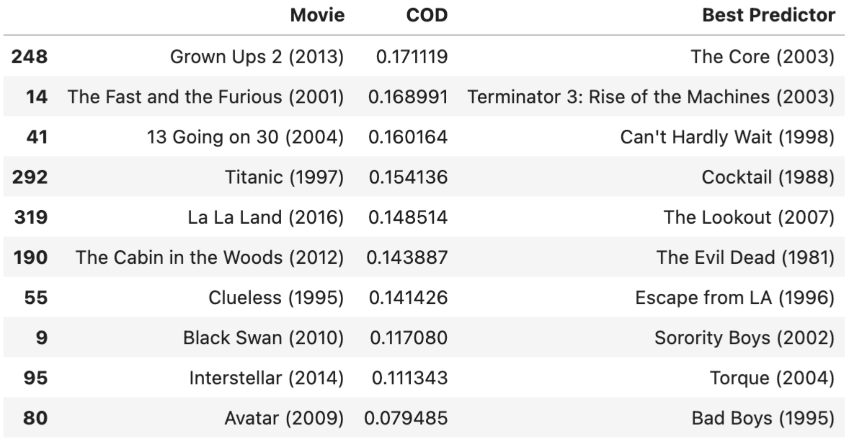
The movies that were best predicted and their predictors are:



	Movie	COD	Best Predictor
203	Erik the Viking (1989)	0.731507	I.Q. (1994)
208	I.Q. (1994)	0.731507	Erik the Viking (1989)
395	Patton (1970)	0.713554	The Lookout (2007)
377	The Lookout (2007)	0.713554	Patton (1970)
240	The Bandit (1996)	0.711222	Best Laid Plans (1999)
249	Best Laid Plans (1999)	0.711222	The Bandit (1996)
282	Congo (1995)	0.700569	The Straight Story (1999)
287	The Straight Story (1999)	0.700569	Congo (1995)
334	The Final Conflict (1981)	0.700188	The Lookout (2007)
300	Ran (1985)	0.692734	Heavy Traffic (1973)

Note that the first eight movies form pairs in which they predict each other with the same COD. However, The Final Conflict is a better predictor of The Lookout than the other way around.

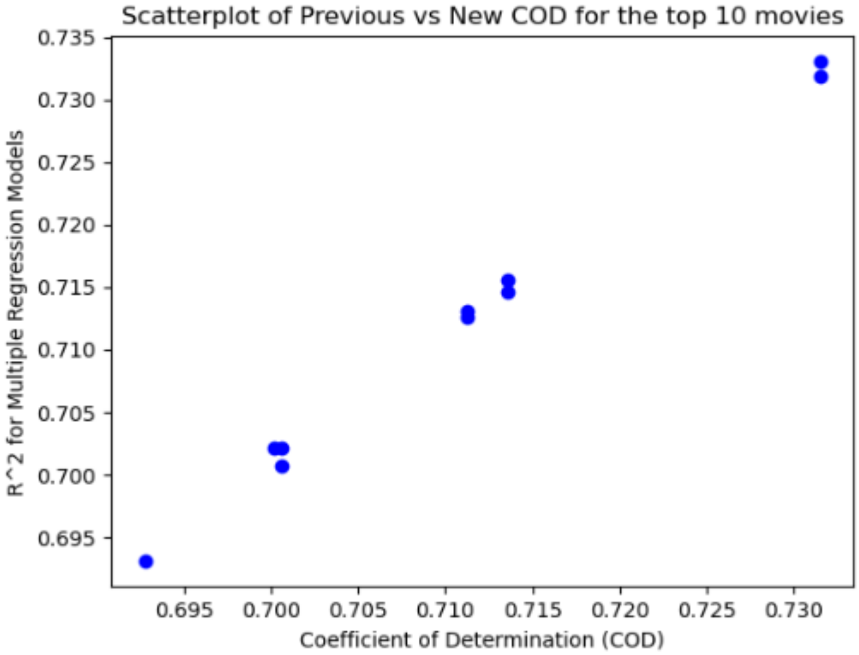
The movies that were the hardest to predict and their best predictors are:

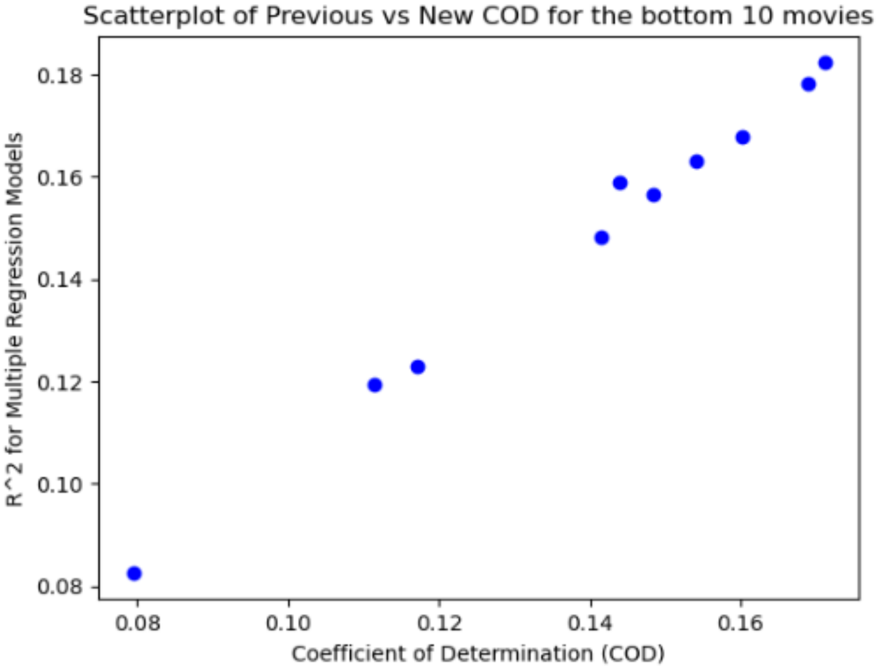


	Movie	COD	Best Predictor
248	Grown Ups 2 (2013)	0.171119	The Core (2003)
14	The Fast and the Furious (2001)	0.168991	Terminator 3: Rise of the Machines (2003)
41	13 Going on 30 (2004)	0.160164	Can't Hardly Wait (1998)
292	Titanic (1997)	0.154136	Cocktail (1988)
319	La La Land (2016)	0.148514	The Lookout (2007)
190	The Cabin in the Woods (2012)	0.143887	The Evil Dead (1981)
55	Clueless (1995)	0.141426	Escape from LA (1996)
9	Black Swan (2010)	0.117080	Sorority Boys (2002)
95	Interstellar (2014)	0.111343	Torque (2004)
80	Avatar (2009)	0.079485	Bad Boys (1995)

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.**

We now seek to improve the results from the single movie predictions by adding the personal information of each user. The information was gender identity, sibship status, and social viewing preferences. Because the additional predictors such as gender identity had categorical answers with 3 answers, we applied one-hot encoding to these categorical predictors, omitting the first column to prevent multicollinearity. Our X in this problem will be the ratings of the best predictor movie we had in question 1 for each of the 20 movies in addition to three categorical answers. Y will be the ratings of each of the top 10 and the bottom 10 movies. We then ran the model for all 20 movies and calculated the R^2 of each movie. We calculated the mean of the difference between the new and the previous R^2. On average the R^2 improved by 0.00494. For the top 10 movies, it improved by 0.0014, and for the bottom 10, it improved by 0.0085. This suggests that with the additional predictors, our model improved. The bottom 10 movies exhibited better improvement which suggests that the new categorical variables added more explanatory power to the bottom 10 movies compared to the top 10 movies. We can also see from the scatterplot that with the new predictors, both top and bottom movies had an increased R² which suggests an improvement of our model, however, the models are likely better at explaining the variance in the top movies compared to the bottom ones. The bottom movies also had a broader range of R² values that suggest more variability in how well the models can explain the variance of the bottom 10 movies. Despite these enhancements, the overall improvement of our model was modest.





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.

We removed the top 10 and bottom 10 movies as calculated in Q1 to form our random sample of 30 movies. Then we removed these 30 movies to again resample 10 more movies, whose ratings became the predictors of our ridge regression model. This model was used to predict corresponding user ratings for each of the 30 movies. While we considered sampling the 10 movies as per those with least null values, it would introduce bias since these 10 movies would be the 10 most popular movies. We performed a grid search on 7 alpha parameter values (10-3, 10-2, … 100, 1000) to determine which alpha value best performed the prediction. We saw that we only obtained alpha values 10 or 100. Thus, we redefined the ridge regression model, considering alpha values from 5 to 120, expecting a better-performing model (lesser RMSE). Although we did not see any notable improvement in the RMSE, we used this model to plot a boxplot for the beta values of each predictor (each of the 10 movies) (figure 1). Of the 10 sampled movies, Goodfellas and Predator were the best predictors, since they had the highest median betas and relatively fewer outliers (and even the outliers were extremely high values). The Proposal was the worst indicator since it had one of the lowest median beta values and many outliers, with one outlier being the least possible beta value. The average RMSE for this model was 0.416.

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

Our problem statement for this question was the same as the previous question, except that we had to define a Lasso Regression model. We performed a grid search on 7 alpha parameter values (10-3, 10-2, … 100, 1000) to determine which alpha value best performed the prediction and obtained 0.010 and 0.001 as the only alphas. Thus, we redefined the model for alpha values ranging from 0.0001 to 0.1. We did not observe any marked improvement in the RMSE but used this model to plot a boxplot for the beta values of the 10 predictors (figure 2). The result we got was almost identical to that of q3 with Goodfellas and Predator as the best predictors.

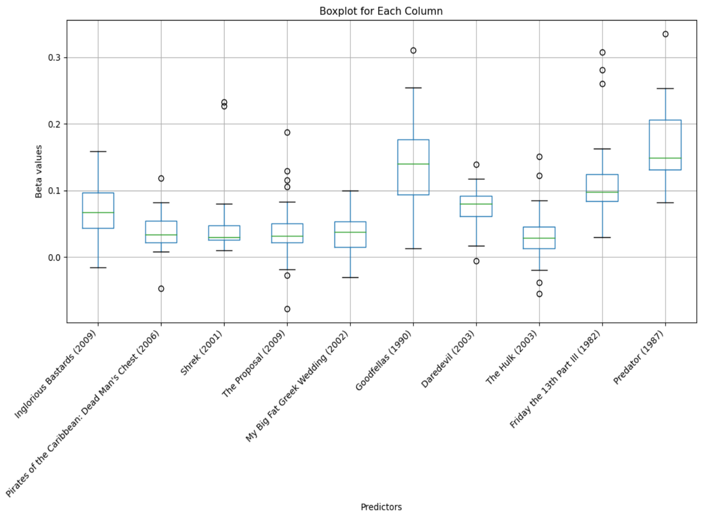


Figure 1. Boxplot of Ridge Regression Model Betas for each predicting movie

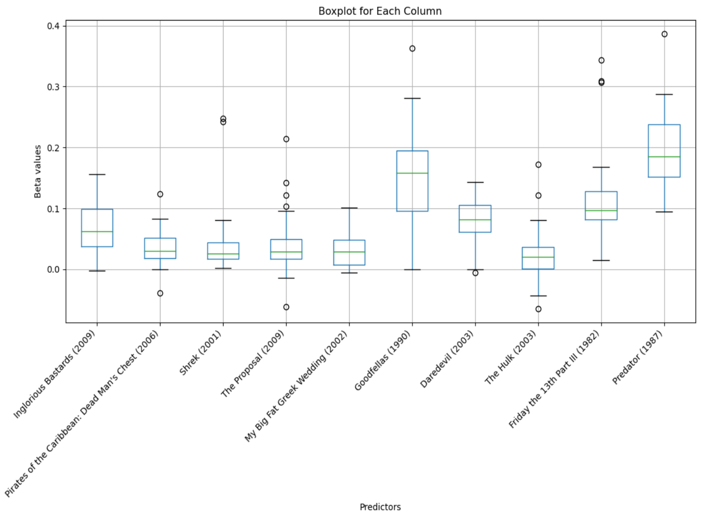
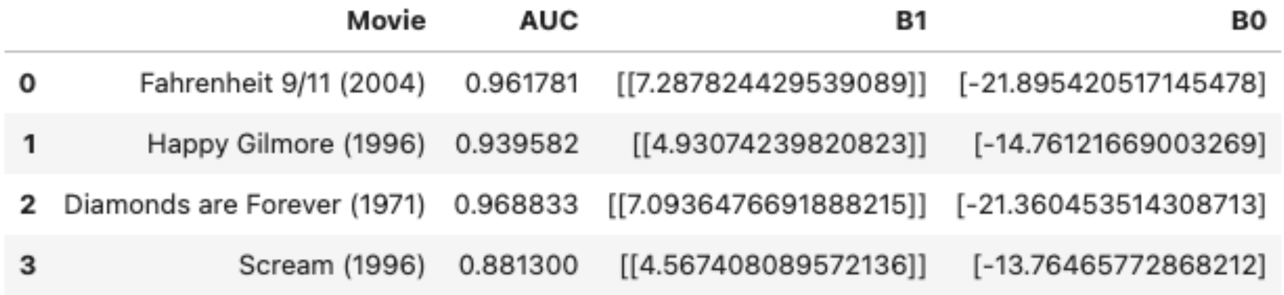


Figure 1. Boxplot of Lasso Regression Model Betas for each predicting movie

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.**

In this question, we first calculated each user's mean ratings across all movies and stored them as X for our logistic regression model. We then sorted all 400 movies according to their mean ratings from all the users. 4 movies in the middle of this sorted list were picked, we found the median rating of each movie and based on this, we labeled every rating of each movie as 0 (= not enjoyed) and 1 (= enjoyed). Every rating that is greater than the median of the movie is labeled as 1 and as 0 if it is smaller than the median. We then built a logistic regression model for each of the four movies where X is the average rating of each user. Y is enjoyed or not of that user with that particular movie. We split the data into training and testing sets with a test size of 0.2. To prevent overfitting, we used K-Fold cross-validation on the training set with 5 folds. We then made the predictions on the test set and generated ROC graphs of each fold and the entire test set for each of the four movies, along with reports to evaluate the model.

For our first movie, in the Appendix First Movie, are the ROC curve graphs for each of the five folds. All our folds performed very well, the lowest AUC we had was 0.95, meaning the model was highly capable of distinguishing between the two classes (enjoyed vs. not enjoyed). The ROC line was bow toward the top left corner of the plot, far from the random line suggesting the model performs significantly better than random guessing. The ROC curve for the test set also achieved a great result with an AUC of 0.96 and the ROC curve far from the random line. It suggests that our model performed very well and is highly effective for predicting movie enjoyment, it also suggests that the average user rating is a very strong predictor of whether a user will enjoy a movie. For our second movie, graphs showed in Appendix Second Movie, our 5 folds had a slightly worse performance than the first movie, with AUC from 0.89 to 0.92. And ROC curves closer to the random line. Our model performed better on the test set with an AUC of 0.94 and a ROC curve far from the random line. This suggests that our model had no indication of overfitting since the performance is even slightly better on the test set and the prediction of enjoyment is very precise. For our third movie, graphs shown in Appendix Third Movie, our model performed better than the previous two movies, with an AUC of 0.92, 0.98, 0.98, 0.97, 0.97 on the CV folds and an AUC of 0.97 on the test set. This suggests that our model performed significantly on the third movie and made great predictions on the enjoyment of the movie. For our last movie, graphs shown in Appendix Fourth Movie, we had an AUC ranging from 0.83 to 0.94 on our CV folds and an AUC of 0.88 on the test set. The model performance was slightly worse than other models but still made predictions at an effective level. It also suggests that for the fourth movie, the average user rating is not as strong of a predictor as what we have seen in other movies.

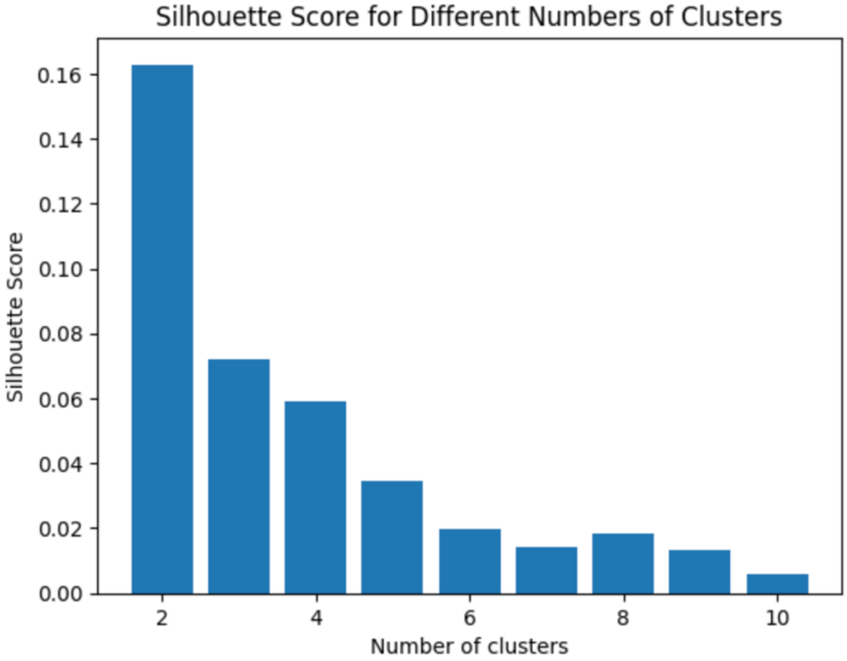


	Movie	AUC	B1	B0
0	Fahrenheit 9/11 (2004)	0.961781	[[7.287824429539089]]	[-21.895420517145478]
1	Happy Gilmore (1996)	0.939582	[[4.93074239820823]]	[-14.76121669003269]
2	Diamonds are Forever (1971)	0.968833	[[7.0936476691888215]]	[-21.360453514308713]
3	Scream (1996)	0.881300	[[4.567408089572136]]	[-13.76465772868212]

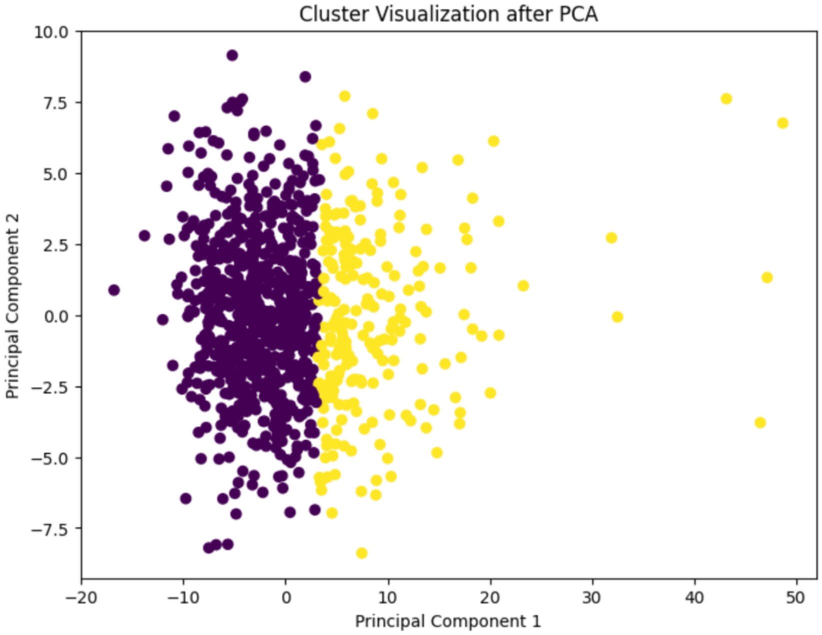
All of our B1 values are positive, indicating a positive relationship that as the average movie rating increases, so does the log odds of the movie being enjoyed. The B0 values are all negative and large in absolute value, suggesting that when we have no information from the average user rating, the model strongly leans towards predicting that the movie will not be enjoyed. The difference in betas cross movies indicates differences in how much the average ratings influence the prediction of enjoyment for different movies. All of our AUC values are relatively high, indicating all four of our models on each movie performed very well. In general, the quality of all four of our models is highly effective.

6. **Extra credit:**

Here we were interested in knowing if users can be grouped, this might be of interest for marketing purposes. To do so we decided to use the whole data set on a K-Means Clustering algorithm. First we eliminated any user that had a NaN value so that the algorithm could be run (only 150 users were lost), then we obtained the silhouette score for different number of cluster to select the best cluster number:



From the silhouette scores it was clear that using 2 clusters would give us the best result. However, a silhouette score of 0.16 is considerably low. Yet, we performed clustering and visualizing the results on a plot of the first two principal components:

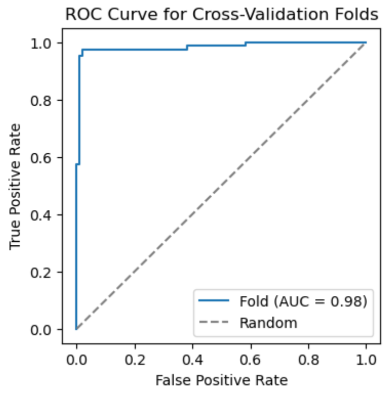


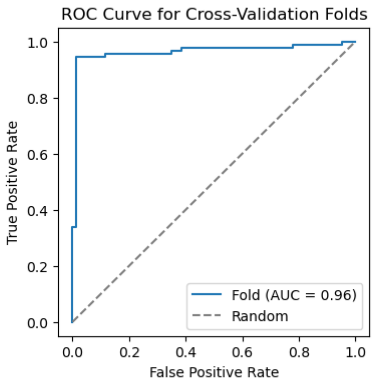
From the plot it is clear that users cannot be easily grouped in clusters since the clusters formed do not really form two clearly separated groups.

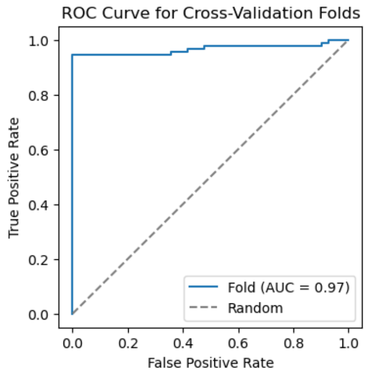
Appendix

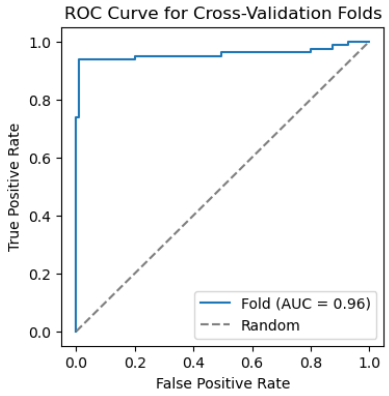
Question 5:

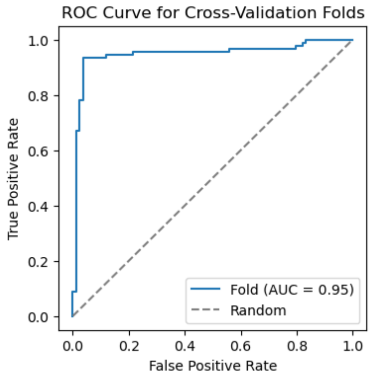
First Movie:

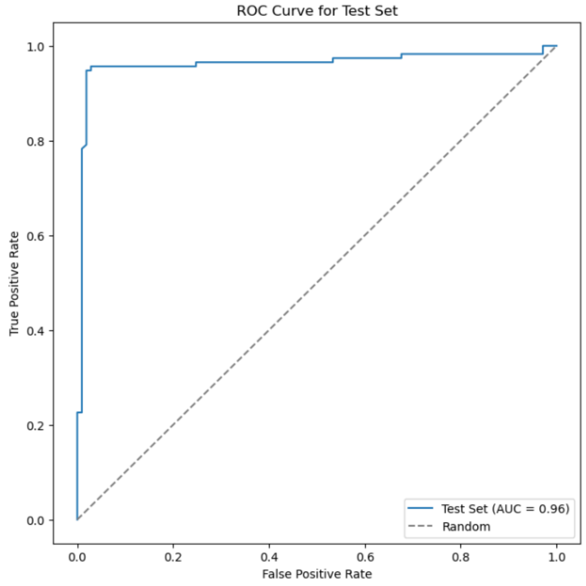




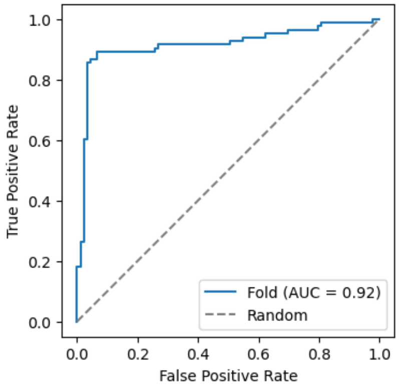


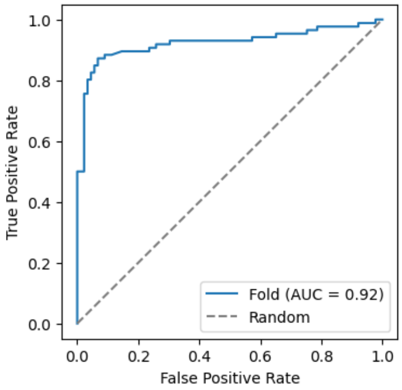


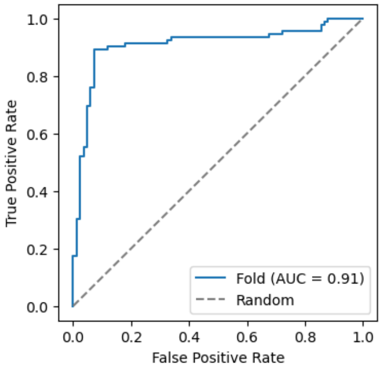


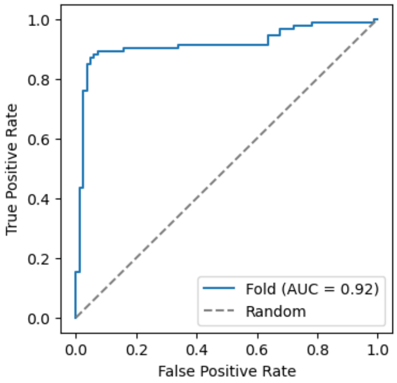


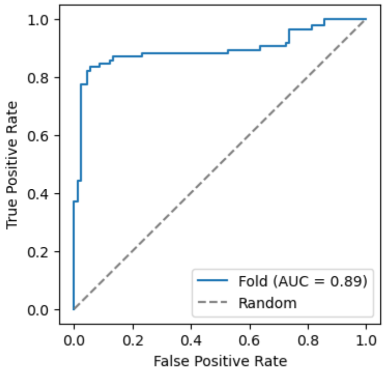
Second Movie:

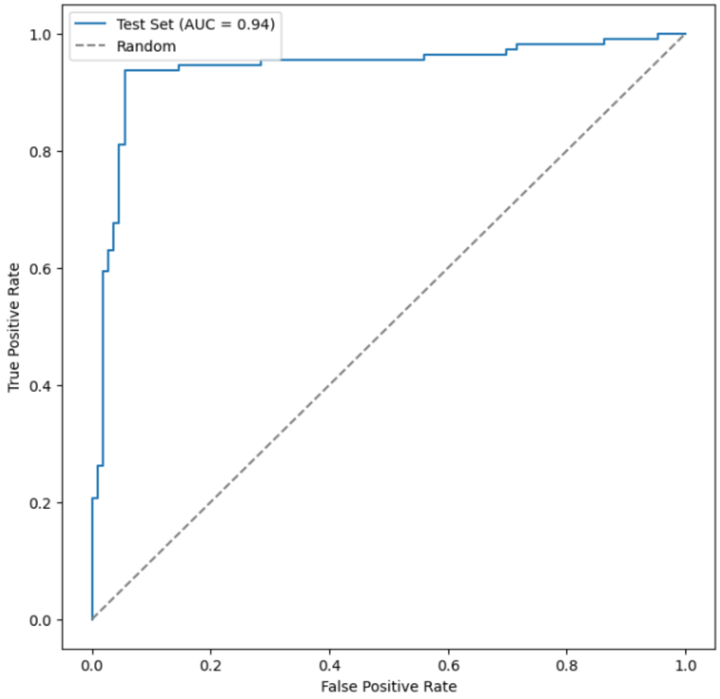

ROC Curve for Cross-Validation Folds


ROC Curve for Cross-Validation Folds

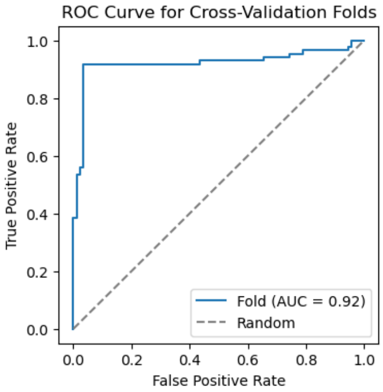

ROC Curve for Cross-Validation Folds

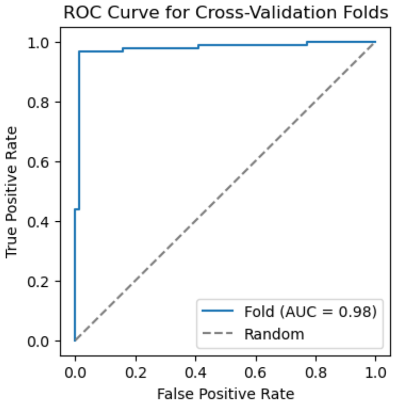

ROC Curve for Cross-Validation Folds

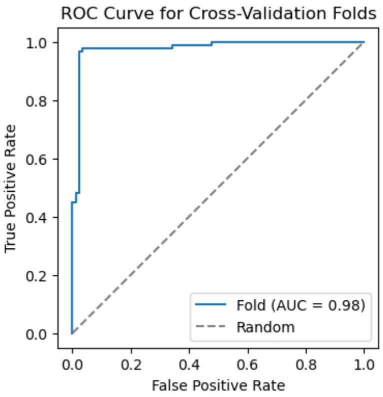

ROC Curve for Cross-Validation Folds

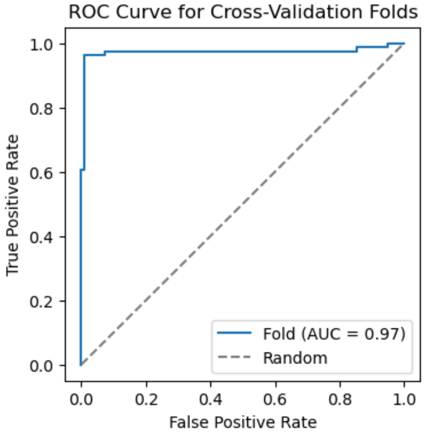

ROC Curve for Test Set

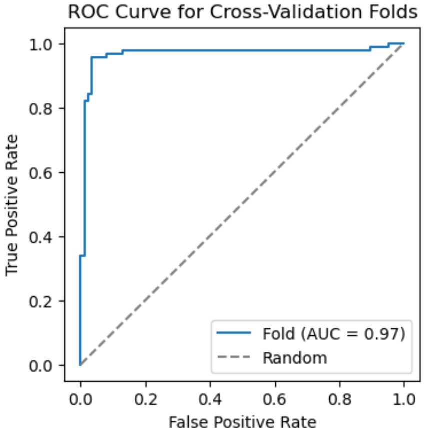
Third Movie:

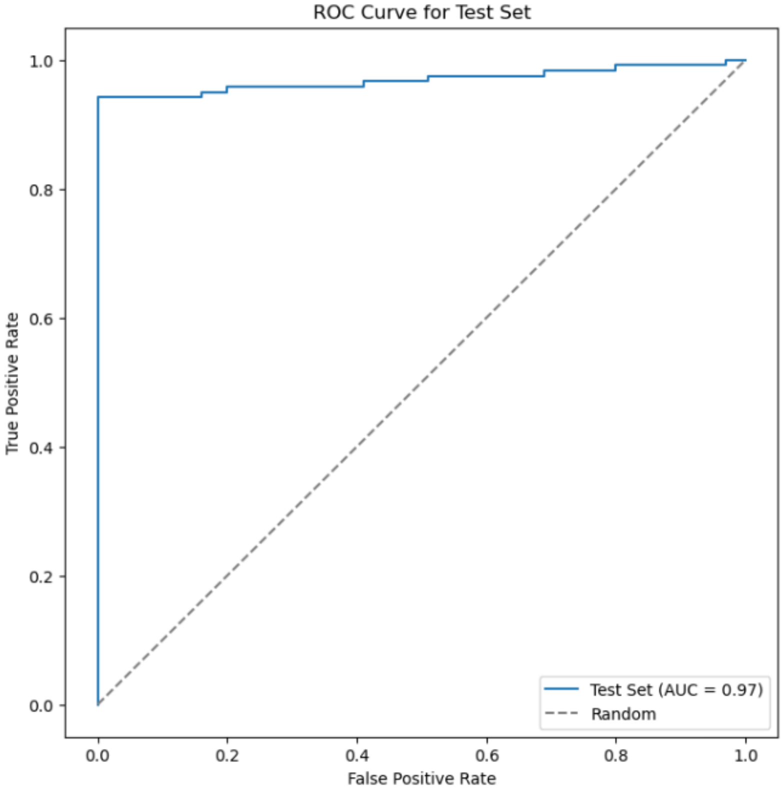

ROC Curve for Cross-Validation Folds
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Fold (AUC = 0.92)
Random


ROC Curve for Cross-Validation Folds
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Fold (AUC = 0.98)
Random

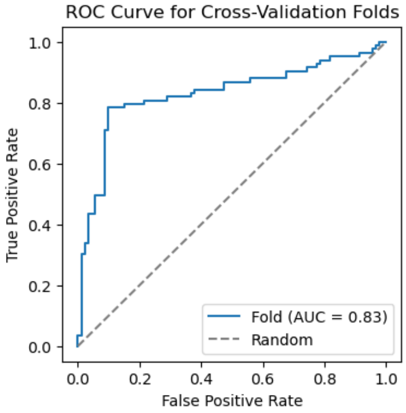

ROC Curve for Cross-Validation Folds
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Fold (AUC = 0.98)
Random

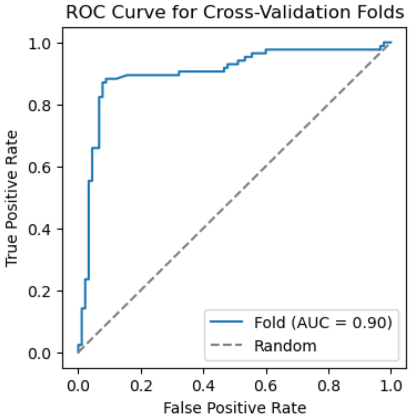

ROC Curve for Cross-Validation Folds
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Fold (AUC = 0.97)
Random

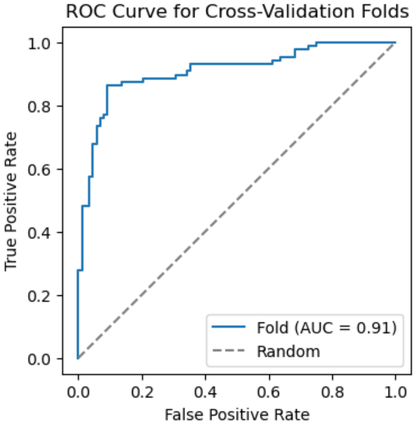

ROC Curve for Cross-Validation Folds
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Fold (AUC = 0.97)
Random

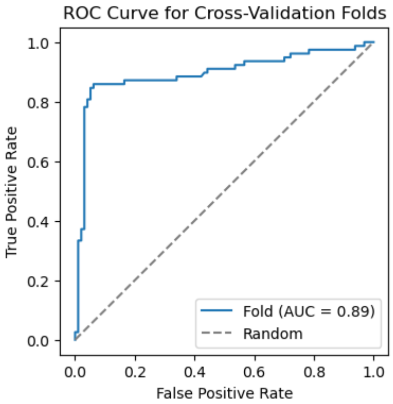

ROC Curve for Test Set
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Test Set (AUC = 0.97)
Random

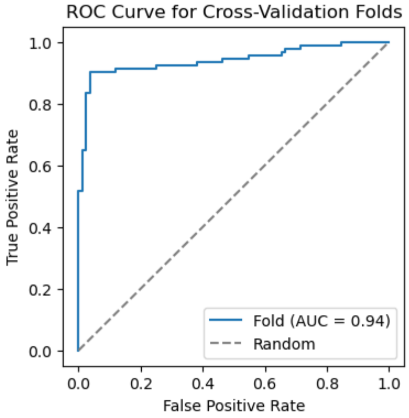
Fourth Movie:

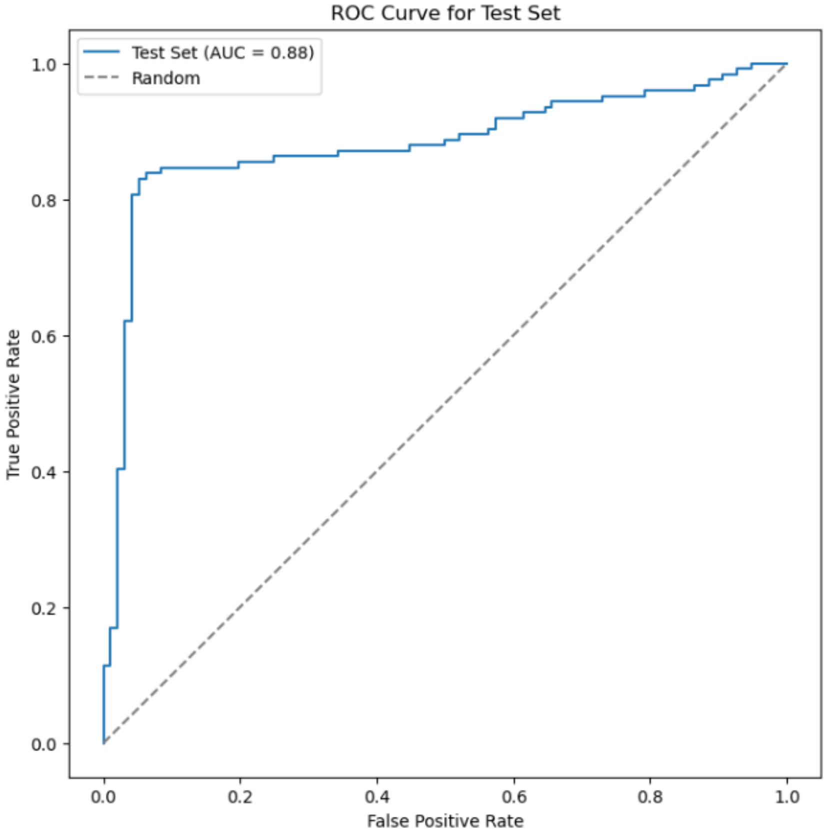

ROC Curve for Cross-Validation Folds


ROC Curve for Cross-Validation Folds

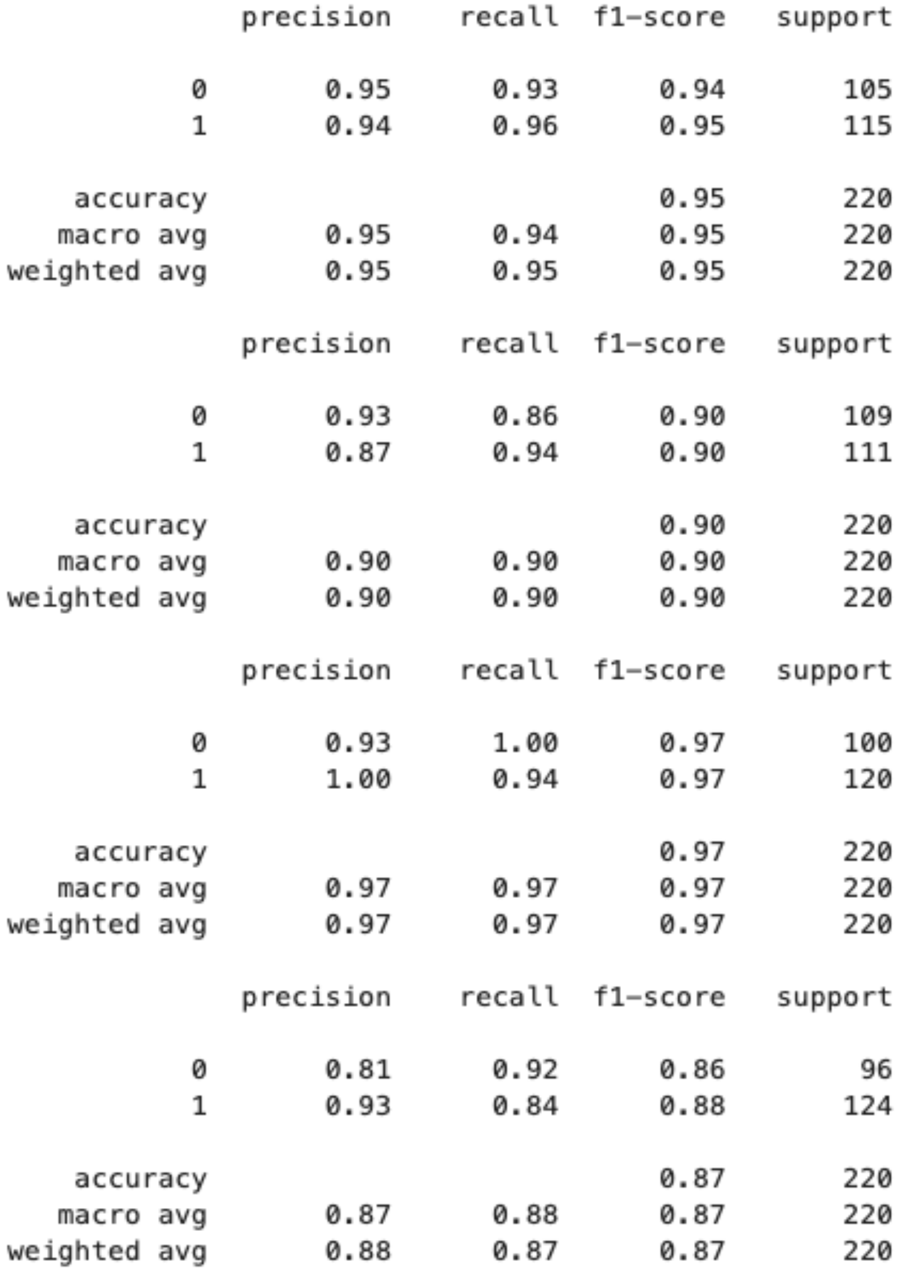

ROC Curve for Cross-Validation Folds


ROC Curve for Cross-Validation Folds


ROC Curve for Cross-Validation Folds


ROC Curve for Test Set

Question 5 report:



```
              precision    recall  f1-score   support

           0       0.95      0.93      0.94       105
           1       0.94      0.96      0.95       115

    accuracy                           0.95       220
   macro avg       0.95      0.94      0.95       220
weighted avg       0.95      0.95      0.95       220

              precision    recall  f1-score   support

           0       0.93      0.86      0.90       109
           1       0.87      0.94      0.90       111

    accuracy                           0.90       220
   macro avg       0.90      0.90      0.90       220
weighted avg       0.90      0.90      0.90       220

              precision    recall  f1-score   support

           0       0.93      1.00      0.97       100
           1       1.00      0.94      0.97       120

    accuracy                           0.97       220
   macro avg       0.97      0.97      0.97       220
weighted avg       0.97      0.97      0.97       220

              precision    recall  f1-score   support

           0       0.81      0.92      0.86        96
           1       0.93      0.84      0.88       124

    accuracy                           0.87       220
   macro avg       0.87      0.88      0.87       220
weighted avg       0.88      0.87      0.87       220
```