1 Introduction

In this project, you are asked to build a recommendation system. You are given a dataset of movie ratings by users. Based on ratings you have, you need to recommend five movies to each user, and your recommendations are evaluated on the held-out dataset.

Associated with the ratings, you also have some information about users and movies. For example, user ages, user genders, and release years of movies. You will also need to analyze your learned model with such data.

2 Collaborative Filtering

In this problem, you are asked to run collaborative filtering to make recommendations.

As we have talked in the class, we can find vector representations of users and items through SVD decomposition of the incomplete matrix of ratings. The training objective is

$$
\min_{(u_i)_{i=1}^N, (v_j)_{j=1}^L} \sum_{i=1}^N \sum_{j=1}^L (M_{ij} - u_i^\top v_j)^2 + \lambda \sum_{i=1}^N \|u_i\|^2 + \lambda \sum_{j=1}^L \|v_j\|^2
$$

Here each user $i$ and each movie $j$ are represented by $u_i \in \mathbb{R}^K$ and $v_j \in \mathbb{R}^K$. $M_{ij}$ is the rating of movie $j$ by user $i$. The formulation essentially uses low dimensional vectors to represent users and items.

You can use other variants of collaborative filtering algorithms to make recommendations. You can either write your own code to solve the problem or use existing python packages (e.g. surprise).

You need to use the model selection procedure to decide the value of model hyperparameters ($\lambda$ and $K$ in the formulation above). One way of doing so is to hold out some ratings as the validation set.

Evaluate Recommendations. After we have trained a recommendation system, we can evaluate it against the test set, which consists of triples in the form of (user, item, rating). The evaluation has two approaches.

In the first approach, we compare model predictions against ratings on user-item pairs in the test set only. For each triple in the test set, we let the model predict the rating and then compare the prediction against the true rating. We measure the quality of the prediction by the difference. Let’s use Mean Absolute Error (MAE) as an overall measure of all differences in this project. The surprise package can directly calculate MAE for you on the test set.
This evaluation method cannot well represent real applications, because we don’t know which item will a user view in the future. The evaluation does not consider implicit feedback either. If a user doesn’t view some items in the test set, it is likely because the user doesn’t like these items. This evaluation method cannot use such information.

In the second approach, we let the model to recommend \( K = 5 \) items for each user, and then evaluate the quality of the \( K \) recommended items by their average rating. To recommend \( K \) items to a user, we can use the model to predict the user’s ratings of all items not reviewed in the training set and then recommend top-ranked ones. We calculate the average rating of the \( K \) recommended items. If the pair of the user and the recommended item is in the test set, then use the true rating. Otherwise, assume the true rating is 2, meaning that the user doesn’t like the item much. The finally evaluation of all recommended items is their overall average rating.

### 3 Vector Analysis

In this project, you are asked to analyze the learned model to get better understanding of collaborative filtering. Especially, you are asked to look into learned vectors and see if there is any meaningful information.

We first compare user vectors and user information. Let \( U \) be the matrix of user vectors, with its \( i \)-th row as the vector \( u_i \) of user \( i \). Each column of \( U \) can be seen as a “learned” feature of users. Now we want to check whether \( U \) brings any information about the user. The method is to use learned features \( U \) to predict user genders \( g \). You can choose any classifier. Then run cross-validation on \((U, g)\) and choose the lowest cross-validation error. We expect to get an error significantly smaller than the error of random guesses. Note that no information about user gender is used in learning \( U \). If it can predict user gender from \( U \), then it means the recommendation model has information about users in \( U \).

Then we compare movie features and movie information in a similar approach. We use movie features \( V \) to predict release years \( y \) of movies. This time you need to choose a regression model, run cross-validation on \((V, y)\), and choose the lowest regression error. Please use Mean Squared Error in this task. Then we compare the error with a naive model that doesn’t use movie features. The naive model uses the label mean (the mean release year) of the training data as the prediction for all test instances. You don’t need to run cross-validation with the naive model – you can just use its training error because the training error of the naive model will be very similar to its cross-validation error. If the error obtained from the model using movie features is significantly better than the error of the naive model, then it means movie features learned from movie ratings contain information about movies.

### 4 Submission requirements

You need to follow submission requirements in this section to guarantee that you submission can be evaluated properly.

In your submission, you need to include 1) your code, 2) your report, and 3) the performance of your model(s).

Your code should clearly show your model training process. You should write clear documentation for your code. Please organize your files in the following directory structure. Your code for recommendation should be in the directory `recommendation_system`, and you code for feature
analyses should be in the directory `feature_analysis`. Zip `proj3-submission` to a file named `proj3-submission.zip` and submit it through provide.

- `proj3-submission`
  |- `code/
     |   |- `recommendation_system`
     |   |- `feature_analysis`
     |   |- `proj3-report.pdf`

Your report should clearly state what you have done for the project.

- State the learning algorithms you have used and the parameter setting of your algorithm. You should have the model selection procedure to decide model hyperparameters. Your report should be clear enough for one to repeat your experiment by only reading your report.

- Report your analysis of learned vectors. Your experiment results either indicate user/movie features have information about users/movies, or otherwise. You need to explain whether the results are what you expect. If so, explain why you expect such results. Otherwise, explain possible reasons that the results are not expected.

- Write a short paragraph to discuss how to incorporate user information and movie information into the recommendation model.

You are encouraged to describe any interesting findings from this project. Try to make the report pleasant to read – not only dry results.

Please report the performances, MAE and the average rating of your model to a shared spreadsheet. The link of the spreadsheet is posted to Piazza resources. There you can also see your classmates’ results. The results are only for your reference and will not be contribute to your grades.

5 Evaluation

In this project, your submission is evaluated based on your code, your analysis of the learned model, and your report.

- You get 40% of full points if you have correctly get a result of recommendations.

- You get 40% of full points if you have analyzed user vectors and movie vectors described in Section 3.

- You get 20% of full points if your report clearly shows what you have done and satisfies the requirements.

If you do not get full points for any grading item, you get partial points according to our estimation of the amount of your effort.
6 The data

The data is from MovieLens. The original dataset has 100,000 ratings by 943 users on 1682 movies. The link to the dataset is provided in Piazza resources. Please refer the README file from the dataset for detailed information. From the provided script prepare.py, we have extracted a training set, a test set, a file containing users’ genders, and a file containing movies’ release years. The script has removed one movie with id 267 from the original dataset, as this movie does not have detailed information. These extracted files are sufficient for you to finish this project if you don’t plan to do more investigation.