

# Movie Recommender - Component Specification

*Group names: Priyanka Bijlani, Sharmeelee Bijlani, Laura Thriftwood, Lakshmi Venkatasubramanian*

## Software Components

### Data Manager

This component extracts the data from the zip file and prepares it for use in the program. It converts the multiple csv-formatted data tables into pandas data frames that will be used by the data processor.

- Inputs: url of zip file
- Outputs: dataframes

### Data Processor

This component processes the data frames to conduct exploratory analysis of the data via a histogram of the ratings. It also builds a sparse matrix that will be used for building the recommendation model.

- Inputs: dataframes
- Outputs: sparse matrix

### Model

This component splits the dataset into train and test sets to prepare it for the machine learning lifecycle. It instantiates a collaborative filtering recommendation model based on the alternating least squares method and fits the train data set to the model.

- Inputs: dataset, split proportion
- Outputs: train dataset, test dataset, ALS model

### Evaluate Model

This component evaluates the recommendation model based on the predictions made from the test set. It compares the masked values in the test set to the training set to determine what the auc percentage is for the model.

- Inputs: predictions, test set, training set
- Outputs: auc score

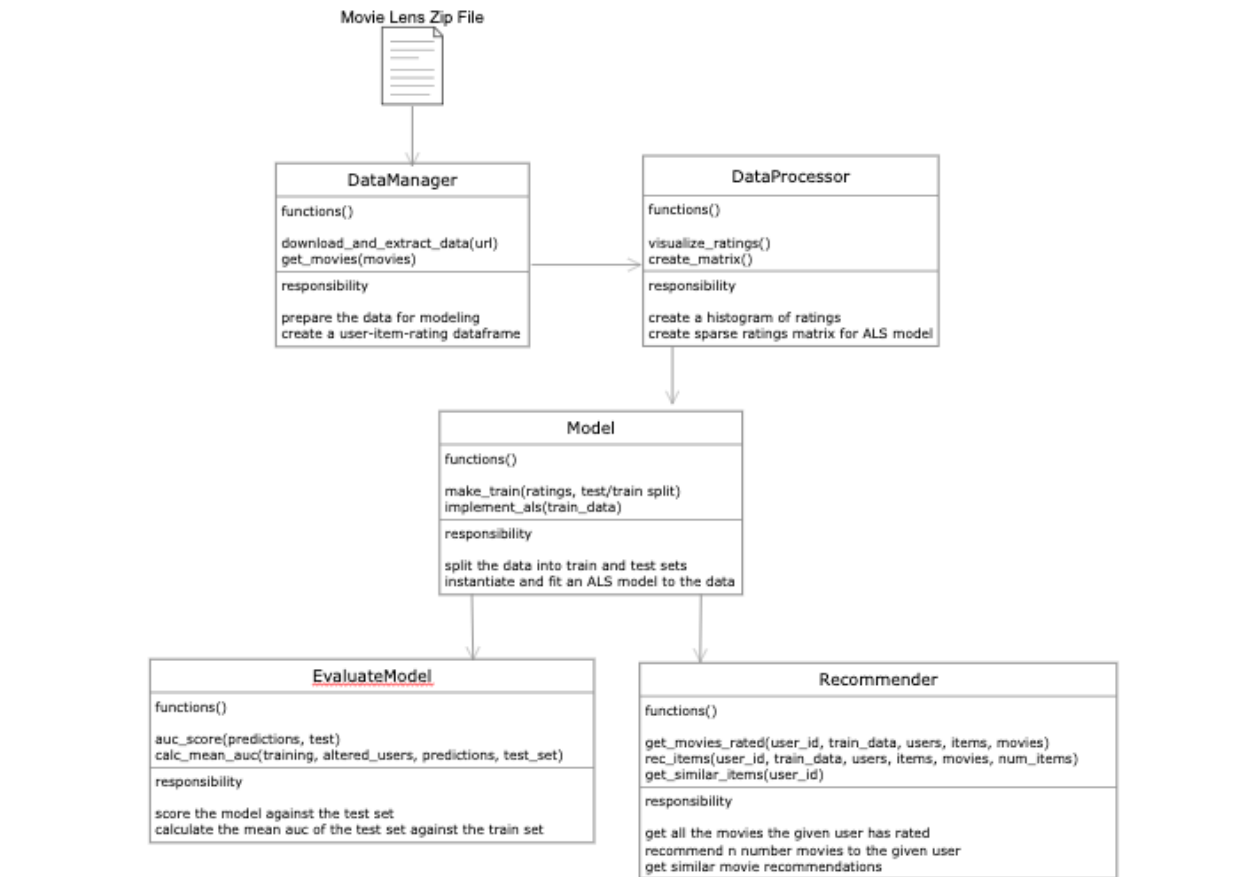
### Recommender

This component contains the recommender function that implements the use cases for this program. It provides recommendations for a user based on their ratings for other movies. It also finds similar movie recommendations for a given user based on what they have already watched.

- Inputs: user, number of recommendations, train data
- Outputs: recommended movies, similar movies

## Interactions to Accomplish Use Cases

The UML diagram shows the activity flow between the components in the recommendation system.



## Preliminary Plan

The below list of actions will be completed in the following order.

1. Complete recommender program.
2. Refactor code into modules.
3. Apply pylint.
4. Write unit tests.
5. Set up a continuous integration pipeline.
6. Add to project Github repo with all required components.
7. Finish project writeup.