STAT535: Statistical Computing **Spring 2019**

Lab 4 : Movie Recommendation

P. Flaherty

Objective The goal of the lab is to use past movie ratings to predict how users will rate movies they haven't watched yet. This type of prediction algorithm forms the underpinning of recommendation engines, such as the one used by Netflix. You have a large set of ratings from real movie-rating data, but holding back 200 ratings for you to predict. This project and the text above is drawn from Widom [1].

Collaboration Formalize your project teams on moodle and work with your team in this project

Readings This lab requires you to explore python packages that have been introduced, but not covered in detail in the class. In a typical data science role, you will need to learn about packages that might be useful and then use them to accomplish your analysis task. You should use google and read package documentation to learn about the various data science packages that have been touched on in class.

Data The following five data files are available for download on moodle. Files `movies` and `allData` are in TSV format (tab-separated values), since movie names may have commas in them. You can use the `read_csv` with the `sep` specified as '\t' in pandas to load tsv files.

- users.csv - Information about 2353 movie watchers. Each line has three fields: userID, age (see notes below), gender ("F" or "M")
- movies.tsv - Information about 1465 movies. Each line has six fields: movieID, name, year, genre1, genre2, genre3. If a movie has fewer than three genres the extra fields are blank.
- ratings.csv - 31,620 movie ratings. Each line has three fields: userID, movieID, rating. The userID and movieID correspond to those in the users.csv and movies.tsv files, respectively. Ratings are integers in the range 1 to 5 (from worst to best).
- allData.tsv - For those who prefer having everything in one place, this file contains the combined information from the previous three files. Each line has 10 fields: userID, age, gender, movieID, name, year, genre1, genre2, genre3, rating
- predict.csv - Ratings for you to predict. Each line has three fields: userID, movieID, rating, with all ratings set initially to 0. There are no ratings for these userID-movieID pairs in ratings.csv.

This data is real: it's a subset of the movie ratings data from MovieLens, collected by GroupLens Research, which Stanford CS102 anonymized. Start by browsing the schema and tables using head to see what's in the various fields. Note that the age field in the users.csv file doesn't contain exact ages but instead one of seven bucketized values as follows: 1 (age is under 18), 18 (age is 18-24), 25 (age is 25-34), 35 (age is 35-44), 45 (age is 45-49), 50 (age is 50-55), 56 (age is 56 or older).

1. **Import the data into pandas data frame** Using pandas, import the data into data frames that you can then explore with head and other pandas methods. Explore what type pandas has read in and consider converting the types for some columns to categorical. https://pandas.pydata.org/pandas-docs/stable/categorical.html

2. **Build a regression model using scikit-learn**

   Predict the rating for each movie between 1 and 5, where each rating may be a real number up to any number of decimal places. (Netflix, for example, predicts user ratings up to one decimal place, i.e., for a given user and movie it might predict "3.8 stars".) The evaluation metric is average absolute distance from the actual rating. Note that actual ratings are always integers from 1 to 5, but each distance (and of course the average distance) may be a real number of any number of decimal places.

   Fit a regression model of your choice and evaluate the metric.

3. **Test prediction performance** Training and testing on the same data set can lead to overfitting. There are two parts to this task. First, generate a random held-out validation set that you will not use for training in any way. Second, do a cross-validation with the training data to measure the prediction performance using only the training data. Third, generate predictions for the held-out validation data set and report your model's performance.

   You may want to return to part 2 and build more sophisticated models. You are encouraged to do so, but make sure you only use your validation set once for the model you select as your final model.

References

[1] Jennifer Widom. *Project 2: Movie Rating Predictions*. 2018. URL: https://web.stanford.edu/class/cs102/projects/project2.htm.