# Python Practical

## Examine the accuracy of a TensorFlow recommendation engine.

**Code:**

1. **Python program that interrogate the model.**

Firstly, for easier data processing, I modify the file “svd\_query.py”, so that model will return the id of recommended movie. Then I use the package telnetlib, which provides a Telnet class implements the Telnet protocol, to communicate with the Tensorflow model. I query the model with user id from 1 to 4999, and store the returned data in a csv file.

*import* ast  
*import* telnetlib  
*import* time  
*import* pandas *as* pd  
  
  
host\_ip = '127.0.0.1'  
tn = telnetlib.Telnet()  
tn.open(host\_ip, port=81)  
results = []  
*for* i *in range*(1, 5000):  
 tn.write(*str*(i).encode("ascii"))  
 time.sleep(0.1)  
 row = tn.read\_very\_eager().decode('ascii')  
 row = ast.literal\_eval(row)  
 user\_sample = [i]  
 *for* \_ *in* row:  
 user\_sample.append(\_[0])  
 results.append(user\_sample)  
results = pd.DataFrame(results)  
results.to\_csv("user\_recomm.csv", header=*None*, index=*False*)  
results = pd.read\_csv("user\_recomm.csv", header=*None*)

1. **Evaluate the model by comparing user’s demographic attributes.**

I write a Python program that evaluate the model by comparing the users’ age, gender and occupation attributes. Firstly, I read the data in “users.dat” and bin the age attribute into four categories: "Children" (182), "Teenagers (2624)", "Middle-aged" (1454), "Elder" (739). Then, I merge the users data with recommendation from TF model, and replace the movie id with corresponding genres. By counting the number of movies for every genre, we can observe the tendency of TF model to recommend movies (The code can be found in file “evaluate.py”).

To better evaluate the model, I visualize the number of top N genres recommended by the model, for people with different demographic attributes (from three dimensions: age, gender and occupation).

**Dimension one: age**

The horizontal axis in the Figure 1. is (movie genre, age category), and the vertical axis represents the total number of movie genres recommended by the model for users of the corresponding age category.

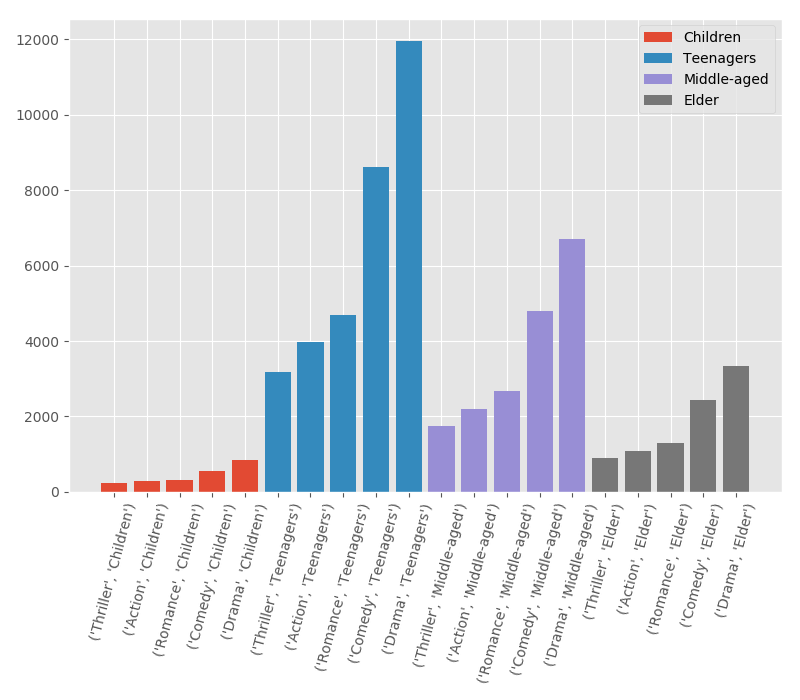


Figure 1. Dimension one - age

We can see the top 5 movie genres recommended by the TF model are all “Drama, Comedy, Romance, Action, Thriller”, regardless of the users’ age category.

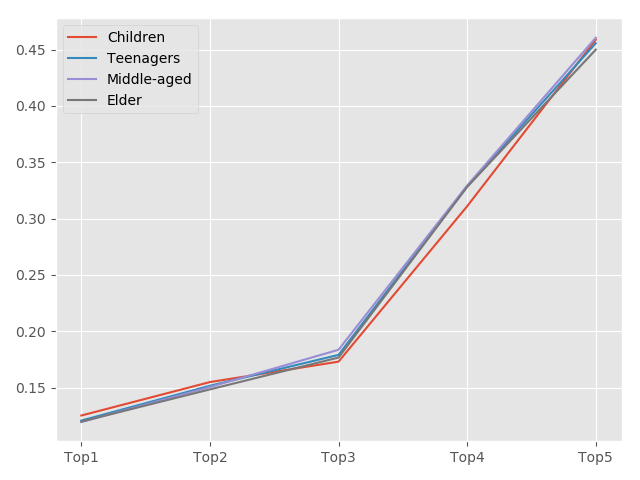


Figure 2. Dimension one - age

The Figure 2. represents the normalized proportion of users for top 5 movie genres recommended by TF model. We can see that the distribution is also very similar.

**Dimension two: gender**

The horizontal axis in the Figure 3. is (movie genre, gender), and the vertical axis represents the total number of movie genres recommended by the model for users of the corresponding gender (Female: 1356, Male: 3643).

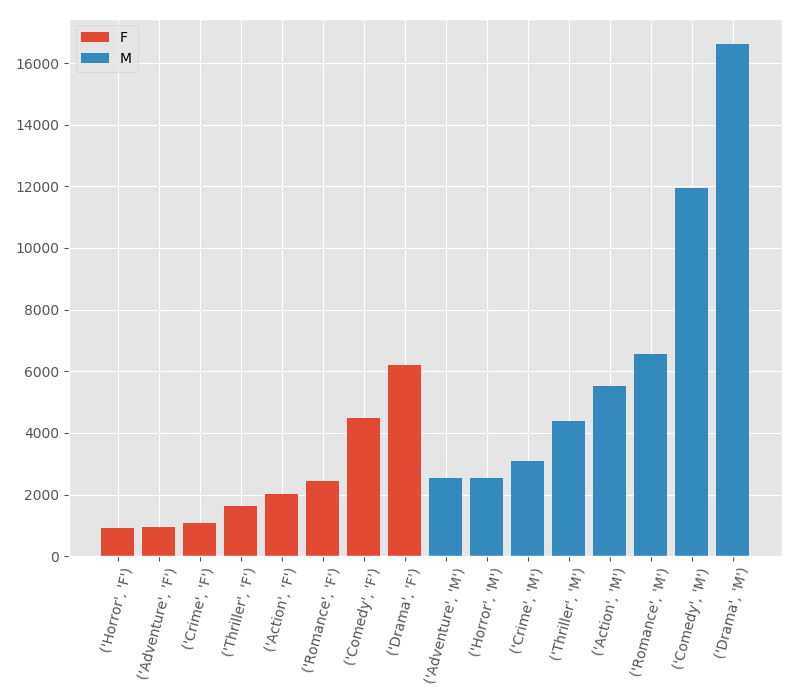


Figure 3. Dimension two - gender

We can see the top 6 movie genres recommended by the TF model remains the same although the gender is different. Only the top 7 and top 8 movie genres are different.

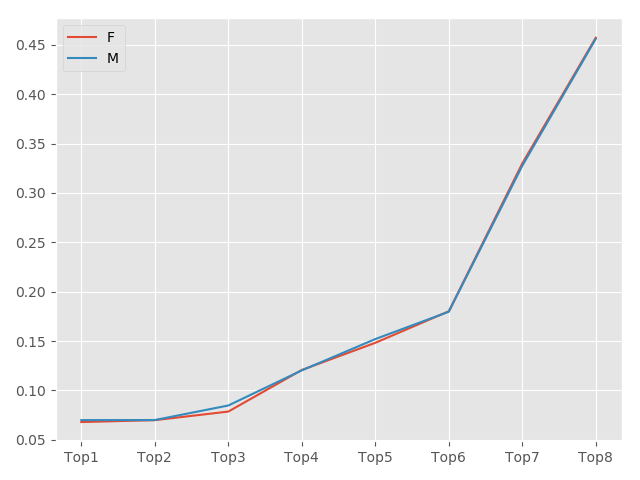


Figure 4. Dimension two – gender

The Figure 4. represents the proportion of users for top 8 genre recommended by TF model.

**Dimension three: occupation**

I also select the 5 most common occupations (0 – “other”, 1- “academic/educator”, 4 – “college/grad student”, 7 – “executive/managerial”, 17 – “technician/engineer”), and select top 3 movie genres recommended by the TF model. The visualization results are similar to the previous two dimensions.

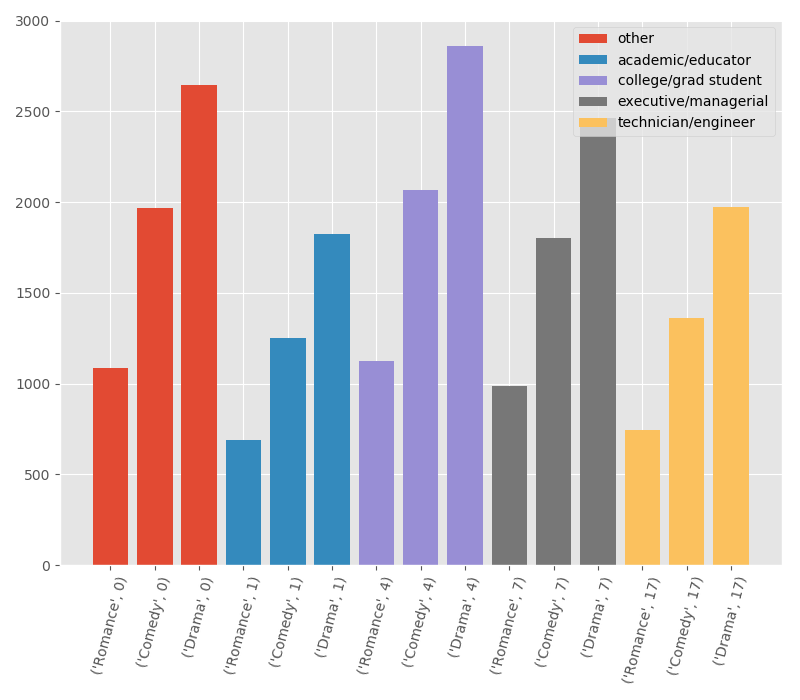
****

Figure 5. Dimension three – occupation

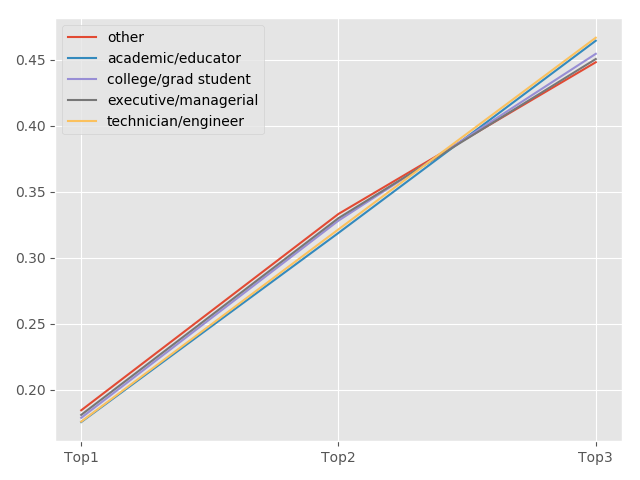


Figure 6. Dimension three - occupation

1. **Visualize the original data.**

For the comparison, I also visualize the number of rated movies in the different movie genres of original data and the corresponding average rating score.

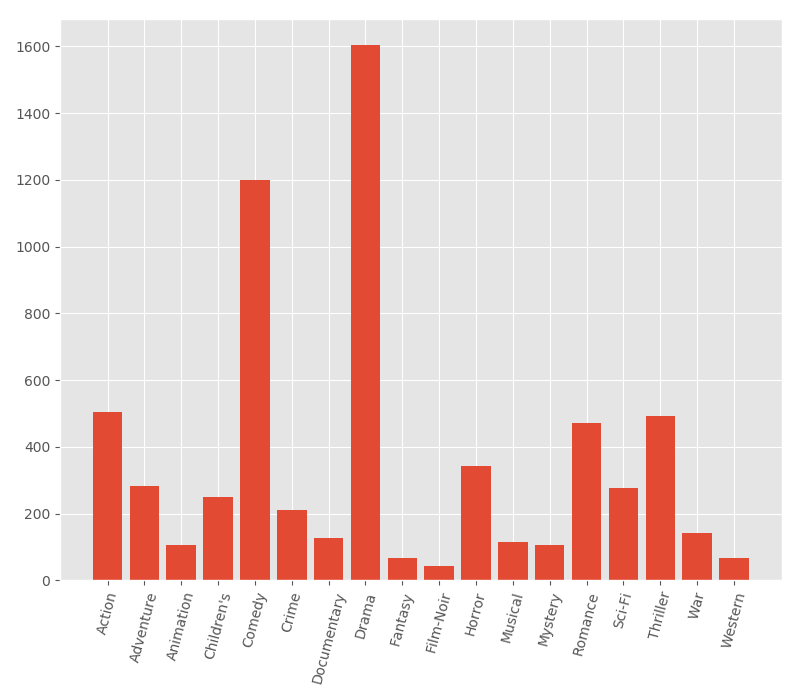
.

Figure 7. Dimension three - occupation

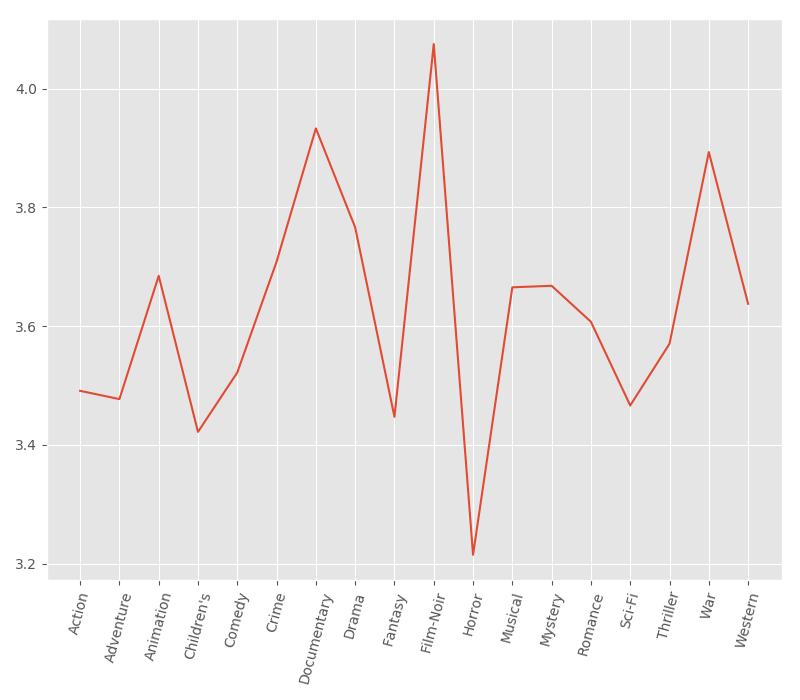
****

Figure 8. Dimension three – occupation

From Figure 7 and Figure 8, we can see that the genre of the most frequently recommended movie from TF model has a great relationship with the corresponding number of rated movies, but has little relationship with the corresponding rating score.

1. **Conclusion:**

From the results generated by three python programs, we can see that the performance of the TF model is not good. Under different demographic, the genres of movies that are recommended to users are still very similar.

The probable reason is that the TF model is trained to predict the rating score of each movie, rather than the top N task. Therefore, we can also see that there is a big difference between the rating regression task and the top N recommendation task.