Movie Recommendation Using Collaborative filtering using Netflix Data

By

Thrinath Nalajala

Collaborative Filtering:

Generally we get Recommendations when ever we watch movie on Netflix or by a product on Amazon. This is all because of the machine learning algorithms through which we are able to build this system.

Now we will use collaborative filtering which filters information and data that is collected from the other users.

Motivation:

I use Netflix a lot and get more recommendations whenever I open it, and interestingly they are all according to my choice.So I felt interested and want to know how it works,this motivated me to to undertake this project.

Recommender Systems:

It is a filtering system that predicts the rating or preference, which is given by user to an item or products like movies etc.

It is of two types:

- Collaboratory Filtering
- Content Based Filtering

Collaboratory Filtering :

User behaviour and item attributes are main concepts in Collaboratory Filtering.This method is done using these two concepts

Content Based Filtering:

Item Attributes is used by the Content based Filtering to find recommendations.

There are two types of Collaboratory Filterings.They are:’

1.User-Based

2.Item-Based

- User-Based is defined as the relationship between target user and other users.
- Whereas Item Based is defined as the relationship between item that targets user rate and other items.

Creating an EMR cluster:

1.Login into aws account

2.Choose Create cluster.

3.On the Create Cluster - Quick Options page, accept the default values except for the following fields:

4. Enter a Cluster name.

5.In Software configuration select the applications as spark.

6.Under Security and access, choose the EC2 key pair that you created in Create an Amazon EC2 Key Pair.

7.Choose Create cluster

Next create a jupyter Notebook by usimg the IPV4 address od the EC2 instance using putty and putty gen.

Add Security groups like 22 and 8888 in the inbound rules of EC2 instance.

The port numbers are used for:

22- to connect to SSH

8888- to connect to jupyter

And also create a S3 bucket and upload the data files, so that the address can be used in jupyter notebook.

Implementation:

1.import the packages

2.Import the Data

```
import os
import numpy as np
import pandas as pd

#location of data files
dbfs_dir = 's3://thrinathfinalp/'
movieTitles_filename = dbfs_dir + 'movie_titles.txt'
trainingRatings_filename = dbfs_dir + 'TrainingRatings.txt'
testingRatings_filename = dbfs_dir + 'TestingRatings.txt'
```

3.Find the mean, min ,max and avg mean rating

```
user_avg.describe('mean_Rating').show()
```

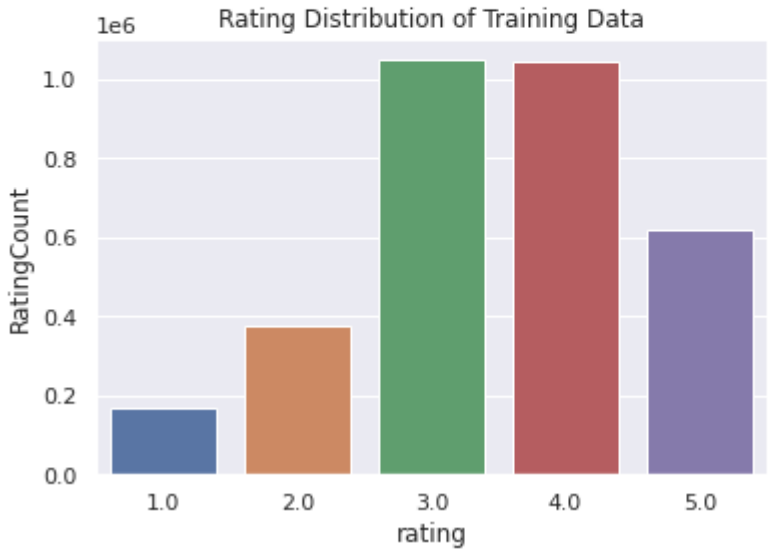
```
[Stage 61:=============================================>      (175 + 8) / 200]
```

```
+-------+------------------+
|summary|       mean_Rating|
+-------+------------------+
|  count|             28978|
|   mean| 3.5160354185440443|
| stddev|0.43468236507005115|
|    min|               1.0|
|    max|               5.0|
+-------+------------------+
```

4.Using Seaborn, I have made some visualization plots for training and also testing datasets

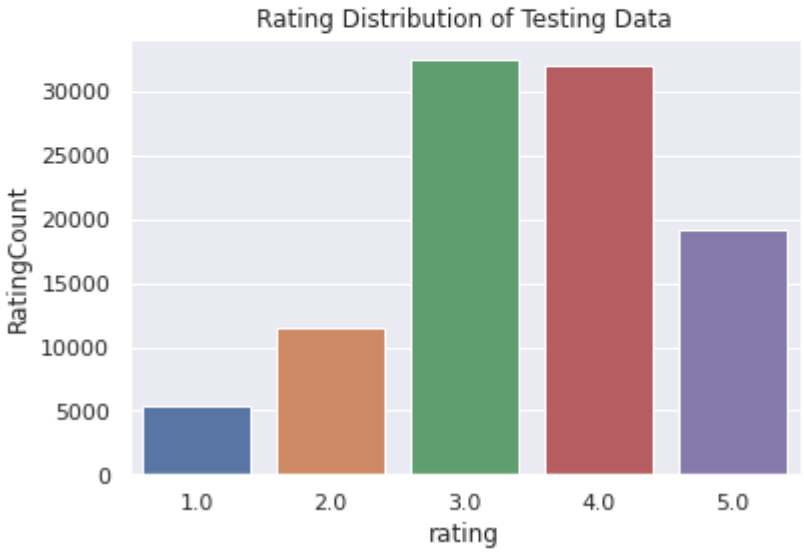
```
: import seaborn as sns
  sns.set_theme(style="darkgrid")
  ax = sns.countplot(x="rating", data=pandas_tr_df)
  ax.set_title('Rating Distribution of Training Data')
  ax.set_ylabel('RatingCount')
```

```
: Text(0, 0.5, 'RatingCount')
```



```
sns.set_theme(style="darkgrid")
ax = sns.countplot(x="rating", data=pandas_ts_df)
ax.set_title('Rating Distribution of Testing Data')
ax.set_ylabel('RatingCount')
```

```
Text(0, 0.5, 'RatingCount')
```



5.Collaborative Filtering Approach is applied

a. Item-Based:

```
#Item Based approach
sim_options = {
    'name': 'pearson',
    'user_based': 'False'
}

clf = KNNBasic(sim_options = sim_options)
cross_validate(clf, dataset, measures=['MAE'], cv=5, verbose=True)
```

b.User-Based:

```
#User Based approach
sim_options = {
    'name': 'MSD',
    'user_based': 'True'
}

clf = KNNBasic(sim_options = sim_options)
cross_validate(clf, dataset, measures=['MAE'], cv=5, verbose=True)
```

We also implemented ALS(Alternating Least Square) in this and found RMSE(Root Mean Square Error).

The RMSE which we got is 0.833000

```
predict_df = my_model.transform(test_df)
predicted_test_df = predict_df.filter(predict_df.prediction != float('nan'))
test_RMSE = reg_eval.evaluate(predicted_test_df)
print('The model had a RMSE on the test set of {0}'.format(test_RMSE))
```

```
The model had a RMSE on the test set of 0.833000416463466
```

Now I have also tested by using a New data which I took randomly and got my data combined with the actual dataset.

```
DataFrame[movieId: bigint, userId: bigint, rating: bigint]
```

movieId	userId	rating
240	0	5
290	0	5
890	0	5
100	0	5
100	0	5
523	0	5
219	0	5
229	0	5
209	0	5
320	0	2

Conclusion:

Collaborative Filtering provides a powerful way for data scientist to recommend new products or items to the users.It helped a lot for MNC’s like Netflix and Amazon by getting the recommendations about what user may need,It also capable to change the users data time to time.

Finally I conclude that movie recommendations really helped many people to find the movie of their choice without having to waste their time by scrolling everything.This also helped the search engine to get better.