MOVIE RECOMMENDATION USING MOVIELENS DATASET

Name: Rithwik S. Vedpathak   Email: vedpathakrithwik@gmail.com

Third Year, Computer,

Vidyavardhini's College of Engineering and Technology

ABSTRACT: With the amount of available online content ever-increasing and all the platforms trying to grab your attention by giving you personalized recommendations, recommendation engines are more important than ever. In this paper will create a recommendation system using Collaborative Filtering, which learns from past user behavior and give unique movie recommendations for every user based on their past ratings.

1. INTRODUCTION

Content based engine suffers from severe limitations. It is only capable of suggesting movies which are close to a certain movie. That is, it is not capable of capturing tastes and providing recommendations across genres.

Also, it doesn't capture the personal tastes and biases of a user.

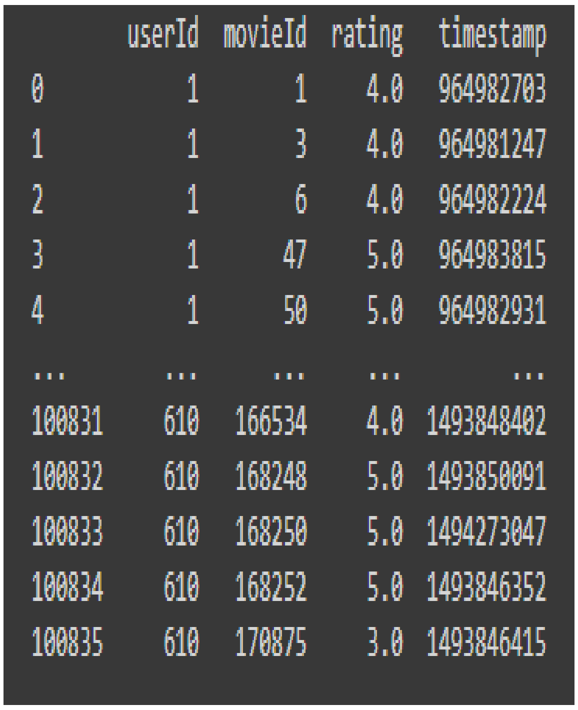
Collaborative Filtering is based on the idea that users similar to me can be used to predict how much I will like a particular product or service those users have used/experienced but I have not.

Will be using publicly available dataset to recommend a movie.

2. DATA COLLECTION

The dataset used for this model is provided on website named grouplens.org.

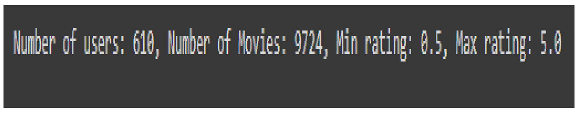
DATA DESCRIPTION


```
        userId  movieId  rating   timestamp
0            1        1     4.0   964982703
1            1        3     4.0   964981247
2            1        6     4.0   964982224
3            1       47     5.0   964983815
4            1       50     5.0   964982931
...        ...      ...     ...         ...
100831     610   166534     4.0  1493848402
100832     610   168248     5.0  1493850091
100833     610   168250     5.0  1494273047
100834     610   168252     5.0  1493846352
100835     610   170875     3.0  1493846415
```

The dataset (ml-latest-small) describes 5-star rating and free-test tagging activity from Movielens, a movie recommendation service. It contains 100826 ratings and 3683 tag applications across 9742 movies. These data were created by 610 users between March 29, 1996 and September 24, 2018.

Users were selected at random for inclusion. All selected users had at least 20 movies. No demographic information is included. Each user is represented by an id, and no other information is provided.

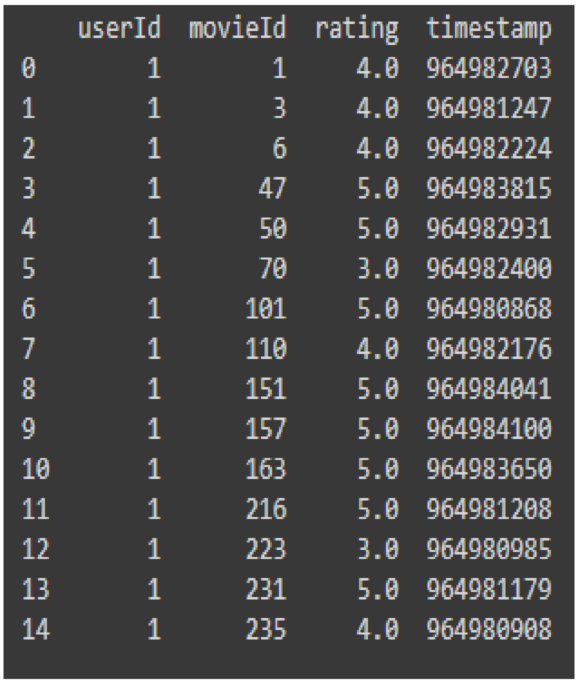
In this paper our aim is to study the ratings of various users in order to draw conclusions which user must be recommended what movie (ml-latest-small).



```
Number of users: 610, Number of Movies: 9724, Min rating: 0.5, Max rating: 5.0
```

CLEANING AND PREPARING DATA

Procuring the data and cleaning was not much of a big task as it was readily available without any invalid data.



```
    userId  movieId  rating  timestamp
0        1        1     4.0  964982703
1        1        3     4.0  964981247
2        1        6     4.0  964982224
3        1       47     5.0  964983815
4        1       50     5.0  964982931
5        1       70     3.0  964982400
6        1      101     5.0  964980868
7        1      110     4.0  964982176
8        1      151     5.0  964984041
9        1      157     5.0  964984100
10       1      163     5.0  964983650
11       1      216     5.0  964981208
12       1      223     3.0  964980985
13       1      231     5.0  964981179
14       1      235     4.0  964980908
```

3. MACHINE LEARNING MODELS

Will be using Model Based Collaborative filtering technique (Deep learning method), TensorFlow and Keras functional model to implement recommendation system.

COLLABORATIVE FILTERING

Collaborative filtering (CF) is a technique used by recommender systems.

It is a method of making automatic predictions about the interests of a user by collecting preferences or taste information from many users (in this case ratings). The underlying assumption of the collaborative filtering approach is that if a person A has the same opinion as a person B on an issue, A is more likely to have B's opinion on a different issue than that of a randomly chosen person.

This differs from the simpler approach of giving an average score for each item of interest, for example based on its number of votes.

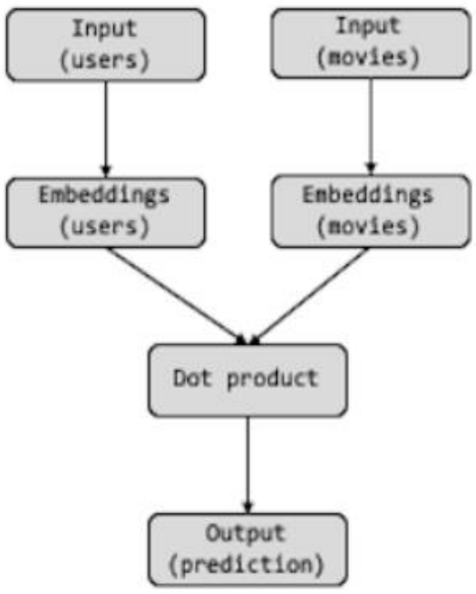
The motivation for collaborative filtering comes from the idea that people often get the best recommendations from someone with tastes similar to themselves. Collaborative filtering encompasses techniques for matching people with similar interests and making recommendations on this basis.

Workflow of a collaborative filtering system is:

1. A user expresses his or her preferences by rating movies. These ratings can be viewed as an approximate representation of the user's interest in the corresponding domain.
2. The ratings match this user's ratings against other user's and finds the people with most 'similar' tastes.
3. With similar users, the system recommends movies that the similar users have related highly but not yet being rated by this user.

NEURAL NETWORK APPROACH

For SVD or PCA, we decompose our original sparse matrix into product of 2 low rank orthogonal matrices. For neural net implementation, we don't need them to be orthogonal, we want our model to learn the values of embedding matrix itself. The user latent features and movie latent features are looked up from the embedding matrices for specific movie-user combination. These are the input values for further linear and non-linear layers. We pass this input to multiple layers and learn the corresponding weights by Adam optimization algorithm.



STEPS

- Create embeddings for both our users and movies.
- Dot product multiplies these matrices of embeddings.
- Use additional bias weights to account for user/product biases
- And we can extend this with typical deep learning layers (i.e., hidden layers, dropout, etc.)

TENSORFLOW

1. **tf.nn**: Provides support for many basic neural network operations(here sigmoid function)
2. **tf.tensordot:** Tensor contraction of user vector and movie vector along specified axes(2).

KERAS FUNCTIONAL MODELS

- Allows flexibility in creating custom models
- Use multiple inputs and subsequent layers along with multiple outputs
- Naming layers allows us to easily view the layer connections.
- Provides loss functions (binary cross entropy) which computes the quantity that my model should seek to minimize during training.
- He normal initializer draws sample from a truncated normal distribution centered on 0 with stddev = sqrt(2 / fan_in) where fan_in is the number of units in the weight tensor.

SIGMOID ACTIVATION

The function maps movie ratings into a value between 0 and 1

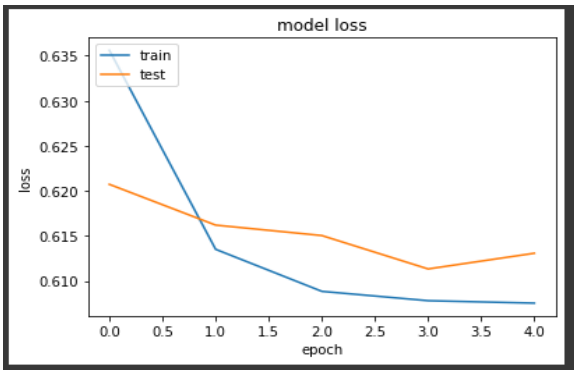
ADAM OPTIMIZATION

Adam is an adaptive learning rate optimization algorithm that’s been designed specifically for training deep neural networks. The algorithm leverages the power of adaptive learning rates methods to find individual learning rates for each parameter.

Its name is derived from adaptive moment estimation as it uses estimations of first(mean) and second (uncentered variance) moments of gradient to adapt the learning rate for each weight of the neural network.

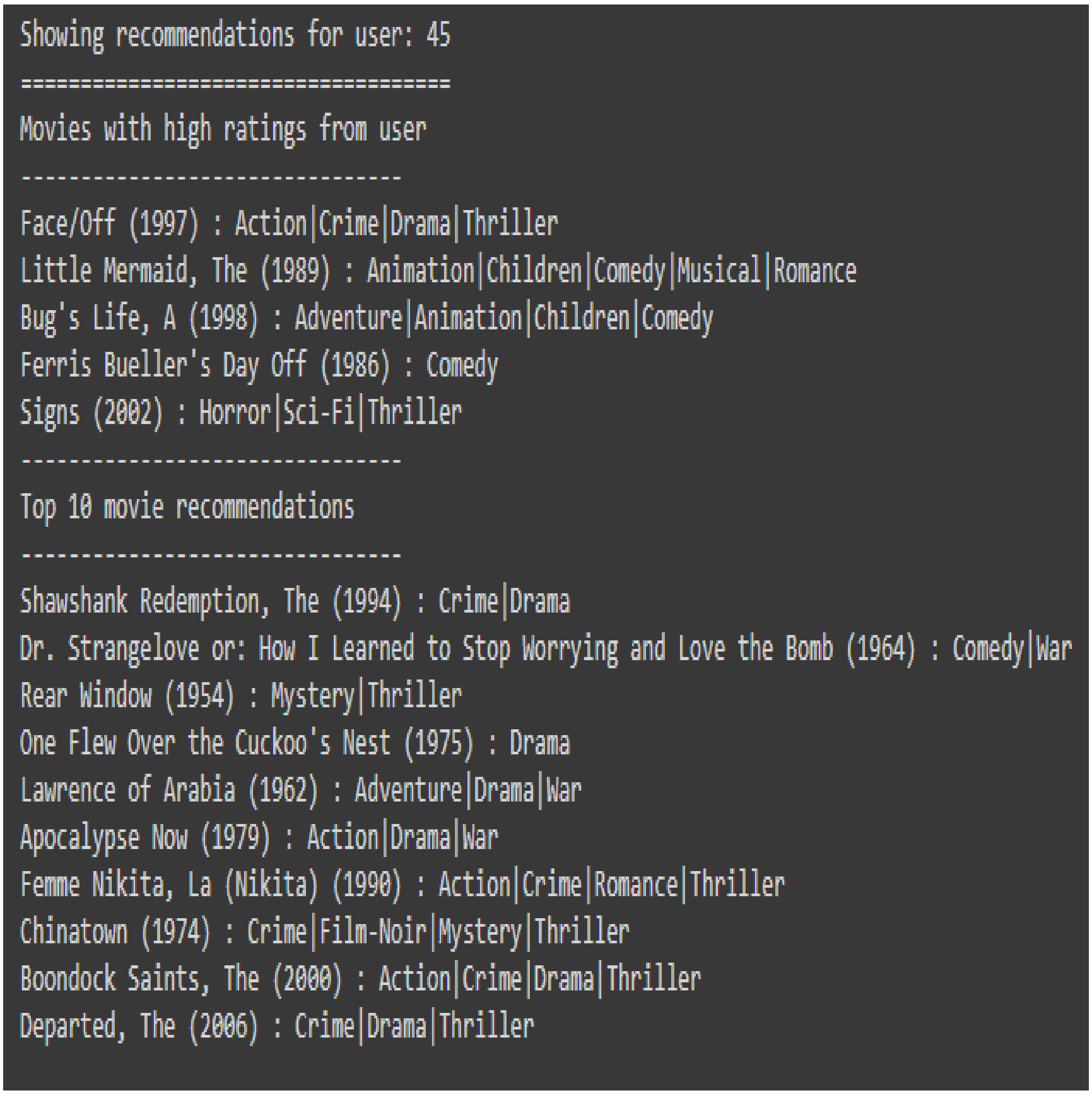
MATPLOTLIB

Used to plot the model loss during training for 5 epochs (epoch vs loss)



4.RESULT:

The model can help user discover new interests i.e., the system may not know the user is interested in a given movie, but the model might still recommend it because similar users are interested in that item. Below is showed top ten movie recommendations for user with id 45. The genre of the movies is similar to what the user had enjoyed watching earlier



```
Showing recommendations for user: 45
====================================
Movies with high ratings from user
--------------------------------
Face/Off (1997) : Action|Crime|Drama|Thriller
Little Mermaid, The (1989) : Animation|Children|Comedy|Musical|Romance
Bug's Life, A (1998) : Adventure|Animation|Children|Comedy
Ferris Bueller's Day Off (1986) : Comedy
Signs (2002) : Horror|Sci-Fi|Thriller
--------------------------------
Top 10 movie recommendations
--------------------------------
Shawshank Redemption, The (1994) : Crime|Drama
Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb (1964) : Comedy|War
Rear Window (1954) : Mystery|Thriller
One Flew Over the Cuckoo's Nest (1975) : Drama
Lawrence of Arabia (1962) : Adventure|Drama|War
Apocalypse Now (1979) : Action|Drama|War
Femme Nikita, La (Nikita) (1990) : Action|Crime|Romance|Thriller
Chinatown (1974) : Crime|Film-Noir|Mystery|Thriller
Boondock Saints, The (2000) : Action|Crime|Drama|Thriller
Departed, The (2006) : Crime|Drama|Thriller
```

5. CONCLUSION: A model-based approach Collaborative filtering is taken to implement the system. We don’t need domain knowledge because embeddings are automatically learned as the system needs only the feedback matrix to train. In particular, the system doesn’t need contextual features.

6. FUTURE WORK:

It works purely on the basis of assigned movie ratings by the users and tries to predict the movies based on how the other users have rated the movie. This approach has a short coming of data sparseness and cold start problem. Hence, to improve the accuracy a hybrid approach can be taken between context based filtering and collaborative filtering.