Movie Recommendations System

By:-

Toluwani Oyewusi (**A20425821**)

Vimarsh Patel (**A20513290**)

Problem Statement

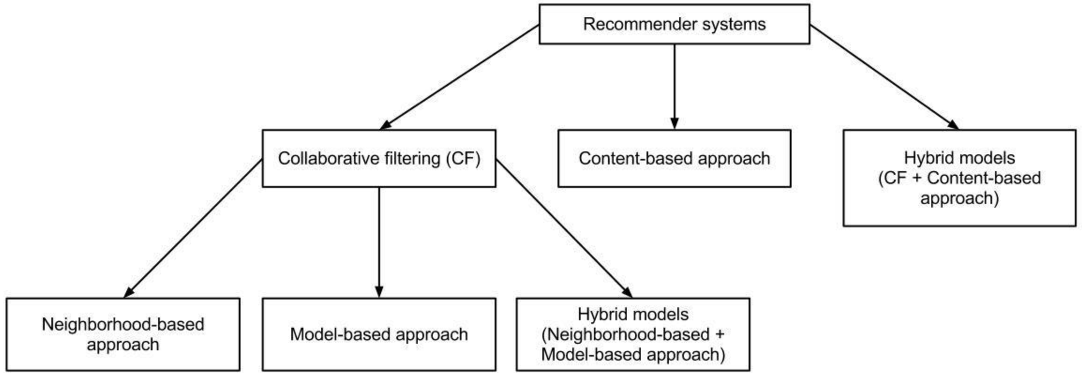
- A large number of movies are made everyday and released in theatres for the public to watch. Everyone does not like the same kind of movies to watch and have different taste. Our aim with this project is to give movie recommendation to the user so they have a better knowledge of the movies they want to see. Recommendation systems make people's life easier by finding new movies the user likes to watch.

Back Ground Material

- Name of the Paper: Movie Recommendations Using the Deep Learning Approach
- Code Implementation: Github
- Software Libraries used:
  - Numpy
  - Pandas
  - TensorFlow
  - Keras
  - Matplotlib
  - SkLearn

Recommendation Systems

- Our life would be easier if our choices were made instantly for whatever purposes.
- Movie recommendation systems help us give suggestions on what to watch. There are numerous ways to build such systems.
- Some of them are:-
- Collaborative Filtering : is the method which is applied the most and easy to implement
- Autoencoders : is a method with deep learning but expensive to perform.
- Deep Learning : approaches have been gaining traction in the recommender system arena as a result of their cutting-edge performances and superior recommendation quality.



Recommender systems

Collaborative filtering (CF)

Content-based approach

Hybrid models (CF + Content-based approach)

Neighborhood-based approach

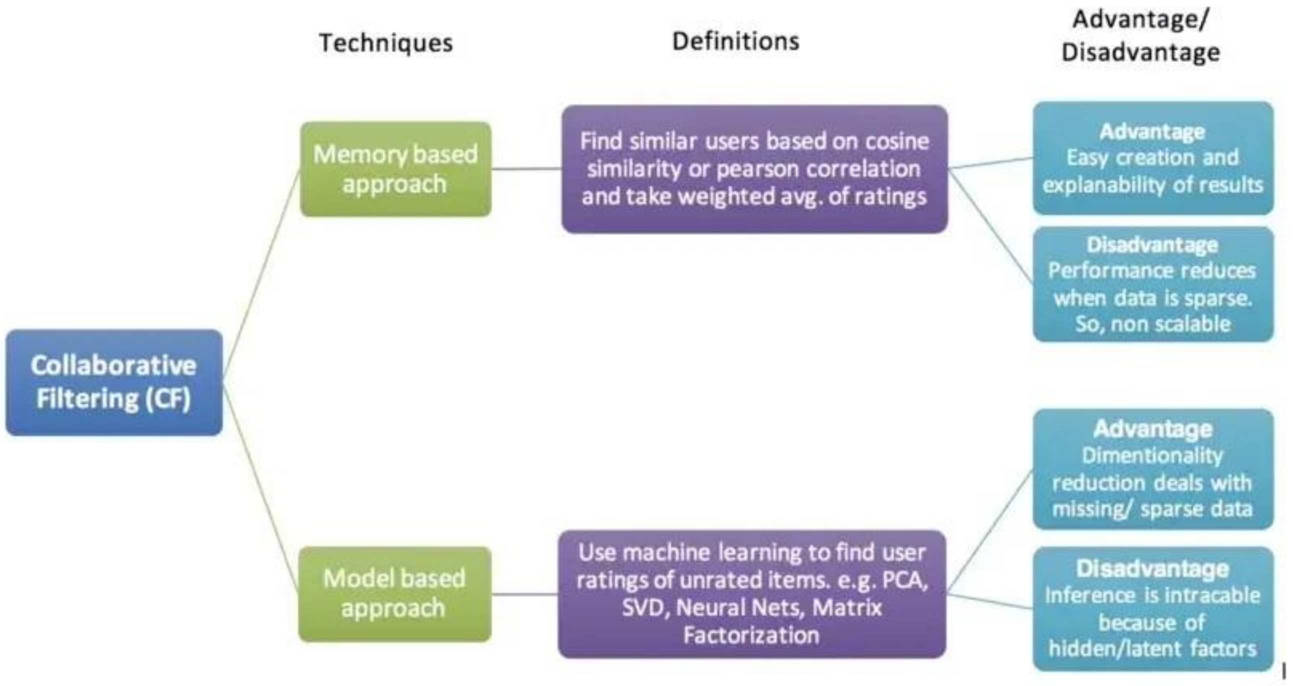
Model-based approach

Hybrid models (Neighborhood-based + Model-based approach)

Collaborative filtering

- Collaborative filtering is also known as social filtering. Collaborative filtering uses algorithms to filter data from user reviews to make personalized recommendations for users with similar preferences. Collaborative filtering is also used to select content and advertising for individuals on social media.
- Collaborative filtering refers to the use of video ratings from specific users or similar users to predict and recommend content that those users are most likely to enjoy. This approach separates and groups users based on similarities and differences in video preferences. It is interesting to dig into because content watched by a specific user can be recommended to a user with similar preferences who has not watched that content yet

- There are numerous ways to perform collaborative filtering:-



Techniques

Definitions

Advantage/ Disadvantage

Collaborative Filtering (CF)

Memory based approach

Find similar users based on cosine similarity or pearson correlation and take weighted avg. of ratings

Advantage
Easy creation and explanability of results

Disadvantage
Performance reduces when data is sparse. So, non scalable

Model based approach

Use machine learning to find user ratings of unrated items. e.g. PCA, SVD, Neural Nets, Matrix Factorization

Advantage
Dimentionality reduction deals with missing/ sparse data

Disadvantage
Inference is intracable because of hidden/latent factors

Deep Learning

- We have tried to implement With Deep Learning in a MultiLayer Perceptron. This system implementation is where deep learning shines because it is optimized for predictive modeling and statistics. Deep learning provides a better understanding of users' preferences, demands and relationships (similarities and differences) with each other.
- Our aim with the model is that we will input the user and the ratings of the movies he has seen. Our model will calculate the ratings of the movie the user has not seen. It will generate a preference score depending on the ratings of the similar movies given by the user and the ratings given to the movie by other users. Our model makes a matrix R and gives the ratings to all the movies with missing values for that particular user.

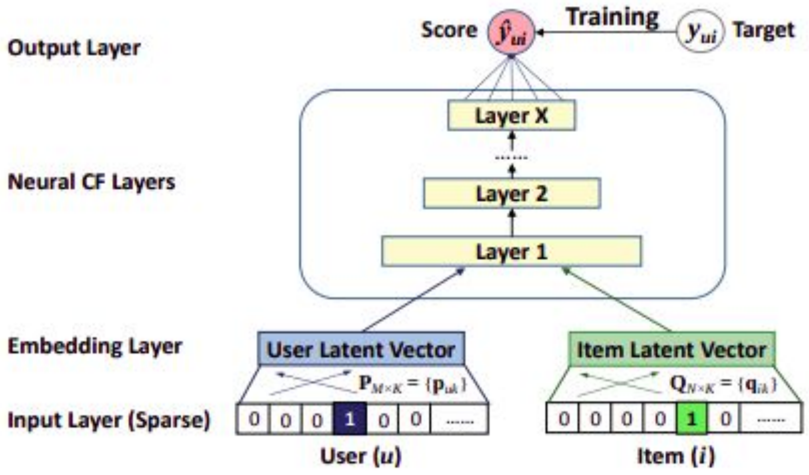
Implementation Details

- Dataset : We have used the MovieLens 100k Dataset which contains 100,000k ratings from 1000 users on1700 movies.
- We consider the user id, movie id, movie name and the ratings of the movie as features.
- We take all these columns and make a dataset. We encode the labels, split them into training and test datasets.
- Model Structure:
- We have 2 input for user and movie to the model
- Embedding Layers for both : turn each word into a fixed length vector of defined size.
- Dropout rate of 0.5: To reduce overfitting by randomly assigns inputs to 0
- 3 dense layers of dimensions 256,512,1024 with ReLu activation have been added.
- L2 Regularization layer with dimension 256 has been used to reduce overfitting.

- Root Mean Square Error is taken as the loss function as we are doing a simple regression analysis.

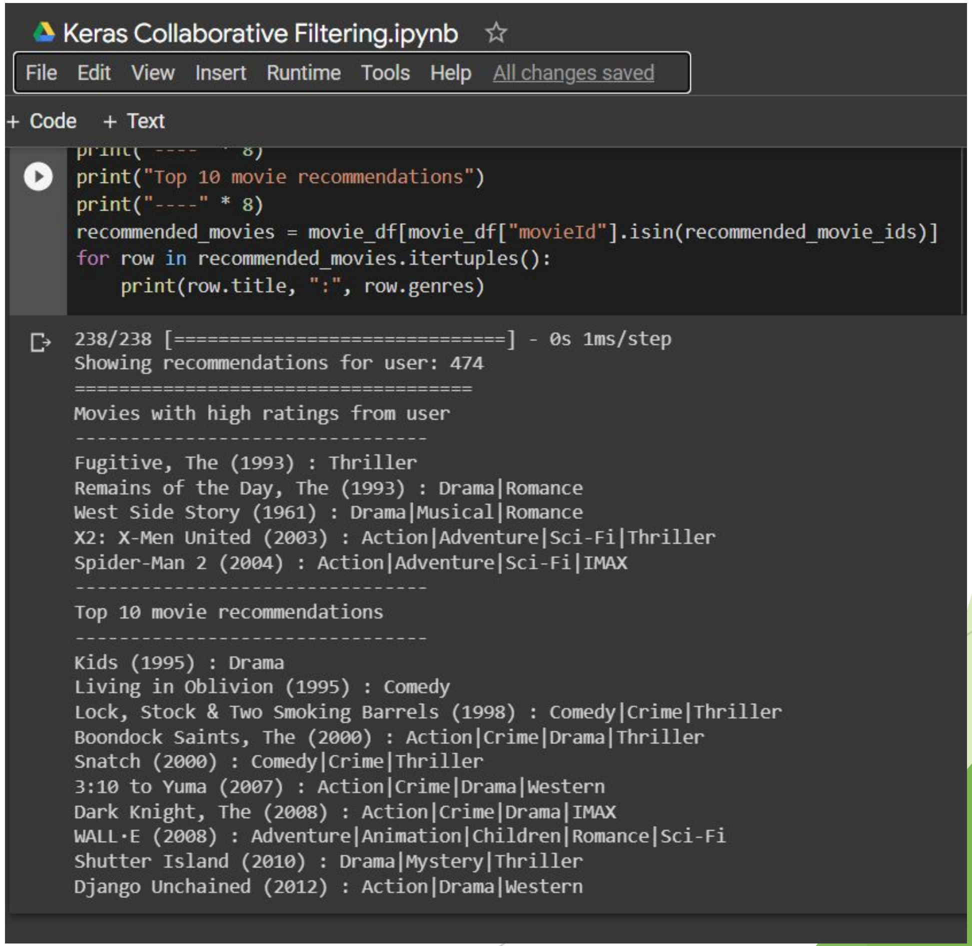
- Our output layer is passed with SoftMax activation

- Finally our model is trained on SGD optimizer.



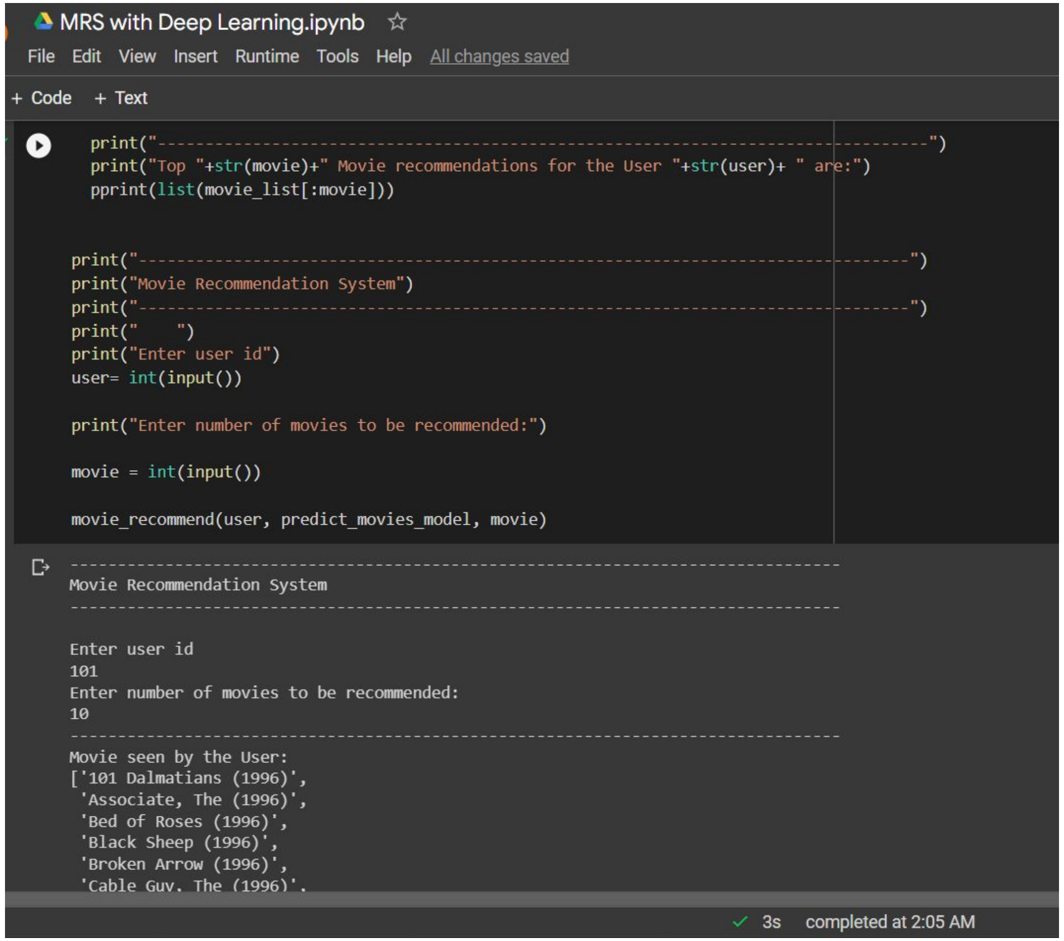
Results

- We are successfully able to predict a list of unseen movies for a specific user.
- We have also implemented the keras collaborative filtering approach with deep learning. We predicted a list of movies from that method as well.



```
print("----" * 8)
print("Top 10 movie recommendations")
print("----" * 8)
recommended_movies = movie_df[movie_df["movieId"].isin(recommended_movie_ids)]
for row in recommended_movies.itertuples():
    print(row.title, ":", row.genres)
```

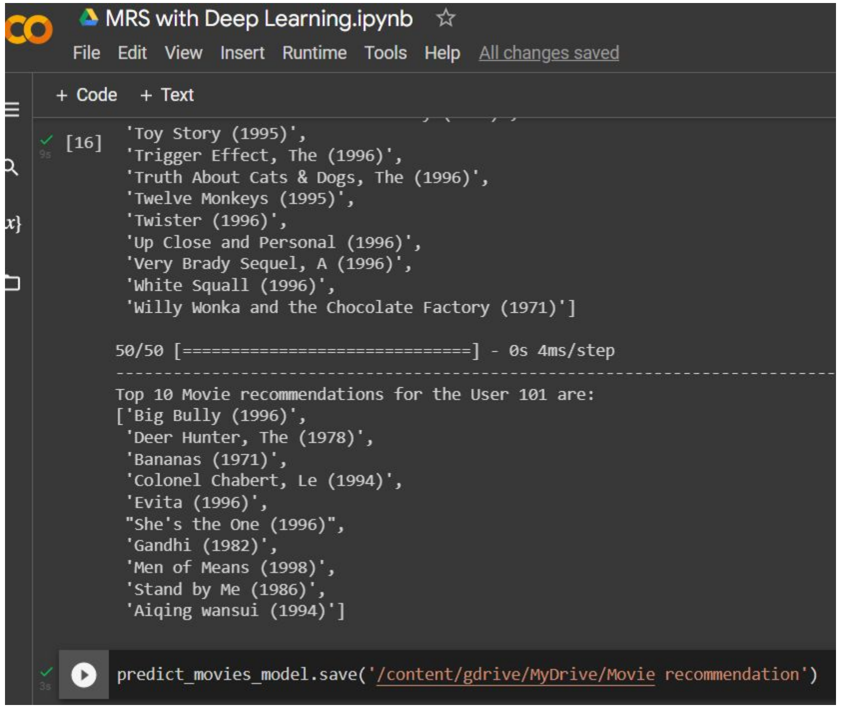
```
238/238 [==============================] - 0s 1ms/step
Showing recommendations for user: 474
====================================
Movies with high ratings from user
--------------------------------
Fugitive, The (1993) : Thriller
Remains of the Day, The (1993) : Drama|Romance
West Side Story (1961) : Drama|Musical|Romance
X2: X-Men United (2003) : Action|Adventure|Sci-Fi|Thriller
Spider-Man 2 (2004) : Action|Adventure|Sci-Fi|IMAX
--------------------------------
Top 10 movie recommendations
--------------------------------
Kids (1995) : Drama
Living in Oblivion (1995) : Comedy
Lock, Stock & Two Smoking Barrels (1998) : Comedy|Crime|Thriller
Boondock Saints, The (2000) : Action|Crime|Drama|Thriller
Snatch (2000) : Comedy|Crime|Thriller
3:10 to Yuma (2007) : Action|Crime|Drama|Western
Dark Knight, The (2008) : Action|Crime|Drama|IMAX
WALL·E (2008) : Adventure|Animation|Children|Romance|Sci-Fi
Shutter Island (2010) : Drama|Mystery|Thriller
Django Unchained (2012) : Action|Drama|Western
```


MRS with Deep Learning.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text

```
    print("--------------------------------------------------------------------------------")
    print("Top "+str(movie)+" Movie recommendations for the User "+str(user)+ " are:")
    pprint(list(movie_list[:movie]))


print("--------------------------------------------------------------------------------")
print("Movie Recommendation System")
print("--------------------------------------------------------------------------------")
print("    ")
print("Enter user id")
user= int(input())

print("Enter number of movies to be recommended:")

movie = int(input())

movie_recommend(user, predict_movies_model, movie)
```

```
--------------------------------------------------------------------------------
Movie Recommendation System
--------------------------------------------------------------------------------

Enter user id
101
Enter number of movies to be recommended:
10
--------------------------------------------------------------------------------
Movie seen by the User:
['101 Dalmatians (1996)',
 'Associate, The (1996)',
 'Bed of Roses (1996)',
 'Black Sheep (1996)',
 'Broken Arrow (1996)',
 'Cable Guy, The (1996)',
```

✓ 3s completed at 2:05 AM


MRS with Deep Learning.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[16]

```
 'Toy Story (1995)',
 'Trigger Effect, The (1996)',
 'Truth About Cats & Dogs, The (1996)',
 'Twelve Monkeys (1995)',
 'Twister (1996)',
 'Up Close and Personal (1996)',
 'Very Brady Sequel, A (1996)',
 'White Squall (1996)',
 'Willy Wonka and the Chocolate Factory (1971)']

50/50 [==============================] - 0s 4ms/step
--------------------------------------------------------------------------------
Top 10 Movie recommendations for the User 101 are:
['Big Bully (1996)',
 'Deer Hunter, The (1978)',
 'Bananas (1971)',
 'Colonel Chabert, Le (1994)',
 'Evita (1996)',
 "She's the One (1996)",
 'Gandhi (1982)',
 'Men of Means (1998)',
 'Stand by Me (1986)',
 'Aiqing wansui (1994)']
```

```
predict_movies_model.save('/content/gdrive/MyDrive/Movie recommendation')
```

Thank You