Presenters:
Anish Gadekar, Parth Shah, Shruti Bendale

Introduction

- Matching consumers with the most appropriate products is key to enhancing user satisfaction and loyalty.

- Therefore, more retailers have become interested in recommender systems, which analyse patterns of user interest in products to provide personalized recommendations that suit a user's taste.

Recommender System Approaches

COLLABORATIVE FILTERING

CONTENT-BASED FILTERING

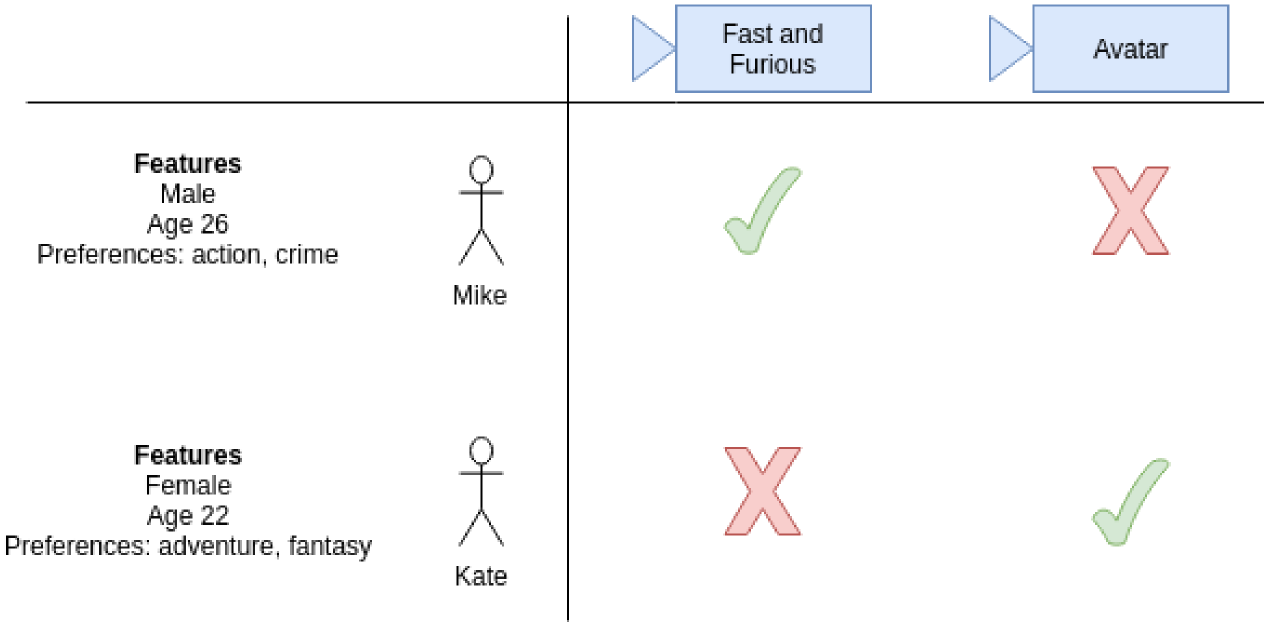
1. Collaborative Filtering:

- This approach is based on the past interactions between users and the target items.
- The input to a collaborative filtering system will be all historical data of user interactions with target items.
- This data is typically stored in a matrix where the rows are the users, and the columns are the items.




2. Content-based Filtering:

- Works with data that the user provides, either explicitly (rating) or implicitly (clicking on a link).
- Based on that data, a user profile is generated, which is then used to make suggestions to the user.



MovieLens Dataset

- ➢20 million ratings
- ➢465,000 tags applications applied to 27,000 movies by 138,000 users

ratings.csv

	userId	movieId	rating	timestamp
0	1	31	2.5	1260759144
1	1	1029	3.0	1260759179
2	1	1061	3.0	1260759182
3	1	1129	2.0	1260759185
4	1	1172	4.0	1260759205

movies.csv

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

Performance Metrics

MAE

RMSE

MEAN ABSOLUTE ERROR

$$\text{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

ROOT MEAN SQUARED ERROR

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

Recommender using Collaborative Filtering

Approach 1:

Stochastic Gradient Descent as a learning algorithm

Overview

- We used stochastic gradient descent for optimization of the algorithm by looping through all ratings in the training set.
- The system predicts the rating of user u for the user i (r_{ui}) and computes the associated prediction error:

$$e_{ui} \overset{def}{=} r_{ui} - q_i^T p_u.$$

- The parameters are then modified by a magnitude proportional to γ in the opposite direction of the gradient, yielding:

$$q_i \leftarrow q_i + g\,(e_{ui}\, p_u - \lambda\, q_i)$$

$$p_u \leftarrow p_u + g\,(e_{ui}\, q_i - \lambda\, p_u)$$

- RMSE after training for 100 epochs: **0.983**

Approach 2:

Deep Learning as a learning algorithm

A high-level design for the flow of the program and the model:

1. **Pre-processing the data:**
   - This includes looking at the shape of the data, removing inconsistencies and garbage values.
2. **Splitting the data into train and test:**
   - We made 80-20 split, using the 'sklearn.train_test_split' library.
3. **Defining the Model**
   - Creating separate embeddings for the Movies matrix and the Users Matrix.
   - Merging the embedded matrices using a dot product.
   - Experimenting with various deep learning layers such a Dense Layers, Batch Normalization, Dropout after the dot product of the matrices. (In a combination that gives the best results)
4. **Deciding parameters such as hidden layers, Dropout, Learning Rate, Optimizers, Epochs.**
5. **Training the model**
6. **Using the trained model for predictions**

Model Summary

Experiments

- We experimented with different values of number of hidden layers, dropout values, optimizers, learning rates.
- We found out that the combination of 1 hidden layer with 50 nodes, 20% dropout, Adam optimizer, 0.001 learning rate and 100 epochs gives the best value of root mean squared error (**0.81**).

No. of Hidden Layers	Dropout Values	Optimization	Learning Rate	Epochs	RMSE
1 (50 nodes)	0.2	sgd	0.001	100	1.099
1 (50 nodes)	0.2	RMSProp	0.001	100	1.118
1 (50 nodes)	0.2	Adam	0.001	100	0.81
1 (50 nodes)	0.2	Adadelta	0.001	100	13.313
1 (50 nodes)	0.2	Adagrad	0.001	100	9.2377
1 (100 nodes)	0.5	Adam	0.001	100	1.372
1 (100 nodes)	0.2	Adam	0.01	100	3.0139
1 (100 nodes)	0.2	Adam	0.001	100	1.984
2 (50 nodes each)	0.2, 0.2	Adam	0.001	100	0.918
2 (100 nodes each)	0.2, 0.2	Adam	0.001	100	0.926

Results

- We used matrix factorization and keras layers to train a deep learning model for our recommendation system.
- Once the model is trained, the system can show the Top N Recommended movies for an input userID.
- In the attached screenshot, we get the top 15 recommended movies for the userID ‘1’.

```
topPredictions(1,rating_data,movie_data)
```

	userId	movieId	rating	prediction	title	genres
0	1	4993	5.0	4.527629	Lord of the Rings: The Fellowship of the Ring,...	Adventure|Fantasy
1	1	7153	5.0	4.516778	Lord of the Rings: The Return of the King, The...	Action|Adventure|Drama|Fantasy
2	1	5952	5.0	4.464870	Lord of the Rings: The Two Towers, The (2002)	Adventure|Fantasy
3	1	1196	4.5	4.249328	Star Wars: Episode V - The Empire Strikes Back...	Action|Adventure|Sci-Fi
4	1	541	4.0	4.208074	Blade Runner (1982)	Action|Sci-Fi|Thriller
5	1	3265	3.5	4.175037	Hard-Boiled (Lat sau san taam) (1992)	Action|Crime|Drama|Thriller
6	1	293	4.0	4.140035	Léon: The Professional (a.k.a. The Professiona...	Action|Crime|Drama|Thriller
7	1	2628	4.0	4.102765	Star Wars: Episode I - The Phantom Menace (1999)	Action|Adventure|Sci-Fi
8	1	260	4.0	4.102588	Star Wars: Episode IV - A New Hope (1977)	Action|Adventure|Sci-Fi
9	1	8368	4.0	4.100841	Harry Potter and the Prisoner of Azkaban (2004)	Adventure|Fantasy|IMAX
10	1	1214	4.0	4.078115	Alien (1979)	Horror|Sci-Fi
11	1	6539	4.0	4.061373	Pirates of the Caribbean: The Curse of the Bla...	Action|Adventure|Comedy|Fantasy
12	1	4306	4.0	4.051948	Shrek (2001)	Adventure|Animation|Children|Comedy|Fantasy|Ro...
13	1	8961	4.0	4.047541	Incredibles, The (2004)	Action|Adventure|Animation|Children|Comedy
14	1	7438	4.0	4.037704	Kill Bill: Vol. 2 (2004)	Action|Drama|Thriller
15	1	3030	3.0	4.017904	Yojimbo (1961)	Action|Adventure

Recommender using Content-based Filtering

A high-level design for the flow of the program and the model:

1. **Pre-processing the data:**

This includes looking at the shape of the data, removing inconsistencies, garbage values, etc.

2. **Splitting the data into train and test:**

We made 80-20 split, using the ‘sklearn.train_test_split’ library.

3. **Defining the Model:**

- We use the MovieLens ratings data where each tag is typically a single word or short phrase. The meaning, value, and purpose of a particular tag is determined by each user.
- We compute the TF-IDF matrix of user ratings data using 'sklearn.feature_extraction.text' TfidfVectorizer.
- We define an AutoEncoder model which gives an Encoding size of 100 with intermediate layers of size 1000.

4. **Deciding parameters such as hidden layers, Dropout, Learning Rate, Optimizers, Epochs.**
5. **Training the Model**
6. **Using the trained model for predictions**

Methodology

- We train an Autoencoder using the TF-IDF values obtained from user rating data.
- After the Autoencoder converges, we obtain the encoded word embeddings from the Encoder half of the model and compute Cosine Similarity within the Embeddings.
- We then query this Cosine Similarity Matrix with the Movie Id as per the dataset and give back top 20 movies for a given user based on the ratings alone.

Observations

- We evaluate the top 20 movies returned for the movies Toy Story and Golden Eye.
- We notice that a considerable number of movies adhere to Toy Story's Adventure, Comedy and Fantasy genres.

	movieId	title	genres	similarity_score
1	2	Jumanji (1995)	Adventure|Children|Fantasy	1.000000
2473	2558	Forces of Nature (1999)	Comedy|Romance	0.855235
10035	33085	Amityville Horror, The (2005)	Horror|Thriller	0.821601
16927	85624	Bohemian Eyes (Boheemi elää - Matti Pellonpää)...	Documentary	0.818860
19244	95670	Tyler Perry's Madea's Witness Protection (2012)	Comedy	0.805679
10556	40015	Aprile (1998)	Comedy	0.801656
7721	8255	Chaos (Kaosu) (1999)	Crime|Mystery|Thriller	0.800484
2027	2111	Man with Two Brains, The (1983)	Comedy	0.796281
2940	3026	Slaughterhouse (1987)	Comedy|Horror	0.790033
3181	3268	Stop! Or My Mom Will Shoot (1992)	Action|Comedy	0.788707
15183	77427	Human Centipede, The (First Sequence) (2009)	Horror	0.787611
4390	4485	Casual Sex? (1988)	Comedy	0.780500
1053	1075	Sexual Life of the Belgians, The (Vie sexuelle...	Comedy|Romance	0.780202
3331	3420	...And Justice for All (1979)	Drama|Thriller	0.778170
16976	86014	Diary of a Wimpy Kid: Rodrick Rules (2011)	Comedy	0.775483

Observations

- However, the model does not provide movies such as other Disney/Pixar/Animated movies or movies which are sequels or prequels to Toy Story which can be safely deemed similar to movies.
- We also notice some outliers such as Horror movies and R rated movies which are not pertinent to our query.

	movieId	title	genres	similarity_score
1	2	Jumanji (1995)	Adventure|Children|Fantasy	1.000000
2473	2558	Forces of Nature (1999)	Comedy|Romance	0.855235
10035	33085	Amityville Horror, The (2005)	Horror|Thriller	0.821601
16927	85624	Bohemian Eyes (Boheemi elää - Matti Pellonpää)...	Documentary	0.818860
19244	95670	Tyler Perry's Madea's Witness Protection (2012)	Comedy	0.805679
10556	40015	Aprile (1998)	Comedy	0.801656
7721	8255	Chaos (Kaosu) (1999)	Crime|Mystery|Thriller	0.800484
2027	2111	Man with Two Brains, The (1983)	Comedy	0.796281
2940	3026	Slaughterhouse (1987)	Comedy|Horror	0.790033
3181	3268	Stop! Or My Mom Will Shoot (1992)	Action|Comedy	0.788707
15183	77427	Human Centipede, The (First Sequence) (2009)	Horror	0.787611
4390	4485	Casual Sex? (1988)	Comedy	0.780500
1053	1075	Sexual Life of the Belgians, The (Vie sexuelle...	Comedy|Romance	0.780202
3331	3420	...And Justice for All (1979)	Drama|Thriller	0.778170
16976	86014	Diary of a Wimpy Kid: Rodrick Rules (2011)	Comedy	0.775483

Comparisons and conclusions

Comparing the models

- **Comparing our results to the benchmark test results** for the MovieLens dataset published by the developers of the Surprise library (A python scikit for recommender systems) in the adjoining table.
- We can see that the **deep learning algorithm performs better** than the other algorithms, but it **takes a long time to train**.
- The deep learning algorithm is also **scalable** to a larger dataset without affecting the RMSE value.

MovieLens	RMSE	MAE
k-NN	1.004	0.744
Centered k-NN	0.968	0.749
k-NN Baseline	0.947	0.743
Co-clustering	0.993	0.753
SVD	0.956	0.737
SVD ++	0.941	0.722
Collaborative(SGD)	**0.983**	**0.739**
Collaborative(Deep Learning)	**0.817**	**0.715**
Content-based	**0.993**	**0.752**

References

- https://www.kdnuggets.com/2019/09/machine-learning-recommender-systems.html
- https://towardsdatascience.com/creating-a-hybrid-content-collaborative-movie-recommender-using-deep-learning-cc8b431618af
- https://nipunbatra.github.io/blog/2017/recommend-keras.html
- https://towardsdatascience.com/deep-autoencoders-for-collaborative-filtering-6cf8d25bbf1d
- https://medium.com/@connectwithghosh/recommender-system-on-the-movielens-using-an-autoencoder-using-tensorflow-in-python-f13d3e8d600d
- https://medium.com/@connectwithghosh/recommender-system-on-the-movielens-using-an-autoencoder-using-tensorflow-in-python-f13d3e8d600d