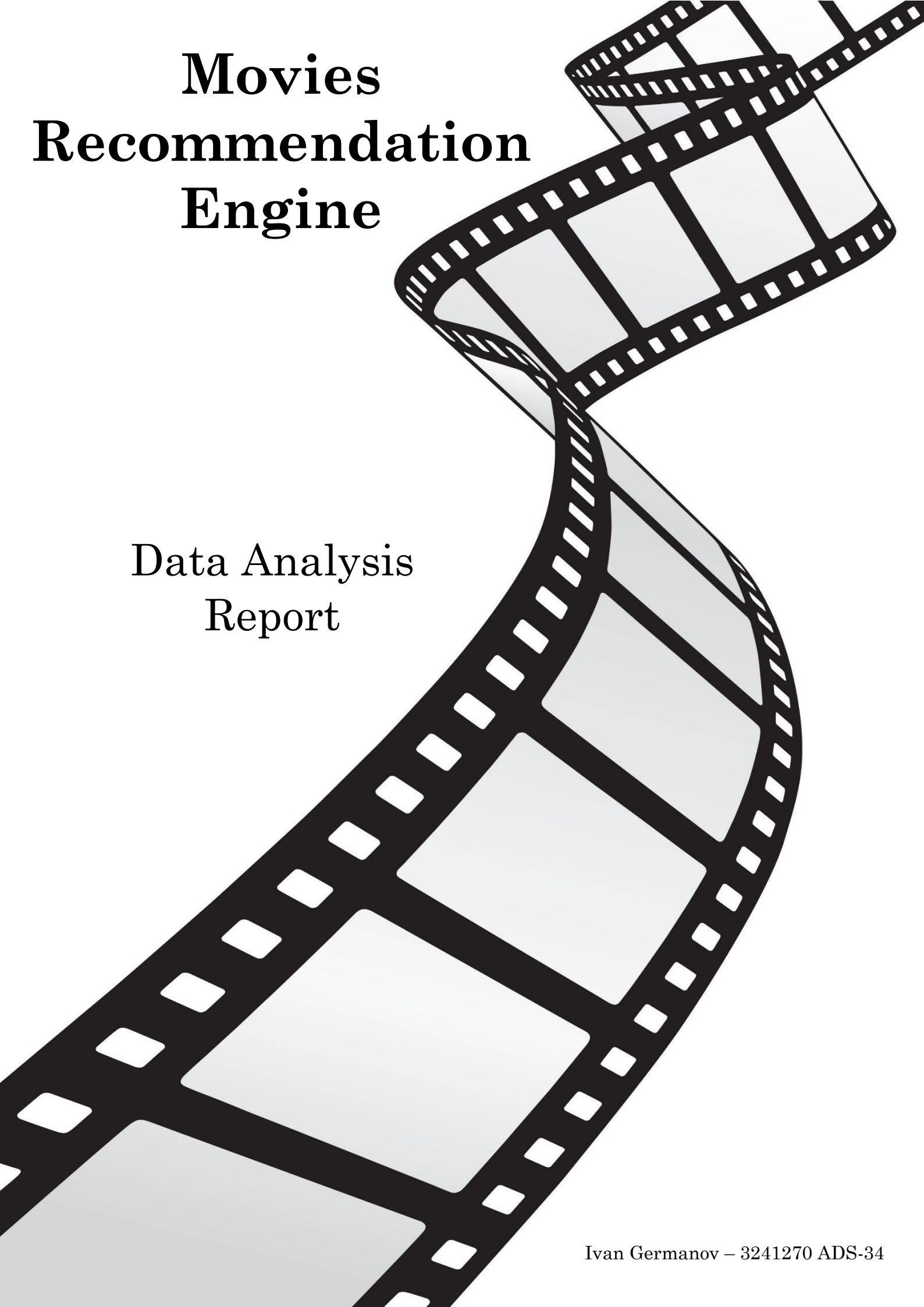
Movies Recommendation Engine



Data Analysis Report

Ivan Germanov – 3241270 ADS-34

DATA

The 3 provided datasets by MovieLens have data about users and movies together with the ratings of the users for those movies. The most relevant features were the rating score users give to movies. The exploration revealed that there are 943 users giving 100000 ratings to 1682 movies. It is important to note, however, that not each movie was rated by each user. In fact, that is a rare occurrence. Ratings are based on a 1-5 scale. Each movie is assigned an ID and has data about its title and genre. Each user is also assigned an ID and has information about his/her age, sex and occupation. There are more features which are irrelevant in this case. In addition, many movies have very few ratings.

METHODS AND ANALYSIS

Since the provided datasets are 3, there was a need to merge them. The first thing which was done was to explore the datasets by viewing their sizes, unique values, averages of features, etc. That was done in order to get a grasp of the datasets.

The data was analyzed, aggregated, plotted using Python and related machine learning libraries (pandas, NumPy, matplotlib, SeaBorn, TuriCreate).

Then an assumption was made that the more ratings a movie has received, the higher the rating of the movie is. The validity of this assumption was examined by the use of a Pearson Correlation.

Afterwards, **correlation** was used once again, this time as a **similarity metric between different movies**, based on their rating. For this model it was observed that many movies have very few ratings. Consequently, for the correlation only the movies with more than 50 ratings were used.

Furthermore, a second recommendation model was created, which is **based on the popularity of a movie**, i.e. the rating. This is useful for new users.

Additionally, a third more sophisticated recommendation model was created, called a **collaborative-filtering model** (Figure 1). The more data about a user, the better this model becomes.

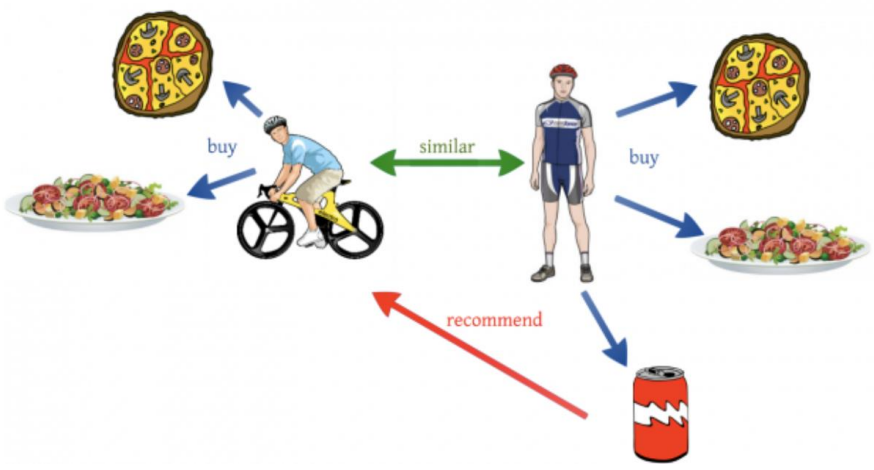


Figure 1

The model **takes into account the latent features in the dataset**. The model learns the latent factors for each user and item and uses them to make rating predictions.

RESULTS

From the descriptive analytics standpoint, a correlation between the number of ratings and the rating itself was found (Figure 2).

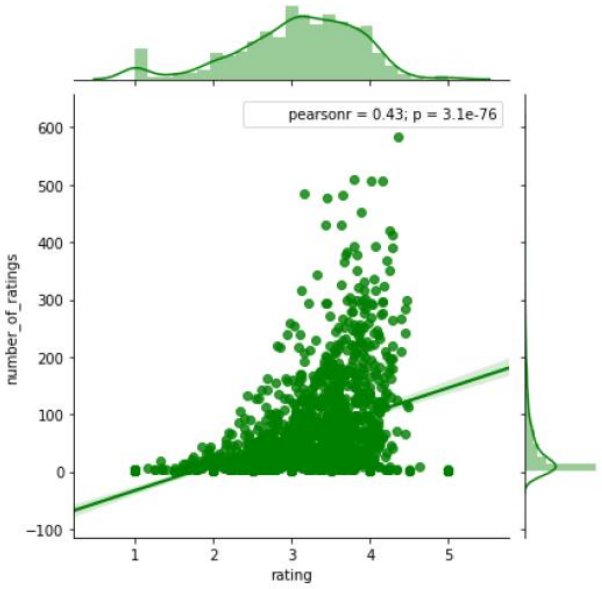


Figure 2

When it comes to predictive analytics, for the first model, which is based on correlation, the correlation results for a randomly chosen example movie (Toy Story) can be seen in Figure 3.

	correlation	number_of_ratings
title		
Toy Story (1995)	1.000000	452
Raise the Red Lantern (1991)	0.641535	58
Flubber (1997)	0.558389	53
Jackal, The (1997)	0.557876	87

Figure 3

The same correlation table can be made for any other movie from the dataset.

For the second model, which is a popularity recommendation model, the most popular movies from the dataset are chosen. The error (RMSE) on testing data is 1. For the following example, the movies are filtered to have at least 50 ratings and the same top 5 movies are recommended to 5 users (Figure 4).

user_id	movie_id	score	rank
1	318	4.484962406015038	1
1	408	4.47	2
1	483	4.446511627906977	3
1	603	4.421875	4
1	513	4.3125	5
2	114	4.540983606557377	1
2	318	4.484962406015038	2
2	169	4.481132075471698	3
2	408	4.47	4
2	483	4.446511627906977	5
3	114	4.540983606557377	1
3	169	4.481132075471698	2
3	408	4.47	3
3	483	4.446511627906977	4
3	64	4.4324324324324325	5
4	114	4.540983606557377	1
4	318	4.484962406015038	2
4	169	4.481132075471698	3
4	408	4.47	4
4	483	4.446511627906977	5
5	114	4.540983606557377	1
5	318	4.484962406015038	2
5	408	4.47	3
5	483	4.446511627906977	4
5	64	4.4324324324324325	5

Figure 4

For the third model, which utilizes collaborative-filtering, predicts the rating a user would give to a movie. For instance, the model gives the top 5 movies it expects to be rated highest. It does that for 5 users (Figure 5).

user_id	movie_id	score	rank
1	286	4.0907851423458705	1
1	302	4.049964782162155	2
1	762	3.966235239370311	3
1	471	3.9660469855503684	4
1	591	3.9560885991291648	5
2	191	4.2581757924841295	1
2	427	4.246132301478232	2
2	12	4.244044440178717	3
2	238	4.243991466550673	4
2	357	4.236781122474516	5
3	153	4.139021735949958	1
3	185	4.1327195282416	2
3	483	4.070365678830589	3
3	211	4.062940310998405	4
3	480	4.058870997591461	5
4	302	4.052674558325256	1
4	313	4.0358977730707775	2
4	127	4.020464282794441	3
4	269	4.006180282993759	4
4	98	3.978991549773658	5
5	195	4.451035175784076	1
5	186	4.374290142519916	2
5	237	4.2118691693381916	3
5	96	4.200936530096973	4
5	478	4.178583406223501	5

Figure 5

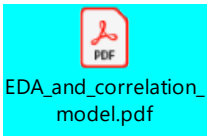
CONCLUSIONS

The 3 provided datasets, collectively called MovieLens, contain data about users, movies and the ratings the users give to the movies.

Three models were used. The first one finds the similarity (via correlation) between movies. The second model recommends movies to users based on the popularity of the movies. The third model predicts how a user would rate any movie based on many latent factors.

APPENDIX

EDA and Correlation Notebook



Recommendation Engines Notebook

Business Case