

Big Data
Project Document
Team 6
Movie Recommendation System

Name	Sec	Bn	Email
Khaled Galal Helmy	1	16	khaled.elnomrosy98@eng-st.cu.edu.eg
Muhammad Ayman	2	11	muhamed.sadek97@eng-st.cu.edu.eg
Yousif Gamal	2	34	Yousif.Ahmed99@eng-st.cu.edu.eg
Youssef Mohamed Ahmed Dawood	2	35	yousef.dawood03@eng-st.cu.edu.eg

Video link:

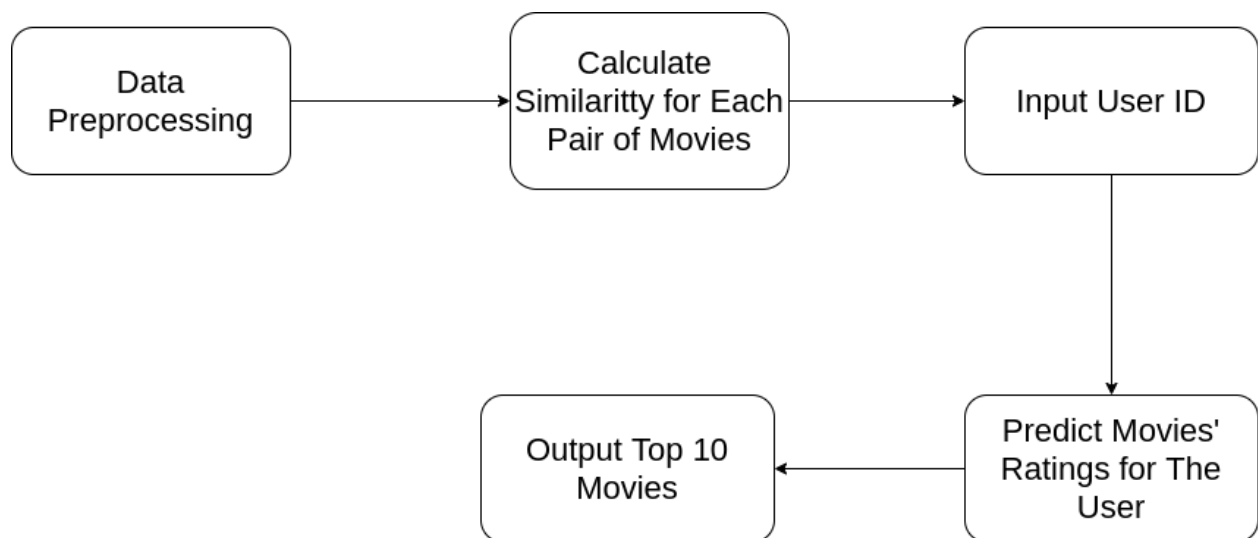
https://drive.google.com/file/d/18_9LP1SIbi2pJxDePOKoEFTux-zoKcvf/view?usp=sharing

Brief Project Description:

Everyone loves movies irrespective of age, gender, or geographical location. We all in a way are connected via this amazing medium.

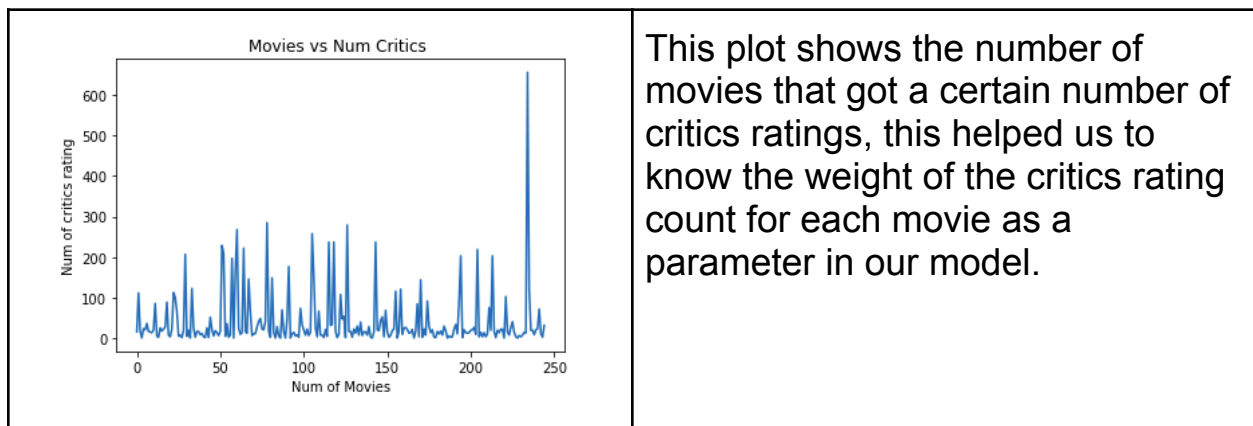
We are building a movie recommendation system that makes use of movies' genres, release year, location, director, actors, and ratings. Also to make the recommendation more personal we use the watch history of the user and the ratings he gave to those movies he watched.

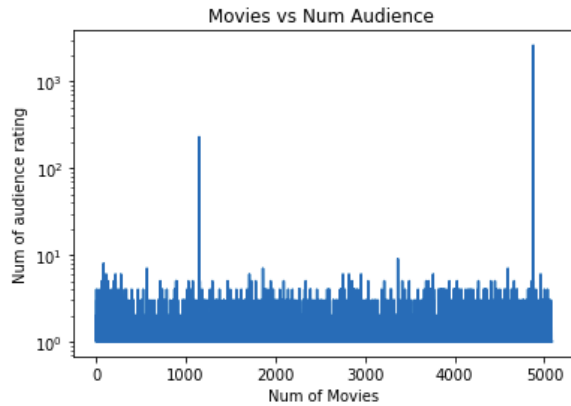
Project Pipeline:



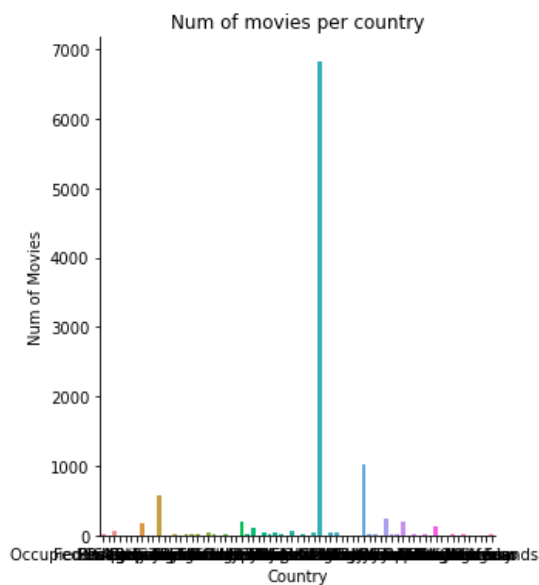
Analysis and solution of the problem:

- Data preprocessing
 - Convert dat files to CSV files
 - Read CSV files into spark Dataframes
 - Filter movies
 - Remove duplicated
 - Keep movies with at least 1800 user reviews and 150 critics review
 - Filter rest of tables based on the filtered movies
 - Change columns types to appropriate types: Example make movieID Int instead of String
 - Drop no needed columns like “spanishTitle” in movies
- Data visualization

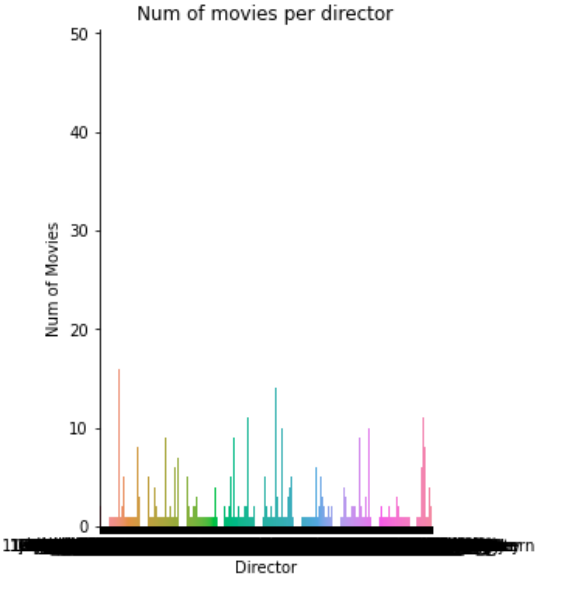
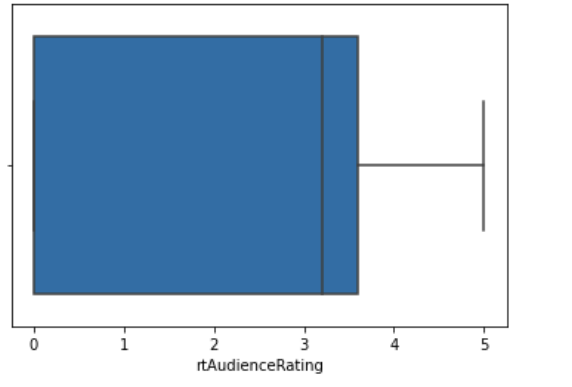
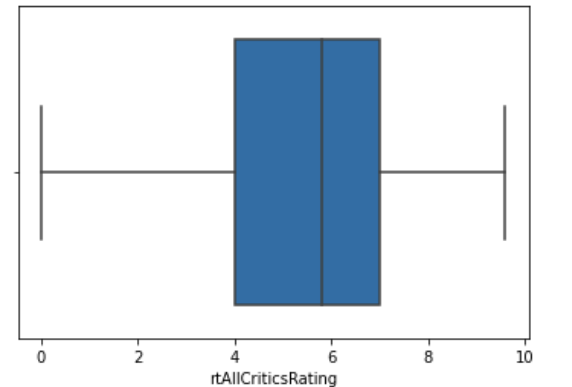


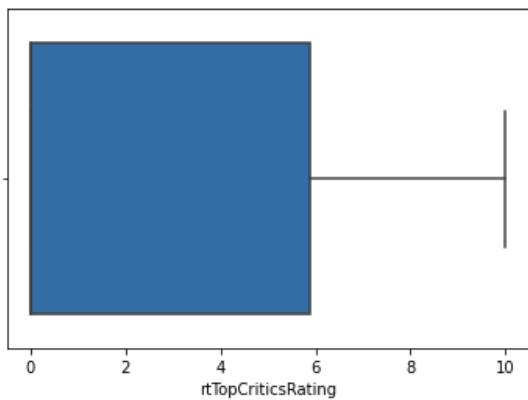


This plot shows the number of movies that got a certain number of audience ratings, this helped us to know the weight of the audience ratings counts for each movie as a parameter in our model.

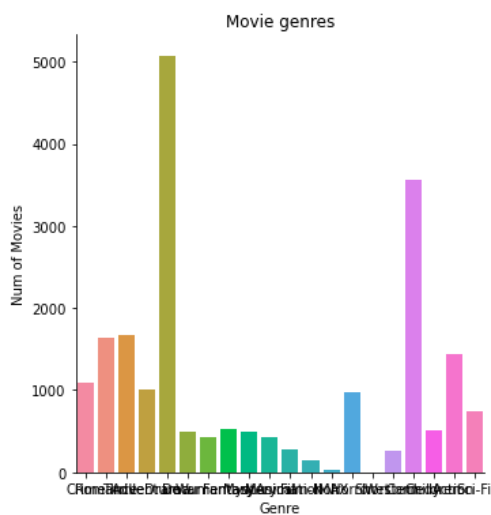


This plot presents the number of movies for each country. It shows that very few countries have many movies and most of the countries have a few movies. But after all the country of the movie matters to the user to some extent that's why the country is included in the prediction system but with low weight.

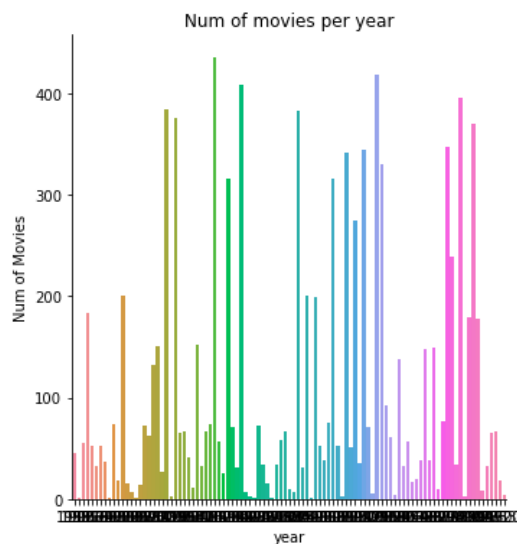
	<p>The graph shows the number of movies made by each director.</p> <p>The graph shows a little diversity for the number of movies made by each director which means there are many directors with money movies that is why we took the director as a parameter in our system with medium weight.</p>
	<p>This is a box plot for the ratings of movies by the audience. From the plot, we can see that the average rating is a little above 2.5 (the half of 5) but also many movies have a very low rating which means bad movies are more than good movies so ratings are important to consider.</p>
	<p>This plot shows the average rating of the critics to different movies which shows that the critics rating falls in the range from 4 to 7 with a maximum rating of 10 and a minimum rating of 0.</p>



This plot shows the average ratings of the top critics of different movies and also the minimum and maximum ratings of the top critics. It shows that most of the movies receive low ratings from the top critics and rarely when a movie receives a high rating.



This plot shows the number of movies for each genre (a movie can have more than one genre). Genres have movies more than others. There is a big diversity shown in the plot which reflects on us that genres matter to the users. That is why genres are included in the system with high weight.



This plot shows the number of movies produced in different years. Some years have a very low number of movies produced in it, And on the other hand, some years have a very large number of movies produced in it compared to other years. Year affects the production capabilities and CGI technologies that is why it's taken as a parameter in the system but with low weight

- Extracting insights from data
 - Movie genres: It is a very important thing in recommending movies so we gave this parameter a high weight in our model.
 - Country: Only a few countries have many movies and most of the countries have very few movies so most of the movies will belong to very few countries that's why it has a low weight in our system
 - Director: many directors have many movies which means that the director of movies matters in the system and that's why it has a medium weight in the system.
 - Actors: Recommending movies based on common actors between the watched and the unwatched movies is a very good parameter to consider. We considered only the top three ranked actors for each movie while finding similar unwatched movies.
 - Year: production capabilities and CGI technologies, which affect the watching experience, differ from year to year and that's why the year is included in our system with low weight.
 - Audience and critics rating: Recommending movies based on audience and critics ratings has a good impact but we gave it a medium weight as not all people like the critics' opinion on movies and people also have different tastes in movies so we depend less on the audience ratings while recommending.
 - Weights:
 - High = .2
 - Medium = .15
 - Low = .05
 - Total 3 high + 2 medium + 2 low = 1
- Model training:
 - The main idea is to calculate the similarity between every two movies based on the parameters specified above.
 - Genres: each movie has a set of genres the similarity is the intersection of the two sets divided by the union

- Director: based on the movies have the same director or not
 - 1 for equal.
 - 0 for not equal.
- Countries: Based on the movies are from the same countries or not
 - 1 for equal.
 - 0 for not equal.
- Actor: each movie has a set of actors we selected the top three actors and the similarity is the intersection of the two sets divided by the union
- Critics, Top critics, and Audience ratings: the movie with min rating divided by the movie with the max rating (each part type of rating with weight $\frac{1}{3}$).
- Year: Based on the difference between the production year of each movie
 - [0-2] years: weight = 1
 - [3-5] years: weight = 0.7
 - [6-7] years: weight = 0.5
 - Greater than 8: weight = 0
- Cosine Similarity

$$Similarity(\vec{A}, \vec{B}) = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| * \|\vec{B}\|}$$

Where A, B are vectors of the ratings to movies A, B by the same users.

- Rating Prediction:

Then we predict the user's ratings to the unwatched movies based on the user's ratings to watched movies and the similarity score between the watched and the unwatched movies. Then we take the top 10 movies with the highest predicted ratings.

$$rating(U, I_i) = \frac{\sum_j rating(U, I_j) * s_{ij}}{\sum_j s_{ij}}$$

Results and evaluation:

There is no scientific way to evaluate the returned results as our approach is called the item-based approach where we predict the ratings of not-watched movies for a specific user by using the ratings he gave to the watched movies and the similarities between those watched and not-watched movies. That's why we can't divide the data into training and validation because this will mean fewer watched movies which will affect the rating prediction of not-watched movies.

Examples for the model recommendations:

User 1:

Watched movies	Recommended movies																																																												
<table><tr><td>0</td><td>Terminator Salvation</td></tr><tr><td>1</td><td>American Beauty</td></tr><tr><td>2</td><td>X-Men</td></tr><tr><td>3</td><td>Bridget Jones's Diary</td></tr><tr><td>4</td><td>Shrek</td></tr><tr><td>5</td><td>Monsters, Inc.</td></tr><tr><td>6</td><td>Harry Potter and the Sorcerer's Stone</td></tr><tr><td>7</td><td>The Lord of the Rings: The Fellowship of the Ring</td></tr><tr><td>8</td><td>About a Boy</td></tr><tr><td>9</td><td>The Bourne Identity</td></tr><tr><td>10</td><td>Road to Perdition</td></tr><tr><td>11</td><td>Bowling for Columbine</td></tr><tr><td>12</td><td>8 Mile</td></tr><tr><td>13</td><td>Far from Heaven</td></tr><tr><td>14</td><td>About Schmidt</td></tr><tr><td>15</td><td>Gangs of New York</td></tr><tr><td>16</td><td>Finding Nemo</td></tr><tr><td>17</td><td>The Italian Job</td></tr><tr><td>18</td><td>Lost in Translation</td></tr></table>	0	Terminator Salvation	1	American Beauty	2	X-Men	3	Bridget Jones's Diary	4	Shrek	5	Monsters, Inc.	6	Harry Potter and the Sorcerer's Stone	7	The Lord of the Rings: The Fellowship of the Ring	8	About a Boy	9	The Bourne Identity	10	Road to Perdition	11	Bowling for Columbine	12	8 Mile	13	Far from Heaven	14	About Schmidt	15	Gangs of New York	16	Finding Nemo	17	The Italian Job	18	Lost in Translation	<table><tr><td colspan="2">suggested</td></tr><tr><td>0</td><td>King Kong</td></tr><tr><td>1</td><td>Across the Universe</td></tr><tr><td>2</td><td>3:10 to Yuma</td></tr><tr><td>3</td><td>Crash</td></tr><tr><td>4</td><td>The Prestige</td></tr><tr><td>5</td><td>Sunshine Cleaning</td></tr><tr><td>6</td><td>Madagascar</td></tr><tr><td>7</td><td>The Number 23</td></tr><tr><td>8</td><td>Melinda and Melinda</td></tr><tr><td>9</td><td>The Matrix Revolutions</td></tr></table>	suggested		0	King Kong	1	Across the Universe	2	3:10 to Yuma	3	Crash	4	The Prestige	5	Sunshine Cleaning	6	Madagascar	7	The Number 23	8	Melinda and Melinda	9	The Matrix Revolutions
0	Terminator Salvation																																																												
1	American Beauty																																																												
2	X-Men																																																												
3	Bridget Jones's Diary																																																												
4	Shrek																																																												
5	Monsters, Inc.																																																												
6	Harry Potter and the Sorcerer's Stone																																																												
7	The Lord of the Rings: The Fellowship of the Ring																																																												
8	About a Boy																																																												
9	The Bourne Identity																																																												
10	Road to Perdition																																																												
11	Bowling for Columbine																																																												
12	8 Mile																																																												
13	Far from Heaven																																																												
14	About Schmidt																																																												
15	Gangs of New York																																																												
16	Finding Nemo																																																												
17	The Italian Job																																																												
18	Lost in Translation																																																												
suggested																																																													
0	King Kong																																																												
1	Across the Universe																																																												
2	3:10 to Yuma																																																												
3	Crash																																																												
4	The Prestige																																																												
5	Sunshine Cleaning																																																												
6	Madagascar																																																												
7	The Number 23																																																												
8	Melinda and Melinda																																																												
9	The Matrix Revolutions																																																												

Runtime:

- for building the similarity, user-independent,: around 6 mins
- For predicting after entering user Id: around 4 mins

Unsuccessful Trials:

We tried to build the item similarity matrix between every two movies but the computation time was very large and we couldn't compute the matrix using spark so instead, we used spark rdd frames to perform any calculations or map functions.

	movID1	movID2	movID3	...
movID1				
movID2				
movID3				
.....				

Any Enhancements:

- To run the model in a fully distributed mode to decrease the computation time and process more movies.