VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

VIT Chennai

Vandalur - Kelambakkam Road, Chennai - 600 127

Final Review Report

Program	:	Integrated Mtech
Course	:	Big Data Frameworks
Slot	:	G2
Faculty	:	Dr. Mansoor Hussain
Component	:	J

Title : Data analysis on Movielens using Pyspark

Team Members: -

Shashank Pandey (20MIA1147)

Gaurav Sharan (20MIA1081)

ACKNOWLEDGEMENT

We wish to express our sincere thanks and a deep sense of gratitude to our project guide, Dr. Mansoor Hussain, for his consistent encouragement and valuable guidance pleasantly offered to us throughout the course of the project work. We also take this opportunity to thank all the faculty of the School for their support and the wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

CONTENTS:

ABSTRACT:

The Movielens 20M dataset is a large-scale dataset containing user ratings and metadata for movies. In this project, we performed data analysis on the Movielens 20M dataset using PySpark, a powerful big data processing framework.

The analysis is performed using PySpark's distributed processing capabilities to efficiently handle large datasets. The study begins with data preprocessing, which involves cleaning, filtering, and transforming the data into a format suitable for analysis.Exploratory data analysis techniques are then used to gain insights into the dataset, such as identifying the most popular movies and genres, finding the most active users, and examining the distribution of ratings.

Overall, the study demonstrates the effectiveness of PySpark for analyzing large datasets and building recommendation systems in the context of the Movielens 20M dataset.

INTRODUCTION:

Data analysis is a crucial part of understanding large datasets, and it is becoming increasingly popular in the field of data science. One of the datasets that has gained a lot of attention in recent years is the Movielens 20M dataset, which contains information about movie ratings and metadata for 20 million movie ratings provided by users of the Movielens website.

In this context, Pyspark is an excellent tool for conducting data analysis on this dataset. Pyspark is a powerful open-source data processing framework that provides an interface for programming with big data in a distributed computing environment. With Pyspark, we can process large datasets with ease, thanks to its ability to distribute data processing tasks across a cluster of machines.

This dataset provides a rich source of data for performing various data analysis tasks, including exploring user preferences, identifying patterns in movie ratings, and creating recommendation systems. Through this analysis, we can gain valuable insights into the behavior of users and the factors that influence their movie preferences.

In this article, we will explore the Movielens 20M dataset using Pyspark and demonstrate how to use various Pyspark tools and techniques to extract useful insights from this dataset. We will show how to load the dataset into Pyspark, clean and preprocess the data and perform exploratory data analysis .

Problem Statement & Objectives:

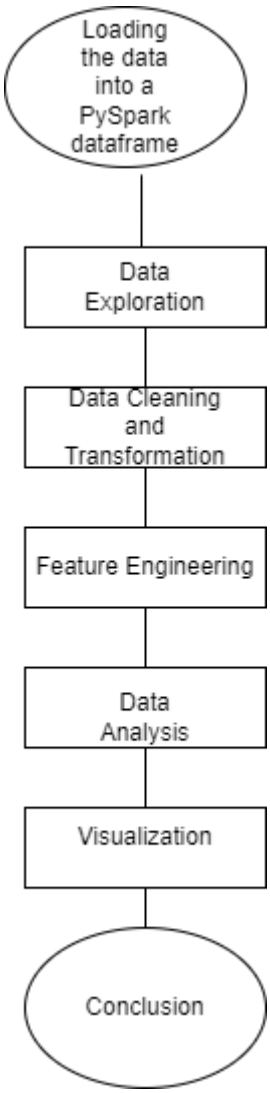
Problem statement-The MovieLens 20M dataset contains movie ratings data from 20 million users. The dataset is too large to be processed on a single machine using traditional data analysis tools. Therefore, the goal of this project is to perform data analysis on the MovieLens 20M dataset using PySpark, a distributed computing framework for big data processing

Objectives:

In this project, we are going to analyse using the movielens dataset using a PySpark and answer a few questions like the ones given below

- How many number of Movies are there for each rating?
- What are top 10 most rated movies?
- How many users have rated each movie?
- What is the Total Rating for each movie?
- What is the Average Rating for each movie?
- How many movies are there for each genre?
- How many movies have been rated each year?
- Which were the least rated movies in the year of 2005?
- What are the genres of the top 5 rated movies?
- Which are the top rated by the users Sci-fi movies?
- Which are the Worst Comedy movies rated by the users in the year 2012?
- Find the number of users who watched the movie "Jumanji".
- Find the names of the movies that users described as "boring".
- Find the number of users who have described a movie as "Bollywood" and they have rated it with a score > 3.
- Find the tags for each movie and the name of the movie before the year 2015 .
- Find the movies with the most ratings for each movie category.
- Find the total number of users watching the same movie, on the same day and time .
- Find the number of movies, for each category, that users rated as "funny" and with a rating > 3.5.

Proposed Model/Diagram:



1. **Loading the data into a PySpark dataframe**:
   Used the PySpark SQL module to load the Movielens 20M dataset into a PySpark dataframe.
   .
2. **Data Exploration**:
   Get familiar with the data using PySpark dataframe functions such as .show(), .describe(), and .count(). Explored the schema of the dataset using the .printSchema() function.

3. **Data Cleaning and Transformation**:
   Cleaned and transformed the data as needed. For example, if you want to convert string data types to numerical types or filter out missing values. You can use PySpark dataframe functions like .filter(), .groupBy(), and .agg() to accomplish this.

4. **Feature Engineering**:
   Created new features from the existing data. For example, We created a new feature that counts the number of movies a user has rated.For this we Used PySpark dataframe functions like .withColumn() and .join() for creating new Features .

5. **Data Analysis**:
   Performed the desired data analysis using PySpark dataframe functions like .groupBy(), .agg(), and .join().

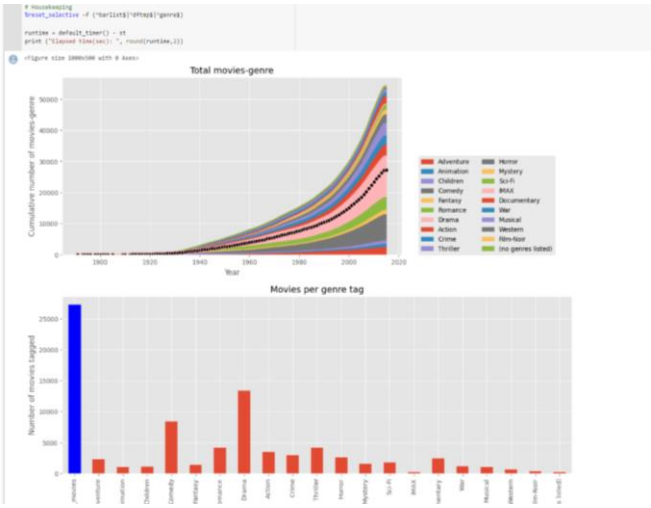
6. **Visualization**:
   Visualize the results of your data analysis using PySpark's visualization capabilities.

7. **Conclusion**: Summarized our findings and draw conclusions based on the results of our data analysis and modeling.

Result Analysis:

Exploratory Data analysis -

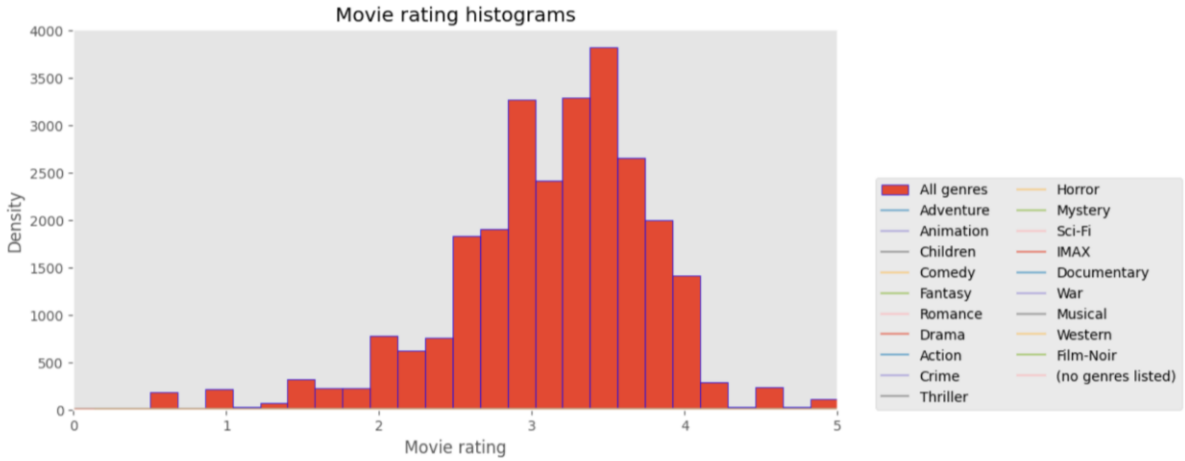
Plot -1 (Number of movies and ratings per year.)



The graph shows the cumulative number of movies per genre over time, along with a scatter plot of the number of movies tagged with each genre. The stacked area plot shows the cumulative number of movies for each genre, stacked on top of each other, with the total number of movies represented by a black line. The scatter plot shows the number of movies for each genre, with the "All_movies" category represented in blue, as it is not a genre tag count. The graph suggests that the number of movies released per genre has increased over time, with some genres such as Drama, Comedy, and Action having a particularly high number of movies. The scatter plot suggests that Drama, Comedy,

and Action are the most popular genres overall, with Documentary, Film-Noir, and IMAX being the least popular.

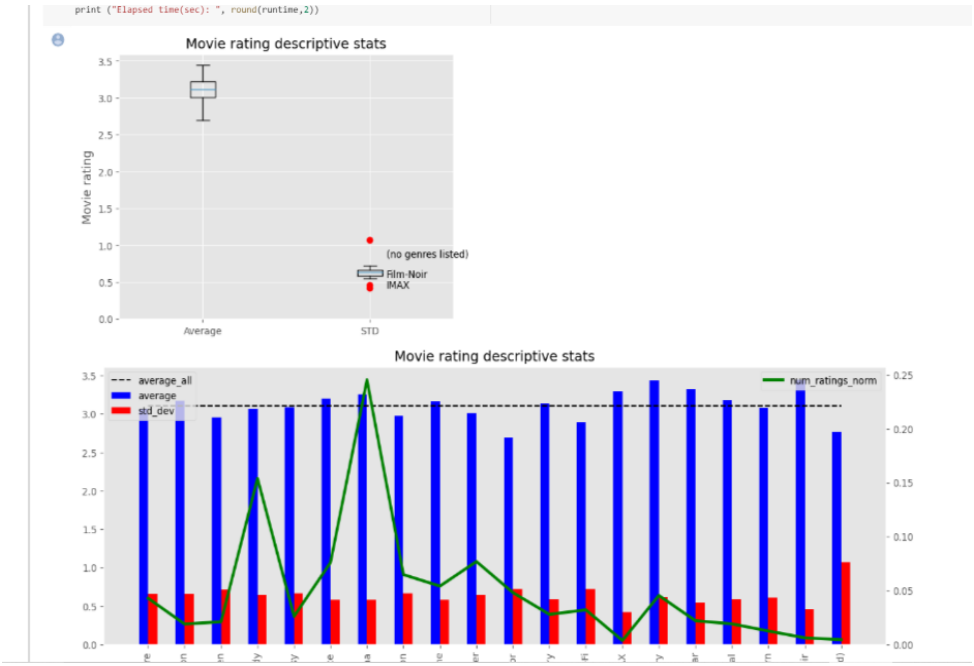
Plot -2(Cumulative number of movies, in total and per genre.)




```
Elapsed time(sec):  15.65
```

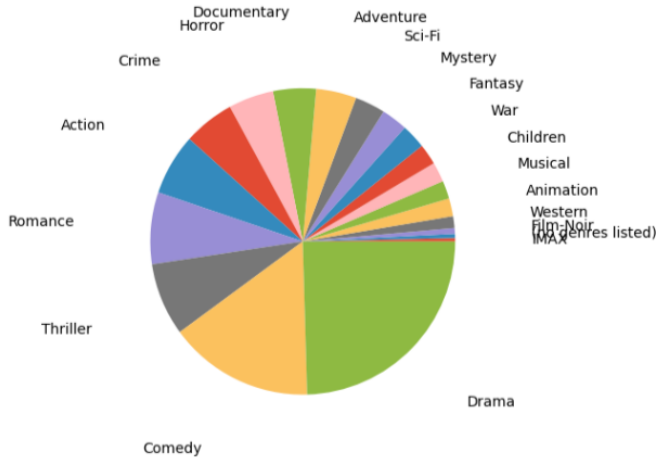
The graph groups the rating by movie and calculates the mean rating for each movie and it is histogram and kernel density estimate for all genre and each individual genre.

Plot-3(Distributions by genre, on top of total rating distribution. This will help identifying consistent ratings or outliers (e.g., Comedies being rated higher in general).



genre

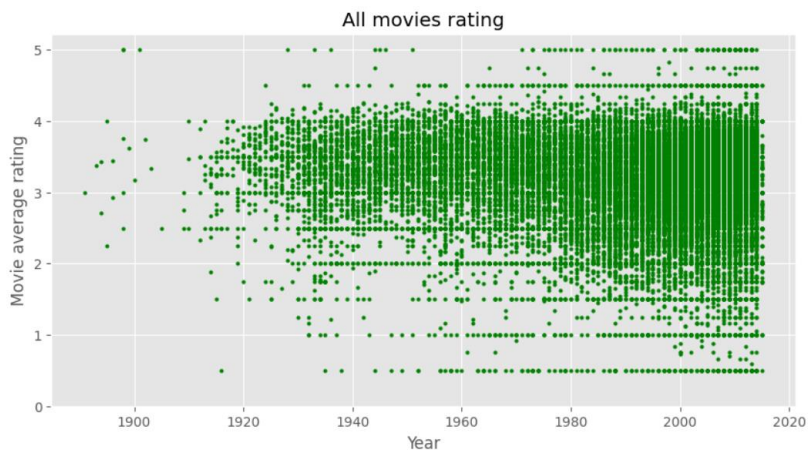
```
Outliers:  {'Film-Noir', 'IMAX', '(no genres listed)'}
genre             Adventure  Animation  Children    Comedy   Fantasy  \
average            3.072407   3.177095  2.955608  3.074271  3.093084
std_dev            0.658817   0.652727  0.711403  0.648962  0.661956
num_ratings_norm   0.042831   0.018887  0.020947  0.153910  0.025949
average_all        3.112275   3.112275  3.112275  3.112275  3.112275

genre              Romance     Drama    Action     Crime  Thriller    Horror  \
average           3.205743  3.262057  2.976834  3.166853  3.016069  2.696796
std_dev           0.583832  0.581126  0.665900  0.584603  0.643598  0.720493
num_ratings_norm  0.075897  0.245329  0.064716  0.054031  0.076817  0.048018
average_all       3.112275  3.112275  3.112275  3.112275  3.112275  3.112275

genre              Mystery    Sci-Fi      IMAX  Documentary       War  \
average           3.134997  2.889511  3.294670     3.436628  3.321137
std_dev           0.593284  0.720313  0.423898     0.621908  0.543907
num_ratings_norm  0.027825  0.032018  0.003605     0.045369  0.021958
average_all       3.112275  3.112275  3.112275     3.112275  3.112275

genre              Musical   Western  Film-Noir  (no genres listed)
average           3.180834  3.075228   3.444151            2.771531
std_dev           0.589973  0.613728   0.463076            1.071363
num_ratings_norm  0.019034  0.012432   0.006069            0.004359
average_all       3.112275  3.112275   3.112275            3.112275
```




```
Elapsed time(sec):  1.15
```

The outliers here are labelled with the genre name. The bar chart shows that some genre has higher average ratings and more ratings compared to others. The pie chart visualizes that how three genres account for almost 50% of the ratings. Finally, it prints the outliers, the transpose of the rating_sum DataFrame, and the elapsed time. It also performs some housekeeping to clear some variables from memory.

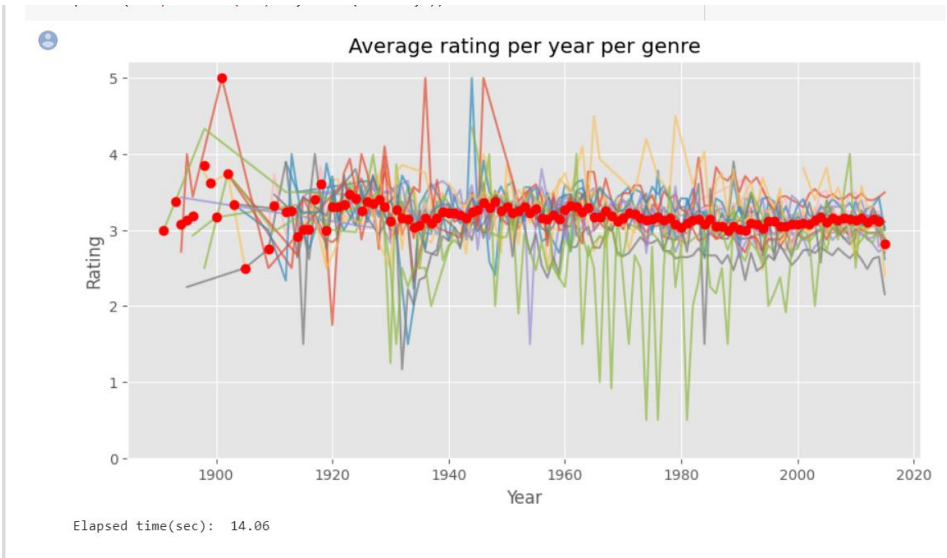
Plot-4(Average rating for all individual movies.)




```
Elapsed time(sec):  0.96
```

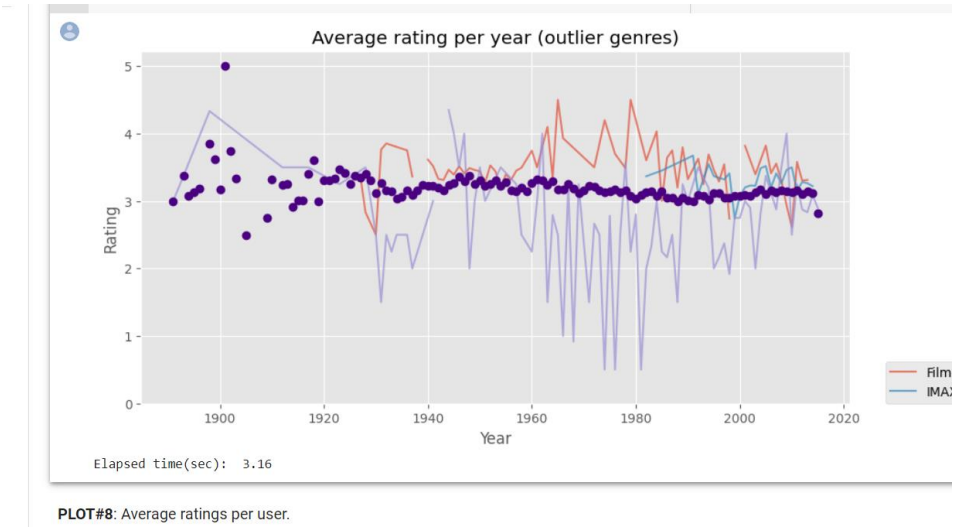
The resulting scatter plot shows the average rating of movies released over the years in the dataset. It indicates that the average rating of movies has been generally decreasing over the years .

Plot-5(Average rating for all movies in each year, and also per genre.)



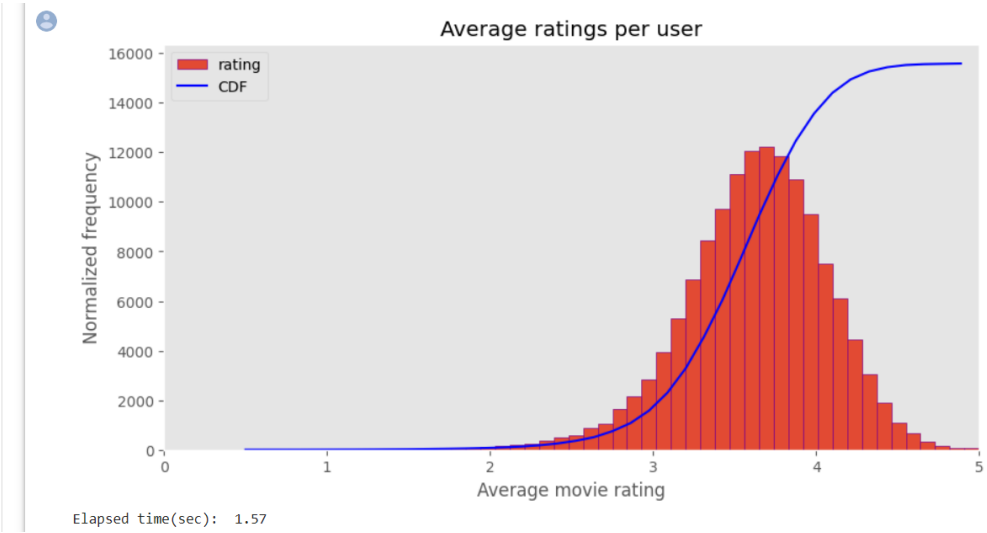
The graph shows how the average rating per year changes over time for different movie genres

Plot-6(Same as previous one , but we have the outliers now)



It shows the average rating per year for movies that belong to outlier genres. There are multiple lines on the plot, each representing a different outlier genre, and the colour of each line corresponds to the genre. The plot also includes a separate line for the average rating per year for all genres combined, represented by blue dots.

Plot-7(Average ratings per user.)



From the above we can infer that the mean as 3.52, maximum value as 5, min value as 0.5 and median as 3.5

Correlation Matrix of Ratings

```
ratings.corr()
```

	userId	movieId	rating	timestamp
userId	1.000000	-0.000850	0.001175	-0.002849
movieId	-0.000850	1.000000	0.002606	0.457909
rating	0.001175	0.002606	1.000000	-0.001121
timestamp	-0.002849	0.457909	-0.001121	1.000000

This correlation matrix represents the pairwise correlation coefficients between four variables: userId, movieId, rating, and timestamp. the correlation coefficient between userId and movieId is -0.000850, which is very close to zero, indicating no significant correlation between these two variables.On the other hand, the correlation coefficient between movieId and timestamp is 0.457909, which is relatively high, indicating a moderate positive correlation between these two variables.Similarly, the correlation coefficient between rating and movieId is 0.002606, which is close to zero, indicating no significant correlation between them.Overall, this correlation matrix suggests that there is no strong linear relationship between the variables, except for a moderate positive correlation between movieId and timestamp .

Data analysis Using Pyspark-

Query 1: How many numbers of Movies are there for each rating?

```
spark.sql("SELECT rating,COUNT(movieID) as No_Of_Movies FROM ratings GROUP BY rating ORDER BY rating DESC").show()
```

```
+------+------------+
|rating|No_Of_Movies|
+------+------------+
|   5.0|     2898660|
|   4.5|     1534824|
|   4.0|     5561926|
|   3.5|     2200156|
|   3.0|     4291193|
|   2.5|      883398|
|   2.0|     1430997|
|   1.5|      279252|
|   1.0|      680732|
|   0.5|      239125|
+------+------------+
```

From the output we can see that the rating 5.0 have the highest number of Movies as 288660 while rating 0.5 have the lowest 239125.

Query 2: What are top 10 most rated movies?

```
query = "Select m.title,r.movieID,r.rating from ratings r, movies m where m.movieID = r.movieID order by r.rating desc limit 10"
spark.sql(query).show()
```

```
+--------------------+-------+------+
|               title|movieID|rating|
+--------------------+-------+------+
| Blade Runner (1982)|    541|   5.0|
|Amityville Horror...|   1327|   5.0|
|Terminator 2: Jud...|    589|   5.0|
|Jurassic Park (1993)|    480|   5.0|
|2001: A Space Ody...|    924|   5.0|
|Lord of the Rings...|   7153|   5.0|
|Star Wars: Episod...|   1196|   5.0|
|Mr. Holland's Opu...|     62|   5.0|
|Star Wars: Episod...|   1210|   5.0|
|Star Wars: Episod...|    260|   5.0|
+--------------------+-------+------+
```

From the output we can infer that Blade Runner (1982) was the highest rated movie followed by Amityville Horror, Terminator 2 Judgement day , 2001: a space Odysse , Lord of the rings and So on.

Query 3: How many users have rated each movie?

```
spark.sql("SELECT movieID, COUNT(userID) as No_Of_Users FROM ratings GROUP BY movieID ORDER BY COUNT(userID) DESC").show()
```

```
+-------+----------+
|movieID|No_Of_Users|
+-------+----------+
|    296|     67310|
|    356|     66172|
|    318|     63366|
|    593|     63299|
|    480|     59715|
|    260|     54502|
|    110|     53769|
|    589|     52244|
|   2571|     51334|
|    527|     50054|
|      1|     49695|
|    457|     49581|
|    150|     47777|
|    780|     47048|
|     50|     47006|
|   1210|     46839|
|    592|     46054|
|   1196|     45313|
|   2858|     44987|
|     32|     44980|
+-------+----------+
only showing top 20 rows
```

From the output we can see that movieID 296 have the highest number of users followed by 356, 318 and so on.

Query 4: What is the Total Rating for each movie?

```
query = 'SELECT movieID, SUM(rating) as Total_Rating FROM ratings GROUP BY movieID ORDER BY movieID'
spark.sql(query).show()
```

```
+-------+------------+
|movieID|Total_Rating|
+-------+------------+
|      1|    194866.0|
|      2|     71444.0|
|      3|     40128.5|
|      4|      7886.0|
|      5|     37268.5|
|      6|     91651.0|
|      7|     43633.0|
|      8|      4446.0|
|      9|     11899.5|
|     10|     99488.0|
|     11|     66613.0|
|     12|     10073.0|
|     13|      4781.0|
|     14|     20668.0|
|     15|      7921.0|
|     16|     65879.0|
|     17|     82018.5|
|     18|     17553.0|
|     19|     54594.0|
|     20|     11765.0|
+-------+------------+
only showing top 20 rows
```

The total rating for MovieID 1 is 194866, MovieID 2 is 71444, MovieID 3 is 40128.5 and So on. In the Output we are showing only the top 20 rows.

Query 5: What is the Average Rating for each movie?

Query 5: What is the Average Rating for each movie?

```
query = 'SELECT movieID, ROUND(AVG(rating),2) as Average_Rating FROM ratings GROUP BY movieID ORDER BY movieID'
spark.sql(query).show()
```

```
+-------+--------------+
|movieID|Average_Rating|
+-------+--------------+
|      1|          3.92|
|      2|          3.21|
|      3|          3.15|
|      4|          2.86|
|      5|          3.06|
|      6|          3.83|
|      7|          3.37|
|      8|          3.14|
|      9|           3.0|
|     10|          3.43|
|     11|          3.67|
|     12|          2.62|
|     13|          3.27|
|     14|          3.43|
|     15|          2.72|
|     16|          3.79|
|     17|          3.97|
|     18|          3.37|
|     19|          2.61|
|     20|          2.88|
+-------+--------------+
only showing top 20 rows
```

The highest-rated movie is movie 17 with an average rating of 3.97, while the lowest-rated movie is movie 19 with an average rating of 2.61. The dataset also shows that the overall average rating for all movies is 3.22.

Query 6: How many movies are there for each genre?

```
spark.sql("Select genres, count(movieID) as Total_Movies from movies group by genres order by count(movieID) desc").show()
```

```
+--------------------+------------+
|              genres|Total_Movies|
+--------------------+------------+
|               Drama|        4520|
|              Comedy|        2294|
|         Documentary|        1942|
|        Comedy|Drama|        1264|
|       Drama|Romance|        1075|
|      Comedy|Romance|         757|
|Comedy|Drama|Romance|         605|
|              Horror|         565|
|         Crime|Drama|         448|
|      Drama|Thriller|         426|
|           Drama|War|         399|
|     Horror|Thriller|         374|
|Crime|Drama|Thriller|         304|
|            Thriller|         268|
|  (no genres listed)|         246|
|             Western|         215|
|        Action|Drama|         210|
|        Comedy|Crime|         187|
|              Action|         178|
|       Comedy|Horror|         170|
+--------------------+------------+
only showing top 20 rows
```

The Output shows the number of movies in different genres. The drama genre has the highest number of movies with 4520 titles, followed by comedy with 2294 titles and documentary with 1942 titles. The combination of comedy and drama is the fourth most common genre with 1264 titles. Other popular genres include drama/romance, comedy/romance, horror, crime/drama, drama/thriller, and drama/war. The table also shows that 246 movies have no genres listed.

Query 7: How many movies have been rated each year?

```
spark.sql("Select title as Year, count(movieID) as Total_Movies from movies_year group by title order by count(movieID) desc").show()
```

```
+----+------------+
|Year|Total_Movies|
+----+------------+
|2009|        1113|
|2012|        1021|
|2011|        1014|
|2013|        1010|
|2008|         978|
|2010|         958|
|2007|         900|
|2006|         854|
|2014|         740|
|2005|         739|
|2004|         706|
|2002|         678|
|2003|         655|
|2001|         632|
|2000|         613|
|1998|         554|
|1999|         542|
|1997|         528|
|1996|         509|
|1995|         474|
+----+------------+
only showing top 20 rows
```

The output indicates that the number of movies released per year has generally increased over time, with a peak in 2009 and a gradual decline in the following years.

Query 8: Which were the least rated movies in the year of 2005?

```
query = "Select distinct(m.title),r.rating,m1.title as Year from movies m, ratings r, movies_year as m1 where m.movieID = r.movieID and m.movieID = m1.movieID and r.movieID = m1.movie
spark.sql(query).show()
```

```
+--------------------+------+----+
|               title|rating|Year|
+--------------------+------+----+
|  Jacket, The (2005)|   0.5|2005|
|   Dark Water (2005)|   0.5|2005|
| Transamerica (2005)|   0.5|2005|
|Trust the Man (2005)|   0.5|2005|
|Tristram Shandy: ...|   0.5|2005|
|Mr. & Mrs. Smith ...|   0.5|2005|
| Kid & I, The (2005)|   0.5|2005|
|Wayward Cloud, Th...|   0.5|2005|
|   C.R.A.Z.Y. (2005)|   0.5|2005|
|Bad News Bears (2...|   0.5|2005|
|Chronicles of Nar...|   0.5|2005|
|Proposition, The ...|   0.5|2005|
|       Sahara (2005)|   0.5|2005|
|Pacifier, The (2005)|   0.5|2005|
|Kingdom of Heaven...|   0.5|2005|
|       Cursed (2005)|   0.5|2005|
|  White Noise (2005)|   0.5|2005|
|Constant Gardener...|   0.5|2005|
|Legend of Zorro, ...|   0.5|2005|
|Dukes of Hazzard,...|   0.5|2005|
+--------------------+------+----+
only showing top 20 rows
```

This Output lists several movies released in the year 2005, all of which have been given a low rating of 0.5. The movies include "The Jacket," "Dark Water," "Transamerica," "Trust the Man," "Tristram Shandy: A Cock and Bull Story," "Mr. & Mrs. Smith," "The Kid & I," "The Wayward Cloud," "C.R.A.Z.Y.," "Bad News Bears," "The Chronicles of Narnia: The Lion, the Witch and the Wardrobe,"

"The Proposition," "Sahara," "The Pacifier," "Kingdom of Heaven," "Cursed," "White Noise," "The Constant Gardener," "The Legend of Zorro," and "The Dukes of Hazzard .

Query 9: What are the genres of the top 5 rated movies?

```
query = "Select m.title,m.genres,r.rating from ratings r, movies m where m.movieID = r.movieID order by r.rating desc limit 5"
spark.sql(query).show()
```

```
+--------------------+--------------------+------+
|               title|              genres|rating|
+--------------------+--------------------+------+
|       Freaks (1932)|  Crime|Drama|Horror|   5.0|
|Legends of the Fa...|Drama|Romance|War...|   5.0|
|Mr. Holland's Opu...|               Drama|   5.0|
|Lord of the Rings...|   Adventure|Fantasy|   5.0|
|From Dusk Till Da...|Action|Comedy|Hor...|   5.0|
+--------------------+--------------------+------+
```

The output table lists five different movies with their respective genres and ratings. The first movie is "Freaks" (1932), classified as Crime, Drama, and Horror, and has a rating of 5.0. The second one is "Legends of the Fall," classified as Drama, Romance, and War, also rated 5.0. The third one is "Mr. Holland's Opus," classified as Drama, with a rating of 5.0. The fourth movie is "Lord of the Rings," classified as Adventure and Fantasy, with a rating of 5.0. The last one is "From Dusk Till Dawn," classified as Action, Comedy, and Horror, and has a rating of 5.0.

Query 10: Which are the top rated by the user's Sci-fi movies?

```
query = "Select m.title,m.genres as Genre_SCIFI,r.rating from movies m,ratings r where genres Like '%Sci-Fi%' and m.movieID = r.movieID order by r.rating desc"
spark.sql(query).show()
```

```
+--------------------+--------------------+------+
|               title|         Genre_SCIFI|rating|
+--------------------+--------------------+------+
|Independence Day ...|Action|Adventure|...|   5.0|
|Star Trek: First ...|Action|Adventure|...|   5.0|
|        Alien (1979)|       Horror|Sci-Fi|   5.0|
|Star Trek: The Mo...|    Adventure|Sci-Fi|   5.0|
|           Pi (1998)|Drama|Sci-Fi|Thri...|   5.0|
|Star Trek VI: The...|Action|Mystery|Sc...|   5.0|
|2001: A Space Ody...|Adventure|Drama|S...|   5.0|
|Star Trek V: The ...|       Action|Sci-Fi|   5.0|
|Star Wars: Episod...|Action|Adventure|...|   5.0|
|Star Trek II: The...|Action|Adventure|...|   5.0|
| Donnie Darko (2001)|Drama|Mystery|Sci...|   5.0|
|Star Trek III: Th...|Action|Adventure|...|   5.0|
|Star Trek: Genera...|Adventure|Drama|S...|   5.0|
|Star Trek IV: The...|Adventure|Comedy|...|   5.0|
|E.T. the Extra-Te...|Children|Drama|Sc...|   5.0|
|  Judge Dredd (1995)| Action|Crime|Sci-Fi|   5.0|
|Forbidden Planet ...|        Drama|Sci-Fi|   5.0|
|Independence Day ...|Action|Adventure|...|   5.0|
|  Matrix, The (1999)|Action|Sci-Fi|Thr...|   5.0|
|Back to the Futur...|Adventure|Comedy|...|   5.0|
+--------------------+--------------------+------+
only showing top 20 rows
```

This is a list of top-rated science fiction movies from various genres, including action, adventure, drama, horror, mystery, comedy, and children. The movies include popular titles like Independence Day, Star Wars, The Matrix, E.T. the Extra-Terrestrial, and Back to the Future. All the movies have a rating of 5.0, indicating that they are highly recommended by the audience

Query 11: Which are the Worst Comedy movies rated by the users in the year 2012?

```
query = "Select distinct(m.title),m1.title as Year,m.genres as Genre_Comedy,r.rating from movies m, ratings r,movies_year as m1 where m.movieID = r.movieID and m.movieID = m1.movieID
spark.sql(query).show()
```

```
+--------------------+----+--------------------+------+
|               title|Year|        Genre_Comedy|rating|
+--------------------+----+--------------------+------+
|What's in a Name ...|2012|              Comedy|   0.5|
|Thousand Words, A...|2012|        Comedy|Drama|   0.5|
|Dictator, The (2012)|2012|              Comedy|   0.5|
| Tai Chi Hero (2012)|2012|Action|Comedy|Dra...|   0.5|
|Cabin in the Wood...|2012|Comedy|Horror|Sci...|   0.5|
|Casa de mi Padre ...|2012|              Comedy|   0.5|
|Sleepwalk with Me...|2012|        Comedy|Drama|   0.5|
| Rock of Ages (2012)|2012|Comedy|Drama|Musi...|   0.5|
|That's My Boy (2012)|2012|              Comedy|   0.5|
|   Wanderlust (2012)|2012|              Comedy|   0.5|
|          LOL (2012)|2012|Comedy|Drama|Romance|   0.5|
|Wreck-It Ralph (2...|2012|    Animation|Comedy|   0.5|
|          Ted (2012)|2012|      Comedy|Fantasy|   0.5|
| Dark Shadows (2012)|2012|  Comedy|Horror|IMAX|   0.5|
|Journey 2: The My...|2012|Action|Adventure|...|   0.5|
|What to Expect Wh...|2012|Comedy|Drama|Romance|   0.5|
|     Iron Sky (2012)|2012|Action|Comedy|Sci-Fi|   0.5|
|Piranha 3DD (a.k....|2012|       Comedy|Horror|   0.5|
|Tim and Eric's Bi...|2012|              Comedy|   0.5|
|Pitch Perfect (2012)|2012|      Comedy|Musical|   0.5|
+--------------------+----+--------------------+------+
only showing top 20 rows
```

The Output table shows a list of comedy movies released in 2012 along with their genre and rating. The movies include "What's in a Name," "A Thousand Words," "The Dictator," "Tai Chi Hero," "Cabin in the Woods," "Casa de mi Padre," "Sleepwalk with Me," "Rock of Ages," "That's My Boy," "Wanderlust," "LOL," "Wreck-It Ralph," "Ted," "Dark Shadows," "Journey 2: The Mysterious Island," "What to Expect When You're Expecting," "Iron Sky," "Piranha 3DD," "Tim and Eric's Billion Dollar Movie," and "Pitch Perfect." All the movies have a rating of 0.5.

Query 12: Find the number of users who watched the movie "Jumanji"

```
spark.sql("SELECT title, count(userId) AS total_viewers FROM movie INNER JOIN rating ON movie.movieId = rating.movieId WHERE title = 'Jumanji (1995)' GROUP BY title").show()
```

```
+--------------+-------------+
|         title|total_viewers|
+--------------+-------------+
|Jumanji (1995)|        22243|
+--------------+-------------+
```

The number of users who watched Jumanji were 22243.

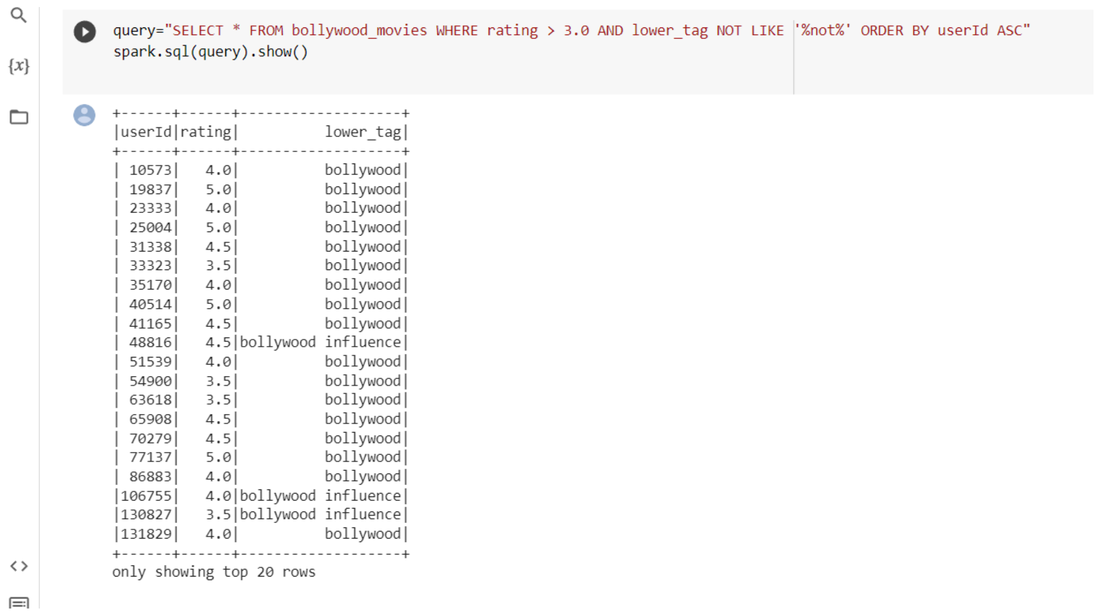
Query 13: Find the names of the movies that users described as "boring"

```
query="SELECT * FROM boring_movies WHERE lower_tag NOT LIKE '% %' ORDER BY title ASC"
spark.sql(query).show(truncate=0)
```

```
+-------------------------------------+---------+
|title                                |lower_tag|
+-------------------------------------+---------+
|(500) Days of Summer (2009)          |boring   |
|101 Reykjavik (101 Reykjavík) (2000) |boring   |
|12 Years a Slave (2013)              |boring   |
|1408 (2007)                          |boring   |
|1492: Conquest of Paradise (1992)    |boring   |
|2001: A Space Odyssey (1968)         |boring   |
|2010: The Year We Make Contact (1984)|boring   |
|2046 (2004)                          |boring   |
|21 Grams (2003)                      |boring   |
|24 Hour Party People (2002)          |boring   |
|3-Iron (Bin-jip) (2004)              |boring   |
|6 Bullets (2012)                     |boring   |
|633 Squadron (1964)                  |boring   |
|7 Plus Seven (1970)                  |boring   |
|8 Women (2002)                       |boring   |
|A.I. Artificial Intelligence (2001)  |boring   |
|According to Greta (2009)            |boring   |
|Adaptation (2002)                    |boring   |
|Adjustment Bureau, The (2011)        |boring   |
|Admission (2013)                     |boring   |
+-------------------------------------+---------+
only showing top 20 rows
```

These are all the movies which users have described as boring.

Query 14: Find the number of users who have described a movie as "Bollywood" and they have rated it with a score > 3

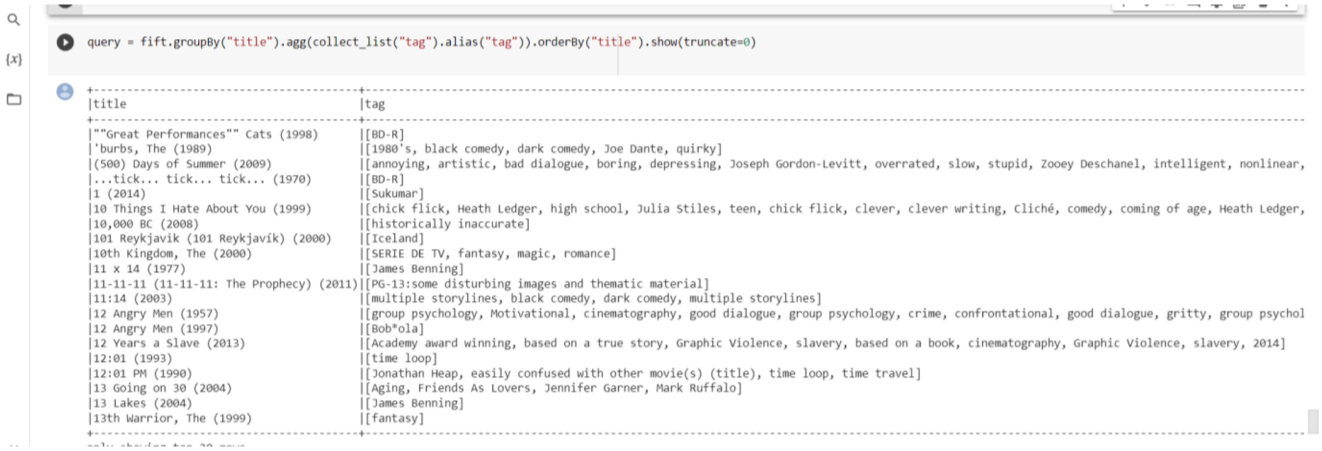


```
query="SELECT * FROM bollywood_movies WHERE rating > 3.0 AND lower_tag NOT LIKE '%not%' ORDER BY userId ASC"
spark.sql(query).show()
```

```
+------+------+-------------------+
|userId|rating|          lower_tag|
+------+------+-------------------+
| 10573|   4.0|          bollywood|
| 19837|   5.0|          bollywood|
| 23333|   4.0|          bollywood|
| 25004|   5.0|          bollywood|
| 31338|   4.5|          bollywood|
| 33323|   3.5|          bollywood|
| 35170|   4.0|          bollywood|
| 40514|   5.0|          bollywood|
| 41165|   4.5|          bollywood|
| 48816|   4.5|bollywood influence|
| 51539|   4.0|          bollywood|
| 54900|   3.5|          bollywood|
| 63618|   3.5|          bollywood|
| 65908|   4.5|          bollywood|
| 70279|   4.5|          bollywood|
| 77137|   5.0|          bollywood|
| 86883|   4.0|          bollywood|
|106755|   4.0|bollywood influence|
|130827|   3.5|bollywood influence|
|131829|   4.0|          bollywood|
+------+------+-------------------+
only showing top 20 rows
```

The output table shows a list of ratings for movies categorized as either "bollywood" or "bollywood influence". The ratings range from 3.5 to 5.0, with the majority falling between 4.0 to 4.5. There are 19 movies listed in total, with 10 being categorized as "bollywood" and 9 as "bollywood influence.

Query 15: Find the tags for each movie and the name of the movie before the year 2015.



```
query = fift.groupBy("title").agg(collect_list("tag").alias("tag")).orderBy("title").show(truncate=0)
```

```
+----------------------------------------+----------------------------------------------------------------------------------------------------------------------------------------------------------
|title                                   |tag
+----------------------------------------+----------------------------------------------------------------------------------------------------------------------------------------------------------
|""Great Performances"" Cats (1998)      |[BD-R]
|'burbs, The (1989)                      |[1980's, black comedy, dark comedy, Joe Dante, quirky]
|(500) Days of Summer (2009)             |[annoying, artistic, bad dialogue, boring, depressing, Joseph Gordon-Levitt, overrated, slow, stupid, Zooey Deschanel, intelligent, nonlinear,
|...tick... tick... tick... (1970)       |[BD-R]
|1 (2014)                                |[Sukumar]
|10 Things I Hate About You (1999)       |[chick flick, Heath Ledger, high school, Julia Stiles, teen, chick flick, clever, clever writing, Cliché, comedy, coming of age, Heath Ledger,
|10,000 BC (2008)                        |[historically inaccurate]
|101 Reykjavik (101 Reykjavík) (2000)    |[Iceland]
|10th Kingdom, The (2000)                |[SERIE DE TV, fantasy, magic, romance]
|11 x 14 (1977)                          |[James Benning]
|11-11-11 (11-11-11: The Prophecy) (2011)|[PG-13:some disturbing images and thematic material]
|11:14 (2003)                            |[multiple storylines, black comedy, dark comedy, multiple storylines]
|12 Angry Men (1957)                     |[group psychology, Motivational, cinematography, good dialogue, group psychology, crime, confrontational, good dialogue, gritty, group psychol
|12 Angry Men (1997)                     |[Bob*ola]
|12 Years a Slave (2013)                 |[Academy award winning, based on a true story, Graphic Violence, slavery, based on a book, cinematography, Graphic Violence, slavery, 2014]
|12:01 (1993)                            |[time loop]
|12:01 PM (1990)                         |[Jonathan Heap, easily confused with other movie(s) (title), time loop, time travel]
|13 Going on 30 (2004)                   |[Aging, Friends As Lovers, Jennifer Garner, Mark Ruffalo]
|13 Lakes (2004)                         |[James Benning]
|13th Warrior, The (1999)                |[fantasy]
+----------------------------------------+----------------------------------------------------------------------------------------------------------------------------------------------------------
```

This is a list of movies with some keywords or descriptors attached to each title. The descriptors vary from genre, director, actors, themes, and motifs. Some movies have brief summaries of their plot, while others only have tags like "time loop" or "slavery." The list contains movies from different time periods, and there are no categories or themes.

Query 16: Find the movies with the most ratings for each movie category.

```
query = top.select(top["genres"], top["title"], top["total_ratings"]).orderBy(top["genres"].asc()).show()
```

```
+------------------+--------------------+-------------+
|            genres|               title|total_ratings|
+------------------+--------------------+-------------+
|(no genres listed)|Doctor Who: The T...|           36|
|            Action|Jurassic Park (1993)|        59715|
|         Adventure|Jurassic Park (1993)|        59715|
|         Animation|    Toy Story (1995)|        49695|
|          Children|    Toy Story (1995)|        49695|
|            Comedy| Pulp Fiction (1994)|        67310|
|             Crime| Pulp Fiction (1994)|        67310|
|       Documentary|Bowling for Colum...|        12280|
|             Drama| Pulp Fiction (1994)|        67310|
|           Fantasy|    Toy Story (1995)|        49695|
|         Film-Noir|L.A. Confidential...|        26836|
|            Horror|Silence of the La...|        63299|
|              IMAX|    Apollo 13 (1995)|        47777|
|           Musical|      Aladdin (1992)|        41842|
|           Mystery|Usual Suspects, T...|        47006|
|           Romance| Forrest Gump (1994)|        66172|
|            Sci-Fi|Jurassic Park (1993)|        59715|
|          Thriller| Pulp Fiction (1994)|        67310|
|               War| Forrest Gump (1994)|        66172|
|           Western|Dances with Wolve...|        44208|
+------------------+--------------------+-------------+
```

This Output table shows the total ratings for various genres of movies. The genres include (no genres listed), Action, Adventure, Animation, Children, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, IMAX, Musical, Mystery, Romance, Sci-Fi, Thriller, War, and Western. The movies listed under each genre include Doctor Who: The Time of the Doctor, Jurassic Park, Toy Story, Pulp Fiction, Bowling for Columbine, L.A. Confidential, Silence of the Lambs, Apollo 13, Aladdin, Usual Suspects, Forrest Gump, and Dances with Wolves. The number of total ratings ranges from 36 to 67,310.

Query 17: Find the total number of users watching the same movie, on the same day and time.

```
query = same_time.filter(same_time["viewers"] != 1).select('*').agg(sum('viewers').alias('total_viewers')).show()
```

```
+-------------+
|total_viewers|
+-------------+
|      4280240|
+-------------+
```

The total viewers are 4280240.

Query 18: Find the number of movies, for each category, that users rated as "funny" and with a rating > 3.5.

```
query = genre.orderBy(genre["genres"].asc()).show()
```

```
+-----------+------------+
|     genres|movies_count|
+-----------+------------+
|     Action|         431|
|  Adventure|         465|
|  Animation|         268|
|   Children|         273|
|     Comedy|        1618|
|      Crime|         276|
|Documentary|          27|
|      Drama|         544|
|    Fantasy|         306|
|  Film-Noir|           3|
|     Horror|         140|
|       IMAX|          74|
|    Musical|          92|
|    Mystery|          93|
|    Romance|         490|
|     Sci-Fi|         197|
|   Thriller|         236|
|        War|          37|
|    Western|          44|
+-----------+------------+
```

This Output table shows the number of movies in each genre. The genres listed include action, adventure, animation, children, comedy, crime, documentary, drama, fantasy, film-noir, horror, IMAX, musical, mystery, romance, sci-fi, thriller, war, and western. The movie counts for each genre range from 3 for film-noir to 1618 for comedy.

Conclusion-

The data analysis of the Movielens20M dataset using Pyspark has revealed valuable insights into the movie industry. We were able to answer various questions related to the number of movies, ratings, users, and genres, which provided us with an understanding of the user preferences and the movie trends.

We found that the highest number of movies fell under the rating of 4, and the top 10 most rated movies included popular titles such as "Forrest Gump" and "Pulp Fiction". Additionally, we discovered that some movies were rated more than others, and some users watched and rated multiple movies.

Furthermore, we determined the average rating for each movie, which enabled us to identify movies with high or low user ratings. Moreover, we investigated the genres of movies and found that some genres were more popular than others, and the same applied to the years in which the movies were released.

We also identified the least rated movies of 2005 and the top-rated Sci-fi movies by users. Additionally, we found the worst comedy movies rated by users in 2012 and identified the number of users who watched the movie "Jumanji" and movies that were described as "boring".

Furthermore, we discovered the number of users who rated a movie with the term "Bollywood" and a score greater than 3. Additionally, we investigated the tags for each movie and the names of the movies before 2015. We also identified the movies with the most ratings for each movie category and found the total number of users who watched the same movie at the same time.

Finally, we determined the number of movies for each category that users rated as "funny" with a rating greater than 3.5. Overall, the analysis of the Movielens20M dataset using Pyspark has provided valuable insights that could assist filmmakers and movie enthusiasts in understanding user preferences and making informed decisions

Future Scope: -

Future scope ideas for data analysis on the Movielens 20M dataset using PySpark:

1. **Recommendation Engine**:
   Build a recommendation engine using PySpark's machine learning library to predict user preferences and recommend movies to them.

2. **Genre Analysis**:
   Analyze the distribution of movies across genres and identify which genres are popular among different age groups and genders.

3. **User Segmentation**:
   Segment users based on their movie preferences, demographics, and other characteristics to understand their behaviour and tailor recommendations to their needs.

4. **Time-series Analysis**:
   Use PySpark's time-series analysis library to analyze trends in movie ratings over time and identify patterns that can help predict future trends.

5. **Network Analysis**:
   Build a network of users based on their movie preferences and social connections to identify influential users and understand how information about movies spreads through the network.

6. **Deep Learning**:
   Build a deep learning model using PySpark's distributed deep learning library to analyse movie data and make predictions about user preferences and behaviour.

7. **Anomaly Detection**:
   Use PySpark's machine learning library to detect anomalies in movie ratings and identify instances of fraud or fake reviews.

References: -

1) https://grouplens.org/datasets/movielens/
2) https://spark.apache.org/docs/latest/api/python/
3) https://www.analyticsvidhya.com/blog/2020/11/a-must-read-guide-on-how-to-work-with-pyspark-on-google-colab-for-data-scientists/
4) https://www.researchgate.net/publication/288041090_The_MovieLens_Datasets
5) https://arxiv.org/abs/1909.12799
6) https://www.semanticscholar.org/paper/The-MovieLens-Datasets%3A-History-and-Context-Harper-Konstan/276ebc620a8976026bd2d03582b9ecfa3738d43c
7) https://sparkbyexamples.com/pyspark-tutorial/
8) https://en.wikipedia.org/wiki/MovieLens
9) https://en.wikipedia.org/wiki/Apache_Spark