VISHNU GOPAL RAJAN

TEAM 12

1001755911

Project 3 Report

Overall status:

I was able to successfully complete the MapReduce part of the program. The setting up of the Hadoop single node cluster was the difficult part since the instructions were not sufficient. I implemented the mapper part of the program by setting the key to Year range and genre. The value was 1 for each of the keys which are summed up matching with its respective key in the reducer.

Based on the values from the mapper and reducer, I was able to plot the graphs for each of the genre groups over the years. I also plotted a graph of the 3 genres over the years depicting the growth of the genres over the years.

SQL

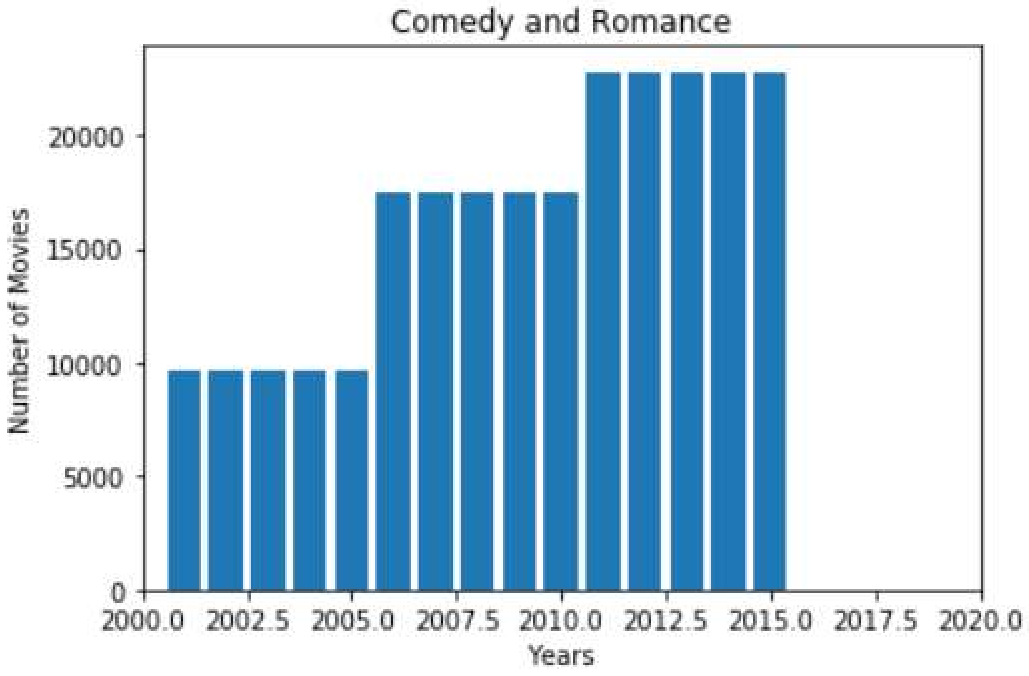
Analysis Results:

Task 1:

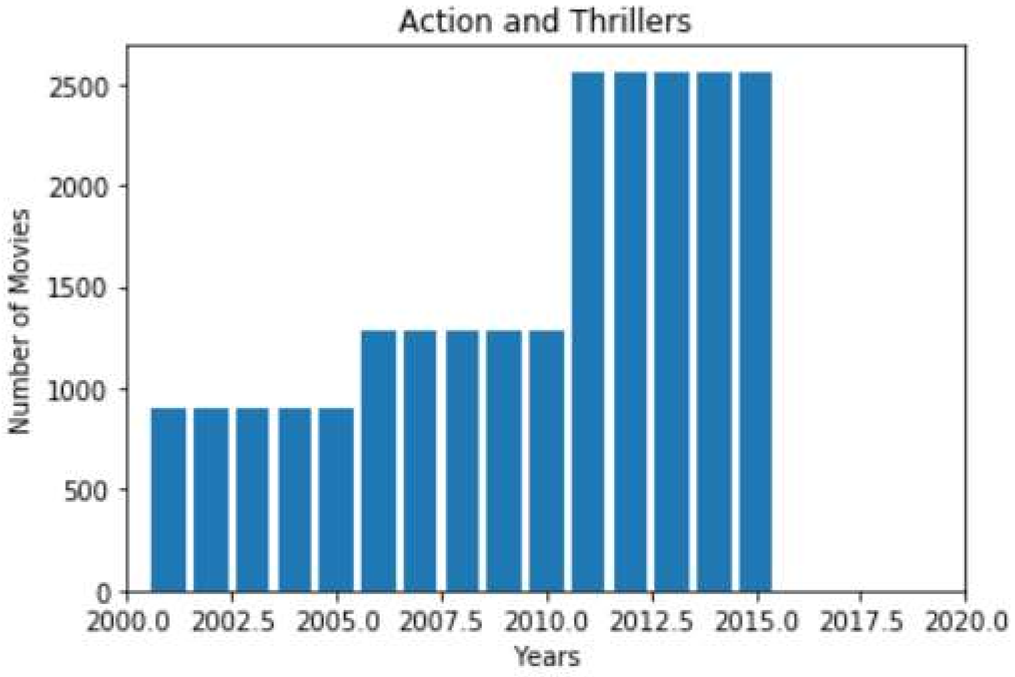
- The MapReduce output for the IMDB data values for the different genres over the 3 periods is :

```
[2001-2005], Action;Thriller        904
[2001-2005], Adventure;Sci-Fi       513
[2001-2005], Comedy;Romance         9674
[2006-2010], Action;Thriller        1279
[2006-2010], Adventure;Sci-Fi       674
[2006-2010], Comedy;Romance         17498
[2011-2015], Action;Thriller        2562
[2011-2015], Adventure;Sci-Fi       1027
[2011-2015], Comedy;Romance         2279
```

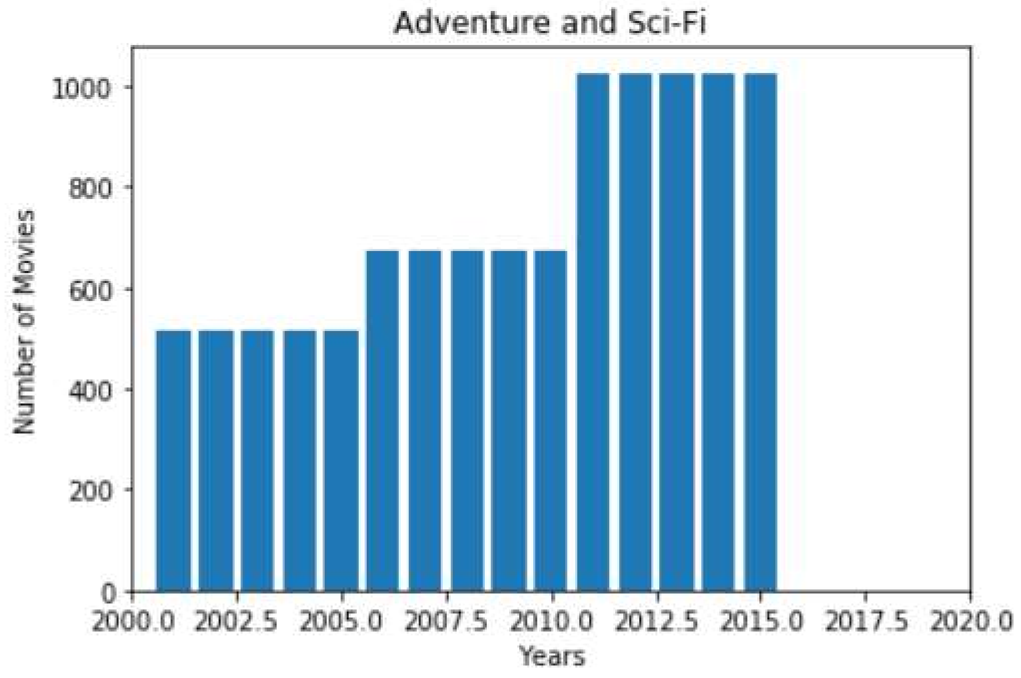
- Graph for the results of genre comedy and romance which shows the growth of this genre over the years.



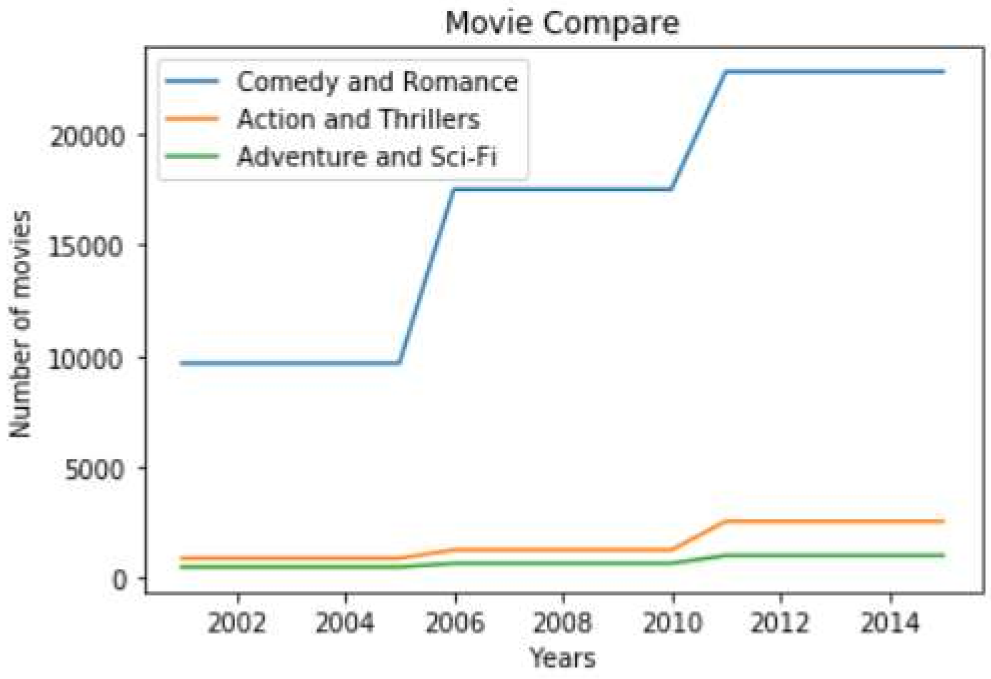
- Graph for the results of genre Action and Thrillers which shows the growth of this genre over the years.



- Graph for the results of Adventure and Sci-Fi which shows the growth of this genre over the years.



- This graph compares the 3 genres and their growth over the years



TASK 2:

SQL QUERIES:

TOP 5 ROMCOM

```
SELECT * FROM
(SELECT B.ORIGINALTITLE, B.GENRES, R.AVERAGERATING
FROM TITLE_BASICS B, TITLE_RATINGS R
WHERE B.ENDYEAR BETWEEN '2001'AND '2005'
AND B.GENRES LIKE '%Romance%Comedy%'
AND B.TCONST=R.TCONST
ORDER BY R.AVERAGERATING DESC)
WHERE ROWNUM <=5;
```

LAST 5 ROMCOM

```
SELECT * FROM
(SELECT B.ORIGINALTITLE, B.GENRES, R.AVERAGERATING
FROM TITLE_BASICS B, TITLE_RATINGS R
WHERE B.ENDYEAR BETWEEN '2001'AND '2005'
AND B.GENRES LIKE '% Romance%Comedy %'
AND B.TCONST=R.TCONST
ORDER BY R.AVERAGERATING)
WHERE ROWNUM <=5;
```

TOP 5 ACTION THRILLER

```
SELECT * FROM
(SELECT B.ORIGINALTITLE, B.GENRES, R.AVERAGERATING
FROM TITLE_BASICS B, TITLE_RATINGS R
WHERE B.ENDYEAR BETWEEN '2001'AND '2005'
AND B.GENRES LIKE '%Action%Thriller%'
AND B.TCONST=R.TCONST
ORDER BY R.AVERAGERATING DESC)
WHERE ROWNUM <=5;
```

LAST 5 ACTION THRILLER

```
SELECT * FROM
(SELECT B.ORIGINALTITLE, B.GENRES, R.AVERAGERATING
FROM TITLE_BASICS B, TITLE_RATINGS R
WHERE B.ENDYEAR BETWEEN '2001'AND '2005'
AND B.GENRES LIKE '%Action %Thriller%'
AND B.TCONST=R.TCONST
ORDER BY R.AVERAGERATING)
WHERE ROWNUM <=5;
```

<u>TOP 5 ADVENTURE SCI-FI</u>

```
SELECT * FROM
(SELECT B.ORIGINALTITLE, B.GENRES, R.AVERAGERATING
FROM TITLE_BASICS B, TITLE_RATINGS R
WHERE B.ENDYEAR BETWEEN '2001'AND '2005'
AND B.GENRES LIKE '%Adventure%Sci-Fi%'
AND B.TCONST=R.TCONST
ORDER BY R.AVERAGERATING DESC)
WHERE ROWNUM <=5;
```

<u>LAST 5 ADVENTURE SCI-FI</u>

```
SELECT * FROM
(SELECT B.ORIGINALTITLE, B.GENRES, R.AVERAGERATING
FROM TITLE_BASICS B, TITLE_RATINGS R
WHERE B.ENDYEAR BETWEEN '2001'AND '2005'
AND B.GENRES LIKE '%Adventure%Sci-Fi%'
AND B.TCONST=R.TCONST
ORDER BY R.AVERAGERATING)
WHERE ROWNUM <=5;
```

- I have used this query plan since it's the most efficient. Instead of searching through the entire database to match the primary key of ORIGINAL_TITLE with TITLE_RATINGS, I have narrowed the search by taking the range of values between the years 2001 and 2005. After narrowing this I have matched the Genres that are required for the search. After this I have matched the primary key TCONST between the ORIGINAL_TITLE table and TITLE_RATINGS table. Once this is done I am sorting the rating values in ascending and descending order and extracting only the top 5 and the last 5 row values.
- This plan is the most efficient because we are not traversing unwanted rows, unnecessarily.

File Descriptions:

- Some of the files I had to create for the MapReduce part of the program were classes folder to hold the compiled java files. I had to use this folder to create a jar folder which I used to run the code on the IMDB data.

Division of Labor:

This project was successfully completed by me.

Logical Errors:

- I had many problems with the set up of Hadoop. This set up took me like few days to successfully finish and run the single node cluster. Video on youtube helped me set up the single node cluster on Hadoop. https://www.youtube.com/watch?v=5rJTPMLKsq0&t=2900s
- Another logical error I faced during implementation of map reduce was NumberFormatException. This exception was thrown because I was using Integer.ParseInt(). After a few searches on google I figured out on how to catch this exception.
- Another logical error I faced was with the implementation of array list to store the genre values in order to run the .contains() function to check for the required genres.