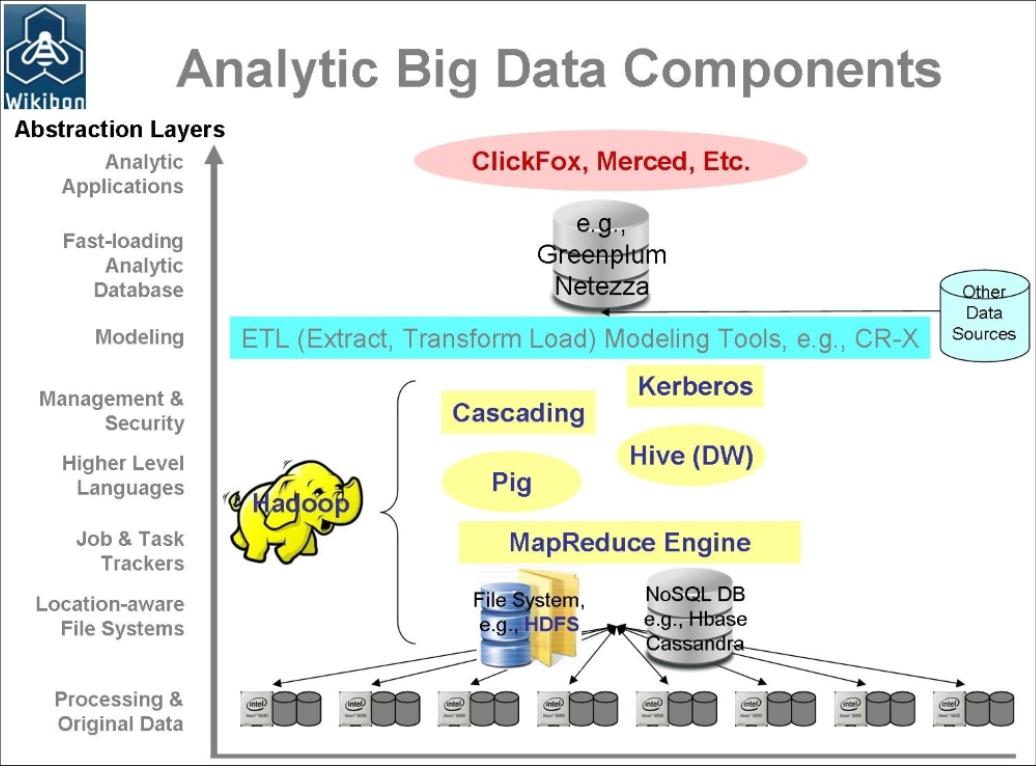
**INTRODUCTION TO BIG DATA TECHNOLOGIES**



**Hadoop:** Hadoop is an open source software programming framework for storing a large amount of data and performing the computation. Its framework is based on Java programming with some native code in C and shell scripts.

**Sqoop:** It is used to import and export data to and from between HDFS and RDBMS.

**Pig:** It is a procedural language platform used to develop a script for MapReduce operations.

**Hive:** It is a platform used to develop SQL type scripts to do MapReduce operations.

**Map reduce:** It is a parallel programming model for processing large amounts of structured, semi-structured, and unstructured data on large clusters of commodity hardware.

|  |  |
| --- | --- |
| **(Poc1.csv )**   * **Movietitle** – name of the movies * **imdbRating­** – rating of the movies * **duration**- duration of movie in seconds * **year**- release year of the movie * **Genre**- (Action, Documentary, Drama, Family, - Fantasy, War, Sport, Show, Adult, Adventure, Thriller, etc.)   **(Doc1.csv)**   * **Movietitle**: tell title of the movie. * **id­** – tell the id of respective movie * **rating**: tell the movie rating * **votecount**: tell the number of vote and review given to the respective movie. |  |

**HADOOP PROBLEM STATEMENTS**

* TO MAKE THE DIRECTORY
* TO DISPLAY THE LIST OF DIRECTORIES
* LOADING FILES IN DIRECTORY
* CHECK FOR FILE EITHER LODED IN OUR DATABASES OR NOT
* RETRIEVE CONTENT OF LOADED FILE
* CREATE THE FILE
* DISPLAY THE FILE
* TO SHOW THE HIDDEN FILES
* TO DISPLAY ALL THE PRESENT WORKING DIRECTORIES
* TO REMOVE THE DIRECTORY

**HIVE PROBLEM STATEMENTS**

1. SHOW THE MOVIE WITH THE MAXIMUM DURATION.
2. SELECT MOVIE WHICH START WITH LETTER “D”
3. SELECT THE MOVIE WITH RATING MORE THAN 8.7
4. SELECT THE MOVIES WHICH IS FORMED BETWEEN 1950-1960
5. SELECT THE MOVIES WHICH IS FORMED BETWEEN 1950-1960 AND OUT OF THEM TELL WHICH IS HORROR, IF HORROR THEN WRITE “1” ELSE “0”
6. PRINT THE LEAST AND HIGHEST RATING IN THE DATASET.
7. PRINT ALL THE MOVIES WITH DIFFERENT GENRE.

**PIG PROBLEM STATEMENTS**

1. LOADING DATA
2. DUMPING DATA (PRINTING DATA)
3. UNION OF TWO DOCUMENT
4. PRINT ALL THE RATING GREATER THAN 8.7
5. FIND OUT THE MAXIMUM, MIN AND AVERAGE RATING

**COMPARISON STUDY IN TIME OF EXECUTION**

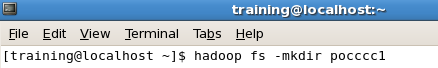
1. DIFFERENCE BETWEEN EXECUTION TIME BETWEEN HIVE AND PIG. COMPARE THE PERFORMANCE OF HIVE AND PIG BY FINDING THE MOVITITLE WITH RATING GREATER THAN 8.7
2. COMPARISON BETWEEN MAP REDUCE AND PIG FOR FINDIND WORD COUNT OF THE DOCUMENT

**PYTHON PROBLEM STATEMENTS**

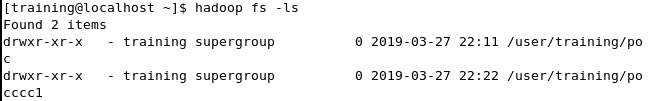
1. Hadoop(HDFS Commands)
2. Hive
3. Pig
4. Python
5. Map Reduce

**HDFS COMMANDS**

* **TO MAKE THE DIRECTORY**



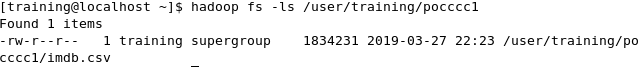
* **TO DISPLAY THE LIST OF DIRECTORIES**



* **LOADING FILES IN DIRECTORY**

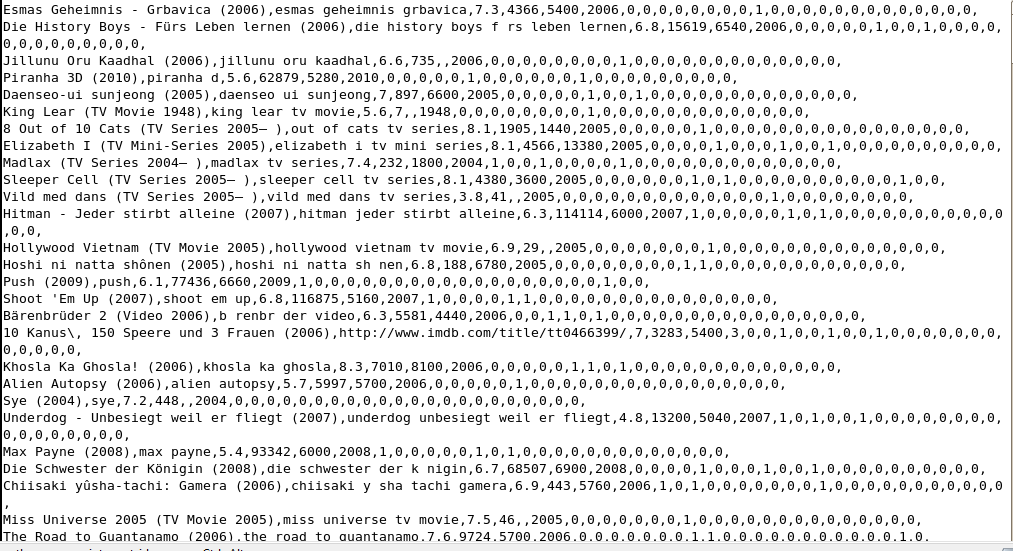


* **CHECK FOR FILE EITHER LODED IN OUR DATABASES OR NOT**

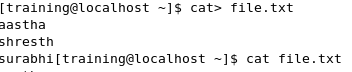


* **RETRIEVE CONTENT OF LOADED FILE**

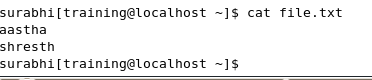




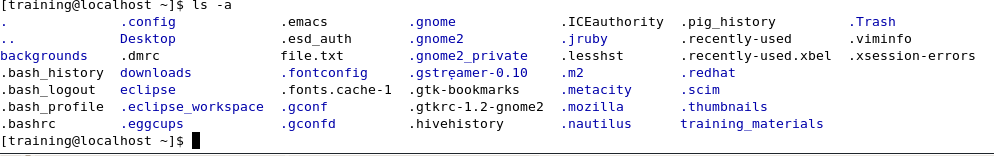
* **CREATE THE FILE**



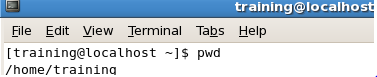
* **DISPLAY THE FILE**



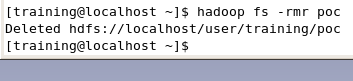
* **TO SHOW THE HIDDEN FILES**



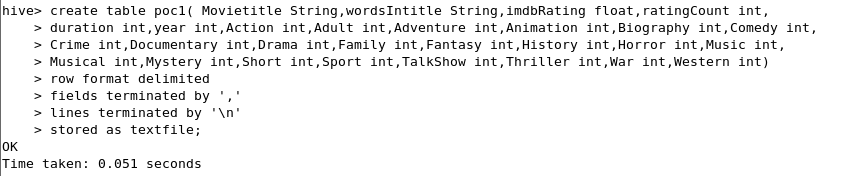
**TO DISPLAY ALL THE PRESENT WORKING DIRECTORIES**



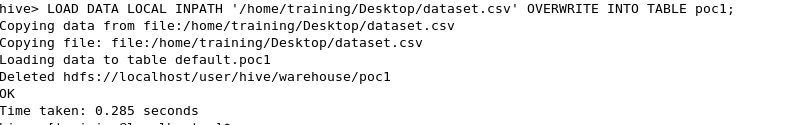
**TO REMOVE THE DIRECTORY**



* **CREATE THE TABLE**

****

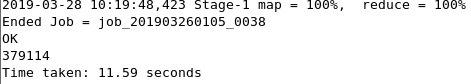
* **LOAD THE DATA FROM DATASET TO THE TABLE**

****

**QUERIES**

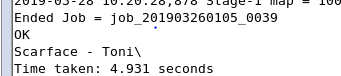
1. **Show the movie with the maximum duration**

**F:\poc\dur.PNG**

****

**F:\poc\dur1.PNG**

* OUTPUT:



1. **Select movie which start with letter “D”**

**F:\poc\2.PNG**

* OUTPUT:

****

1. **Select the movie with rating more than 8.7**

**F:\poc\select.PNG**

* OUTPUT

****

1. **Select the movies which is formed between 1950-1960**

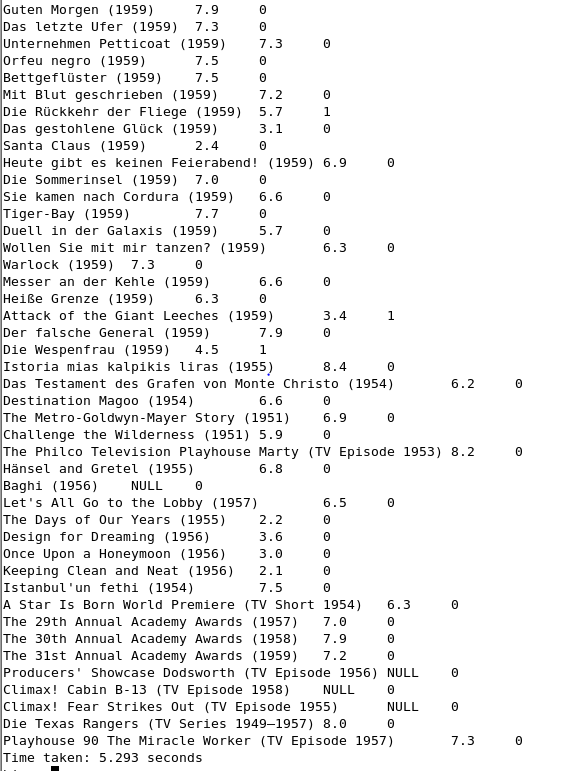
**F:\poc\4.PNG**

* OUTPUT:

****

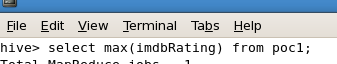
1. **Select the movies which is formed between 1950-1960 and out of them tell which is horror, if horror then write “1” else “0”**

**F:\poc\5.PNG**

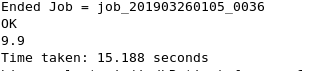
* OUTPUT:  
  

1. **Print the least and Highest Rating in the dataset.**

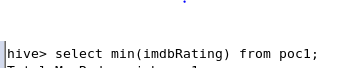
**MAX RATING**

****

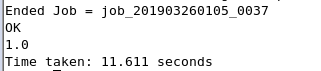
* OUTPUT:

****

**MIN RATING**



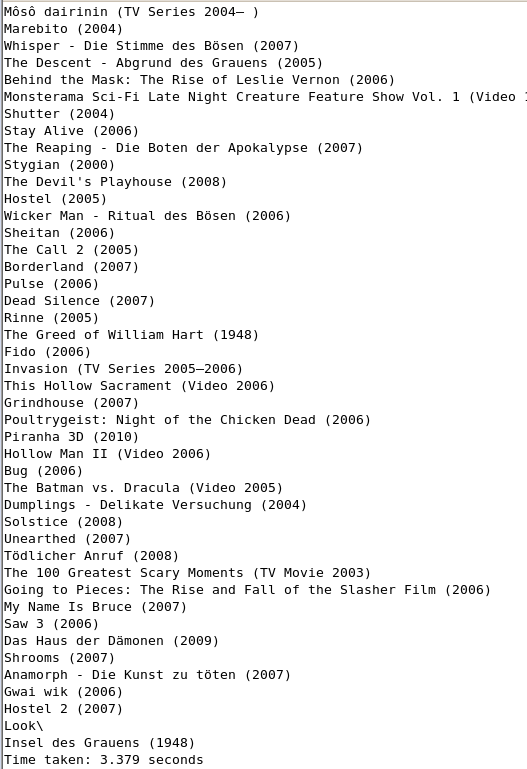
* OUTPUT:

****

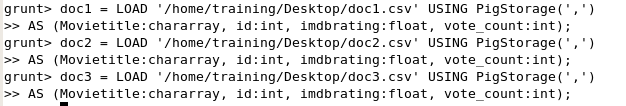
1. **Print all the movies with different genre.**

**F:\poc\h.PNG**

* OUTPUT:



**Loading data**

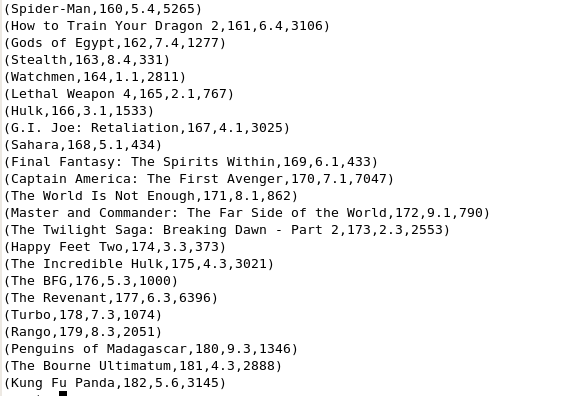




**DUMPING DATA (PRINTING DATA)**



OUTPUT:



* **Union of two document**

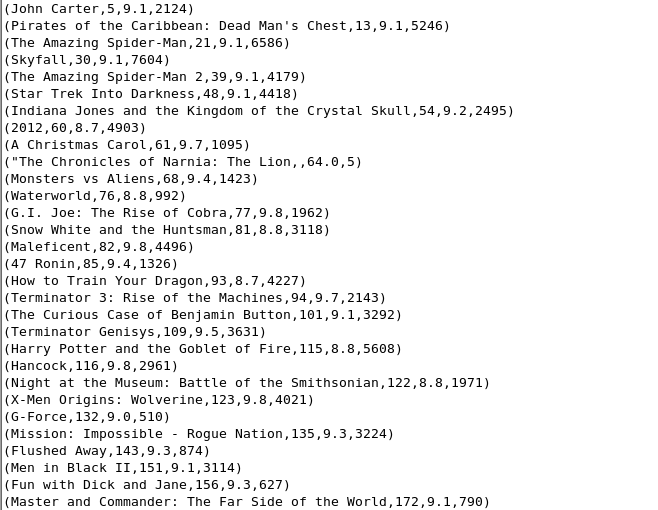


OUTPUT:



* **Print all the rating greater than 8.7**





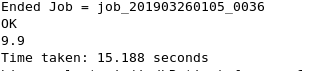
* **Find out the maximum, min and average rating**

**MAXIMUM**





OUTPUT:

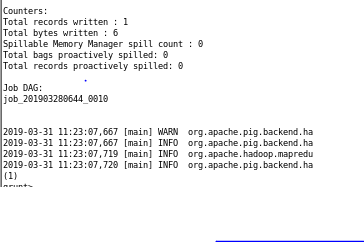
****

**MINIMUM**





OUTPUT:

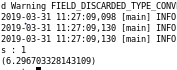


**AVERAGE**





OUTPUT:



* **Difference between execution time between hive and pig. Comparison between performance of hive and pig while finding the Movietitle with rating greater than 8.7**

Therefore Hive perform the query in 3.745 seconds and pig perform it in the 4 seconds. Therefore instead of being the most suitable language for large dataset, it take more time as compared to hive.

As shown in the picture:

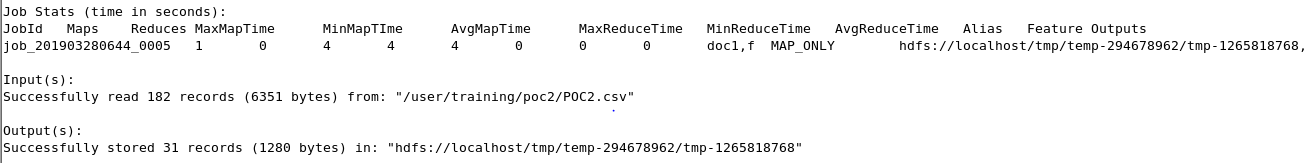
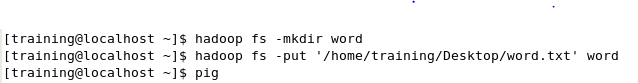


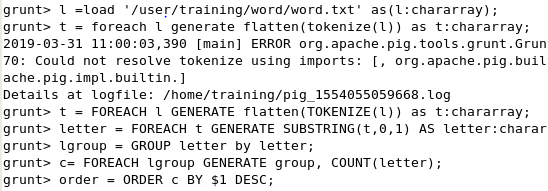
Figure. Pig



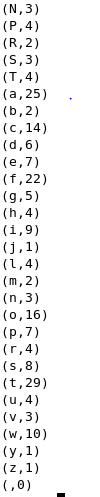
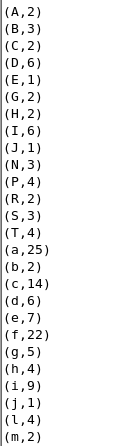
Figure. Hive

* **Comparison between map reduce and pig for finding word count of the document**
* **Pig**





OUTPUT:



**MAP REDUCE STATEMENTS**

* WORD COUNT OF THE FILE

**COMMAND:**

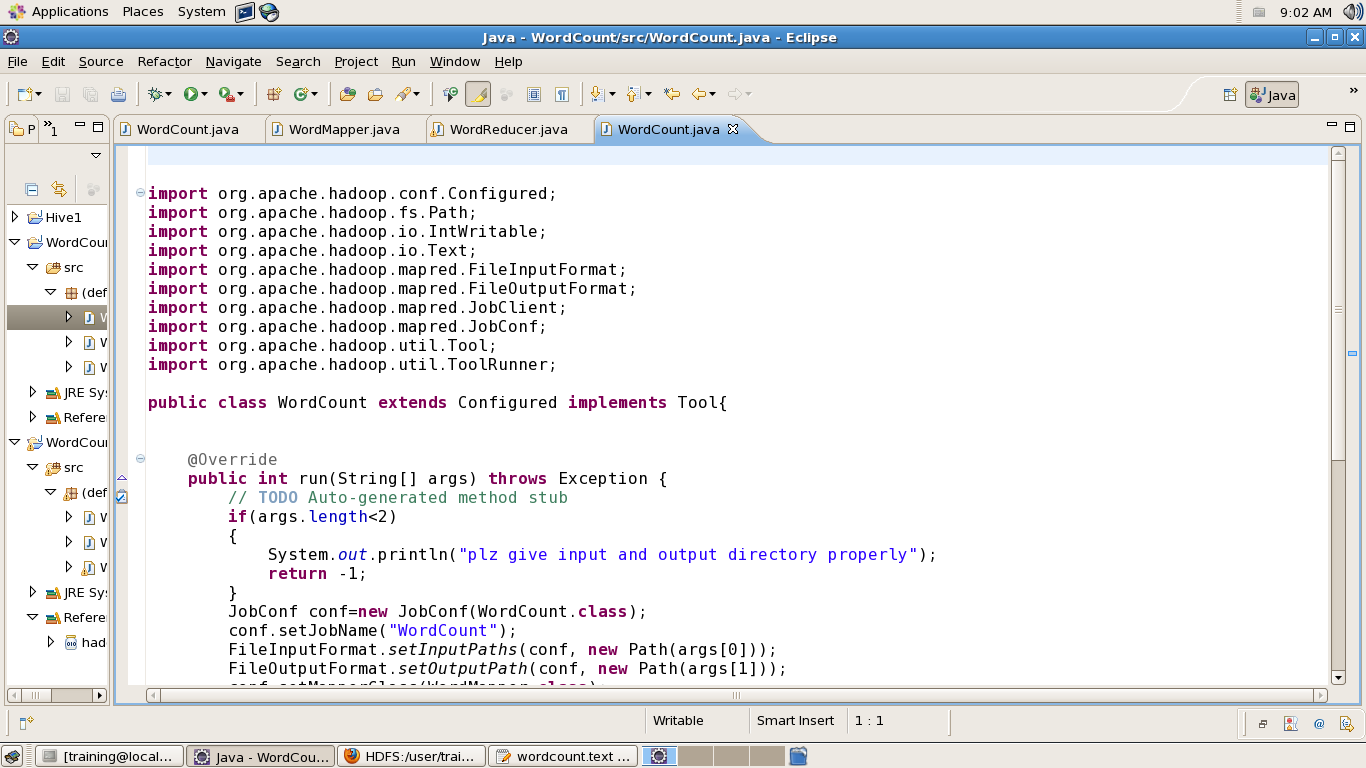
1. **Create a Java Project :WordCount**
2. **Add External Jar files:**

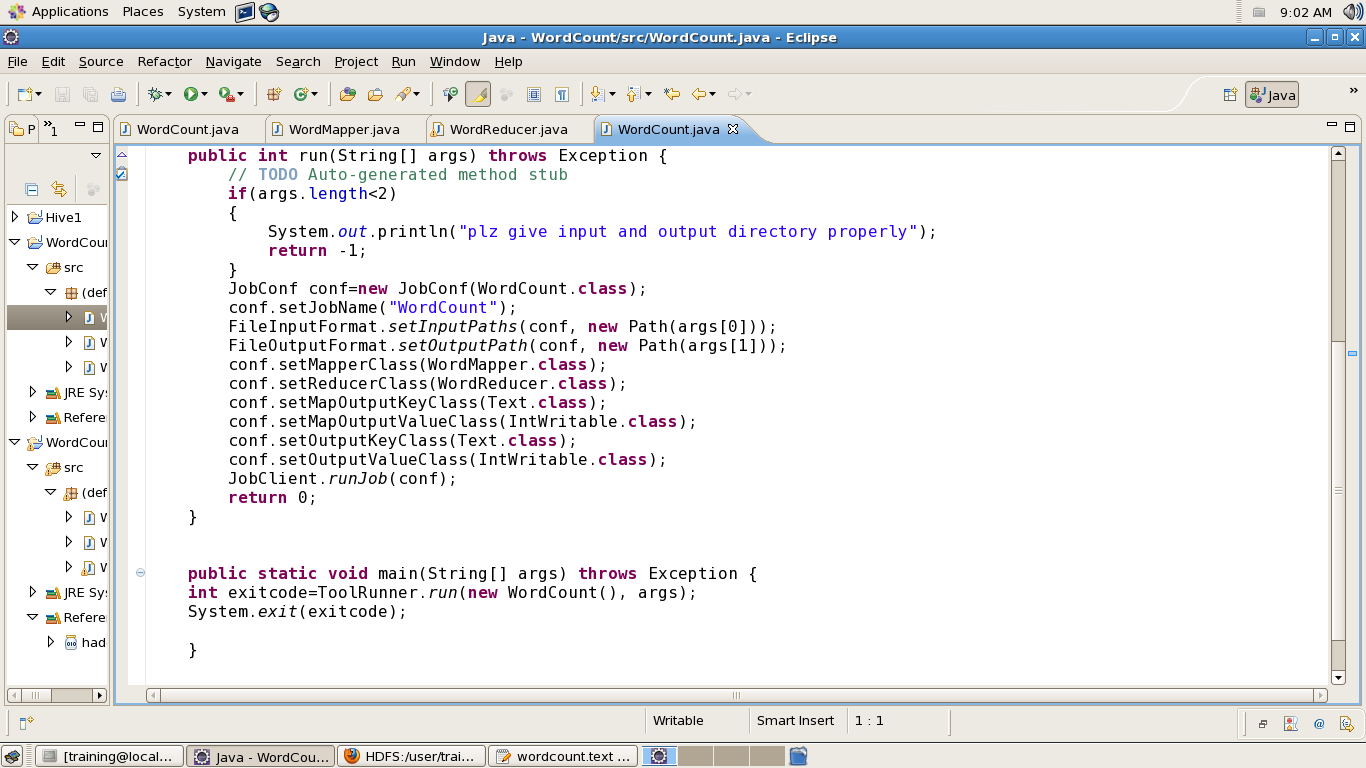
**Rightclick on project 🡪Properties 🡪Java BuildPath 🡪Add External jar Files 🡪**

**Then:**

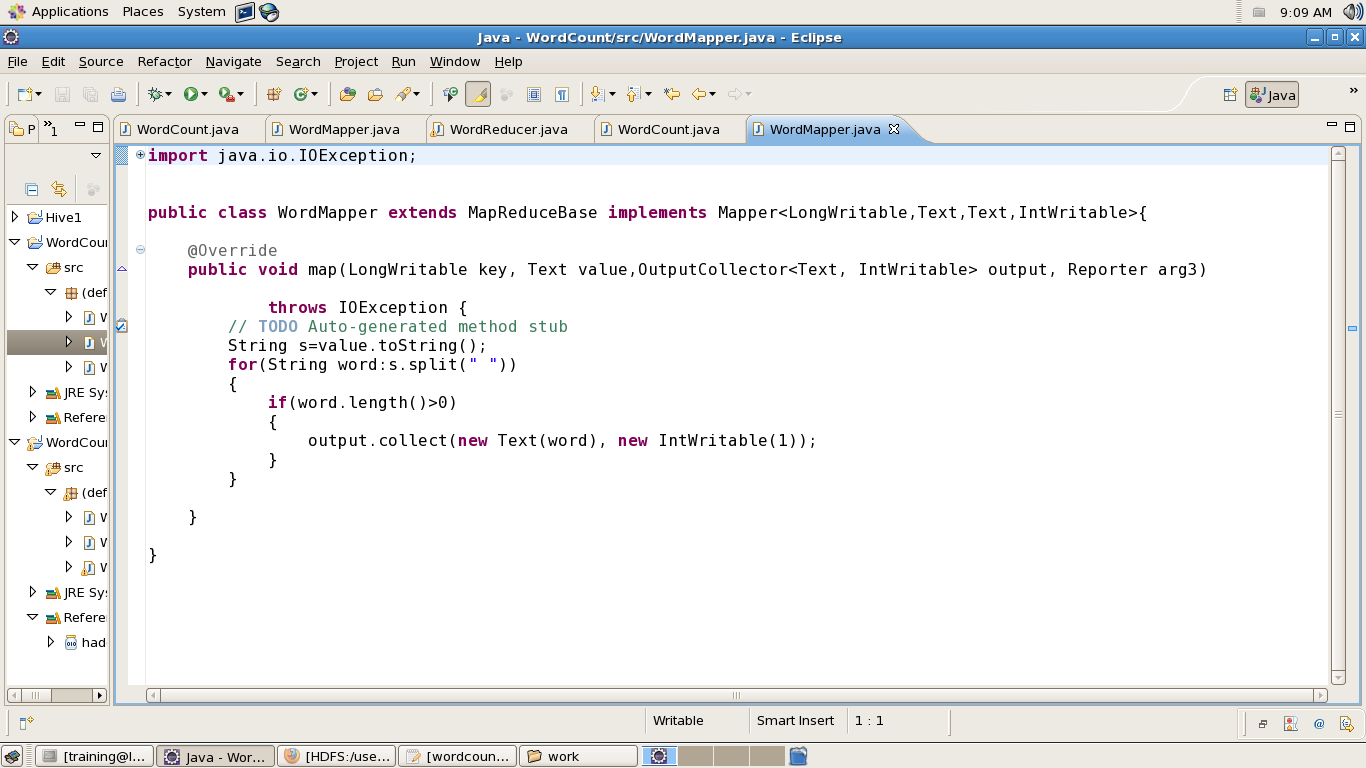
**click on FileSystem 🡪usr 🡪lib 🡪hadoop-0.20 🡪hadoop-core.jar 🡪ok 🡪ok**

1. **Create a class WordCount.java i.e your Driver Code**
2. **Create a class Mapper i.e WordMapper.java**
3. **Create a class Reducer i.e WordReducer.java**

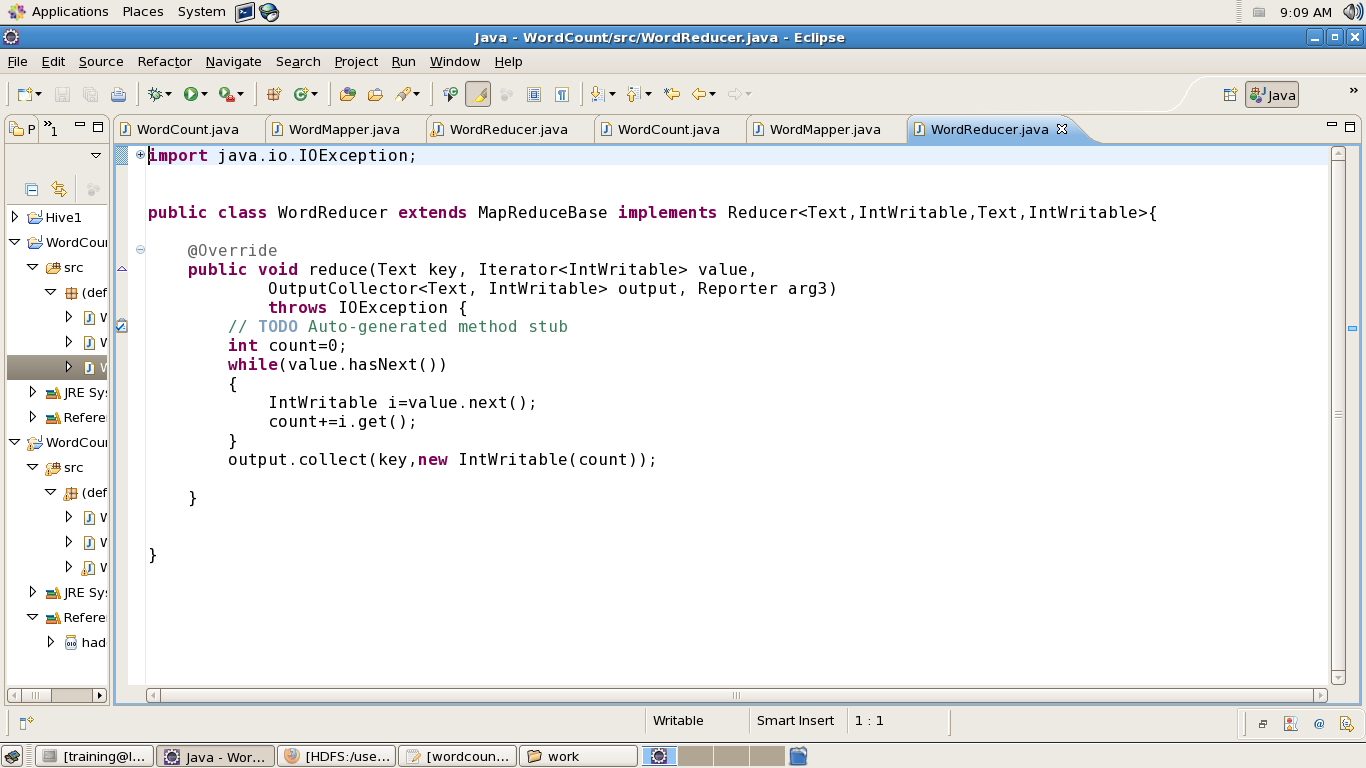




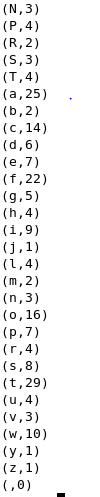
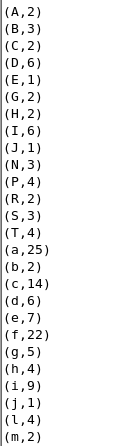
**REDUCER CLASS**



MAPPER CLASS



OUTPUT:



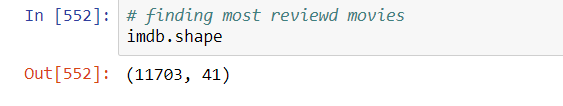
**Python**

**Tools used**: Anaconda package 5.0

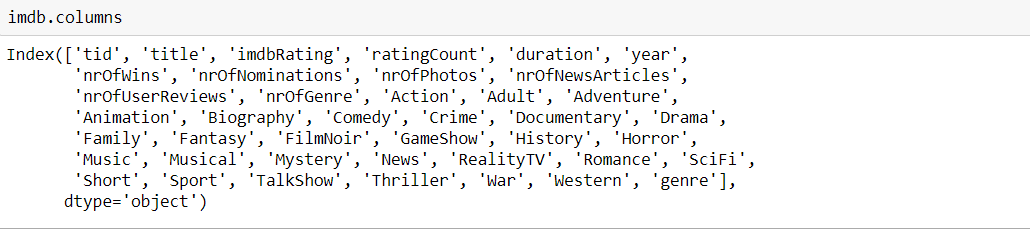
**Dataset Specifications:**

The dataset used for the given analysis is the dataset of the movie reviews and ratings given by the imdb.

It contains the data of 11703 movies with 41 different fields-



The columns used for analysis are as follows-

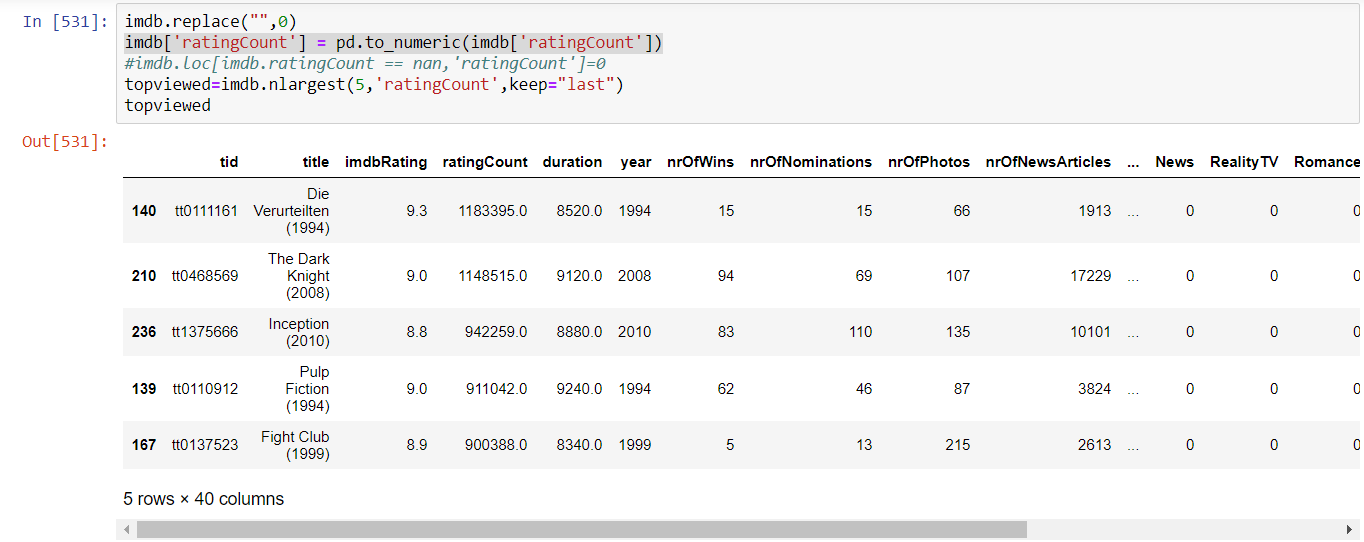


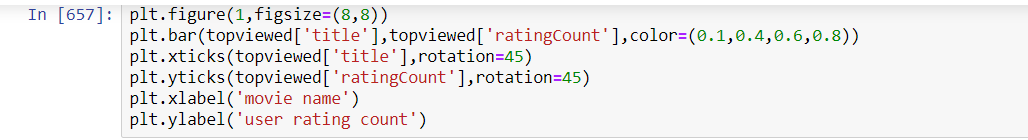
**Queries:**

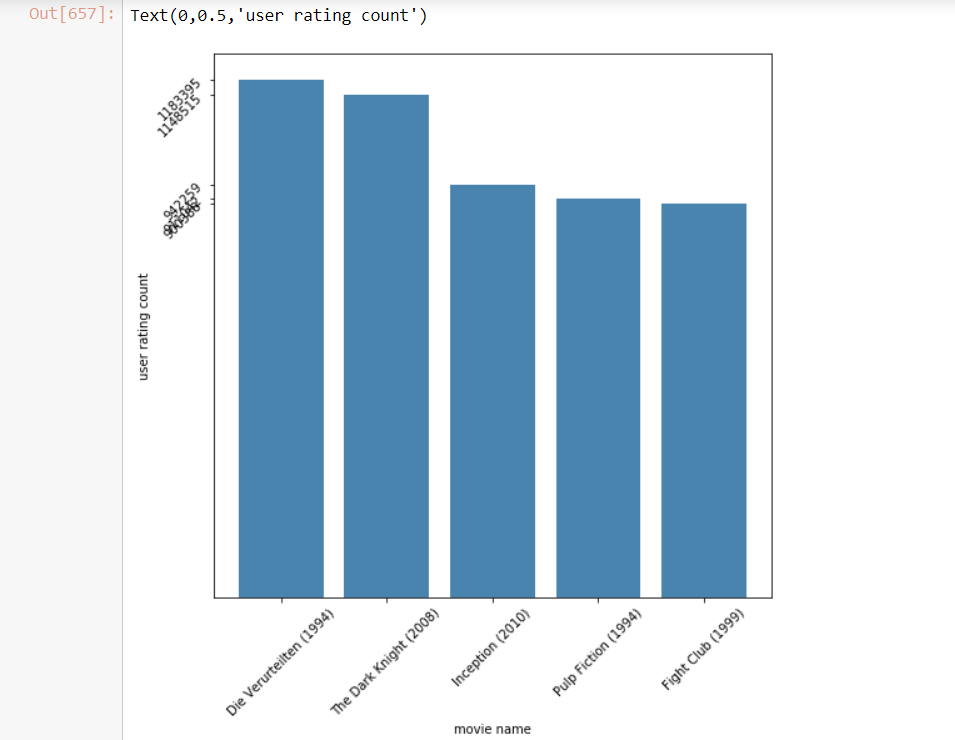
Following queries have been performed on the above dataset using the analysis and visualisation libraries of python in jupyter notebook-

1. **Maximum watched movies as per the user rating count-**

The number of user ratings can give us an insight over the number of people who actually went to watch the movie. Thus, using this field we can get an idea about the audience of the most watched movies. Below are the 5 movies with the maximum rating count-



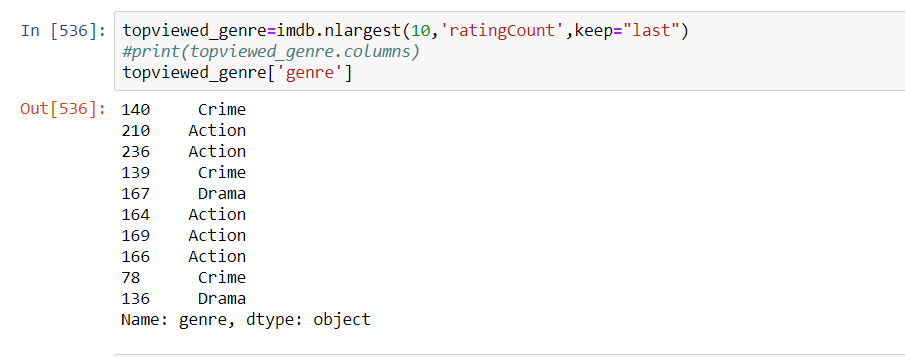




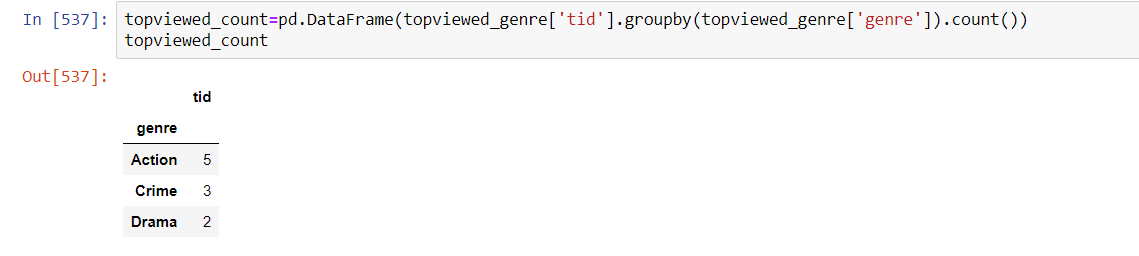
Based on above insight, it can also be deduced that the number of wins, rating and user rating count of a movie are closely correlated.

1. **Genre with maximum number of audience**-

Using the user rating count of the movies, it can also be deduced that which genres are more popular among audience. Below are the genre of the top 10 movies with highest user rating count.



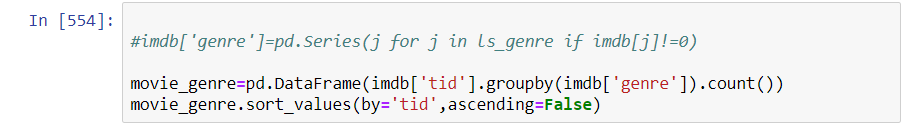
Looking at the above results, maximum number of movies watched by audience are of action genre.

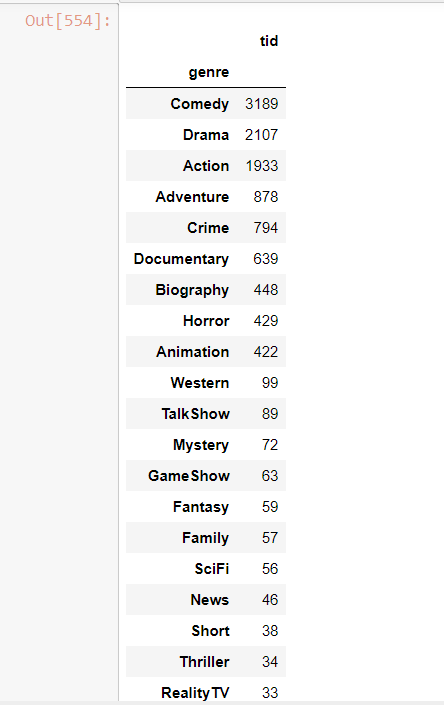


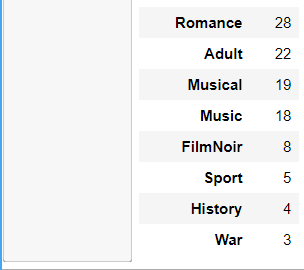
Thus Action is the most preferred genre of movies to watch among the audience followed by Crime and Drama.

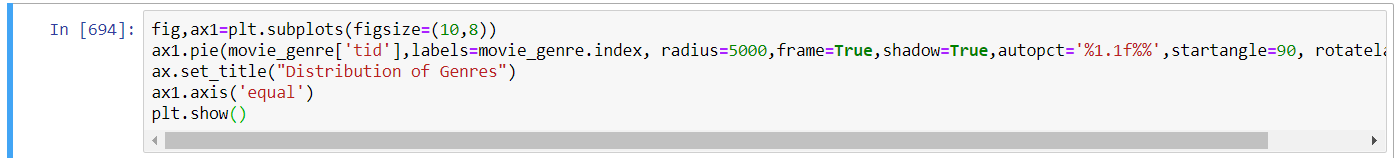
1. **Most popular genre in production-**

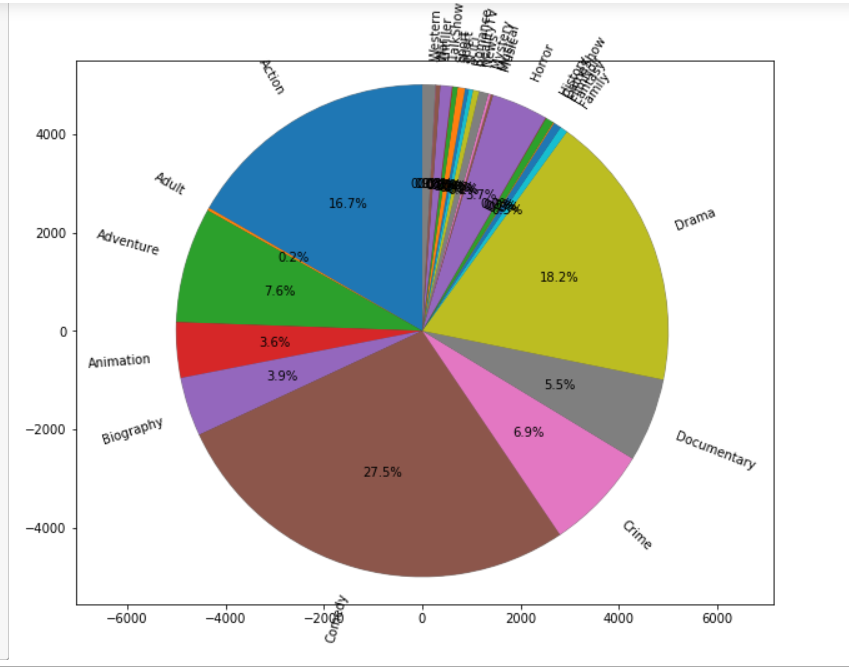
In the given dataset, the movies have been classified 28 into different genres which are as follows-





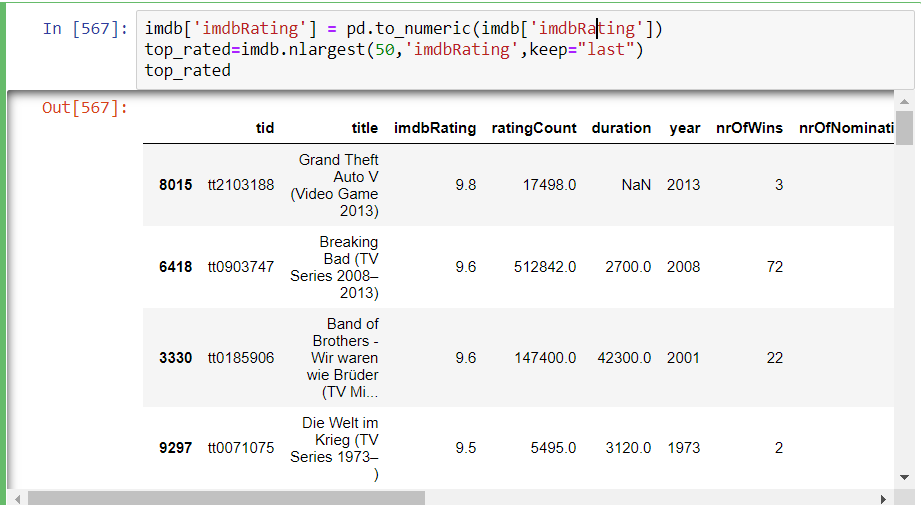






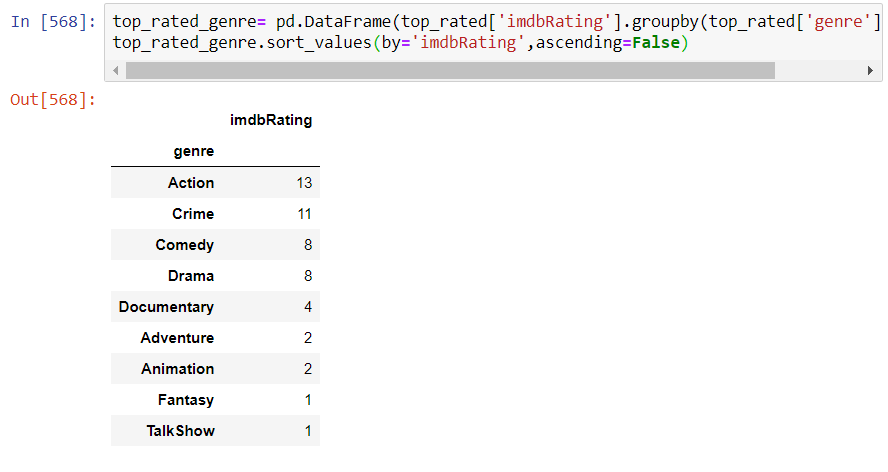
Based on the above results, it can be conferred that comedy is a preferred choice when it comes to media industry. It enjoys a major share of 27.5 % in total movie production followed by drama with a share of 18.2%.

1. **Genres of the highest rated movies-**



It can be observed from the above results that the highest rating goes as far as to 9.8

The genre of these 50 movies is distributed as follows-



It can be conferred that ‘Action’ movies are most liked by the audience. Most of the highest rated movies are action movies.