|  |
| --- |
|  |
| Movie Recommendation System |
| A Project Report |

|  |
| --- |
| Ariba Khan (k163831, gr2)  5/5/2019 |

**Problem Statement:**

In this day and age, as the online media is on the rise, more and more people are opting to watch movies online rather than downloading them or watching them on Television. Therefore it is vital for some movie based applications like Netflix to build a mechanism that classifies movies according to a users preference and make it easier for them to find a movie they prefer watching.

**Proposed Solution:**

A well known solution to the problem stated above is a movie recommendation system. This system helps users by recommending movies to them based on their previous ratings or movies they have already watched.

**Applying The Data Science Process:**

1. **Setting the research goal**

The goal of this project is stated above in the problem statement and the proposed solution.

1. **Retrieving Data**

The data used in this project was extracted from <https://grouplens.org/datasets/movielens/1m/>. It consisted of 3 files, namely:

* Movies.dat
* Users.dat
* Ratings.dat

* 1. **About The Data**
* **Movies.dat**

This file consisted of three movie attributes, namely, the movie title, the year of release and the genre (s)

* **ratings.dat**

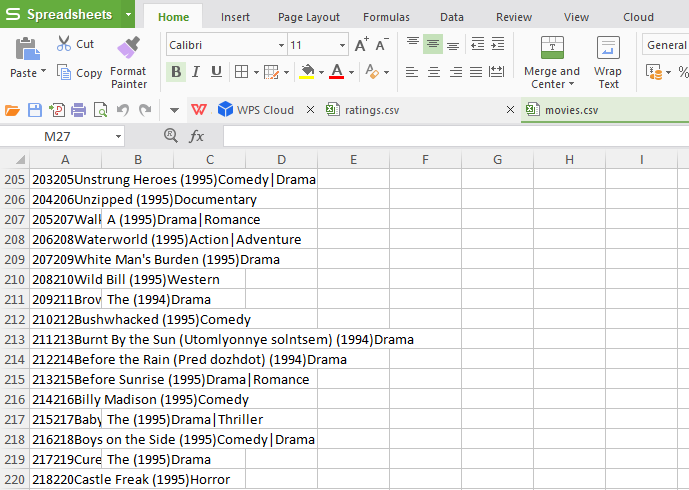
This file consisted of three attributes, namely user id, of the user who is giving the rating to a movie, the movie id to which a rating is being given to, the ratings, ranging from 1 to 5 stars with 1 being the lowest rating and 5 being the highest and lastly, the timestamp.

* **users.dat**

This file consists of all the demographic details of a user. These consist of their gender (M = male and F=female), their occupation (total 21 options), the area zip code and their age, ranging from under 18 to 56 and above.

1. **Data Preparation:**

* The files used in the dataset were in the form of “.dat”, so for ease of use they were converted into .csv format.
* Ad shown in Fig 1.0 there was no proper division of data in the csv file, therefore in the code, the columns were sperated using ‘/t’.

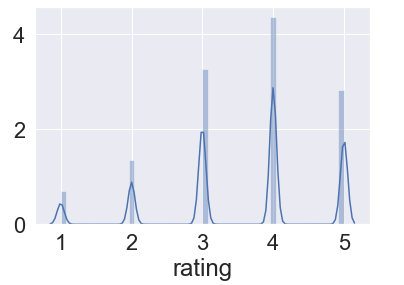


**Fig 1.0**

* While converting the ratings.dat file into ratings.csv, the timestamp column was deleted as the seconds at which a movie was rated is irrelevant to our project.
* Unique values in our dataset, such as a movie rated twice were dropped in our code.
* In order to filter out the most preferred movies in our dataset, the three files had to be merged in such a way that that the movies with the highest rating given to a particular movie by the same user could be deduced from the data

1. **Exploratory Data Analysis:**

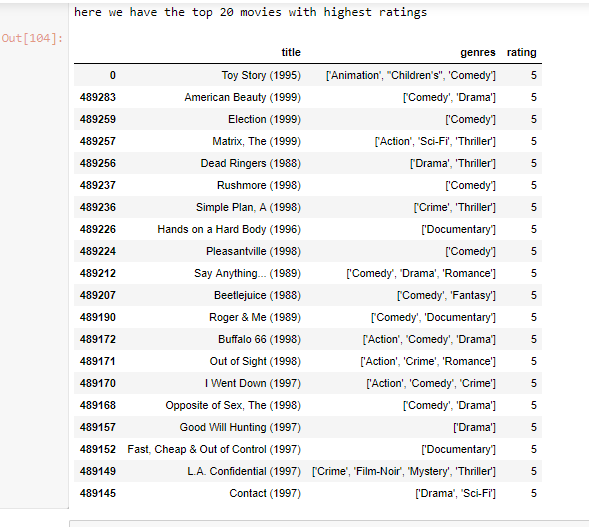
The data was analyzed in order to understand what was going around in our data set.



**Fig 2.0**

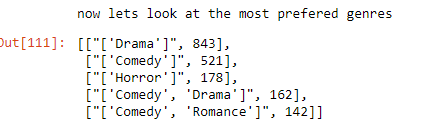
Figure 2.0 tells us that the users surveyed in our dataset were fairly neutral; this confirms the credibility of our data. They didn’t give extreme ratings to our users

( as low as 1 or as high as 5).

****

**Fig 3.0**

**Fig 3.0**

****

**Fig 3.1**

From Figures 3.0 and 3.1 we can see that the movies that are highly preferred in the dataset include movies falling in the genre of Drama, comedy, horror and romance.

1. **Building Models**There are two models that are widely used in recommendation systems, either collaborative filtering or content based models. My project has utilized Content Based Filtering
   1. **Content Based Filtering**

This model works on the logic that if a person likes a certain object (a movie in our case) then it is likely that they may prefer another object that is similar to the first object (i.e the rating of a movie).

**5.2 Term Frequency and Inverse Document Frequency (TFIDF)**

TFIDF are very popular when it comes to recommendation systems. They are used to find the importance of a word in a text, or in our project, the importance of genres, movie names and ratings.

Consider we come across a movie (M) that user A likes to watch. For this we will look at the ratings they have given to a movie of a particular genre. We then see which movies “come closer” to the movie M by user A. that is, we look at the movies to check which of them are the most similar to movie A.

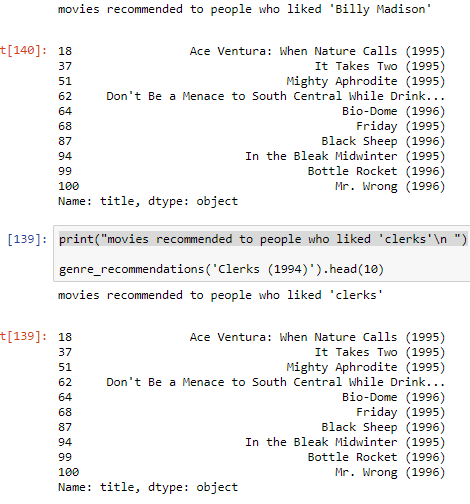
For our model we used cosine similarity to find out which movies were closer to the movies a particular user likes to watch.

* 1. **The Working**

In order to calculate TFIDF scores, our project used **cosine similarity** to find out which movie was closer to the preferred movie of a particular user.

We broke up the genres given in out datasets to string values for comparison purposes. We imported TfidVectorizer in our project which converts the strings to be an input in the vectors used in out model.

**6.The Results**



**Fig 4.0**

Figure 4.0 shows the final output of our project. A user enters a movie they prefer watching and in response 10 movies with a high similarity score to that movie are shown in the output and recommended to the user.

**References**

* <https://medium.com/python-pandemonium/introduction-to-exploratory-data-analysis-in-python-8b6bcb55c190>
* <https://stackoverflow.com/questions/29530232/how-to-check-if-any-value-is-nan-in-a-pandas-dataframe>
* <https://towardsdatascience.com/evaluation-metrics-for-recommender-systems-df56c6611093>
* <https://www.youtube.com/watch?v=ZOX18HfLHGQ>  
  https://www.youtube.com/watch?v=iYie42M1ZyU&t=60s