MOVIE RECOMMENDATION SYSTEM

USING MACHINE LEARNING

**A Project Report for Industrial Training**

###### **Submitted by**

**ABHIRUP NANDI[BPPIMT-IT(2ND YR)]**

***In the partial fulfillment for the award of the degree of***

##### **B.Tech**

At

**Ardent Computech Pvt. Ltd.**



**Dec - Jan 2017-2018**



**CERTIFICATE FROM SUPERVISOR**

This is to c ertify that *ABHIRUP NANDI* successfully completed the project titled **"** MOVIE RECOMMENDATION SYSTEM USING MACHINE LEARNING" under my supervision during the period from DEC-JAN.

*Signature of the Supervisor*

**Date:**  **Upasak Pal**

**Project Supervisor**

**Ardent Collaborations**

**Acknowledgement**

The achievement that is associated with the successful completion of any task would be incomplete without Stuti ma’am and Joyjit Sir whose endless cooperation made it possible. Their constant guidance and encouragement made all our efforts successful.

We take this opportunity to express our deep gratitude towards our project mentor, *[Stuti ma’am]* for giving such valuable suggestions, guidance and encouragement during the development of this project work. We are truly grateful to have ma’am as our mentor because she encourages us to implement new things in the project.

Last but not the least we are grateful to all the faculty members of Ardent Computech Pvt. Ltd. for their support.

**LIST OF CONTENTS :**

* PYTHON AND IT’S FEATURES-
* ANACONDA AND THE IPYHTON SHELL-
* THE PYTHON PACKAGES-
* INTRODUCTION OF MACHINE LEARNING
* TYPES OF MACHINE LEARNING
* TYPES OF OF SUPERVISED MACHINE LEARNING
* APPLICATION OF MACHINE LEARNING-RECOMMENDATION SYSTEM
* MOVIE RECOMMENDATION SYSTEM USING CF
* INTRODUCTION TO MOVIE RECOMMENDATIO N SYSTEM
* PROBLEM STATEMENT
* DATA USAGE
* PROGRAM CODE
* EXPERIMENTAL RESULTS
* SOURCES
* **PYTHON :**

****

* Python is an interpreted high-level programming language for general purpose programming.
* Python was developed by Guido van Rossum and was released in 1991.
* The Python Software Foundation (PSF) is the organization behind Python.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural. Python interpreters are available for many operating systems.

* PYTHON VERSION:
* First version released in Jan 1994 i.e. Python 1.0
* Second version released in Oct 2000 i.e. Python 2.0
* Third version released in Dec 2008 i.e. Python 3.0

**CURRENT VERSION:**

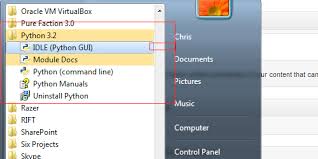
* Python 3.6.3
* Python 2.7.14
* PYTHON FEATURES:

Python provides lots of features that are listed below.

1. Easy to Learn and Use
2. Expressive Language
3. Interpreted Language
4. Cross-platform Language
5. Free and Open Source
6. Object-Oriented Language
7. Extensible
8. Large Standard Library
9. GUI Programming Support
10. Integrate

Python is widely used in many domains.

* Web Development
* Data Analysis
* Machine Learning
* Internet Of Things
* GUI Development
* Image Processing
* Data Visualization
* Game Development
* IDLE :



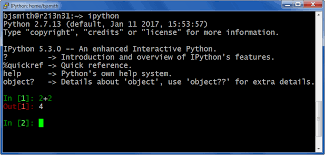
IDLE (Integrated Development Environment or Integrated Development and Learning Environment) is an integrated development environment for Python, which has been bundled with the default implementation of the language. IDLE has been criticized for various usability issues, including losing focus, lack of copying to clipboard feature, lack of line numbering options, and general user interface design.

* ANACONDA :



Anaconda is a freemium open source distribution of the Python and R Programming languages for large-scale data processing, predictive analytics, and scientific computing, that aims to simplify package management and deployment. Package versions are managed by the package management system conda.

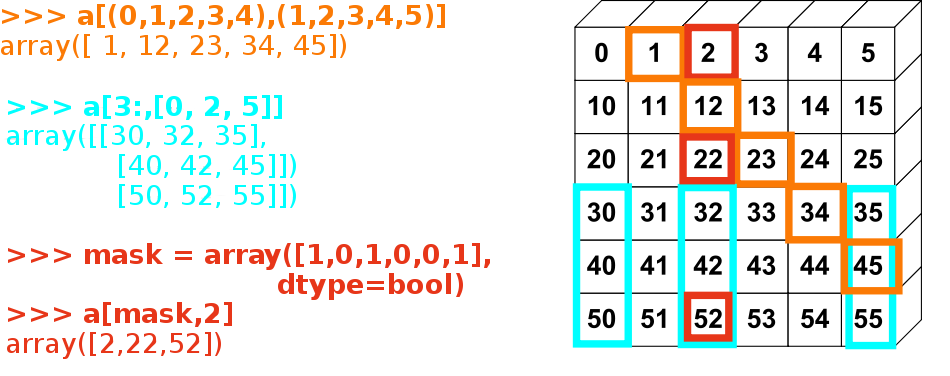
* IPYTHON:



IPython is a command shell for interactive computing in multiple programming languages, originally developed for the Python Programming Language that offers introspection , rich media , shell syntax, tab completion and history.

It provides following features :

* Interactive shells(terminal and Qt-based).
* A browser-based notebook with support for code, text, mathematical expressions, inline plots and other media.
* Support for interactive data visualization and use of GUI toolkits.
* Flexible, embeddable interpreters to load into one’s own projects.
* Tools for parallel computing.
* PACKAGES:
* NUMPY:



Numpy is an acronym for "Numeric Python" or "Numerical Python". It is an open source extension module for Python, which provides fast precompiled functions for mathematical and numerical routines. Furthermore, Numpy enriches the programming language Python with powerful data structures for efficient computation of multi-dimensional arrays and matrices. The implementation is even aiming at huge matrices and arrays. Besides that the module supplies a large library of high-level mathematical functions to operate on these matrices and arrays.

Numpy is the fundamental package for scientific computing with Python. It contains among other things:

* a powerful N-dimensional array object
* sophisticated (broadcasting) functions
* tools for integrating C/C++ and Fortran code
* useful linear algebra, Fourier transform, and random number capabilities

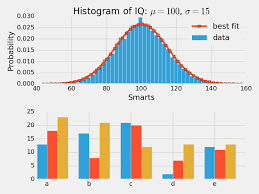
Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows Numpy to seamlessly and speedily integrate with a wide variety of databases.



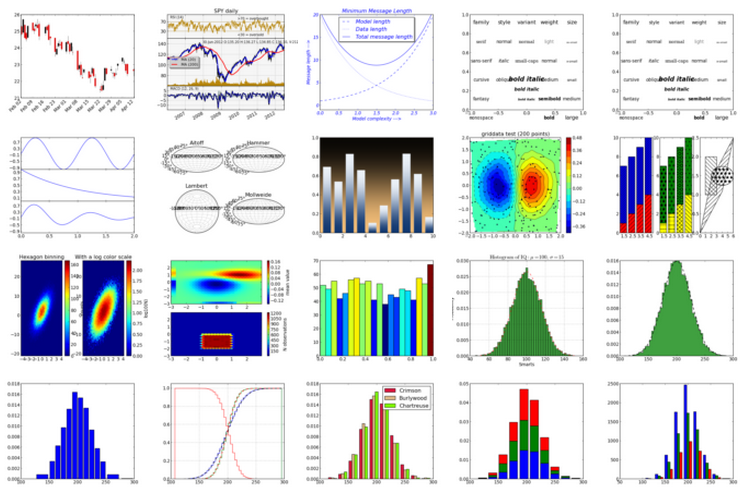
Scipy (Scientific Python) is often mentioned in the same breath with Numpy. Scipy extends the capabilities of Numpy with further useful functions for minimization, regression, Fourier-transformation and many others.

* MATPLOTLIB:

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [IPython](http://ipython.org/) shell, the [jupyter](http://jupyter.org/index.html) notebook, web application servers, and four graphical user interface toolkits.

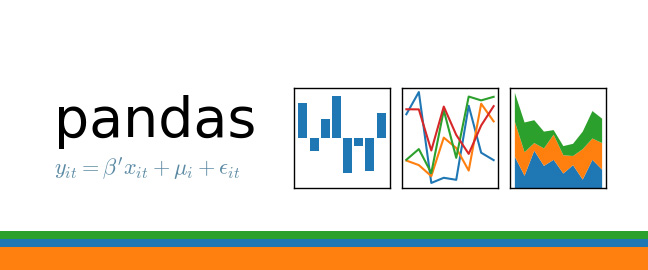


Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter charts, etc., with just a few lines of code.



For simple plotting the pyplot module provides a MATLAB like interface , particularly when combined with IPython. For the power user, you have full control of the line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

* PANDAS:



Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

* SCIKIT-LEARN:

**Scikit-learn**  is a [free software](https://en.wikipedia.org/wiki/Free_software" \o "Free software) [machine learning](https://en.wikipedia.org/wiki/Machine_learning" \o "Machine learning) [library](https://en.wikipedia.org/wiki/Library_(computing)" \o "Library (computing)) for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)" \o "Python (programming language)) programming language. It features various [classification](https://en.wikipedia.org/wiki/Statistical_classification" \o "Statistical classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis" \o "Regression analysis) and [clustering](https://en.wikipedia.org/wiki/Cluster_analysis" \o "Cluster analysis) algorithms including [support vector machines](https://en.wikipedia.org/wiki/Support_vector_machine" \o "Support vector machine), [random forests](https://en.wikipedia.org/wiki/Random_forests" \o "Random forests), [gradient boosting](https://en.wikipedia.org/wiki/Gradient_boosting" \o "Gradient boosting),

*[k](https://en.wikipedia.org/wiki/K-means_clustering" \o "K-means clustering)*[-means](https://en.wikipedia.org/wiki/K-means_clustering" \o "K-means clustering) and [DBSCAN](https://en.wikipedia.org/wiki/DBSCAN" \o "DBSCAN), and is designed to interoperate with the Python numerical and scientific libraries [Numpy](https://en.wikipedia.org/wiki/NumPy" \o "NumPy) and [Scipy](https://en.wikipedia.org/wiki/SciPy" \o "SciPy).

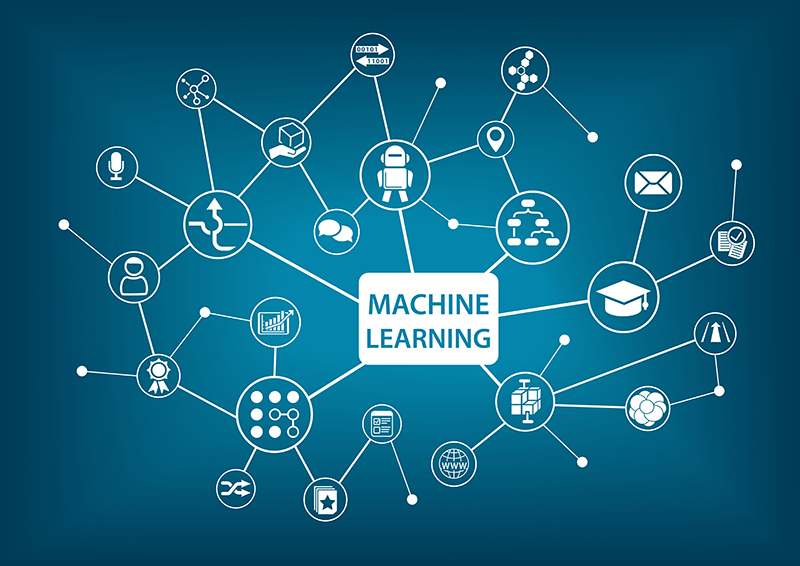
* SEABORN :

Seaborn is a Python Library that allows to build beautiful data visualizations. Seaborn turns out to be very effective during data exploration as it allows to do the following:

* Building histograms.
* Plotting Kernel Density Estimates.
* Building beautiful scatterplots in order to get a feeling about the data.
* Boxplots are also available in this library as they give you a sense on how data is distributed.
* Building Heatmaps…

In short, Seaborn would allow to quickly understand the data on hand in a more visual way

* **MACHINE LEARNING:**



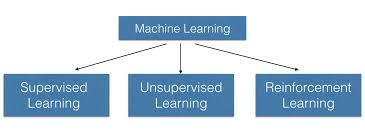
Machine learning is a type of artificial intelligence ([AI](http://searchcio.techtarget.com/definition/AI)) that allows software applications to become more accurate in predicting outcomes without being explicitly programmed. The basic premise of machine learning is to build [algorithms](http://whatis.techtarget.com/definition/algorithm) that can receive input data and use [statistical analysis](http://whatis.techtarget.com/definition/statistical-analysis) to predict an output value within an acceptable range.

Machine Learning can be used in -

* Data Security
* Personal Security
* Financial Trading
* Healthcare
* Marketing Personalization
* Fraud Detection
* Recommendations
* Online Search
* Natural Language Processing(NLP)
* Smart Cars

There are three types of Machine learning :

* Supervised Machine Learning
* Unsupervised Machine Learning
* Reinforced Machine Learning



**Supervised Machine Learning** : **Supervised learning** is the **machine learning** task of inferring a function from labeled training data.

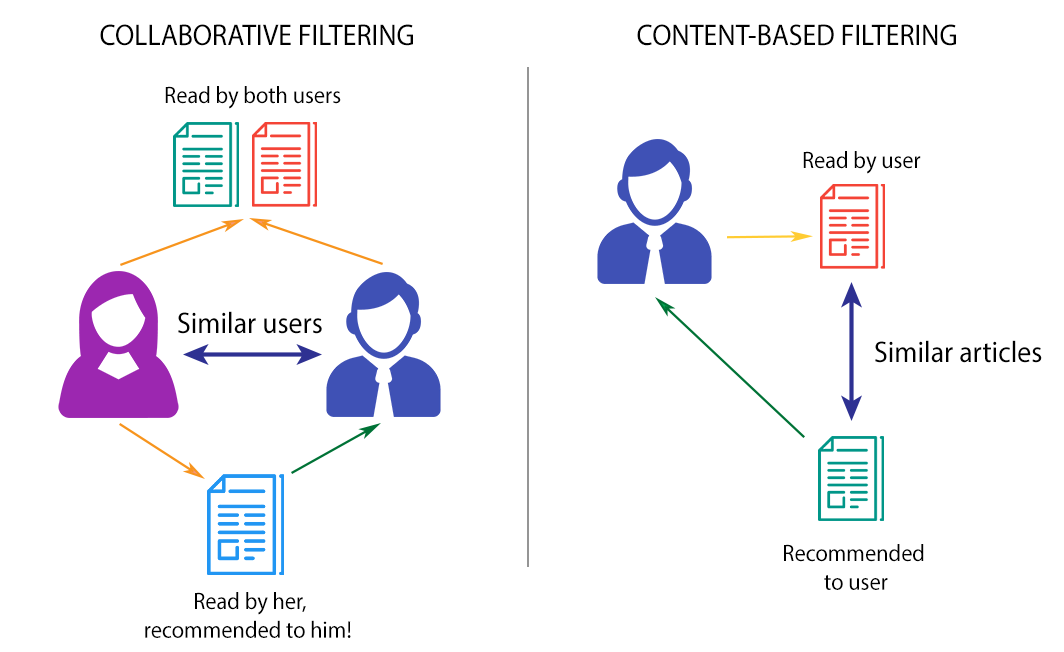
**Unsupervised Machine Learning : Unsupervised machine learning** is the [machine learning](https://en.wikipedia.org/wiki/Machine_learning" \o "Machine learning) task of inferring a function to describe hidden structure from "unlabeled" data (a classification or categorization is not included in the observations).

**Reinforced Machine Learning** : **Reinforcement learning** (**RL**) is an area of [machine learning](https://en.wikipedia.org/wiki/Machine_learning" \o "Machine learning) inspired by [behaviourist psychology](https://en.wikipedia.org/wiki/Behaviorism" \o "Behaviorism), concerned with how [software agent s](https://en.wikipedia.org/wiki/Software_agent" \o "Software agent)ought to take *[actions](https://en.wikipedia.org/wiki/Action_selection" \o "Action selection)* in an *environment* so as to maximize some notion of cumulative *reward.*

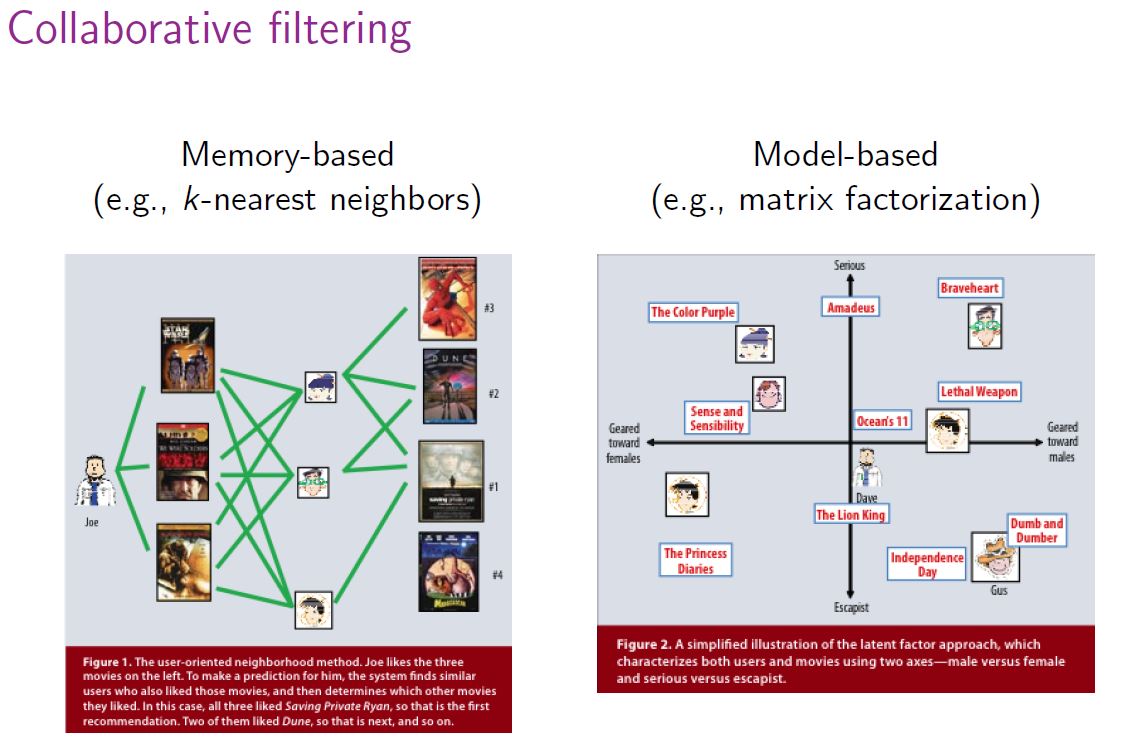
* SUPERVISED MACHINE LEARNING :

Supervised Machine Learning contains of two parts:

* Collaborative Filtering- “Tell me what’s popular among my peers”
* Content-Based Filtering-“show me more of the same what i have liked”



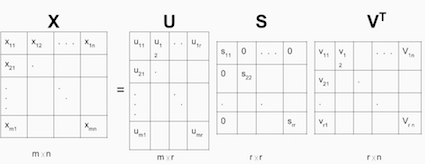
* COLLABORATIVE FILTERING:
* Model-Based Methods
* Memory-Based Methods



* **MODEL-BASED METHODS:**

Model-based Collaborative Filtering is based on **matrix factorization (MF)** which has received greater exposure, mainly as an unsupervised learning method for latent variable decomposition and dimensionality reduction. Matrix factorization is widely used for recommender systems where it can deal better with scalability and sparsity than Memory-based CF.

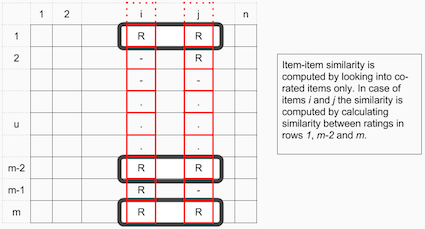
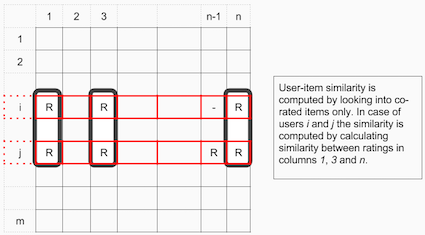
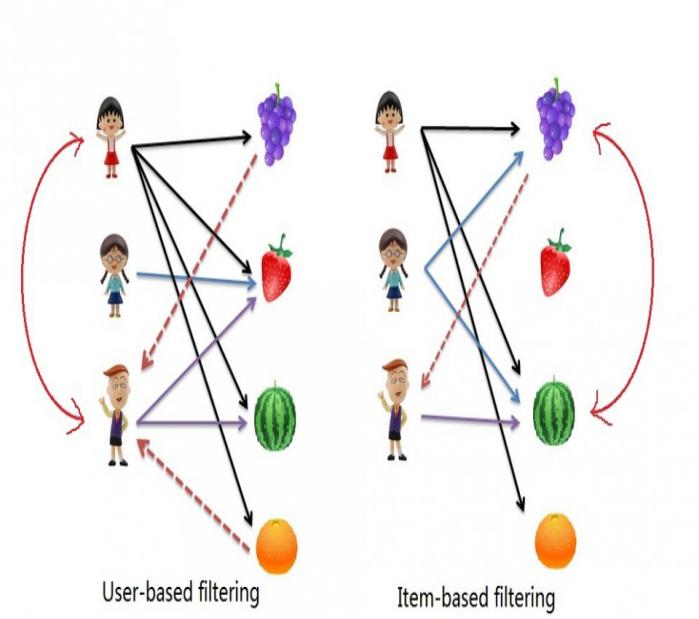
https://latex.codecogs.com/gif.latex?X=USV%5eT



* **MEMORY-BASED METHODS:**

Memory-Based Collaborative Filtering approaches can be divided into two main sections: **user-item filtering** and **item-item filtering**.

A user-item filtering will take a particular user, find users that are similar to that user based on similarity of ratings, and recommend items that those similar users liked.



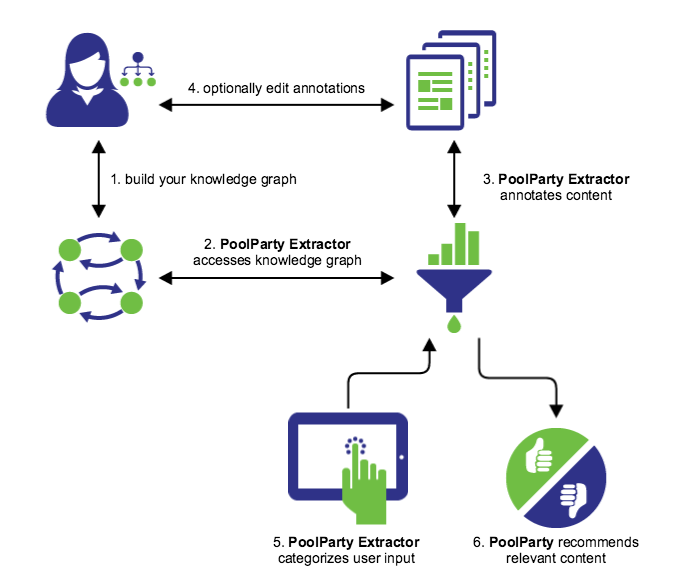
Cosine similiarity for users a and m can be calculated using the formula below, where you take dot product of the user vector uk and the user vector ua and divide it by multiplication of the Euclidean lengths of the vectors.

gif.gif

To calculate similarity between items m and b you use the formula:

gif (1).gif

* APPLICATION OF MACHINE LEARINING:
* RECOMMENDATION SYSTEM:



A **recommender system** or a **recommendation system** is a subclass of [information filtering system](https://en.wikipedia.org/wiki/Information_filtering_system" \o "Information filtering system) that seeks to predict the "rating" or "preference" that a user would give to an item.

Recommender systems have become increasingly popular in recent years, and are utilized in a variety of areas including movies, music, news, books, research articles, search queries, social tags, and products in general. There are also recommender systems for experts, collaborators, jokes, restaurants, garments, financial services, life insurance, romantic partners ([online dating](https://en.wikipedia.org/wiki/Online_dating" \o "Online dating)), and Twitter pages.

* CONTENT:
* INTRODUCTION
* PROBLEM STATEMENT
* EXPERIMENT RESULTS
* CONCLUSION
* INTRODUCTION:

• People seek information via words, recommendation letters, news reports etc.

• Recommendation systems imitate this social process to enable quick filtering of the information on the web

• Lots of companies try to offer services that involve recommendations to address the right user groups.

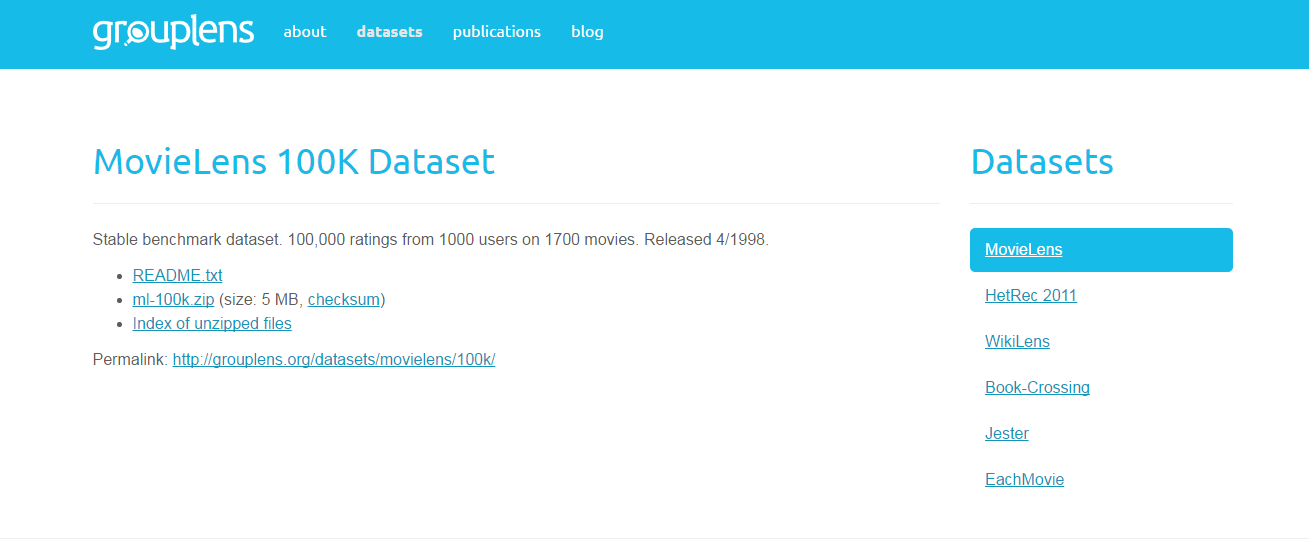
* PROBLEM STATEMENT:

**We want to build a movie recommendation system using the collaborative filtering method with the help of python.**

* DATA USAGE :

We use the below link for the datasets.

<http://grouplens.org/datasets/movielens/100k/>



Here are brief descriptions of the data.

u.data -- The full u data set, 100000 ratings by 943 users on 1682 items.

Each user has rated at least 20 movies. Users and items are

numbered consecutively from 1. The data is randomly

ordered. This is a tab separated list of

user id | item id | rating | timestamp.

The time stamps are unix seconds since 1/1/1970 UTC

u.info -- The number of users, items, and ratings in the u data set.

u.item -- Information about the items (movies); this is a tab separated

list of

movie id | movie title | release date | video release date |

IMDb URL | unknown | Action | Adventure | Animation |

Children's | Comedy | Crime | Documentary | Drama | Fantasy |

Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi |

Thriller | War | Western |

The last 19 fields are the genres, a 1 indicates the movie

is of that genre, a 0 indicates it is not; movies can be in

several genres at once.

The movie ids are the ones used in the u.data data set.

u.genre -- A list of the genres.

u.user -- Demographic information about the users; this is a tab

separated list of

user id | age | gender | occupation | zip code

The user ids are the ones used in the u.data data set.

u.occupation -- A list of the occupations.

* **PROGRAM:**

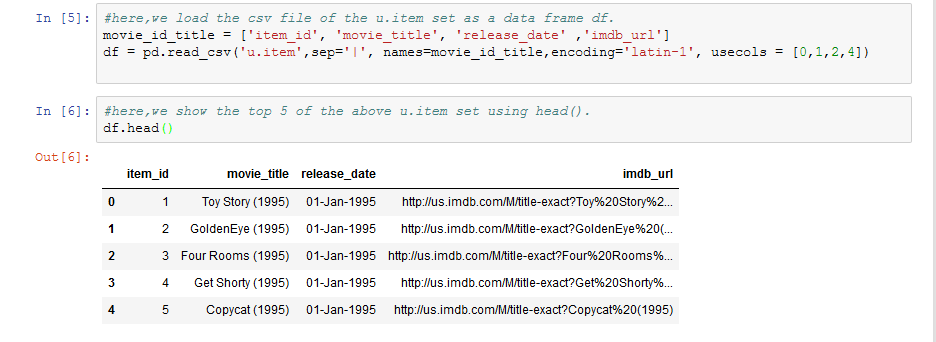
Here, we importing the python libraries then loading the csv file of u.data set as a data frame which is used to store tabular data,such as a spreadsheet or database. df.head() is displaying the top 5 data’s of u.data from the dataset.

* **u.data** – The full dataset, 100000 ratings by 943 users on 1682 items.



Here, we are loading the csv file of u.item set as a dataframe df and displaying the top 5 data’s of u.item from dataset.

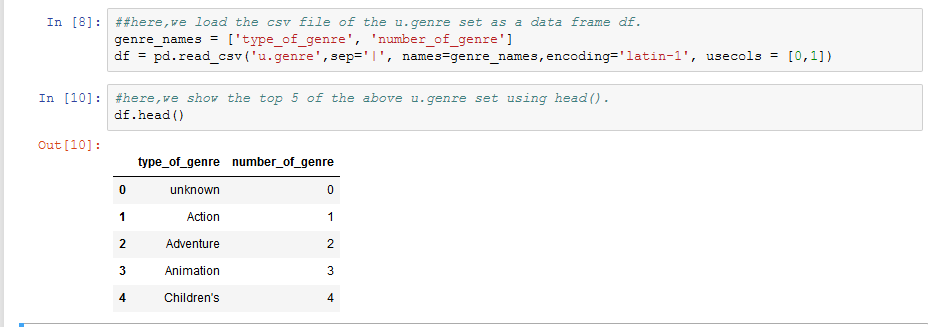
* **u.item** – Information about the items.



Here, we are loading the csv file of u.genre set as a data frame and displaying the top 5 data’s of u.genre from the dataset.

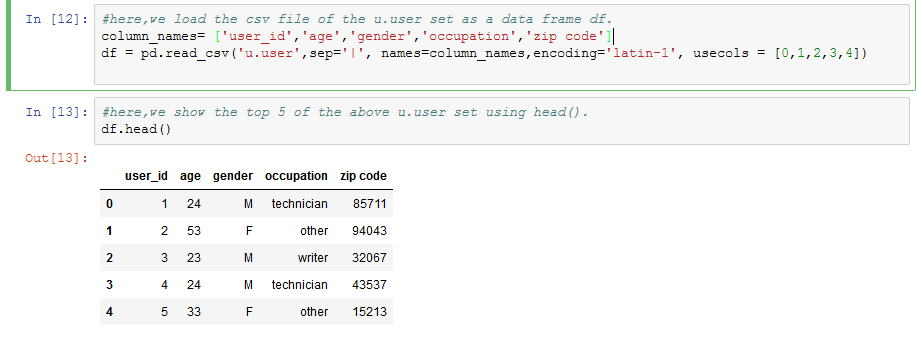
Sep is used for separator between the arguments to print the function & ‘|’ represents delimeter. Latin-1 is used for converting the code to byte values. Usecols is used to return the subset of the columns.

* **u.genre** – A list of genres.



Here, we are loading the csv file of u.user set as a dataframe df and displaying the top 5 data’s of u.user from dataset.

* **u.user** – Information about the user.

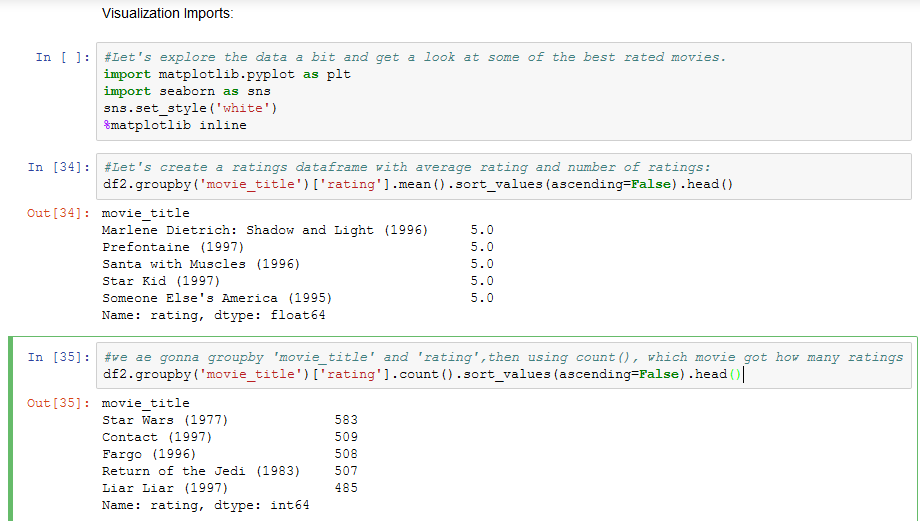


Here, we are merging both the dataset on a common key i.e item\_id and displaying the results of top data’s in dataset with df.head().



Here, we are importing matplotlib as well as seaborn package. Using groupby() we are grouping the rows of movie\_title together based off a column Rating and taking mean of it and displaying the result.[%matplotlib.inline is only for jupyter notebooks,if you are using another editor,you will use plt.show() at the end of all your plotting commands to have the figure pop up in another window.

* Seaborn – It is a visualization library like matplotlib.

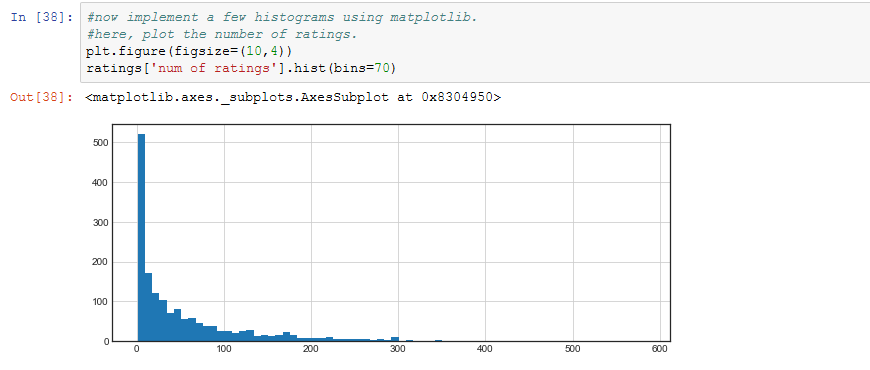


Here, we are calculating the mean of ratings as well as sum of ratings and displaying there results.



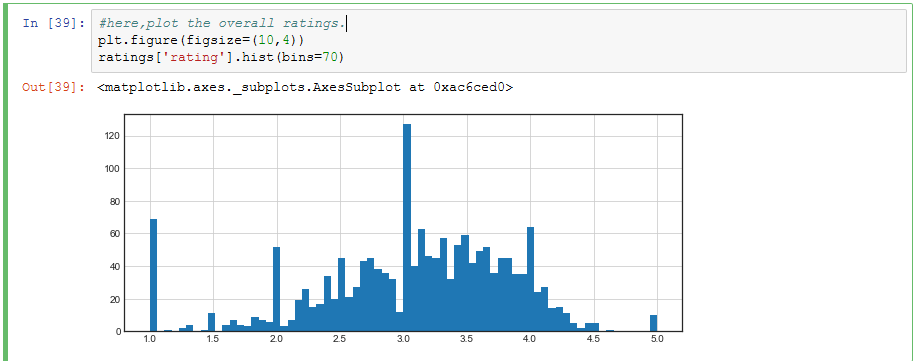
**Displaying Matplotlib Graph :**

Plot the number of ratings.



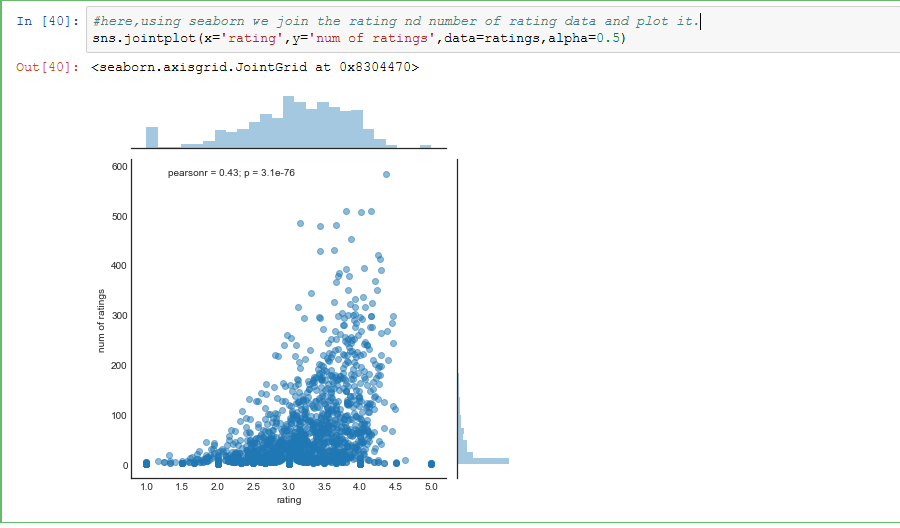
**Displaying Mathplotlib Graph :**

Plot the overall ratings.

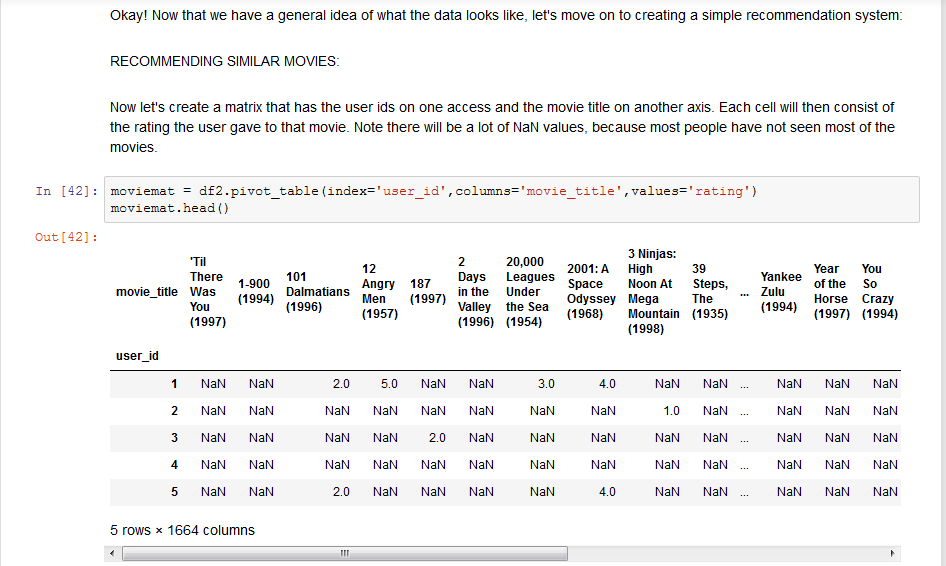


Here, we are joining both the plots and displaying the result in form of graph . Alpha is used for transparecy.

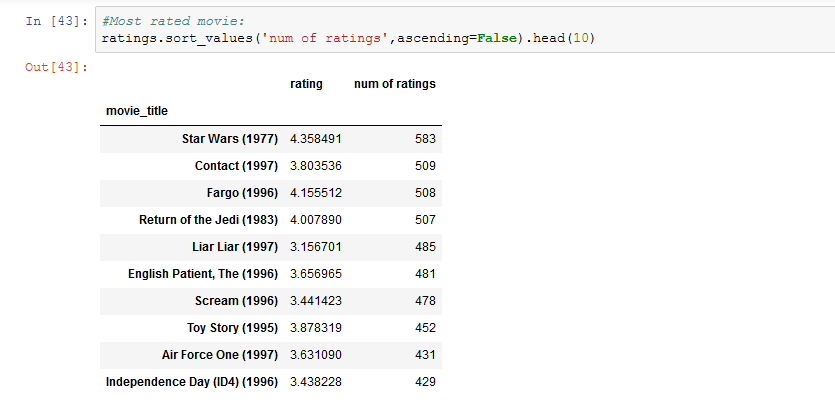
Here,we use the Seaborn package.



**Recommending Similar Movies :**

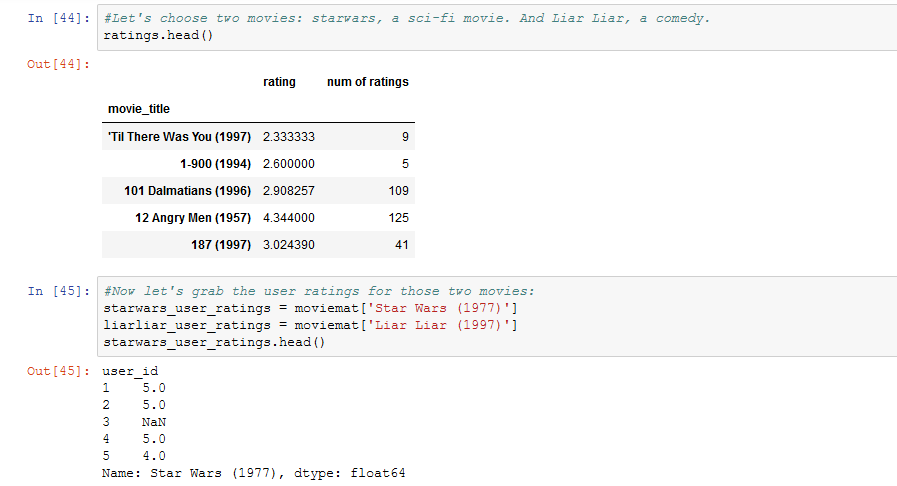


Here, we are sorting the values of num of ratings

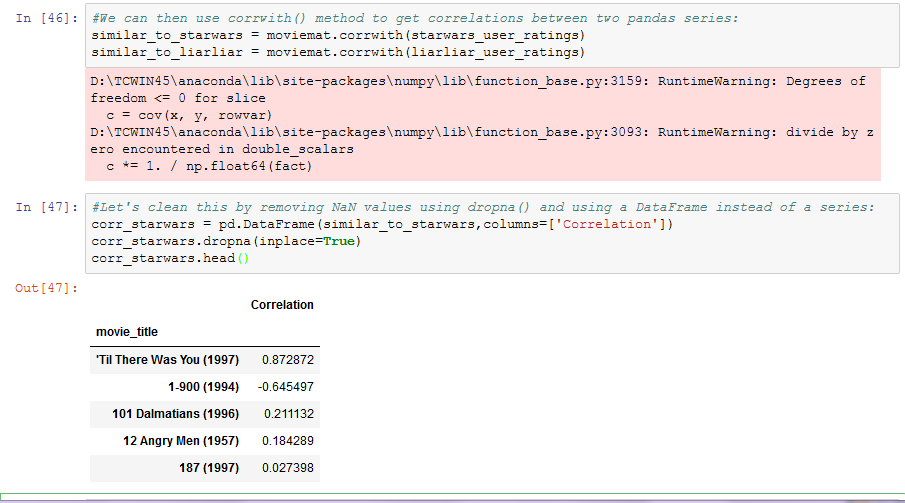


Displaying the results of two movies :

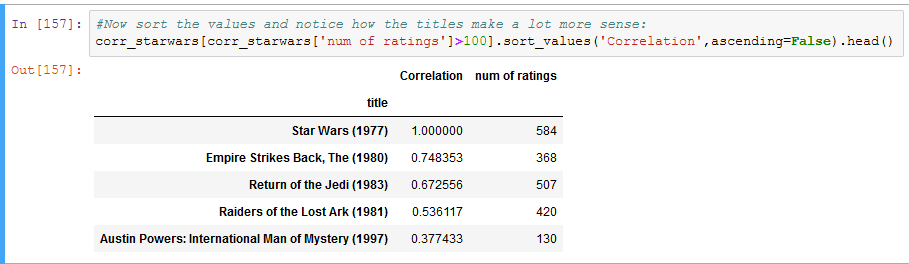
Here,we choose two movies STARWARS and LIAR LIAR to find out the co\_relation of the movies with the other movies.



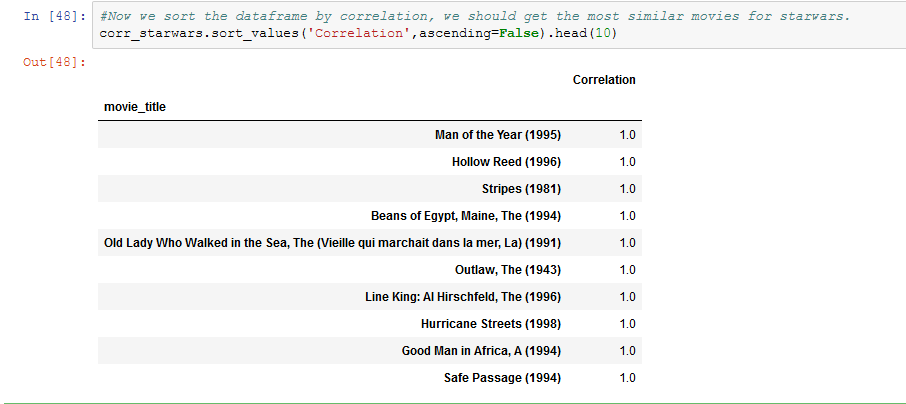
Calculating the correlation of two movies and removing NAN values using dropna() of any one of the movie and calculating correlation of that particular movie and displaying its results.First we consider STARWARS.we use the corrwith() function to find the co-relation.the function of (inplace=True) is to save the replace data.



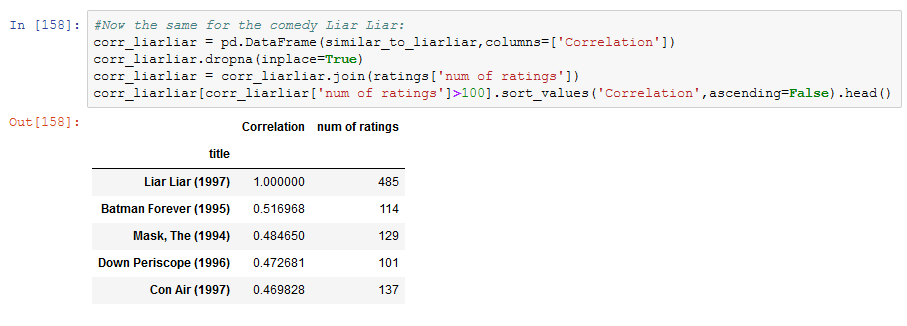
Sorting the values and noticing how the titles make a lot more sense.



Now,sorting the DataFrame,we get the most similar movies for STARWARS.We can recommend this movies to that users who like the movie STARWARS.

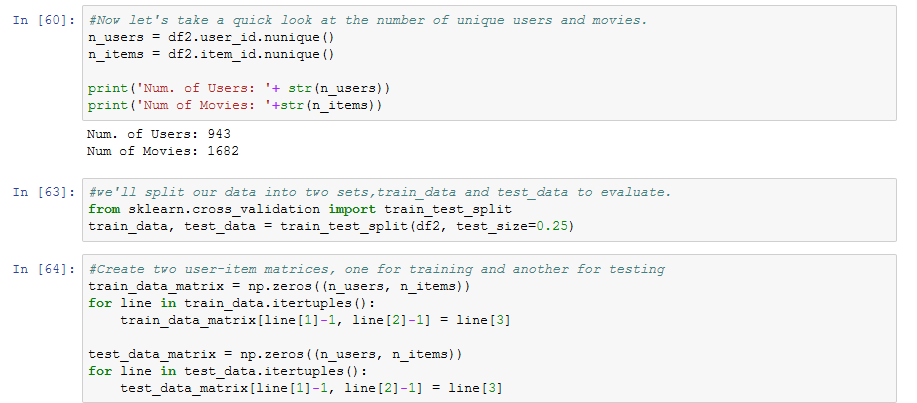


Here,we recommend similar movies for LIAR LIAR, using the same method as STARWARS.



NOW,WE WANT TO PREDICT DATA AND FIND OUT THE ERROR THAT HOW MUCH IT DIFFERS FRO THE ACTUAL DATA.

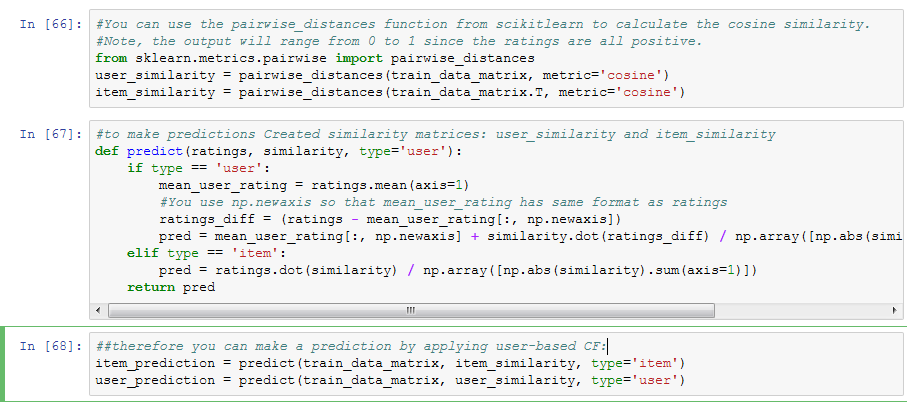
Here,we are Displaying no. of unique users and movies.To evaluate this data set,we split the dataset into two sets of data,TRAIN\_DATA and TEST\_DATA and create the USER\_ITEM matrices for train and test data sets.we use the memory based CF.



Importing pairwise distances from SCIKITLEARN.

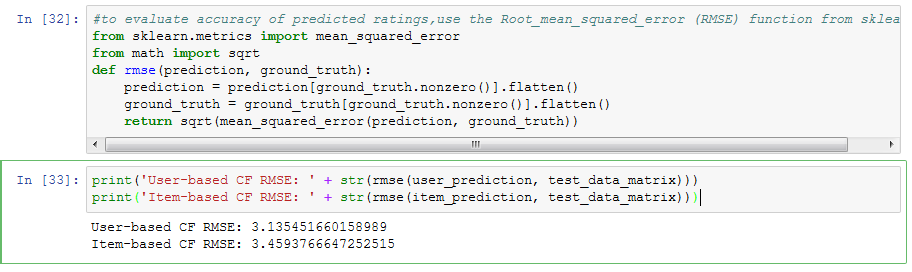
Here,we use the Cosine similarity to calculate pairwise distance function.

To make the predictions,we create user and item similarity matrices.Using user based collaborative filtering,we predict the data,using train data set.



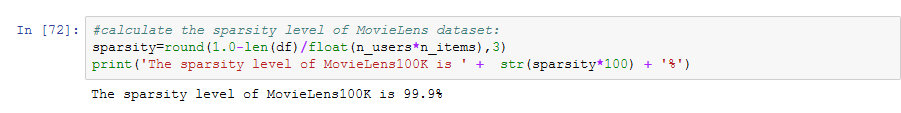
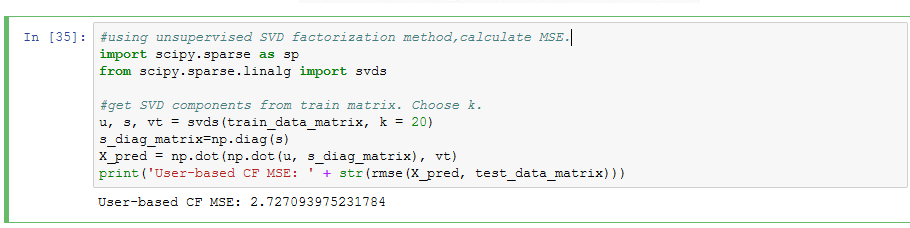
Here,to evaluate the accuracy of the predicted ratings,we use the RMSE(Root mean squared error) which is originally the root of MSE(mean squared error) from Sklearn.

Since you only want to consider predicted ratings that are in the test dataset, you filter out all other elements in the prediction matrix with prediction[ground\_truth.nonzero()].



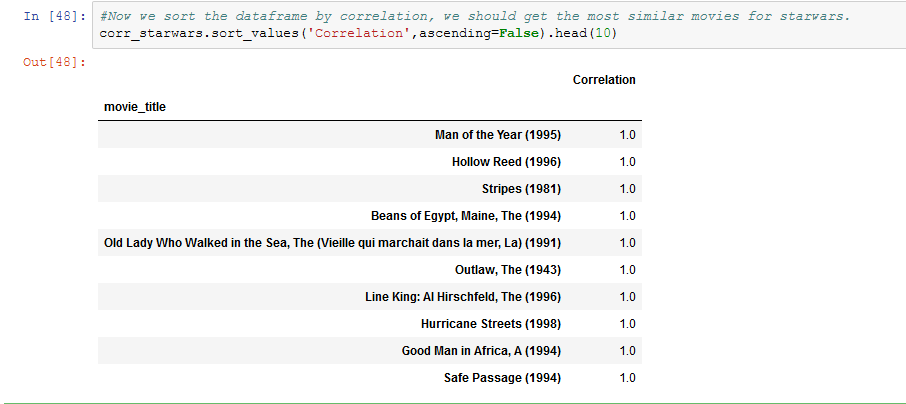
Here,we use the model based collaborative filtering to Calculate the sparsity level of movie lens. Model-based Collaborative Filtering is based on **matrix factorization (MF)** which has received greater exposure, mainly as an unsupervised learning method for latent variable decomposition and dimensionality reduction. Matrix factorization is widely used for recommender systems where it can deal better with scalability and sparsity than Memory-based CF. The goal of MF is to learn the latent preferences of users and the latent attributes of items from known ratings (learn features that describe the characteristics of ratings) to then predict the unknown ratings through the dot product of the latent features of users and items.

Using SVD (Singular Value Decomposition)factorization method,calculate the MSE.We use Scipypackage for the sparsity matrix.

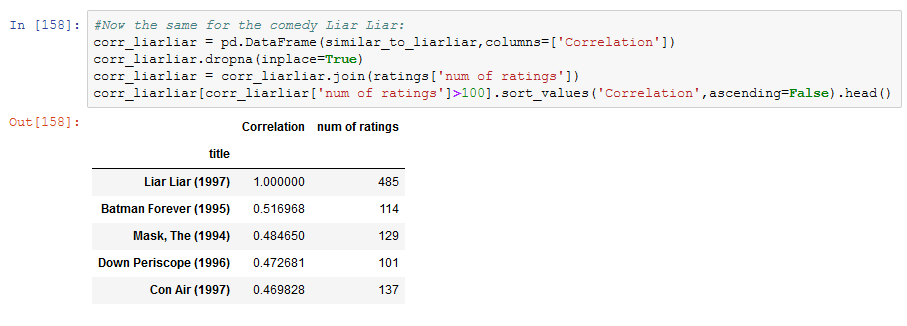
 

* **EXPERIMENTAL RESULTS:**

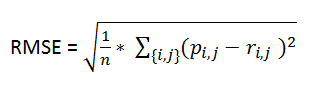
Here,we get the most similar movies like STARWARS.we get recommendation for STARWARS.



Here,we get the most similar movies like LIAR LIAR.we get recommendation for LIAR LIAR.



Root Mean Square Error (RMSE) as evaluation measure



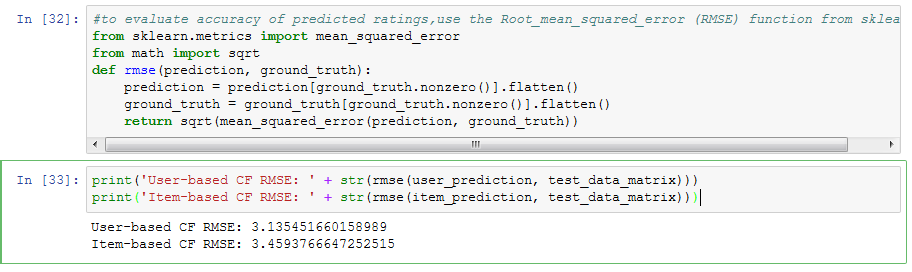
where n is the number of total ratings

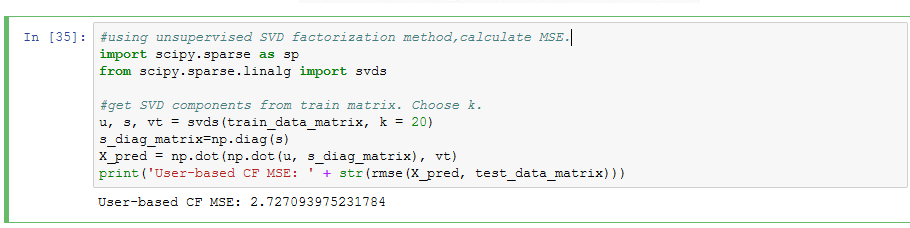
pi,j is the prediction for user i on item j

ri,j is the the actual rating.

First,we use memory based CF,to evaluate the Error.

Second,we use model based CF to evaluate the error.





* **CONCLUSION :**

In this project movie recommendation system has been implemented based on collaborative filtering engine. The system can be highly improved by making use of caching mechanisms, user clustering which will definitely boost the speed of the system, using yahoo term extraction web service to parse and get important keywords from the feeback provided by the user for an item and utilizing these keywords in context based engine. Further enhancements include storing users past history of results, contexts for future predictions.

Compared with extreme tree boosting model, improved latent factor model has lower MSE when exploit full features. One important reason is that it captures the correlation between all features besides users and movie.

* **SOURCES :**

<https://www.python.org/>

<https://anaconda.org/anaconda/python>

<http://www.numpy.org/>

<https://matplotlib.org/>

<http://scikit-learn.org/>

<https://ipython.org/>