INTRODUCTION:-

A recommendation system helps user’s find compelling content in a large corpora. For example, the Google Play Store provides millions of apps, while YouTube provides billions of videos. More apps and videos are added every day. How can users find new compelling new content? Yes, one can use search to access content. However, a recommendation engine can display items that users might not have thought to search for on their own.

Working Principle:-

STEP-1 DATA PRE-PROCESSING

Steps in Data Pre-processing

Step 1: Import the libraries

Step 2: Import the data-set

Step 3: Check out the missing values

Step 4: See the Categorical Values

Step 5: Splitting the data-set into Training and Test Set

Step 6: Feature Scaling

Step 1: Import the libraries

**Some basic libraries of python that are used in data pre-processing are listed below:-**

1. **Numpy** is the fundamental package for scientific computing with Python. It is useful in linear algebra, Fourier transform, and random number capabilities
2. **Pandas** is for data manipulation and analysis. Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for tthe [Python](https://www.python.org/) programming language. Pandas is a [NumFOCUS](https://www.numfocus.org/open-source-projects.html" \t "_blank) sponsored project.
3. **Matplotlib** is a Python 2D plotting library which produces publication quality figures in a variety of hard copy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [IPython](http://ipython.org/" \t "_blank) shells, the [Jupyter](http://jupyter.org/" \t "_blank) notebook, web application servers, and four graphical user interface toolkits.

1. **Seaborn**is a Python data visualization library based on [matplotlib](https://matplotlib.org/" \t "_blank). It provides a high-level interface for drawing attractive and informative statistical graphics.

Step 2: Import the data-set:- In this step we will import the dataset with respect to its type such as .csv,.txt,.html etc.

Example- for .csv we will use pandas.read\_csv(‘file name’)

Step 3: Check out the missing values: - The concept of missing values is important to understand in order to successfully [manage](http://www.statisticssolutions.com/academic-solutions/resources/dissertation-resources/data-entry-and-management/multiple-imputation-for-missing-data/) data. If the missing values are not handled properly by the researcher, then he/she may end up drawing an inaccurate inference about the data. Due to improper handling, the result obtained by the researcher will differ from ones where the missing values are present.

SYNTAX- Data.isnull ()

Here data is the variable where csv file read by panadas

Step 4: See the Categorical Values: -

* For this step we will use loc and iloc concept of python to separate the dependent variable and independent variable.
* After separating both the dependent variable and independent variable.To convert Categorical variable into Numerical data we can use LabelEncoder() class fromsklearn.preprocessing library.
* If these Categories and there is no relational order between the them. So , we have to prevent this, we’re going to use what are **Dummy Variables.**
* **To create dummy variable we are going to use OneHotEncoder Class from sklearn.preprocessing or we can use pandas get dummies method** to use **get\_dummies( )**for creating Dummy Variables.

Step 5: Splitting the data-set into Training and Test Set

In this we are going to split our dataset into 2 parts

1.Training set

2.Test set

I have mention a small part of code from my one project of customer behaviour analysis.

from sklearn.model\_selection import train\_test\_split

x\_train, x\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size=0.2,random\_state=0)

here I have 10,000 data in csv format and I have split the dataset into 8:2 ratio that is 8000 training set and 2000 test set.

Step 6: Feature Scaling:-

**Feature scaling** is the method to limit the range of variables so that they can be compared on common grounds.

For feature scaling in general we need to use:-

from sklearn.preprocessing import StandardScaler

sc=StandardScaler() (**we have to create StandardScaler object to call fit\_transform)**

x\_train=sc.fit\_transform(x\_train)

x\_test=sc.fit\_transform(x\_test)

RECOMMENDATION SYSTEM:-

**To build a recommendation system for blogs in machine learning. Assume we have a chrome extension, when the user clicks on that, it will show them recommendation –**

We can use any of 2 neural networks i.e-

1. Boltzmann Machines
2. AutoEncoder

Both of the neural network comes under Unsupervised neural network and most frequently used in recommendation system.

Some commonly used algorithm for recommendation system are-

# **1. Collaborative filtering**

Collaborative filtering (CF) and its modifications is one of the most commonly used recommendation algorithms. Even data scientist beginners can use it to build their personal movie recommender system, for example -Movie recommendation.

When we want to recommend something to a user, the most logical thing to do is to find people with similar interests, analyse their behaviour, and recommend our user the same items. Or we can look at the items similar to ones which the user bought earlier, and recommend products which are like them.

# **Matrix decomposition for recommendations**

The next interesting approach uses matrix decompositions. It’s a very elegant recommendation algorithm because usually, when it comes to matrix decomposition, we don’t give much thought to what items are going to stay in the columns and rows of the resulting matrices. But using this recommender engine, we see clearly that **u** is a vector of interests of **i-th user,** and **v** is a vector of parameters for **j-th film.**

**I have created a Netflix Reccomendation system**

**Link-** <https://github.com/simranrout/netflix-reccomendation-system>