# MACHINE LEARNING AND

# DEEP LEARNING

## INTRODUCTION TO MACHINE LEARNING:

Machine Learning(ML) is the scientific study of algorithms and statistical models that computer systems use in order to perform a specific task effectively without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of Artificial Intelligence(AI).

## IMPORTANCE OF MACHINE LEARNING:

Consider some of the instances where machine learning is applied: the self-driving Google car, cyber fraud detection, online recommendation engines—like friend suggestions on Facebook, Netflix showcasing the movies and shows you might like, and “more items to consider” and “get yourself a little something” on Amazon—are all examples of applied machine learning. All these examples echo the vital role machine learning has begun to take in today’s data-rich world.

Machines can aid in filtering useful pieces of information that help in major advancements, and we are already seeing how this technology is being implemented in a wide variety of industries.

With the constant evolution of the field, there has been a subsequent rise in the uses, demands, and importance of machine learning. Big data has become quite a buzzword in the last few years; that’s in part due to increased sophistication of machine learning, which helps analyze those big chunks of big data. Machine learning has also changed the way data extraction, and interpretation is done by involving automatic sets of generic methods that have replaced traditional statistical techniques.

The process flow depicted here represents how machine learning works

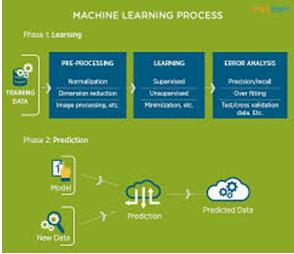


Figure 1 : The Process Flow

## USES OF MACHINE LEARNING:

Earlier in this article, we mentioned some applications of machine learning. To understand the concept of machine learning better, let’s consider some more examples: web search results, real-time ads on web pages and mobile devices, email spam filtering, network intrusion detection, and pattern and image recognition. All these are by-products of applying machine learning to analyze huge volumes of data

Traditionally, data analysis was always being characterized by trial and error, an approach that becomes impossible when data sets are large and heterogeneous. Machine learning comes as the solution to all this chaos by proposing clever alternatives to analyzing huge volumes of data.

By developing fast and efficient algorithms and data-driven models for real-time processing of data, machine learning can produce accurate results and analysis.

## TYPES OF LEARNING ALGORITHMS:

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

## Supervised Learning :

When an algorithm learns from example data and associated target responses that can consist of numeric values or string labels, such as classes or tags, in order to later predict the correct response when posed with new examples comes under the category of supervised learning.

Supervised machine learning algorithms uncover insights, patterns, and relationships from a labelled training dataset – that is, a dataset that already contains a known value for the target variable for each record. Because you provide the machine learning algorithm with the correct answers for a problem during training, it is able to “learn” how the rest of the features relate to the target, enabling you to uncover insights and make predictions about future outcomes based on historical data.

Examples of Supervised Machine Learning Techniques are Regression, in which the algorithm returns a numerical target for each example, such as how much revenue will be generated from a new marketing campaign.

Classification, in which the algorithm attempts to label each example by choosing between two or more different classes. Choosing between two classes is called binary classification, such as determining whether or not someone will default on a loan. Choosing between more than two classes is referred to as multiclass classification.

## Unsupervised Learning:

When an algorithm learns from plain examples without any associated response, leaving to the algorithm to determine the data patterns on its own. This type of algorithm tends to restructure the data into something else, such as new features that may represent a class or a new series of uncorrelated values. They are quite useful in providing humans with insights into the meaning of data and new useful inputs to supervised machine learning algorithms.

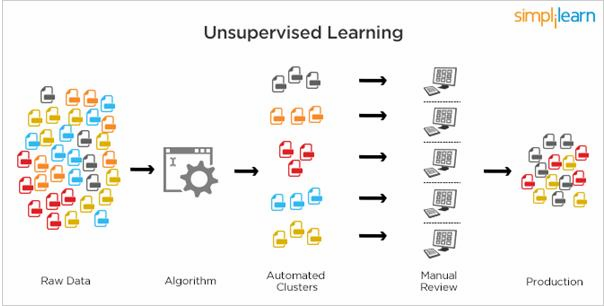


Figure 2 : Unsupervised Learning

Popular techniques where unsupervised learning is used also include self-organizing maps, nearest neighbor mapping, singular value decomposition, and k-means clustering. Basically, online recommendations, identification of data outliers, and segment text topics are all examples of unsupervised learning.

## Semi Supervised Learning:

As the name suggests, semi-supervised learning is a bit of both supervised and unsupervised learning and uses both labeled and unlabeled data for training. In a typical scenario, the algorithm would use a small amount of labeled data with a large amount of unlabeled data.

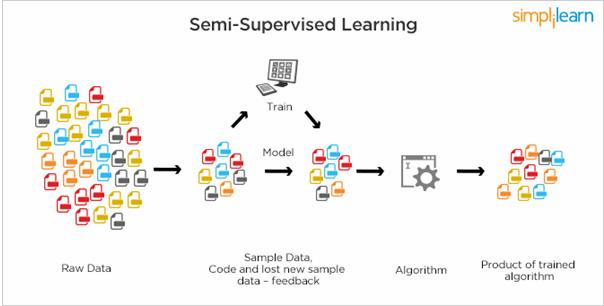


Figure 3 : Semi Supervised Learning

## RELATION BETWEEN DATA MINING,MACHINE LEARNING AND DEEP LEARNING:

Machine learning and data mining use the same algorithms and techniques as data mining, except the kinds of predictions vary. While data mining discovers previously unknown patterns and knowledge, machine learning reproduces known patterns and knowledge—and further automatically applies that information to data, decision-making, and actions.

Deep learning, on the other hand, uses advanced computing power and special

types of neural networks and applies them to large amounts of data to learn, understand, and identify complicated patterns. Automatic language translation and medical diagnoses are examples of deep learning.

2. **DEEP LEARNING**

Deep learning is an artificial intelligence function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence (AI) that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network.

2.1KEY TAKEAWAYS

Deep learning is an AI function that mimics the workings of the human brain in processing data for use in detecting objects, recognizing speech, translating languages, and making decisions.

Deep learning AI is able to learn without human supervision, drawing from data that is both unstructured and unlabeled.

Deep learning, a form of machine learning, can be used to help detect fraud or money laundering, among other functions.

**2.2How Deep Learning Works**

Deep learning has evolved hand-in-hand with the digital era, which has brought about an explosion of data in all forms and from every region of the world. This data, known simply as big data, is drawn from sources like social media, internet search engines, e-commerce platforms, and online cinemas, among others. This enormous amount of data is readily accessible and can be shared through fintech applications like cloud computing.

However, the data, which normally is unstructured, is so vast that it could take decades for humans to comprehend it and extract relevant information. Companies realize the incredible potential that can result from unraveling this wealth of information and are increasingly adapting to AI systems for automated support.

**2.3Deep Learning vs. Machine Learning**

One of the most common AI techniques used for processing big data is machine learning, a self-adaptive algorithm that gets increasingly better analysis and patterns with experience or with newly added data.

If a digital payments company wanted to detect the occurrence or potential for fraud in its system, it could employ machine learning tools for this purpose. The computational algorithm built into a computer model will process all transactions happening on the digital platform, find patterns in the data set, and point out any anomaly detected by the pattern.

Deep learning, a subset of machine learning, utilizes a hierarchical level of artificial neural networks to carry out the process of machine learning. The artificial neural networks are built like the human brain, with neuron nodes connected together like a web. While traditional programs build analysis with data in a linear way, the hierarchical function of deep learning systems enables machines to process data with a nonlinear approach.

**2.4Uses:**

Deep learning models are widely used in extracting high-level abstract features, providing improved performance over the traditional models, increasing interpretability and also for understanding and processing biological data. To predict splicing action of exons, a fully connected feedforward neural network was designed by Xiong et al. [60]. In recent years, CNNs were applied on the DNA dataset directly without the requirement of defining features a priori [2], [44]. Compared to a fully connected network, CNNs use less parameters by applying a convolution operation on the input data space and also parameters are shared between the regions. Hence, large DNA sequence data can be trained using these models and also improved pattern detection accuracy can be obtained. Deepbind, a deep architecture based on CNNs, was proposed by Alipanathi et al. [57], which predicts specificities of DNA and RNA binding proteins. CNNs were also used for predicting chromatin marks from a DNA sequence [44]. Angermueller et al. [35] have incorporated CNNs for predicting DNA methylation states. Like CNNs, other deep architectures were also applied for extracting features from raw DNA sequence data and for processing the data.

**2.5RELATION BETWEEN DATA MINING,MACHINE LEARNING AND DEEP LEARNING:**

Machine learning and data mining use the same algorithms and techniques as data

mining, except the kinds of predictions vary. While data mining discovers previously

unknown patterns and knowledge, machine learning reproduces known patterns and

knowledge—and further automatically applies that information to data, decision-making, and

actions.

Deep learning, on the other hand, uses advanced computing power and special

5

types of neural networks and applies them to large amounts of data to learn, understand, and

identify complicated patterns. Automatic language translation and medical diagnoses are

examples of deep learning.

**2.6 Algorithms in Deep Learning**

Deep Learning is a new area of Machine Learning research, which has been introduced with the objective of moving Machine Learning closer to one of its original goals: Artificial Intelligence. See these course notes for a brief intro of ml of ai and an introduction to deep learning algorithms

Deep Learning is about learning multiple levels of representation and abstraction that help to make sense of data such as images, sound, and text. For more about deep learning algorithms, see for example:

THINGS presented here will introduce you to some of the most important deep learning algorithms and will also show you how to run them using theano. Theano is a python library that makes writing deep learning models easy, and gives the option of training them on a GPU.

The algorithm tutorials have some prerequisites. You should know some python, and be familiar with numpy. Since this tutorial is about using Theano.

The purely supervised learning algorithms are meant to be read in order:

1. Logistic Regression - using Theano for something simple
2. Multilayer percepton - introduction to layers
3. Deep Convolution Network- a simplified version of LeNet5

The unsupervised and semi-supervised learning algorithms can be read in any order (the auto-encoders can be read independently of the RBM/DBN thread):

* [Auto Encoders, Denoising Autoencoders](http://deeplearning.net/tutorial/dA.html#daa) - description of autoencoders
* [Stacked Denoising Auto-Encoders](http://deeplearning.net/tutorial/SdA.html#sda) - easy steps into unsupervised pre-training for deep nets
* [Restricted Boltzmann Machines](http://deeplearning.net/tutorial/rbm.html#rbm) - single layer generative RBM model
* [Deep Belief Networks](http://deeplearning.net/tutorial/DBN.html#dbn) - unsupervised generative pre-training of stacked RBMs followed by supervised fine-tuning

Building towards including the mcRBM model, we have a new tutorial on sampling from energy models:

* [HMC Sampling](http://deeplearning.net/tutorial/hmc.html#hmc) - hybrid (aka Hamiltonian) Monte-Carlo sampling with scan()

Building towards including the Contractive auto-encoders tutorial, we have the code for now:

* [Contractive auto-encoders](https://github.com/lisa-lab/DeepLearningTutorials/blob/master/code/cA.py) code - There is some basic doc in the code.

Recurrent neural networks with word embeddings and context window:

* [Semantic Parsing of Speech using Recurrent Net](http://deeplearning.net/tutorial/rnnslu.html#rnnslu)

LSTM network for sentiment analysis:

* [LSTM network](http://deeplearning.net/tutorial/lstm.html#lstm)

Energy-based recurrent neural network (RNN-RBM):

* [Modeling and generating sequences of polyphonic music](http://deeplearning.net/tutorial/rnnrbm.html#rnnrbm)

Segmentation for medical imagery (meant to be read in order):

* [Fully Convolutional Networks (FCN) for 2D segmentation](http://deeplearning.net/tutorial/fcn_2D_segm.html#fcn-2d-segm)
* [U-Net](http://deeplearning.net/tutorial/unet.html#unet)
* 1D segmentation

# PYTH0N

Basic programming language used for machine learning is : PYTHON

## 3.1INTRODUCTION TO PYHTON:

* + - Python is a high-level, interpreted, interactive and object-oriented scripting language.
    - Python is a general purpose programming language that is often applied in scripting roles
    - Python is Interpreted: Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is like PERL and PHP.
    - Python is Interactive: You can sit at a Python prompt and interact with the interpreter directly to write your programs.
    - Python is Object-Oriented: Python supports the Object-Oriented style or technique of programming that encapsulates code within objects.

## 3.2HISTORY OF PYTHON:

* + - Python was developed by GUIDO VAN ROSSUM in early 1990’s
    - Its latest version is 3.7 , it is generally called as python3

## 3.3FEATURES OF PYTHON:

* + - Easy-to-learn: Python has few keywords, simple structure, and a clearly defined syntax,

This allows the student to pick up the language quickly.

* + - Easy-to-read: Python code is more clearly defined and visible to the eyes.
    - Easy-to-maintain: Python's source code is fairly easy-to-maintaining.
    - A broad standard library: Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
    - Portable: Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
    - Extendable: You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
    - Databases: Python provides interfaces to all major commercial databases.
* GUI Programming: Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

## 3.4HOW TO SETUP PYTHON:

* + - Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.
    - The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python.

## 3.4.1Installation(using python IDLE):

* + - * Installing python is generally easy, and nowadays many Linux and Mac OS distributions include a recent python.
      * [Download python from www.python.org](http://www.python.org/)
      * When the download is completed, double click the file and follow the instructions to install it.
      * When python is installed, a program called IDLE is also installed along with it. It provides a graphical user interface to work with python.

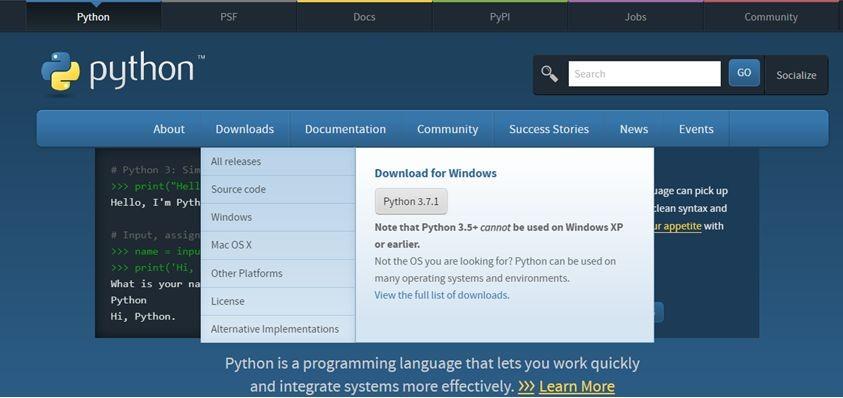


Figure 4 : Python download

## 3.4.2Installation(using Anaconda):

* + - * Python programs are also executed using Anaconda.
      * Anaconda is a free open source distribution of python for large scale data processing, predictive analytics and scientific computing.
      * Conda is a package manager quickly installs and manages packages.
      * In WINDOWS:
      * In windows
        + Step 1: Open Anaconda.com/downloads in web browser.
        + Step 2: Download python 3.4 version for (32-bitgraphic installer/64 -bit graphic installer)
        + Step 3: select installation type( all users)
        + Step 4: Select path(i.e. add anaconda to path & register anaconda as default python 3.4) next click install and next click finish
        + Step 5: Open jupyter notebook ( it opens in default browser)

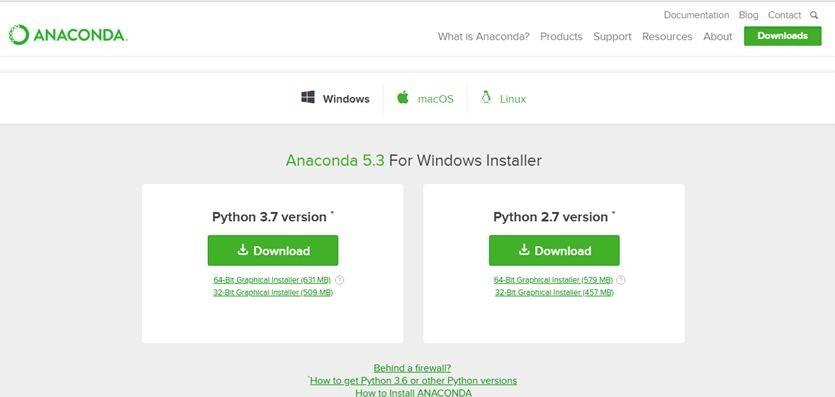


Figure 5 : Anaconda download



Figure 6 : Jupyter notebook

## 3.5PYTHON VARIABLE TYPES:

* Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.
* Variables are nothing but reserved memory locations to store values.
* Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory.
* Python variables do not need explicit declaration to reserve memory space. The declaration happens automatically when you assign a value to a variable.
* Python has various standard data types that are used to define the operations possible on them and the storage method for each of them.
* Python has five standard data types –
  + Numbers
  + Strings
  + Tuples
  + Dictionary

## 3.5.1Python Numbers:

* + - * Number data types store numeric values. Number objects are created when you assign a value to them.
      * Python supports four different numerical types − int (signed integers) long (long integers, they can also be represented in octal and hexadecimal) float (floating point real values) complex (complex numbers).

## 3.5.2Python Strings:

* + - * Strings in Python are identified as a contiguous set of characters represented in the quotation marks.
      * Python allows for either pairs of single or double quotes.
      * Subsets of strings can be taken using the slice operator ([ ] and [:] ) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.
      * The plus (+) sign is the string concatenation operator and the asterisk (\*) is the repetition operator.

## 3.5.3Python Lists:

* + - * Lists are the most versatile of Python's compound data types.
      * A list contains items separated by commas and enclosed within square brackets

([]).

* + - * To some extent, lists are similar to arrays in C. One difference between them is that all the items belonging to a list can be of different data type.
      * The values stored in a list can be accessed using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1.
      * The plus (+) sign is the list concatenation operator, and the asterisk (\*) is the repetition operator.

## 3.5.4Python Tuples:

* + - * A tuple is another sequence data type that is similar to the list.
      * A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses.
      * The main differences between lists and tuples are: Lists are enclosed in brackets ( [

] ) and their elements and size can be changed, while tuples are enclosed in parentheses ( ( ) ) and cannot be updated.

* + - * Tuples can be thought of as read-only lists.
      * For example − Tuples are fixed size in nature whereas lists are dynamic. In other words, a tuple is immutable whereas a list is mutable. You can't add elements to a tuple. Tuples have no append or extend method. You can't remove elements from a tuple. Tuples have no remove or pop method.

or hashes found in Perl and consist of key-value pairs. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.

* + - * Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([]).
      * You can use numbers to "index" into a list, meaning you can use numbers to find out what's in lists. You should know this about lists by now, but make sure you understand that you can only use numbers to get items out of a list.
      * What a dict does is let you use anything, not just numbers. Yes, a dict associates one thing to another, no matter what it is.

## 3.6PYTHON FUNCTION:

**3.6.1Defining a Function:**

You can define functions to provide the required functionality. Here are simple rules to define a function in Python. Function blocks begin with the keyword def followed by the function name and parentheses (i.e.()).

Any input parameters or arguments should be placed within these parentheses.

You can also define parameters inside these parentheses

The code block within every function starts with a colon (:) and is indented. The statement returns [expression] exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as return None.

## 3.6.2Calling a Function:

Defining a function only gives it a name, specifies the parameters that are to be included in the function and structures the blocks of code. Once the basic structure of a function is finalized, you can execute it by calling it from another function or directly from the Python prompt.

## 3.7PYTHON USING OOP’s CONCEPTS:

**3.7.1Class:**

* + - * Class: A user-defined prototype for an object that defines a set of attributes that characterize any object of the class. The attributes are data members (class variables and instance variables) and methods, accessed via dot notation.
      * Class variable: A variable that is shared by all instances of a class. Class variables are defined within a class but outside any of the class's methods. Class variables are not used as frequently as instance variables are.
      * Data member: A class variable or instance variable that holds data associated with a class and its objects.
      * Instance variable: A variable that is defined inside a method and belongs only to the current instance of a class.
      * Defining a Class:
        + We define a class in a very similar way how we define a function.
        + Just like a function ,we use parentheses and a colon after the class name(i.e. ():) when we define a class. Similarly, the body of our class is

indented like a functions body is.



Figure 7 : Defining a Class

## 3.7.2 init method in Class:

* + - * The init method — also called a constructor — is a special method that runs when an instance is created so we can perform any tasks to set up the instance.
      * The init method has a special name that starts and ends with
      * two underscores: init ().

**SONGS RECOMMENDER SYSTEM**

## 4.1PROBLEM STATEMENT:

To predict the songs for a particular user and to predict similar songs if a user chooses a song.

## 4.2 OBJECTIVE OF THE CASE STUDY:

The number of songs available exceeds the listening capacity of an individual in their lifetime. It is tedious for an individual to sometimes to choose from millions of songs and there is also a good chance missing out on songs which could have been the favourites.

Music service providers like Spotify need an efficient way to manage songs and help their customers to discover music by giving a quality recommendation. For building this recommendation system, they deploy machine learning algorithms to process data from a million sources and present the listener with the most relevant songs.

There are mainly three types of recommendation system: content-based, collaborative and popularity.

**MODEL BUILDING**

## 4.3PREPROCESSING OF THE DATA:

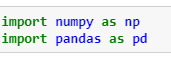
Preprocessing of the data actually involves the following steps:

## 4.3.1GETTING THE DATASET:

We can get the data set from the database or we can get the data from client.

## 4.3.2IMPORTING THE LIBRARIES:

We have to import the libraries as per the requirement of the algorithm.



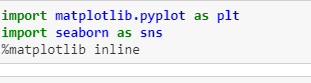


Figure 8:importing libraries

## 4.3.3IMPORTING THE DATA-SET:

Pandas in python provide an interesting method read\_csv(). The read\_csv function reads the entire dataset from a comma separated values file and we can assign it to a DataFrame to which all the operations can be performed. It helps us to access each and every row as well as columns and each and every value can be access using the dataframe. Any missing value or NaN value have to be cleaned.

Reading the dataset

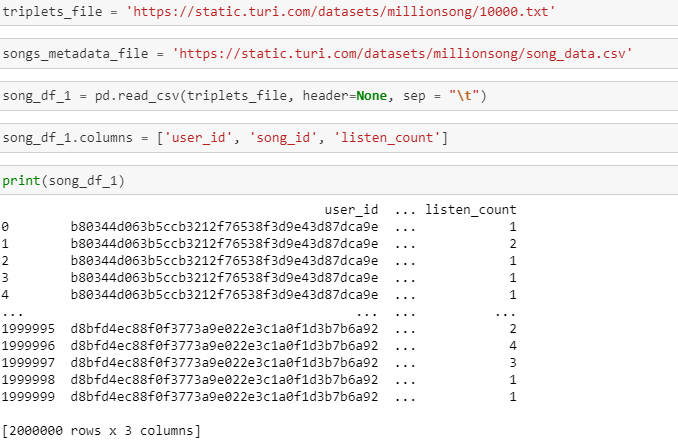




Figure 9: Reading the dataset

Removing dupilicates in metadata file and merging two datasets



Figure 10: Removing dupilicates in metadata file and merging two datasets

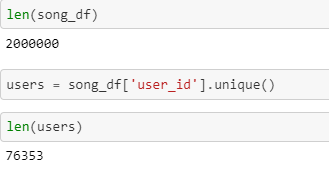


Figure 11:Finding no of users and of dataset

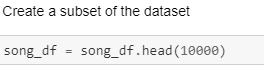


Figure 12:Creating a subset of dataset

## 4.3.4HANDLING MISSING VALUES:

Missing values can be handled in many ways using some inbuilt methods: (a)dropna()

(b)fillna()

(c)interpolate()

* + - 1. mean imputation and median imputation

### (a)dropna():

dropna() is a function which drops all the rows and columns which are having the missing values(i.e. NaN)

* + - * + dropna() function has a parameter called how which works as follows
* if how = ’all’ is passed then it drops the rows where all the columns of the particular row are missing
* if how = ’any’ is passed then it drops the rows where all the columns of the particular row are missing

### (b)fillna():

fillna() is a function which replaces all the missing values using different

ways.

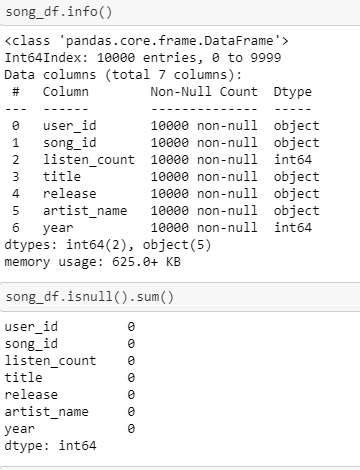
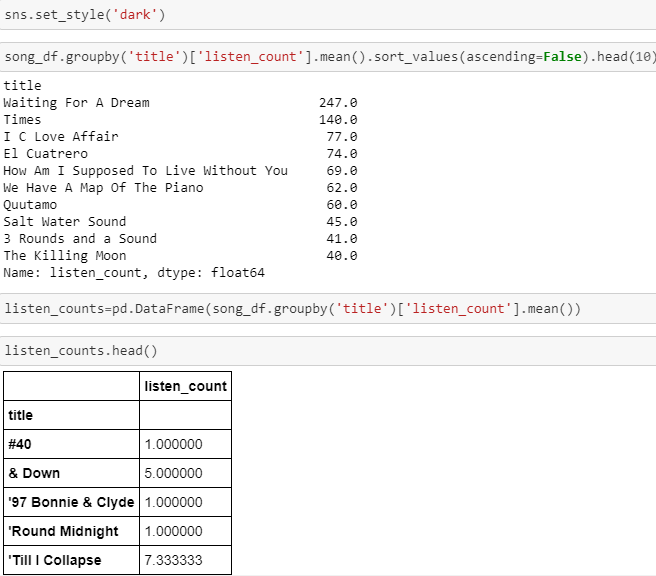
****

Figure 13: finding the information about dataset and checking if any null values

4.3.5Data Visualization:



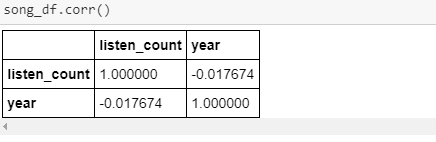


Figure 14: Checking the corr() of the dataset

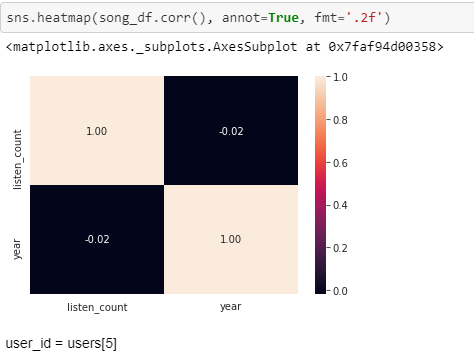


Figure 15 : heatmap of the dataset

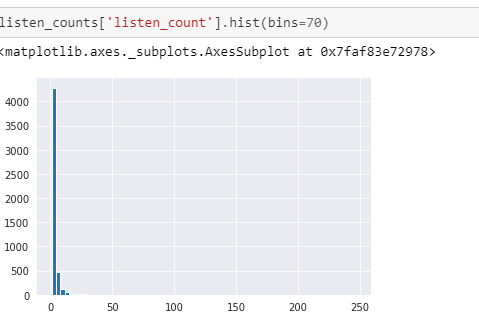


Figure 16: histogram of listen\_count

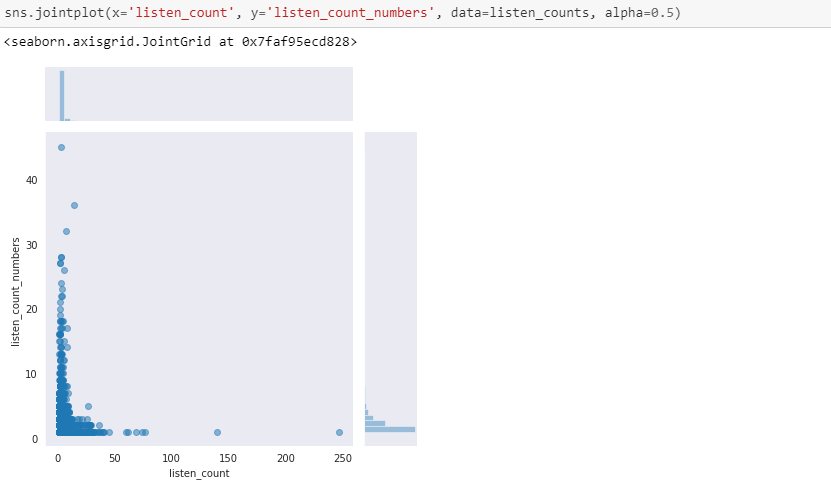
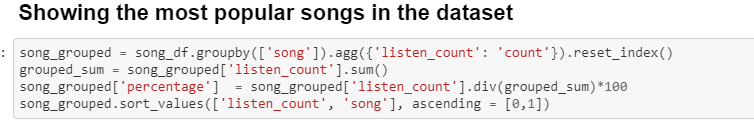


Figure 17: join-plot for listen count and listen count numbers

4.3.6 Popular songs



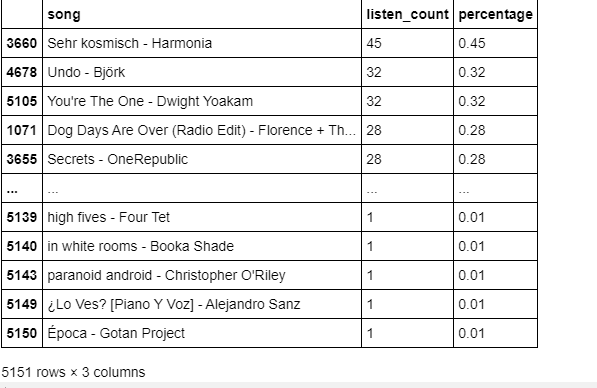


Figure 18: Finding most popular songs

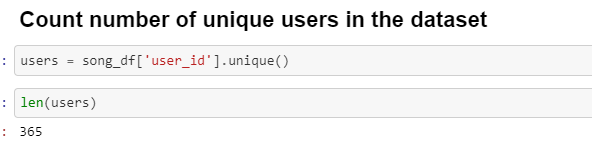


Figure 19: Count the number of Unique users



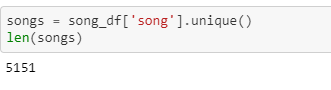


Figure 20: Count the number of Unique songs

## 4.4TRAINING THE MODEL:

**4.4.1Method 1:**

* Splitting the data : after the preprocessing is done then the data is split into train and test sets
* In Machine Learning in order to access the performance of the classifier. You train the classifier using 'training set' and then test the performance of your classifier on unseen 'test set'. An important point to note is that during training the classifier only uses the training set . The test set must not be used during training the classifier. The test set will only be available during testing the classifier.
* training set - a subset to train a model.(Model learns patterns between Input and Output)
* test set - a subset to test the trained model.(To test whether the model has correctly learnt )
* The amount or percentage of Splitting can be taken as specified (i.e. train data = 75%

, test data =25% or train data = 80% , test data= 20%)

* First we need to identify the input and output variables and we need to separate the input set and output set
* In scikit learn library we have a package called model\_selection in which train\_test\_split method is available .we need to import this method
* This method splits the input and output data to train and test based on the percentage specified by the user and assigns them to four different variables(we need to mention thevariables)

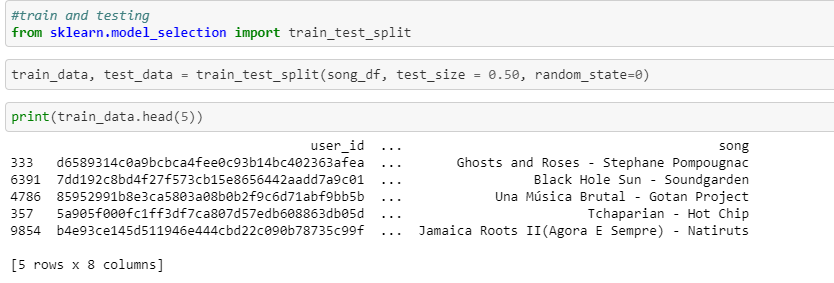


Figure 21: Training the dataset

**4.5Modelling:**

A package as been written for recommending and evalusting the models

The package for recommending is given below.

4.5.1Building Recommenders model based on popularity:



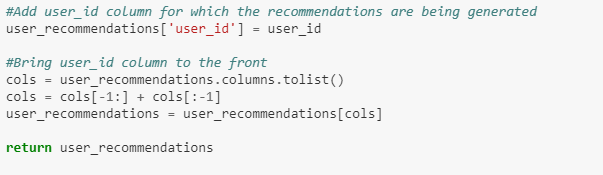


Figure 22: Recommenders model based on personalized algorithm in reco2 package

4.5.2Building Recommenders model based on Similarity:

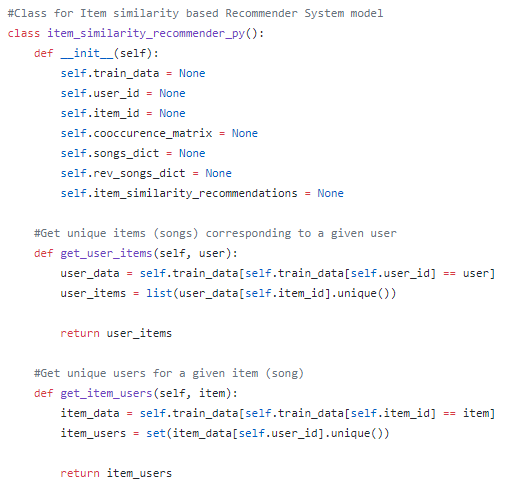


Figure 23: Similarity based Algorithm

## Figure 24: Importing the recommenders package and creating instance of Popularity Class

## 

## 

4.5.3 Making some predictions using popularity model

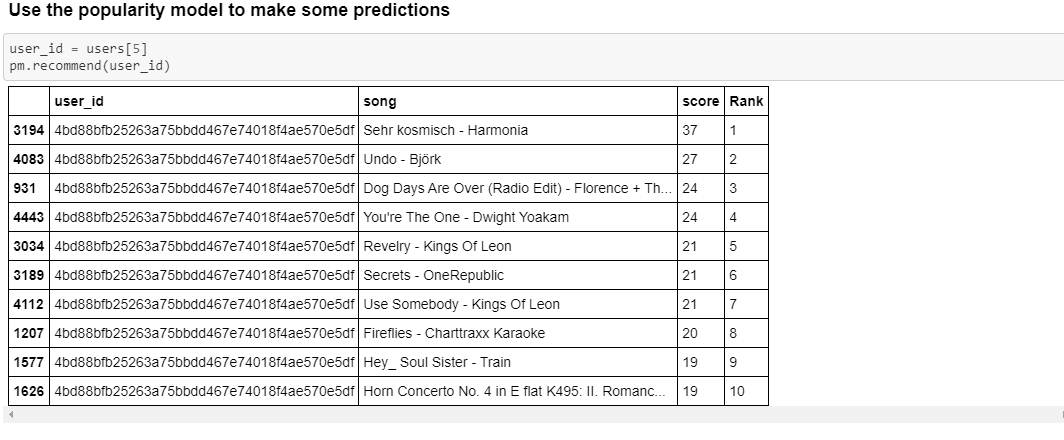
****

Figure 25: Making some predictions using popularity model

4.5.4 Build a song recommender with personalization

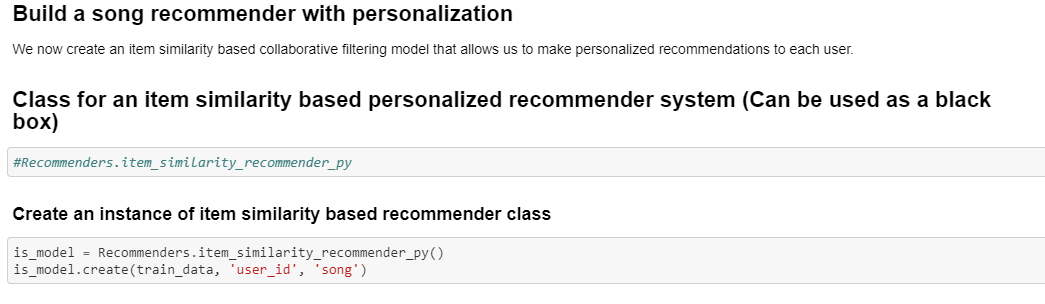
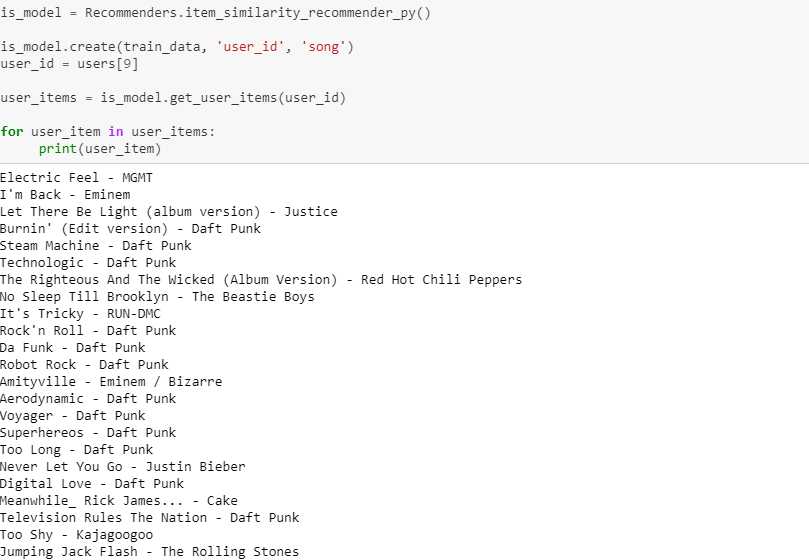
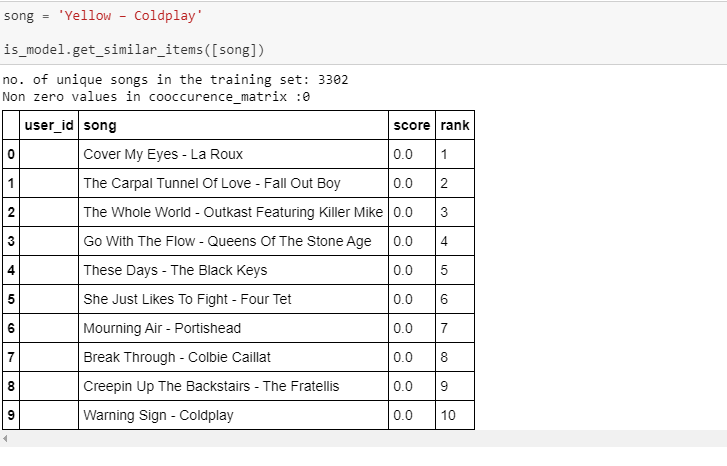
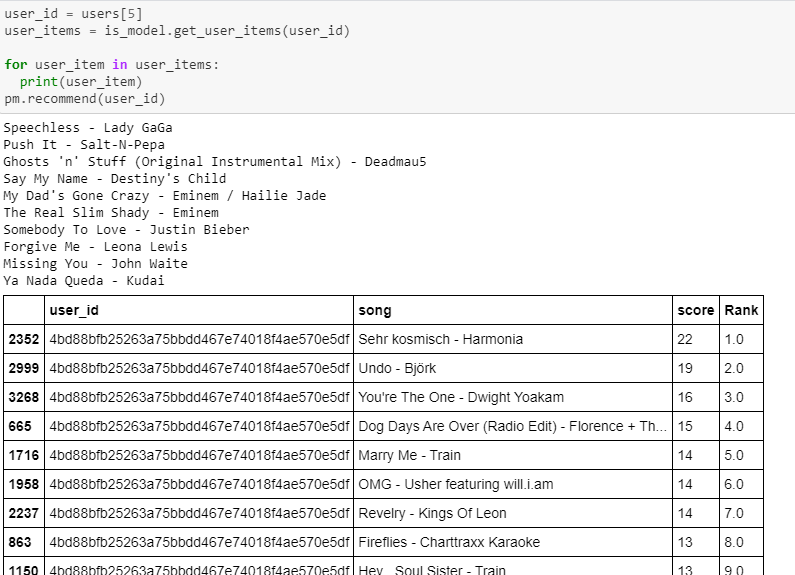
****

Figure 26: Creating an instance for Similarity class of Personalized model

**4.5.5 **

**Figure 27: Using personalized model to make some predictions **

**Figure 28: Recommending songs based on users given song**

****

**Figure 29: Recommending a song to user based on personalized model**

**4.6Evaluation of model using precision recall curve:**

A package has has written for evaluation and is given below.

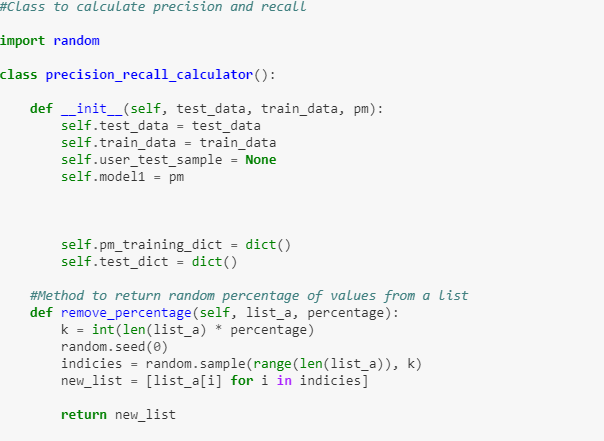
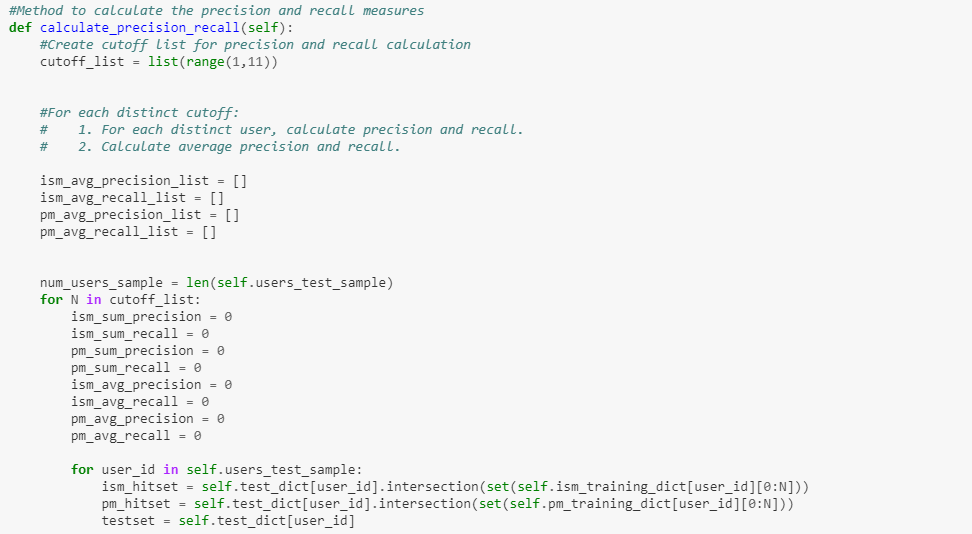


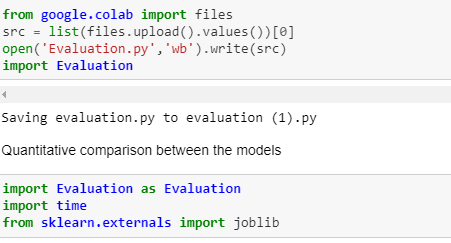


Figure 30: Evaluating the models we are using Precision and Recall Curve

****

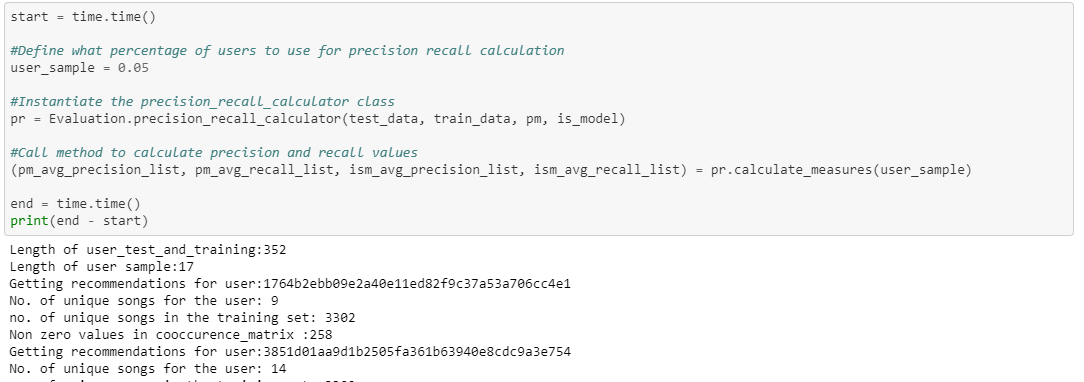
****

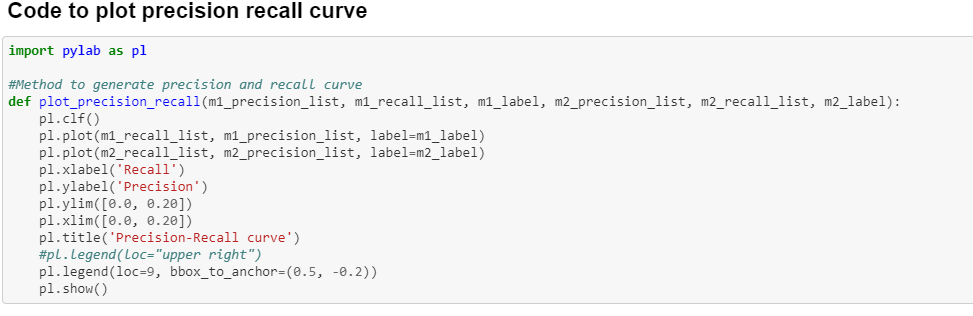
**Figure 31:** Evaluating the models we are using Precision and Recall Curve

****

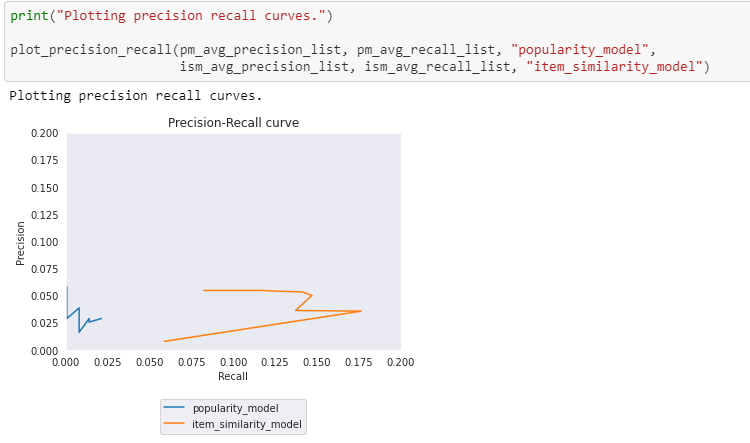
**Figure32: Importing the Evaluation Algorithm in evaluation package**

**4.6.1 Code to plot a precision recall curve**

****

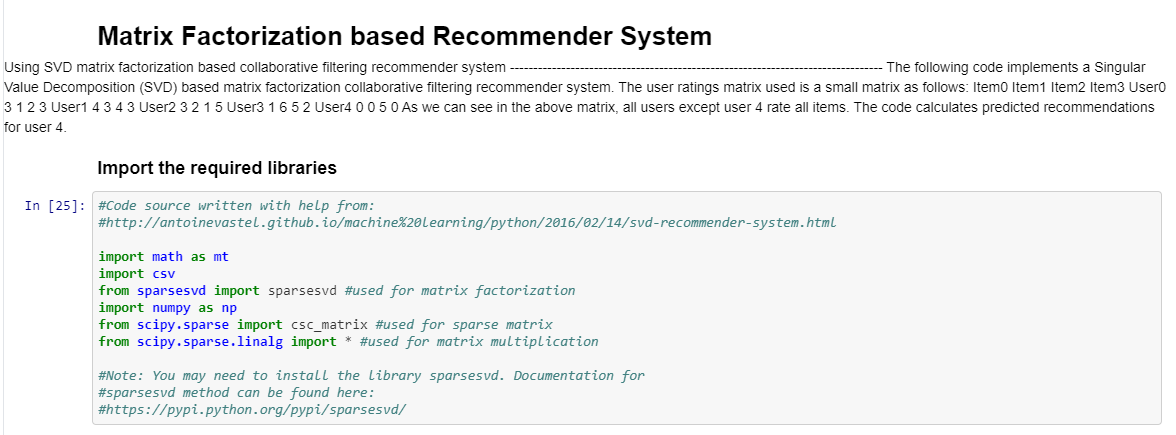
****

**Figure 33: Code to plot a precision recall curve**

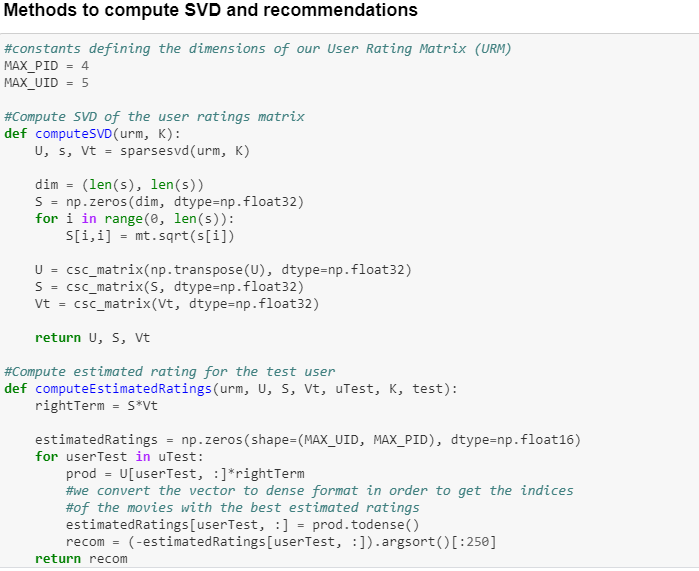
****

**Figure 34: Plotting precision recall curves**

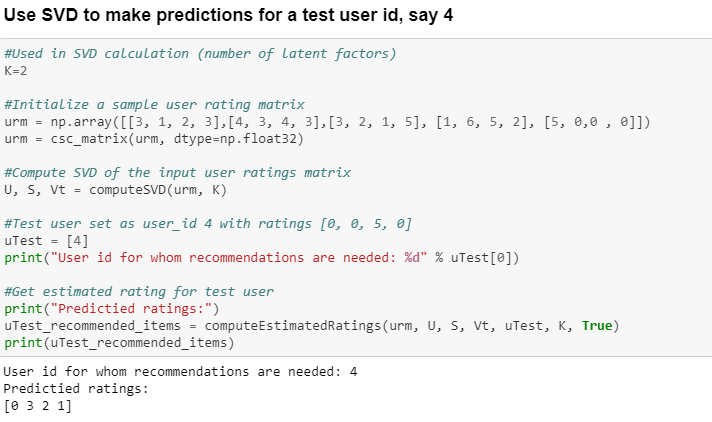
**4.7 SVD and Matrix Factorization based Recommender System**

****

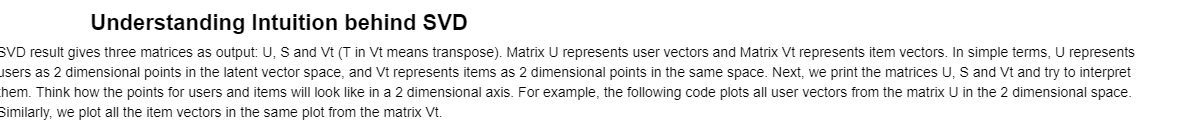
**Figure 35: Matrix Factorization based Recommender System**

****

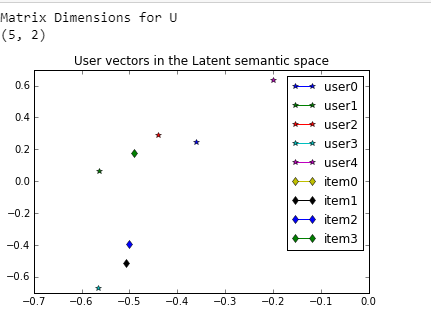
**Figure 36: Methods to compute SVD Recommendations**

****

**Figure 37: Code using SVD to make predictions for a test user id**

****



****

**Figure 38: Understanding Institution behind SVD**

**5 Conclusion:**

The curve shows that the personalized model provides much better performance over the popularity mode

By this method or project one can easily recommend songs to user by his history of listening songs and by giving a song to the model and the model predicts similar songs.

# 6 REFERENCES:

1. https://en.wikipedia.org/wiki/Machine\_learning
2. <https://en.wikipedia.org/wiki/Deep_learning>
3. https://en.wikipedia.org/wiki/pythonbasics