Utsav Chandresh Savaliya

Utsavchandresh.savaliya@sjsu.edu

Lab 2 report

CMPE 257 MACHINE LEARNING

**Task 1 Artificial Neural Network**

1. **Load the dataset (CIFAR – 10) from the given link:**

Data = cifar10.load\_data( )

1. **Explain in your own words:**
2. **Gradient Descent**

Gradient descent is an optimization algorithm which is commonly-used to train [machine learning](https://www.ibm.com/cloud/learn/machine-learning) models and [neural networks](https://www.ibm.com/cloud/learn/neural-networks).  Training data helps these models learn over time, and the cost function within gradient descent specifically acts as a barometer, gauging its accuracy with each iteration of parameter updates.

1. **Drop out**

The term “dropout” refers to dropping out units (both hidden and visible) in a neural

network. Dropout refers to ignoring units (i.e. neurons) during the training phase of certain set of neurons which is chosen at random. By “ignoring”, I mean these units are not considered during a particular forward or backward pass.

1. **Activation functions**

Activation function decides, whether a neuron should be activated or not by calculating weighted sum and further adding bias with it. The purpose of the activation function is to **introduce non-linearity** into the output of a neuron.

1. **Back Propagation**

When training a neural network by gradient descent, a [loss function](https://deepai.org/machine-learning-glossary-and-terms/loss-function) is calculated, which represents how far the network's predictions are from the true labels. Backpropagation allows us to calculate the gradient of the loss function with respect to each of the weights of the network. This enables every weight to be updated individually to gradually reduce the loss function over many training iterations.

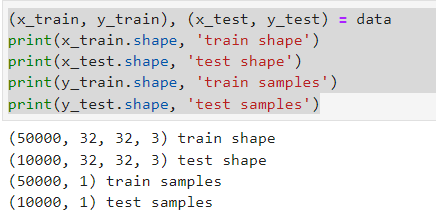
1. **Epochs, Iterations and Batch size**

An **epoch** is a term used in machine learning that refers to the number of passes the machine learning algorithm has made over the entire training dataset.

The **batch size** is the number of samples processed before updating the model. The number of epochs represents the total number of passes through the**training dataset.**

**Iterations** is the number of batches needed to complete one epoch.

1. **Visualize / summarize the data:**
2. Number of entities in training and testing set and number of classes in target variable



b. Number of pixels in the image (Height and width individually)

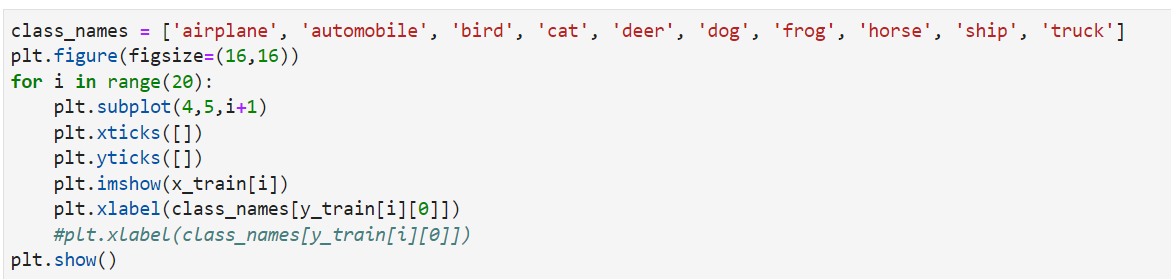
Rows = 32 and Columns=32. Therefore number of pixels= 32\*32

c. Number of images per class

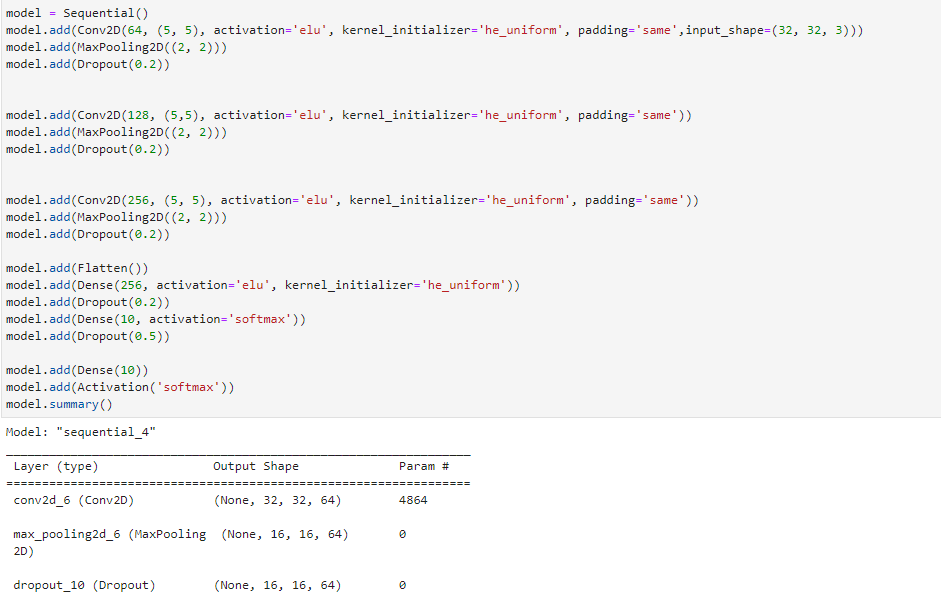
There are 6000 images per class which makes a total of 60,000 images in the whole

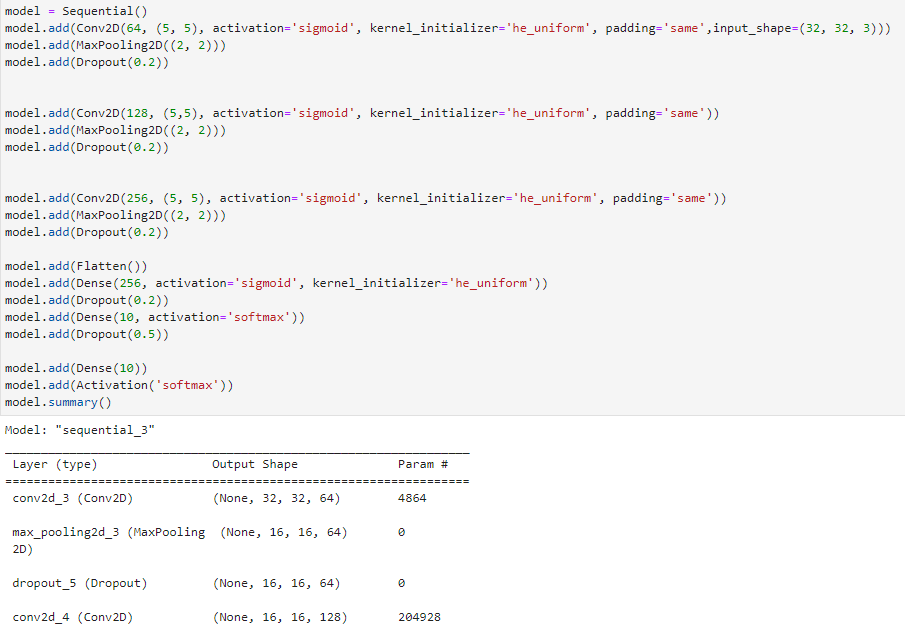
Dataset

d. Display at least 2 images of each class



1. Train a neural network
2. Decide the number of layers and neurons in each layer
3. Try out different activation functions

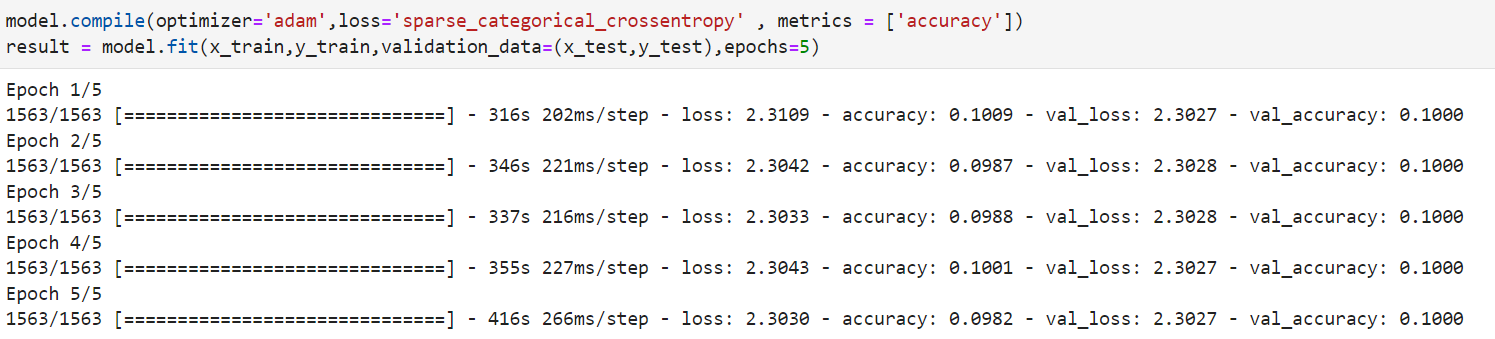
Relu, sigmoid, elu etc activation function is being tried out



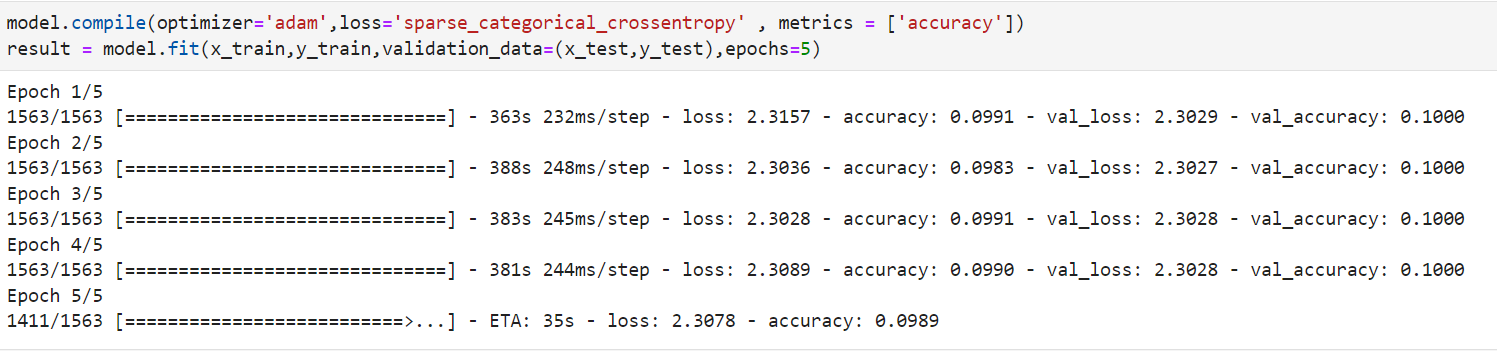


1. Try with and without using drop out

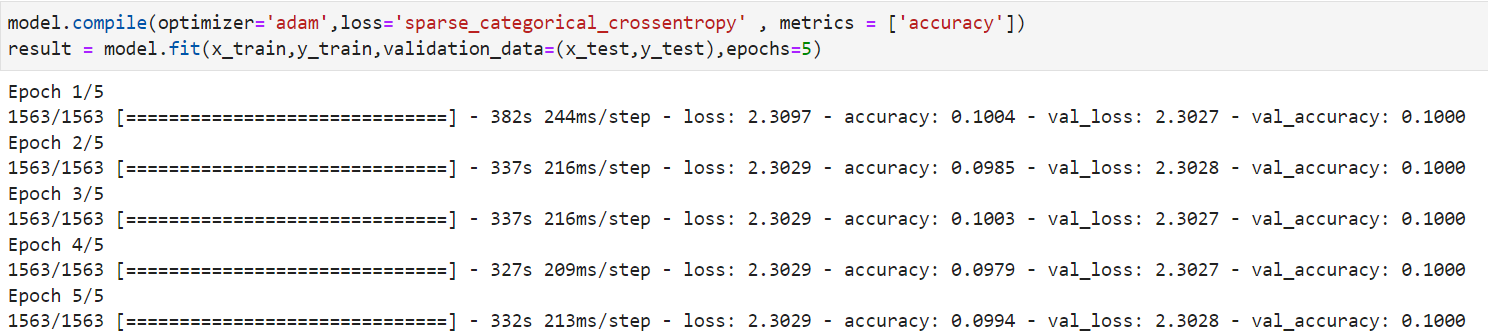
Here the Dropout is 0.3



Dropout as 0.5



Dropout is not used



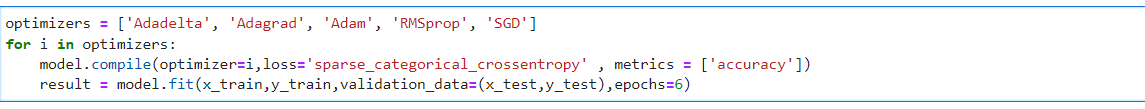
1. Try different regularizations apart from dropout:

kernel reguliser = l1(0.0000001)

kernel reguliser= l2(0.0000001)



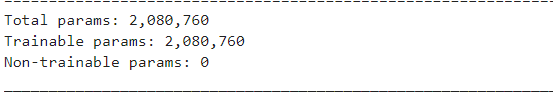
e. Try different optimization algorithms (such as Gradient Descent, Adam etc.)

Adadelta, Adagrad, Adam, RMSprop, SGD etc are used here.

1. Create a graph of f1 score vs epochs for training and validation set.

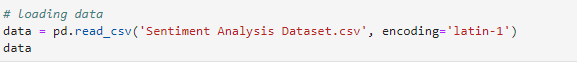
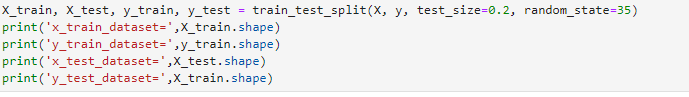


I have used only 5 epochs here as the running is very high due to some limitations.

g. Calculate the number of trainable parameters in your final model.

**Task 2 Natural Language Processing**

1. Load the movie reviews sentiment analysis dataset and split into 80:20 ratio for training and test data

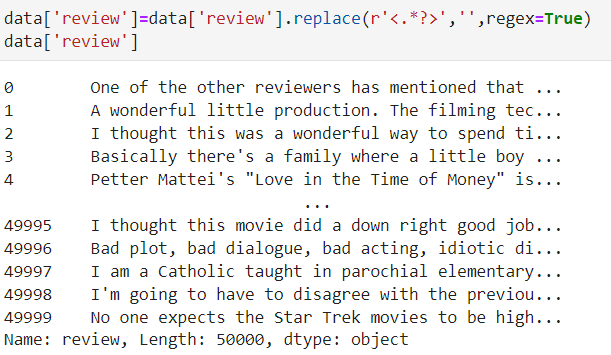
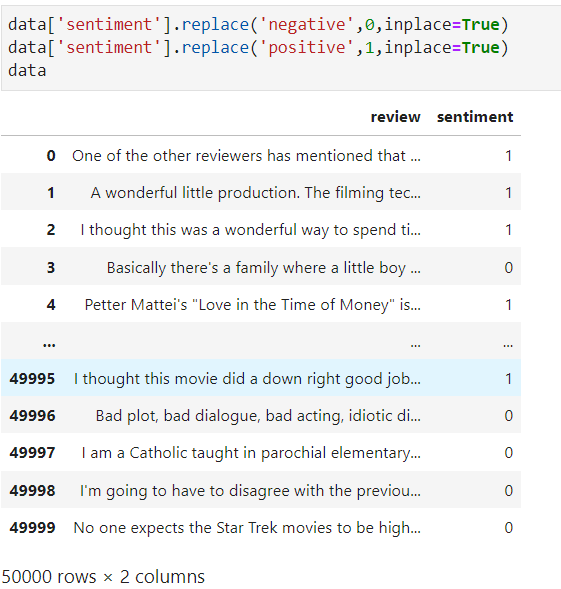


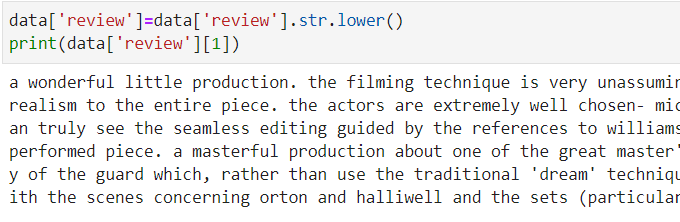
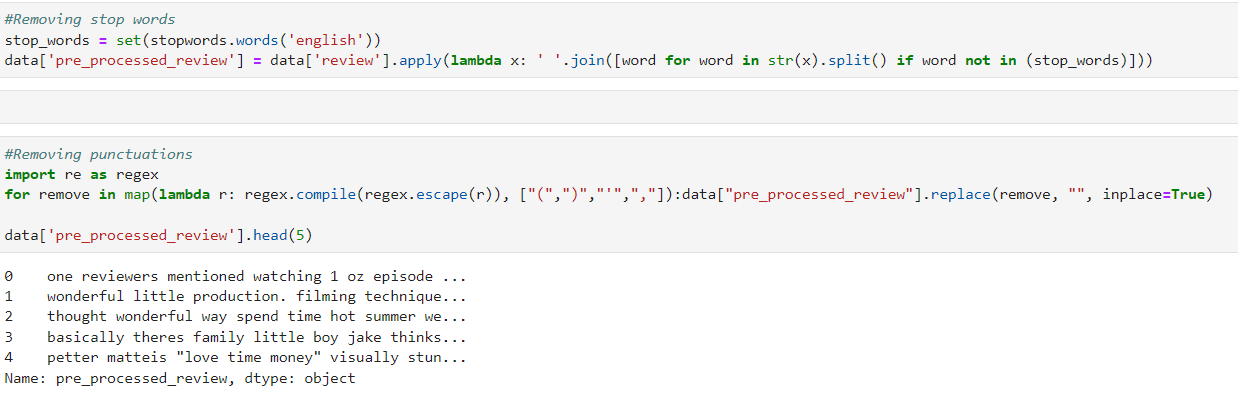
2. What is lexical vs semantic text analysis

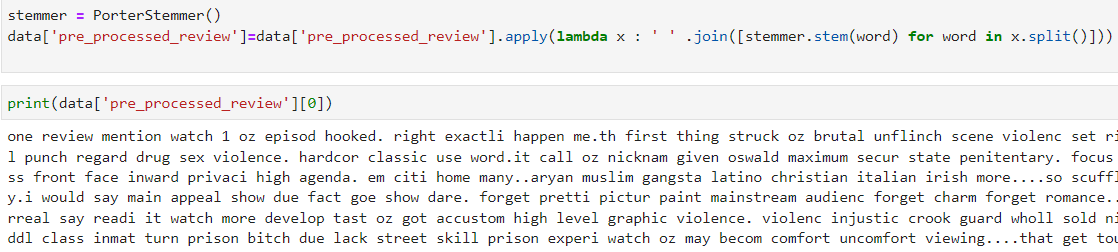
**Lexical analysis is based on smaller token but on the other side semantic analysis focuses on larger chunks.** Semantic Analysis captures the meaning of the given text considering into account context, of sentences grammar roles and logical structuring. Its aim is to find the meaning from the given text.

In other words the purpose of semantic analysis is to draw exact meaning, or you can say dictionary meaning from the text. **Lexical analysis is the process of trying to understand what words mean, intuit their context, and note the relationship of one word to others. It is often the entry point to many NLP data pipelines.**

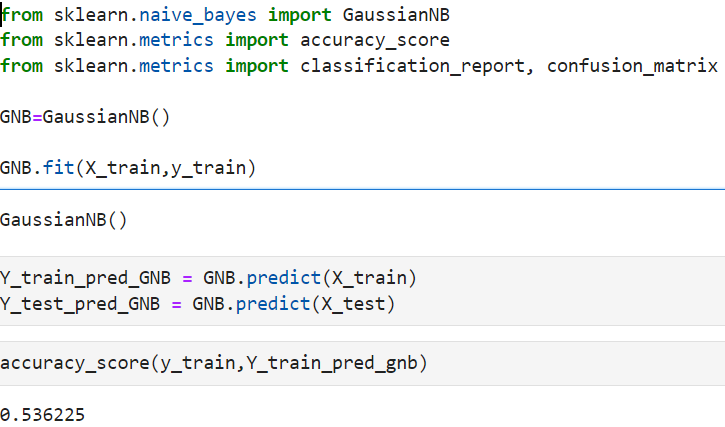
The first step of compilation, called lexical analysis, is to convert the input from a simple sequence of characters into a list of tokens of different kinds, such as numerical and string constants, variable identifiers, and programming language keywords.

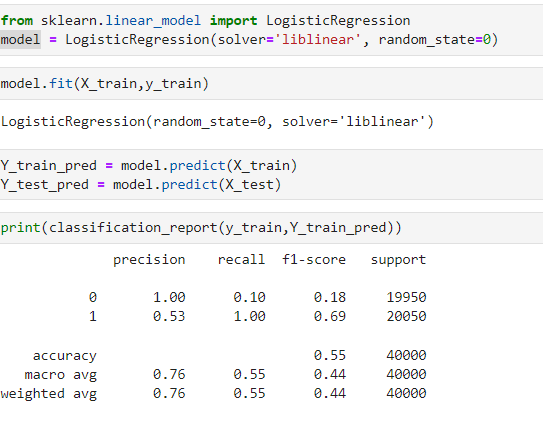
3. Perform required preprocessing on the dataset (removing stop words, vectorization)



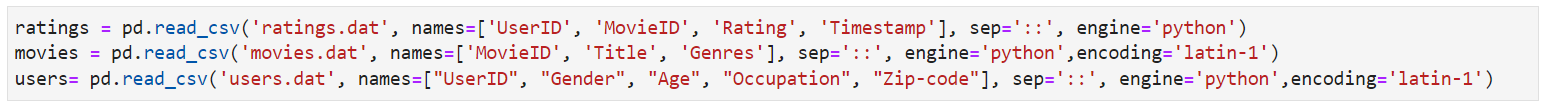


4.Build a model to classify the rows





**Task 3 Recommender System**

1. Download m1.zip file from the link
2. Load the movies and ratings data
3. What do you mean by Singular Value Decomposition

In [linear algebra](https://en.wikipedia.org/wiki/Linear_algebra), the singular value decomposition (SVD) is a [factorization](https://en.wikipedia.org/wiki/Matrix_decomposition) of a [real](https://en.wikipedia.org/wiki/Real_number) or [complex](https://en.wikipedia.org/wiki/Complex_number) [matrix](https://en.wikipedia.org/wiki/Matrix_(mathematics)). It generalizes the [eigen decomposition](https://en.wikipedia.org/wiki/Eigendecomposition) of a square [normal matrix](https://en.wikipedia.org/wiki/Normal_matrix) with an orthonormal eigen basis to any matrix. It is related to the [polar decomposition](https://en.wikipedia.org/wiki/Polar_decomposition#Matrix_polar_decomposition).

* A = U W V^T
* U: m x n matrix of the orthonormal eigenvectors of A A^T .
* W: n x n diagonal matrix of the singular values which are the square roots of the eigenvalues of A^T A
* V^T: transpose of a n x n matrix containing the orthonormal eigenvectors of A^T A.

It is used for : Calculation of Pseudo-inverse , Solving a set of Homogeneous Linear Equations (Mx =b), Curve Fitting Problem

1. What do you mean by Principal Component Analysis ?

* Principal Component Analysis (PCA) is a statistical technique used for data

reduction without losing its properties.

* Basically, it describes the composition of variances and covariances through several

linear combinations of the primary variables, without missing an important part of

the original information.

* In another term, it is about obtaining a unique set of orthogonal axes where the data

has the largest variance. Its main aim is to overcome the dimensionality of the

problem.

* The reduction of dimensionality should be such that when dropping higher

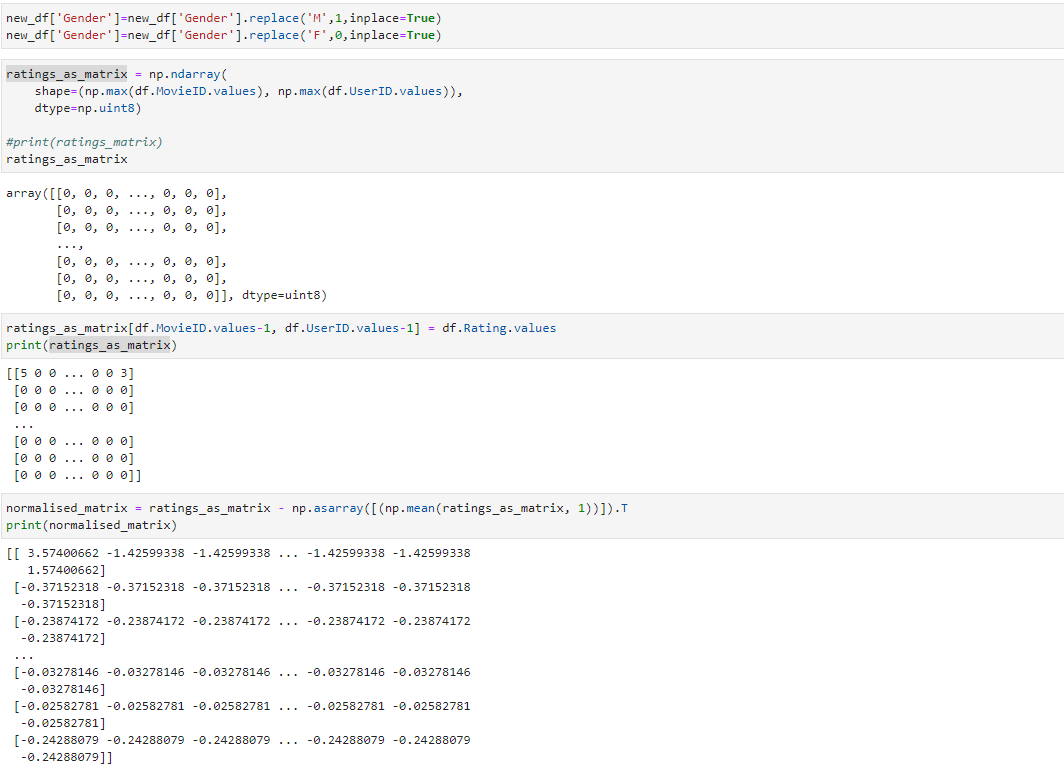
dimensions, the loss of data is minimum.

1. Explain content-based vs collaborative recommendation?

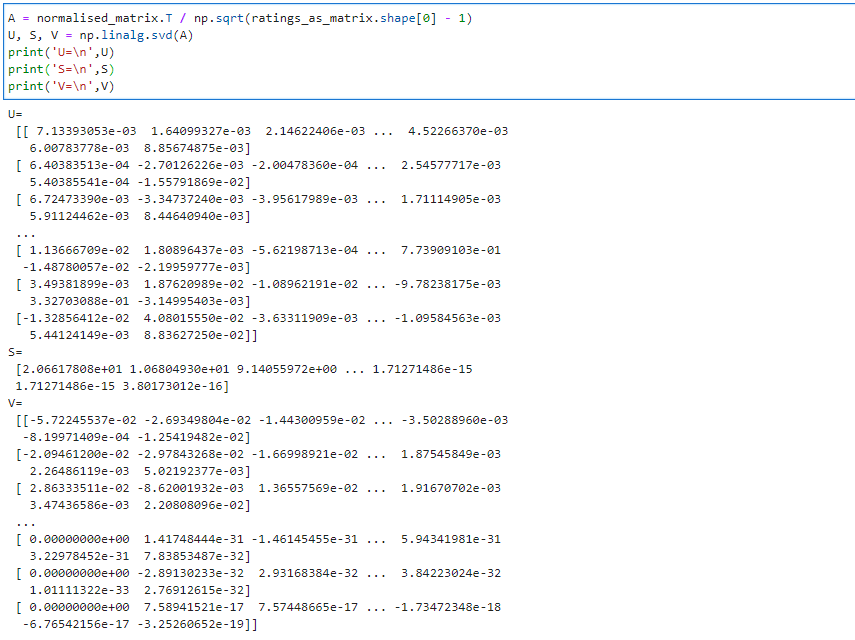
**Content-based filtering system:** Content-Based recommender system tries to guess the features or behavior of a user given the item’s features, he/she reacts positively to.

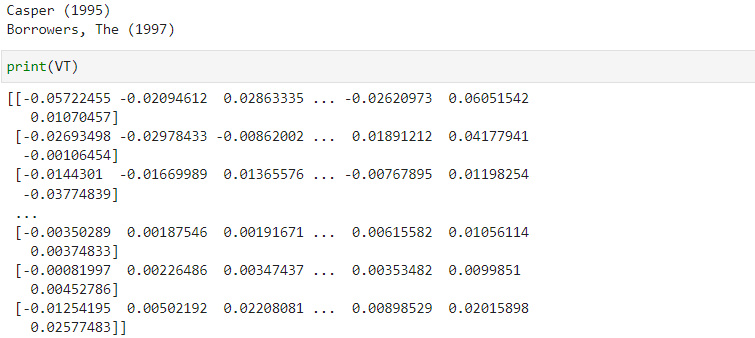
**Collaborative filtering System:** Collaborative does not need the features of the items to be given. Every user and item is described by a feature vector or embedding.

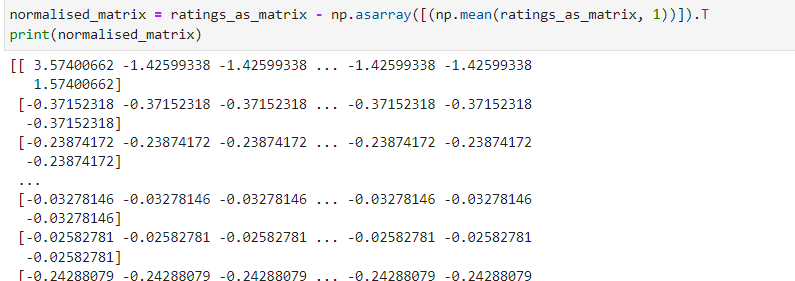
1. Create m x u matrix with movies as row and users as column. Normalize the matrix.

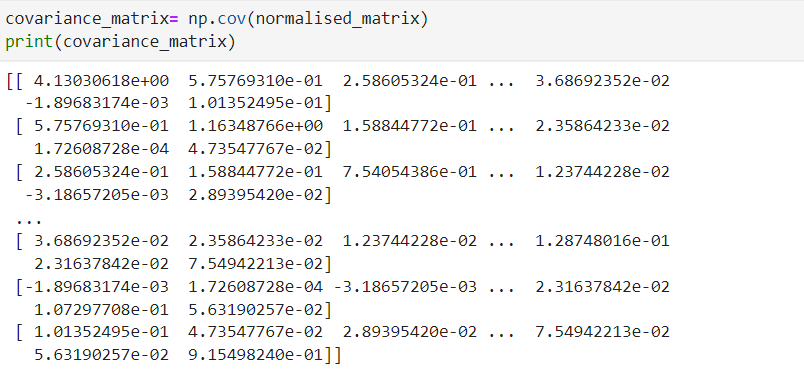


1. Perform SVD to get U, S and V

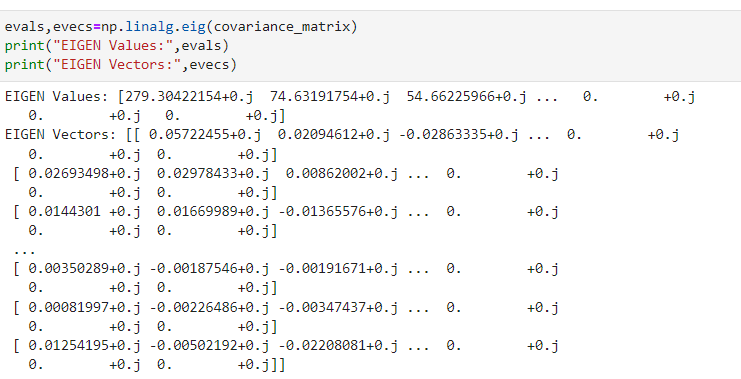


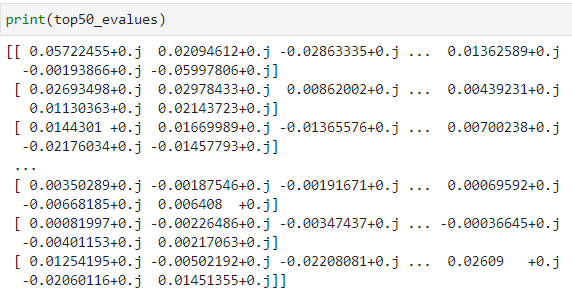
1. Select top 50 components from V.T
2. Calculate the covariance matrix for the entire dataset (from step 6)



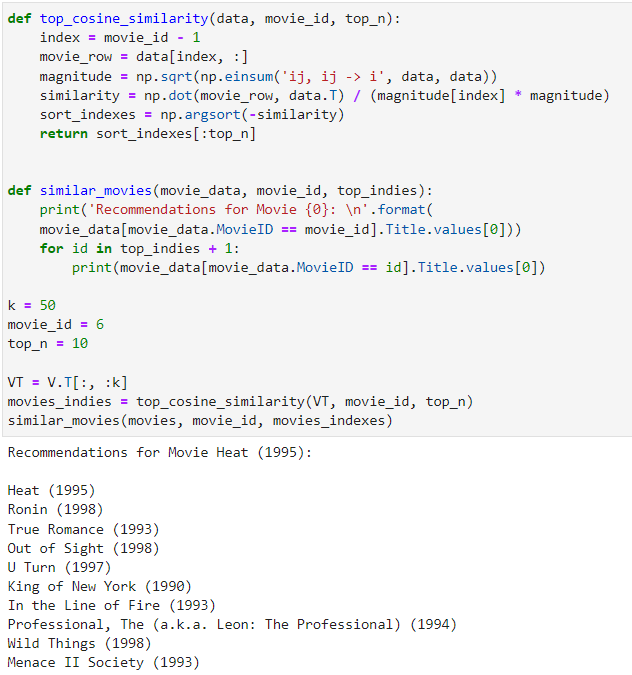


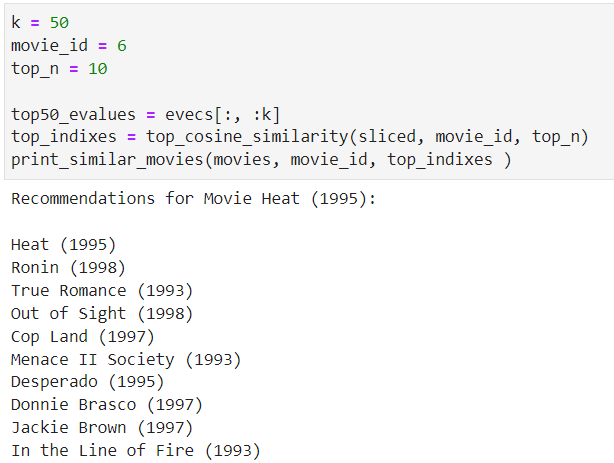
1. Get the eigen vectors from the covariance matrix



11. Get the top 50 eigen vectors using eigen values

12. Using cosine similarity find 10 closest movies using the 50 components from SVD (step 8)

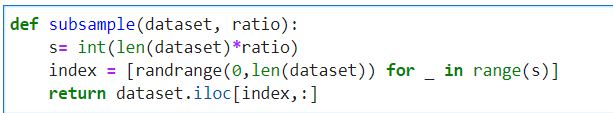
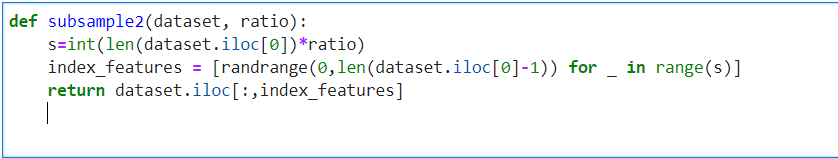


13. Using cosine similarity find 10 closest movies using the 50 components from PCA (step 11)

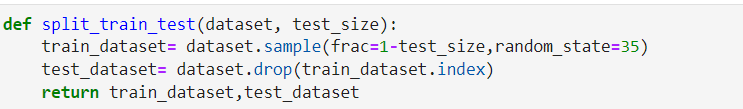
14. Compare the results of the above two methods

* I find that singular value decomposition seems faster than the covariance matrix method
* The eigenvalues are not sorted in ascending order and it needs to be sorted while the singular values from SVD are already sorted
* In results also there is a slight variation in order of the movies which is suggested

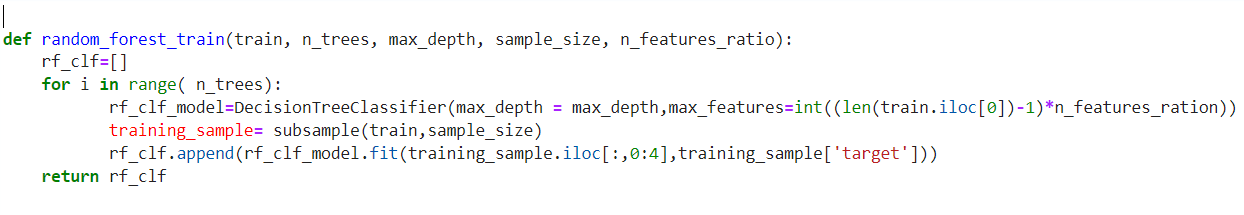
**Task 4 Random Forest – Self implementation:**

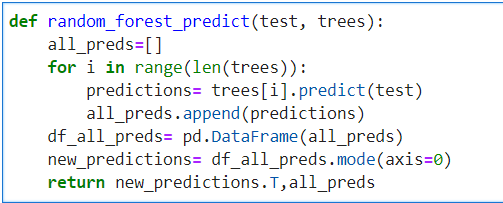
1. **Complete the following function that creates subsample of a dataset with replacement**
2. Complete the following function that creates subsample of the dataset with feature size reduced as per the given ratio

3. Perform train test split without using sklearn (complete the following function)



1. Perform training using random forest algorithm by completing the function below



1. Complete the following prediction function of random forest

**Note about my learnings:**

* The task given for this lab is very useful covers the broad segment of topics which are useful in order to learn ML or DL course.
* For task1 the training time was taking too much time to hardware constraints but I learned how to figure alternate options if there are those issues.
* For task2, the preprocessing part is main part which taught me that for any ML algorithm to run if the dataset is cleaned and in proper format than it is much easier to run our model.