**MINI PROJECT REPORT**

**Name:** Adarsh Kumar

**Section:** A

**Semester:** 6

**University Roll no:** 2013225

**Project Name:** Movie Recommendation System

**What are Recommendation Systems?**

**Recommender systems** are an important class of machine learning algorithms that offer “relevant” suggestions to users. Amazon, Netflix, all function on recommendation systems where the system recommends you the next video or product based on your past activity (**Content-based Filtering**) or based on activities and preferences of other users similar to you (**Collaborative Filtering**). Likewise, Facebook also uses a recommendation system to suggest Facebook users you may know offline. Recommendation Systems work based on the **similarity**between either the content or the users who access the content. There are several ways to measure the similarity between two items. The recommendation systems use this **similarity matrix** to recommend the next most similar product to the user.

In this project, I have built a machine learning algorithm that would recommend movies based on a movie the user likes. This Machine Learning model would be based on Cosine Similarity.

**Tools, Language and Libraries Used:**

Python

Visual Studio Code

Tkinter

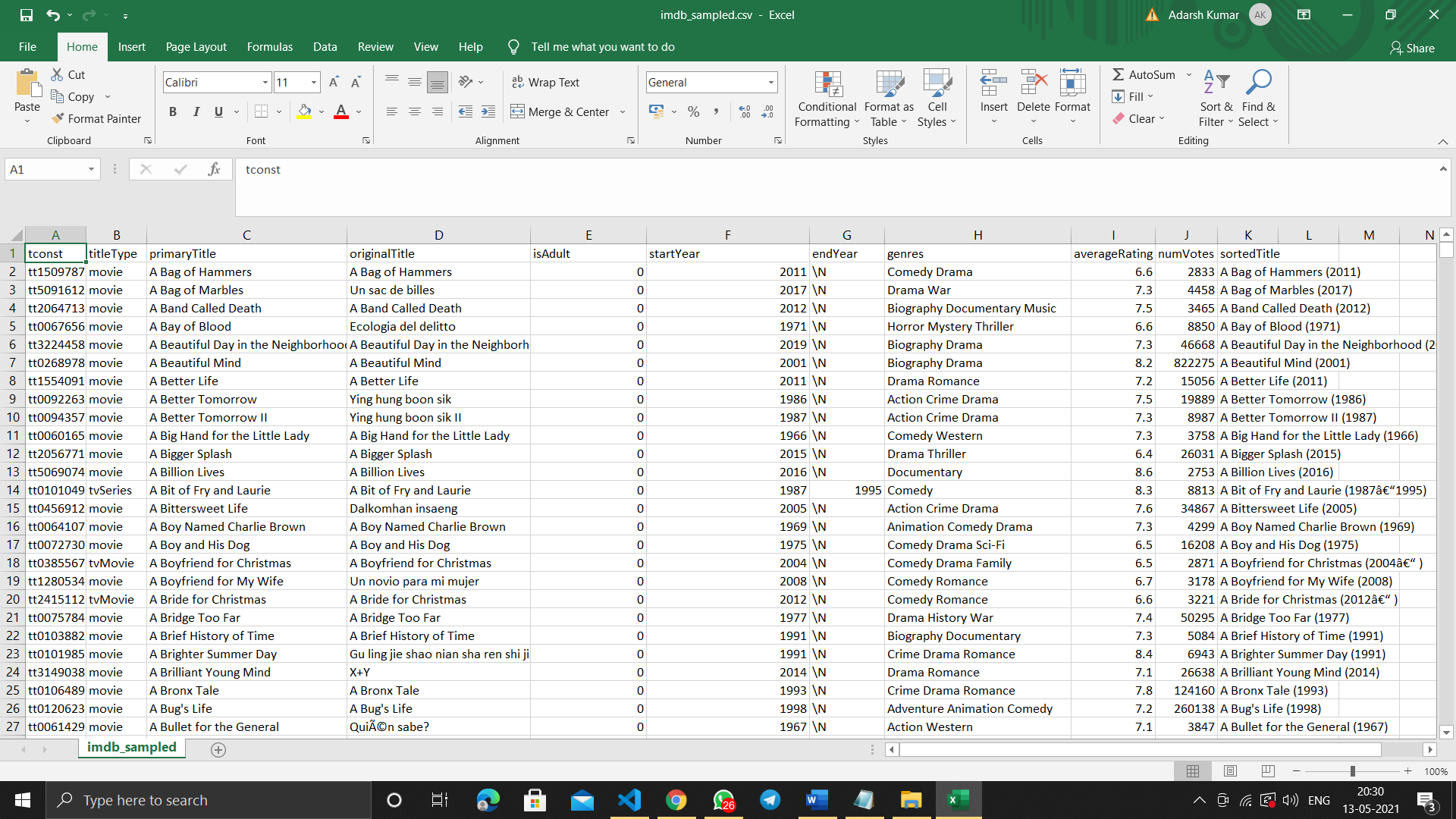
Numpy

Pandas

Sklearn

* The first step to build a movie recommendation system is getting the appropriate data.

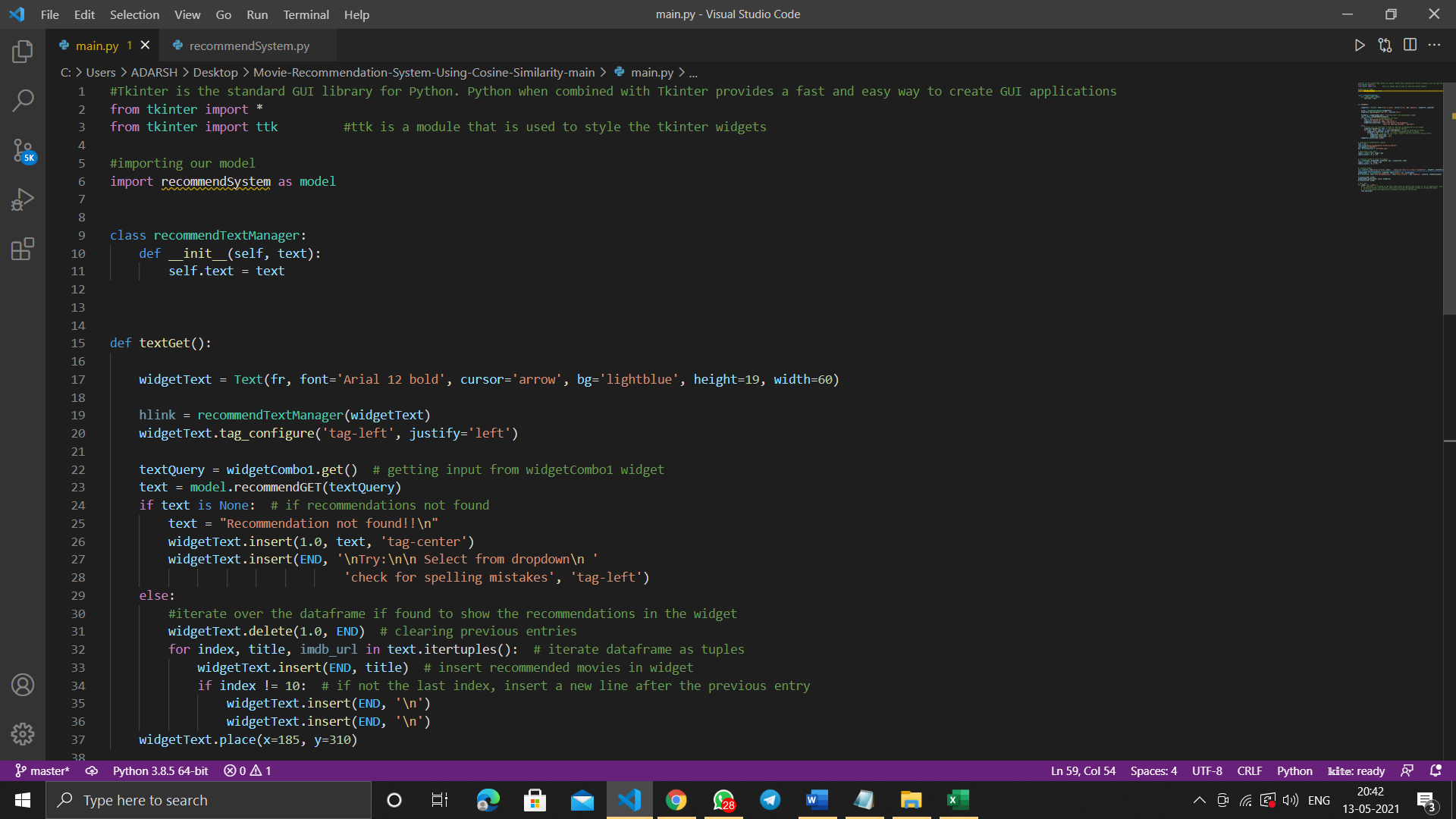
I have taken dataset named “imdb\_sampled.csv”

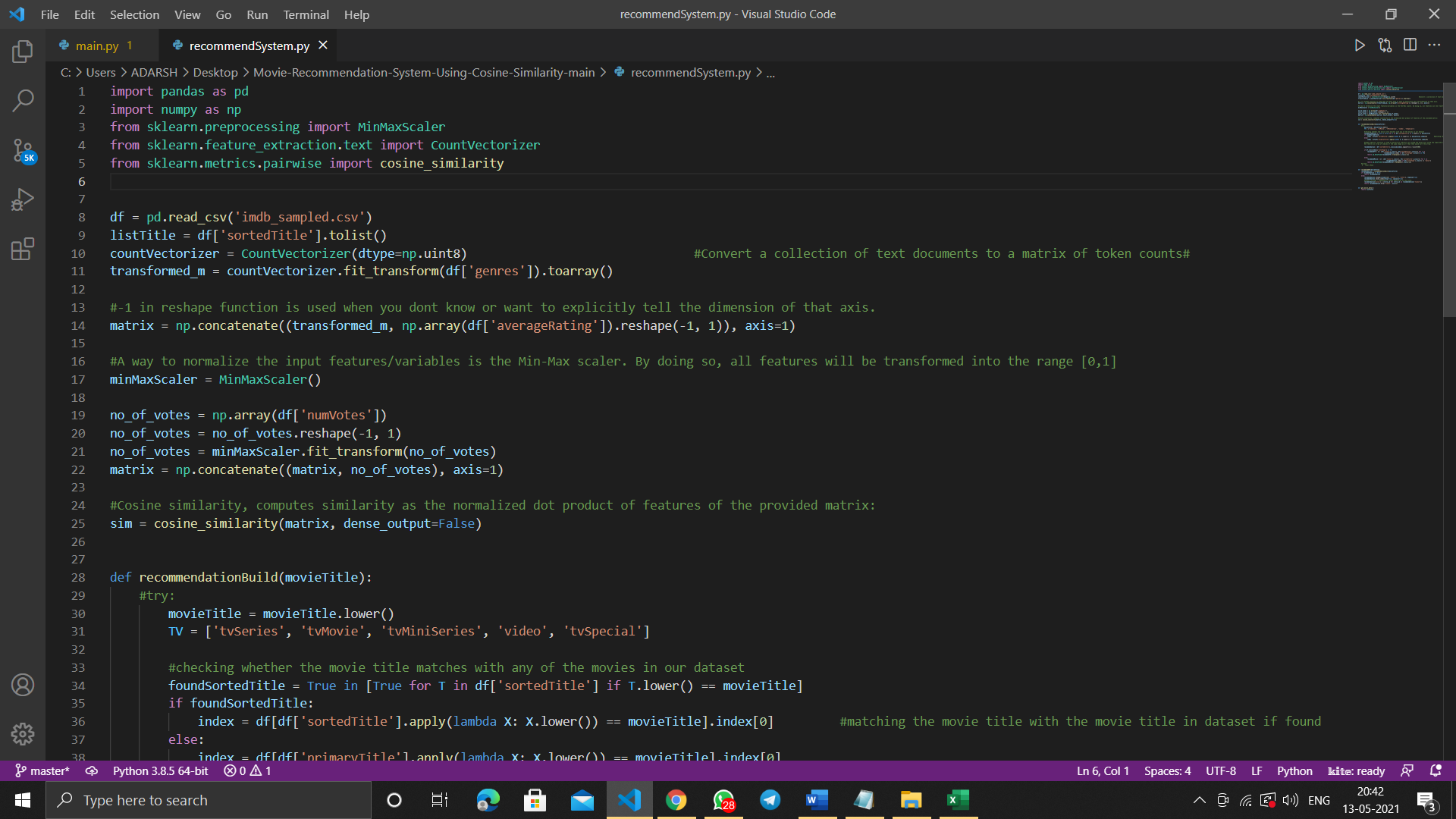


* tconst (string) - alphanumeric unique identifier of the title
* titleType (string) – the type/format of the title (e.g. movie, short, tvseries, tvepisode, video, etc)
* primaryTitle (string) – the more popular title / the title used by the filmmakers on promotional materials at the point of release
* originalTitle (string) - original title, in the original language
* isAdult (boolean) - 0: non-adult title; 1: adult title
* startYear (YYYY) – represents the release year of a title. In the case of TV Series, it is the series start year
* endYear (YYYY) – TV Series end year. ‘\N’ for all other title types
* genres (string array) – includes up to three genres associated with the title
* averageRating – weighted average of all the individual user ratings
* numVotes - number of votes the title has received.

Now I have imported some important libraries numpy and pandas for data analysis and manipulation. Scikit-learn’s CountVectorizer, used to convert a collection of text documents to a vector of term/token counts. Also cosine\_similarity from sklearn, as the metric of our similarity matrix.

Also tkinter for GUI.



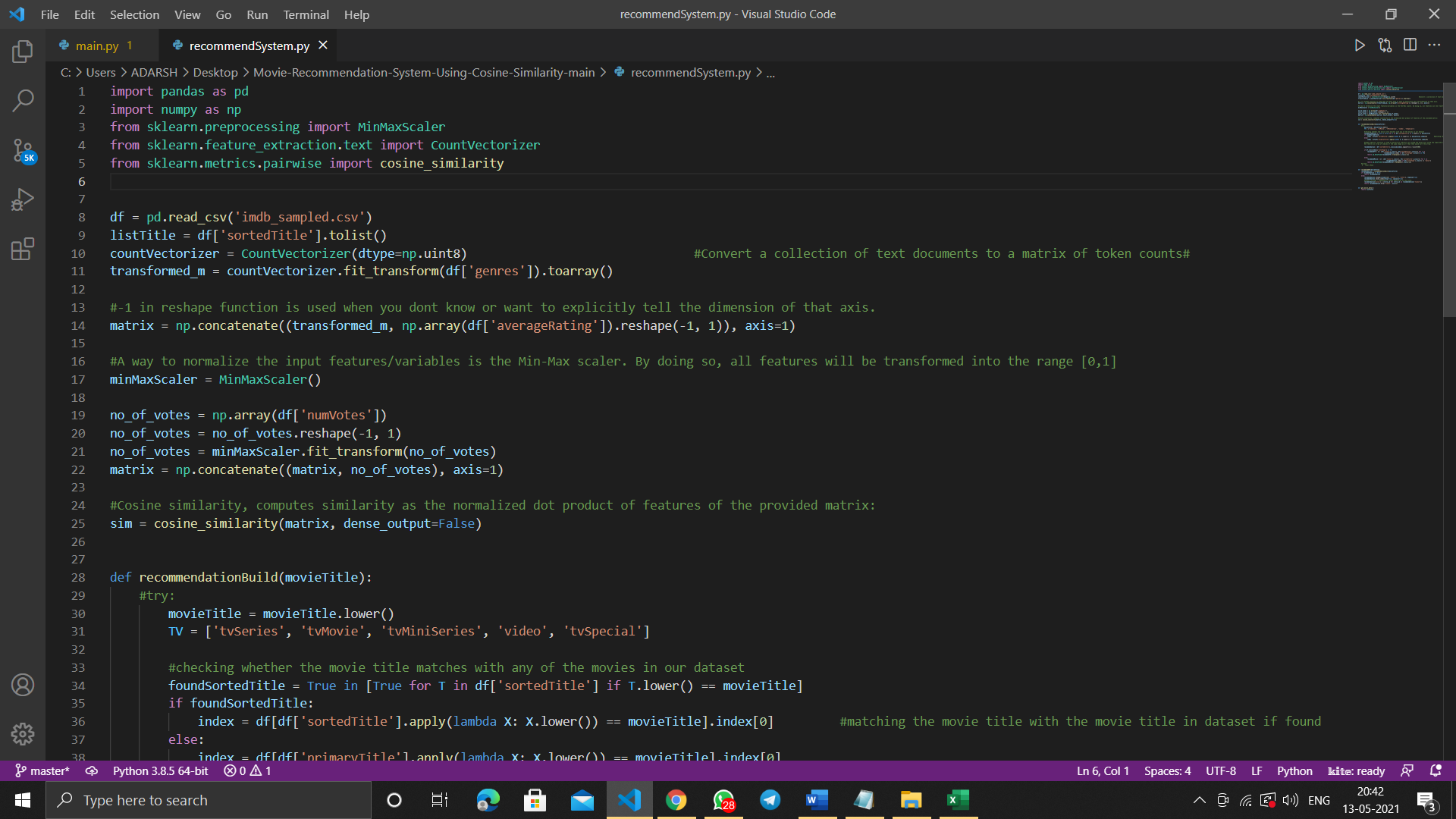


Some important methods used:

**Sklearn’s Count Vectorizer**

In order to use textual data for predictive modeling, the text must be parsed to remove certain words – this process is called tokenization. These words need to then be encoded as integers, or floating-point values, for use as inputs in machine learning algorithms. This process is called feature extraction (or vectorization).

Scikit-learn’s CountVectorizer is used to convert a collection of text documents to a vector of term/token counts. It also enables the ​pre-processing of text data prior to generating the vector representation. This functionality makes it a highly flexible feature representation module for text.

****

**MinMaxScaler**

For each value in a feature, MinMaxScaler subtracts the minimum value in the feature and then divides by the range. The range is the difference between the original maximum and original minimum. MinMaxScaler preserves the shape of the original distribution. It keeps the values between 0 and 1.

It is highly effected by outliers.

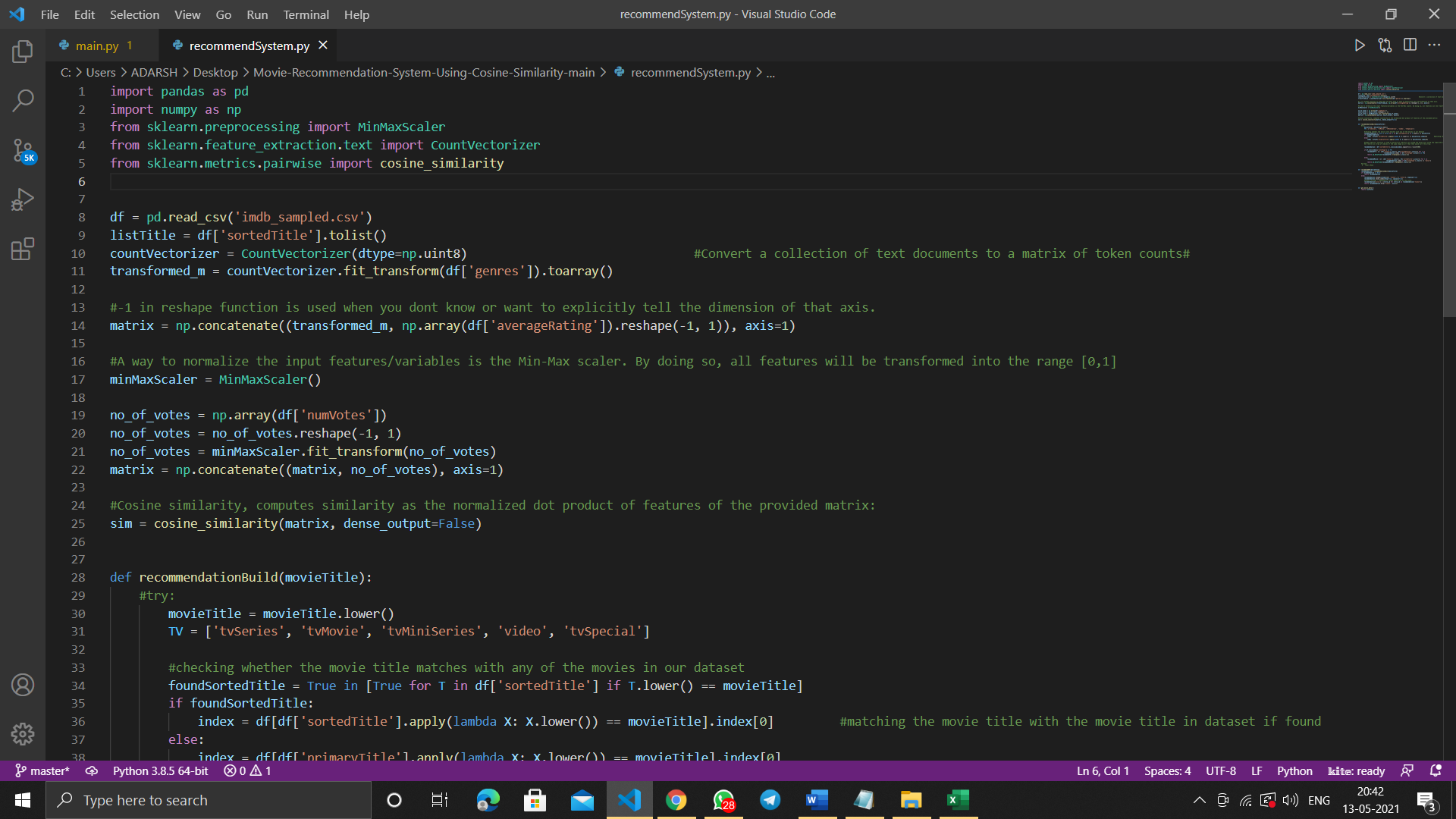
**cosine\_similarity**

We will use the Cosine Similarity from Sklearn, as the metric to compute the similarity between two movies.

Cosine similarity is a metric used to measure how similar two items are. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. The output value ranges from 0–1.

0 means no similarity, where as 1 means that both the items are 100% similar.

The python Cosine Similarity or cosine kernel, computes similarity as the normalized dot product of input samples X and Y. We will use the sklearn cosine\_similarity to find the cos θ for the two vectors in the matrix.



The sim matrix is a numpy array with calculated cosine similarity between each movies. As you can see in the image below, the cosine similarity of movie 0 with movie 0 is 1; they are 100% similar (as should be).

Similarly the cosine similarity between movie 0 and movie 1 is 0.105409 (the same score between movie 1 and movie 0 — order does not matter).

Movies 0 and 4 are more similar to each other (with a similarity score of 0.23094) than movies 0 and 3 (score = 0.0377426).

The diagonal with 1s suggests what the case is, each movie ‘x’ is 100% similar to itself.



**Now let’s look at the code:-**

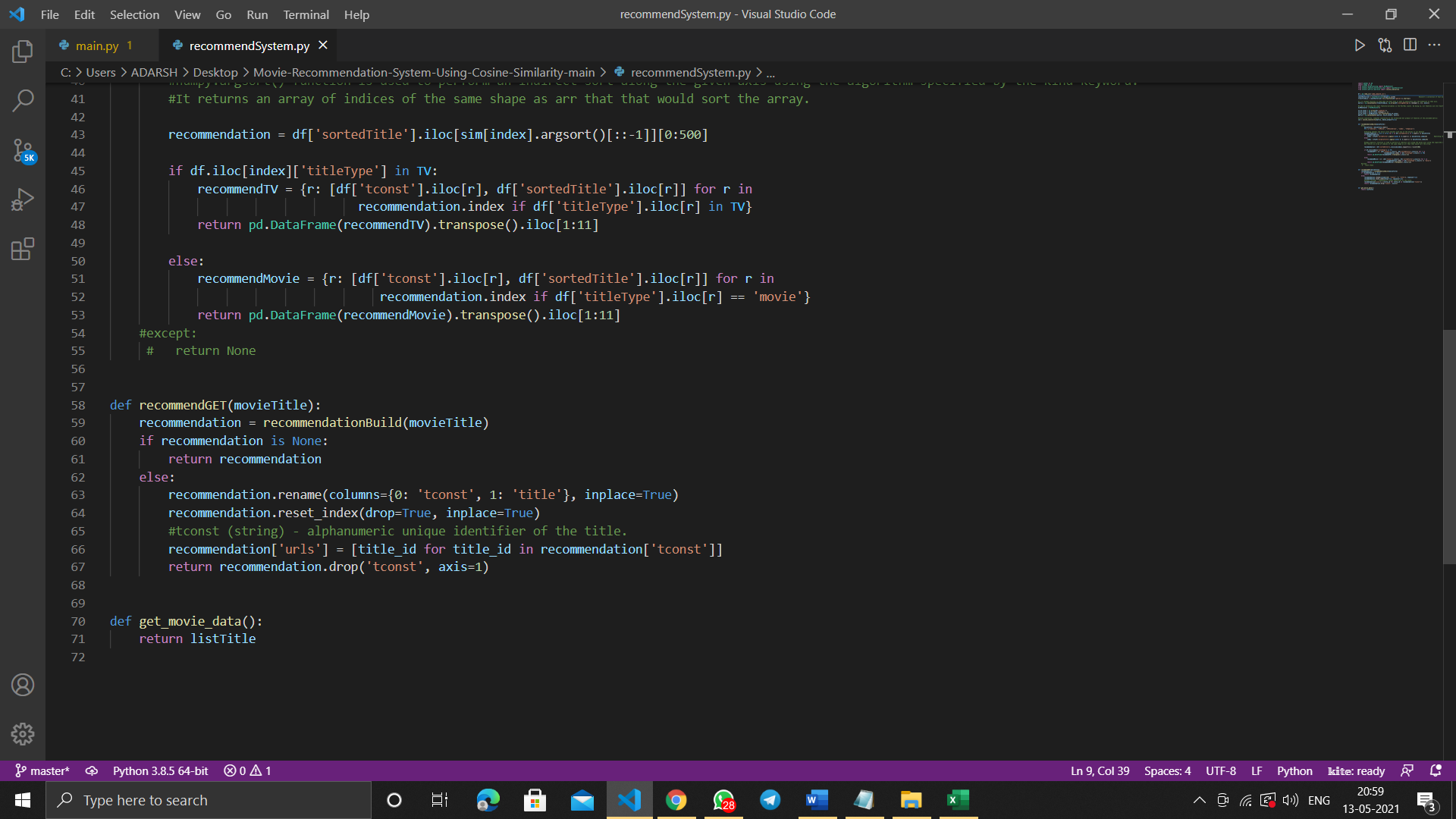
* **recommendSystem.py**

Now I have defined a function recommendGET which takes the input movie title which the user enters. From here function recommendationBuild will be called which contains the main logic for the code.

recommendationBuild will check whether there are movies that can be recommended from the dataset. The dataset would be containing many movies to be recommended ,but I have displayed the top 10 movies based on the ratings, genre and number of votes.

This is done by seeing the value in the sim matrix made by using cosine similarity.

The movies having high value will be selected and then returned back to the function recommendGET in decreasing order of rating and number of votes.

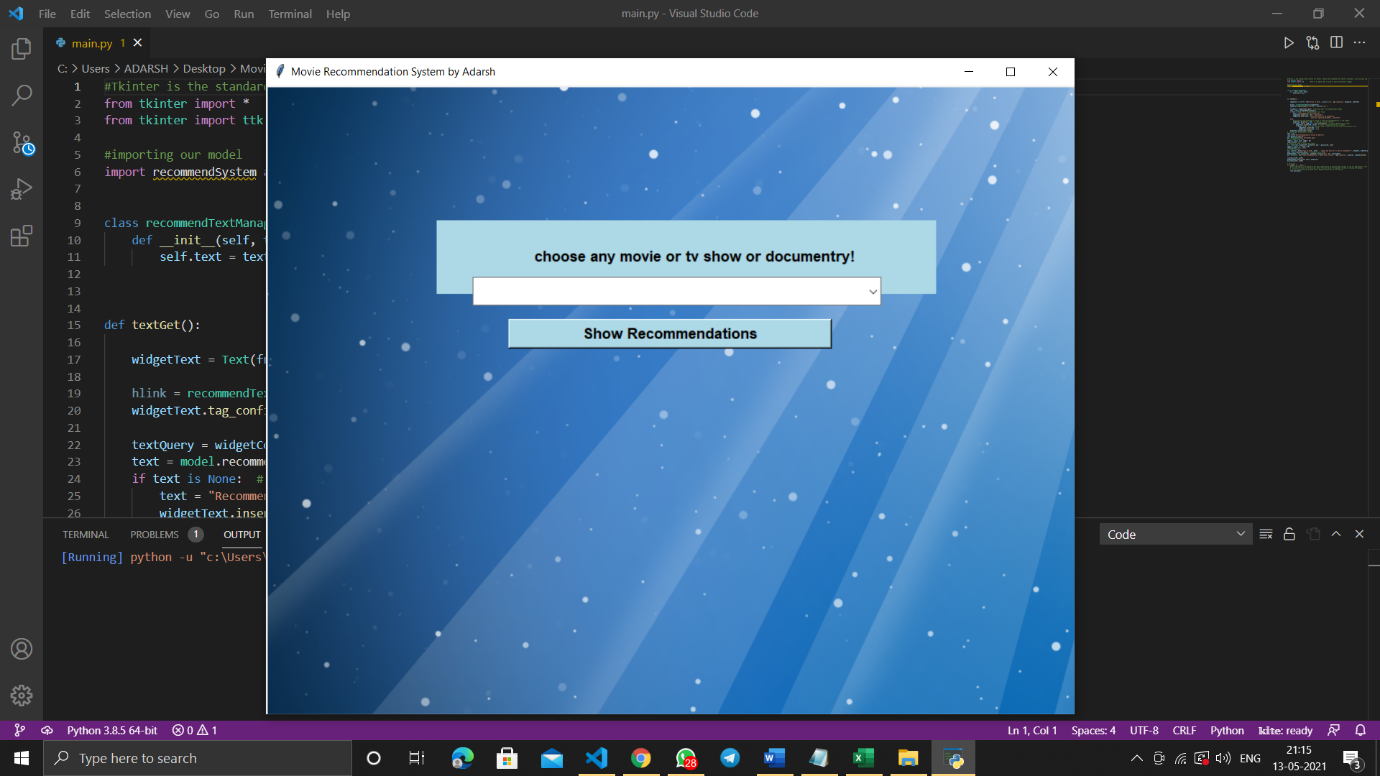




* **main.py**

This file contains the GUI code using tkinter.

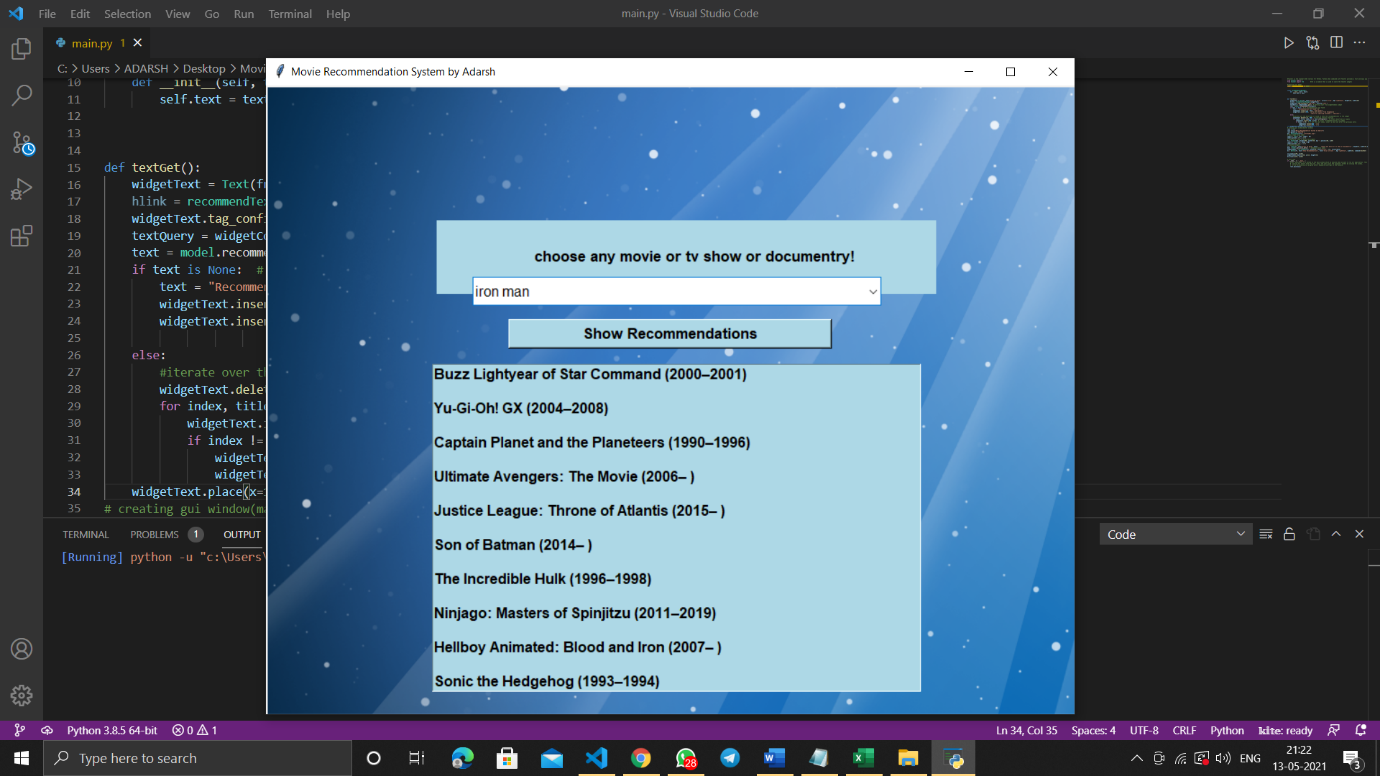
Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

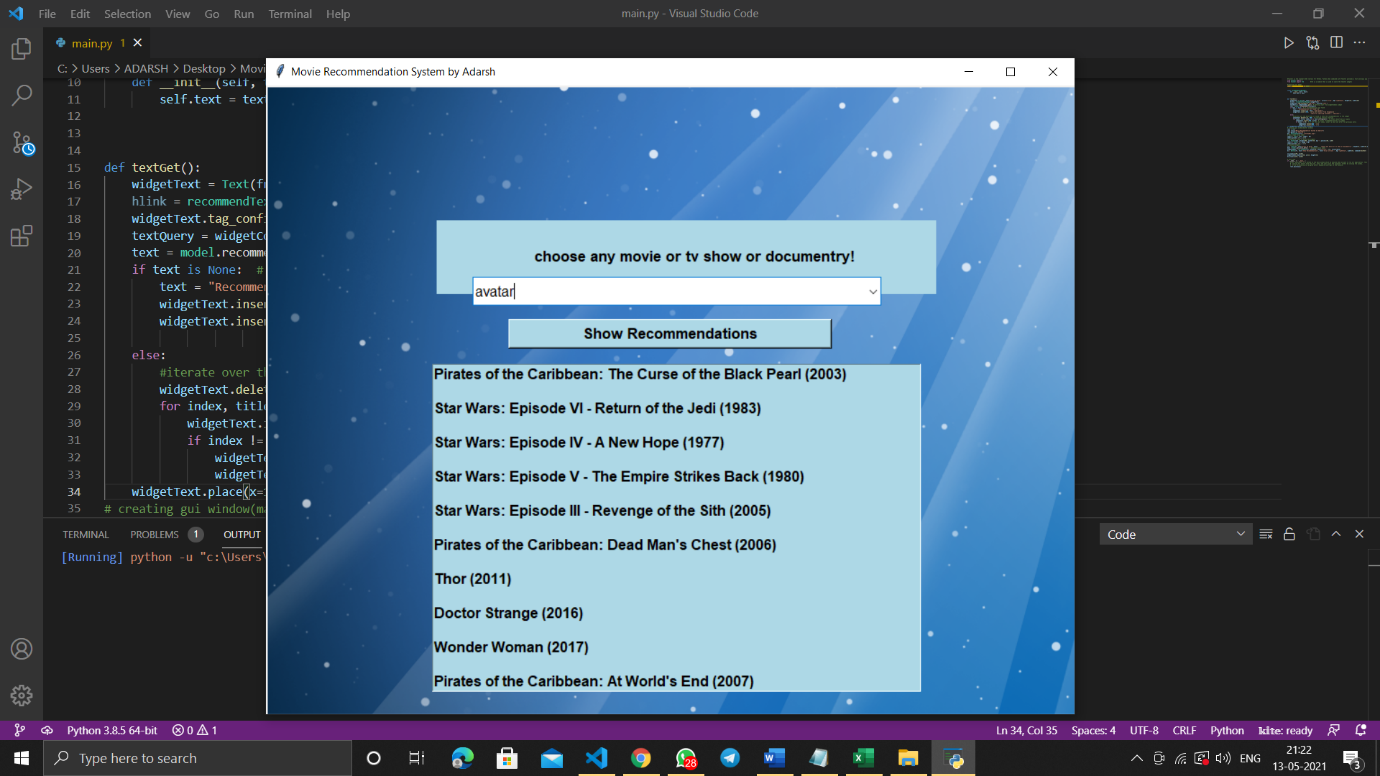


This is the GUI window created using tkinter with some widgets( label, combobox and button).

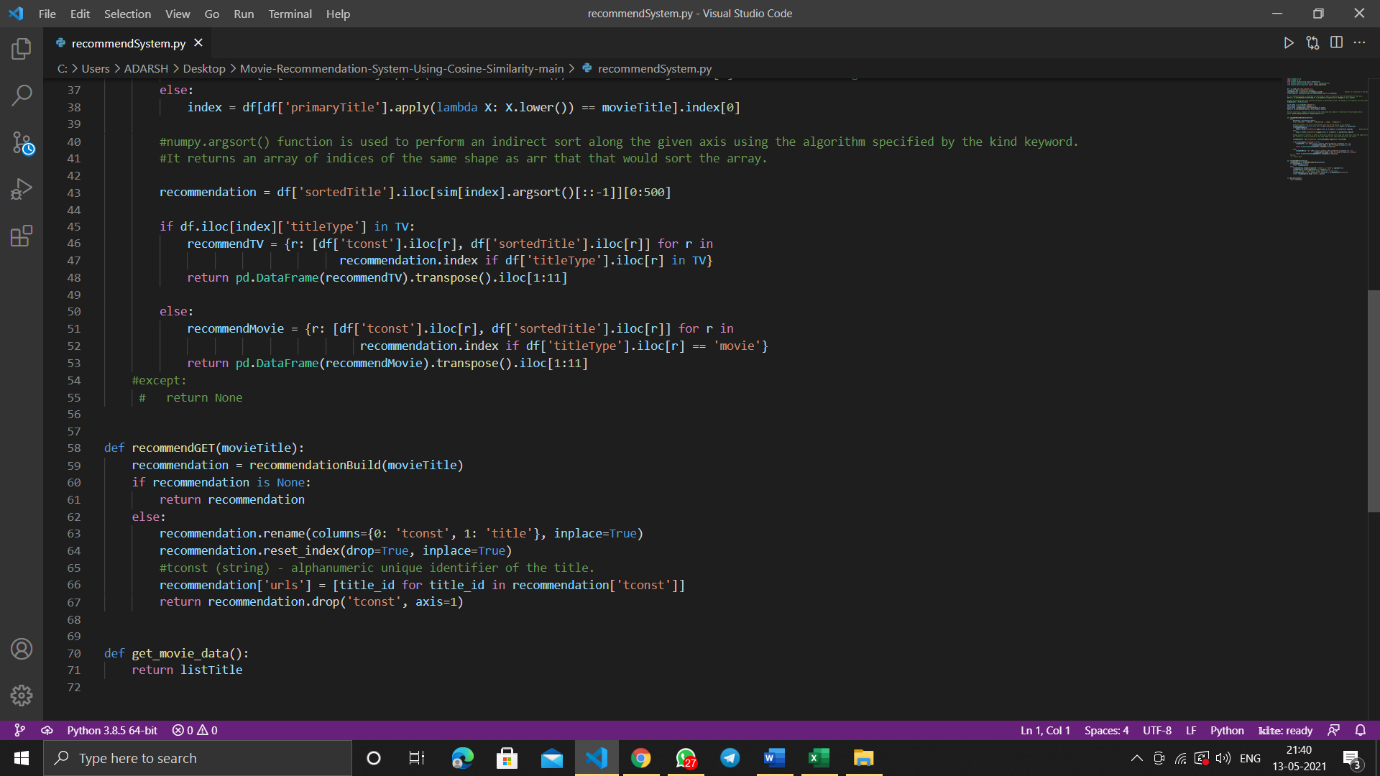
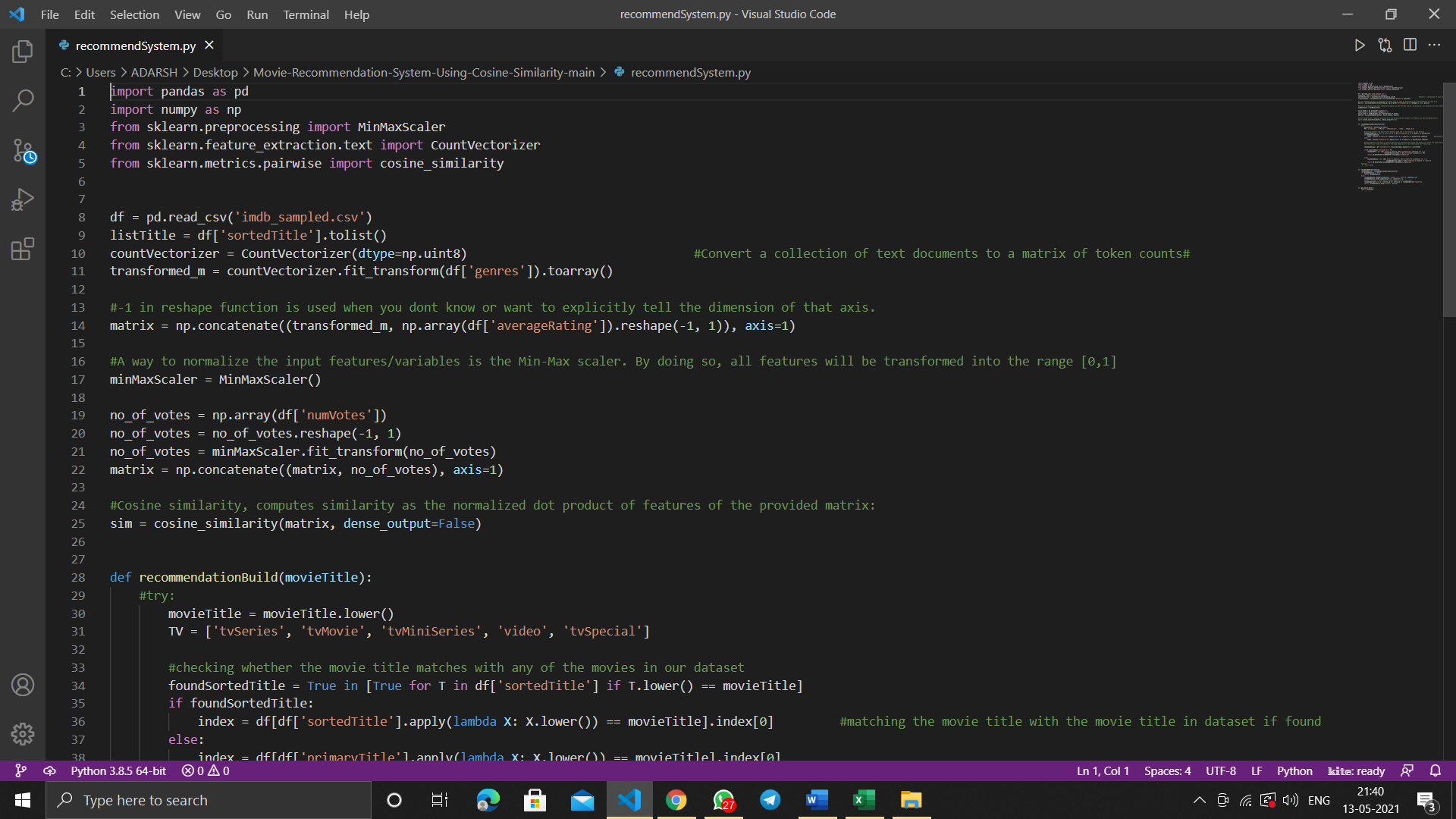
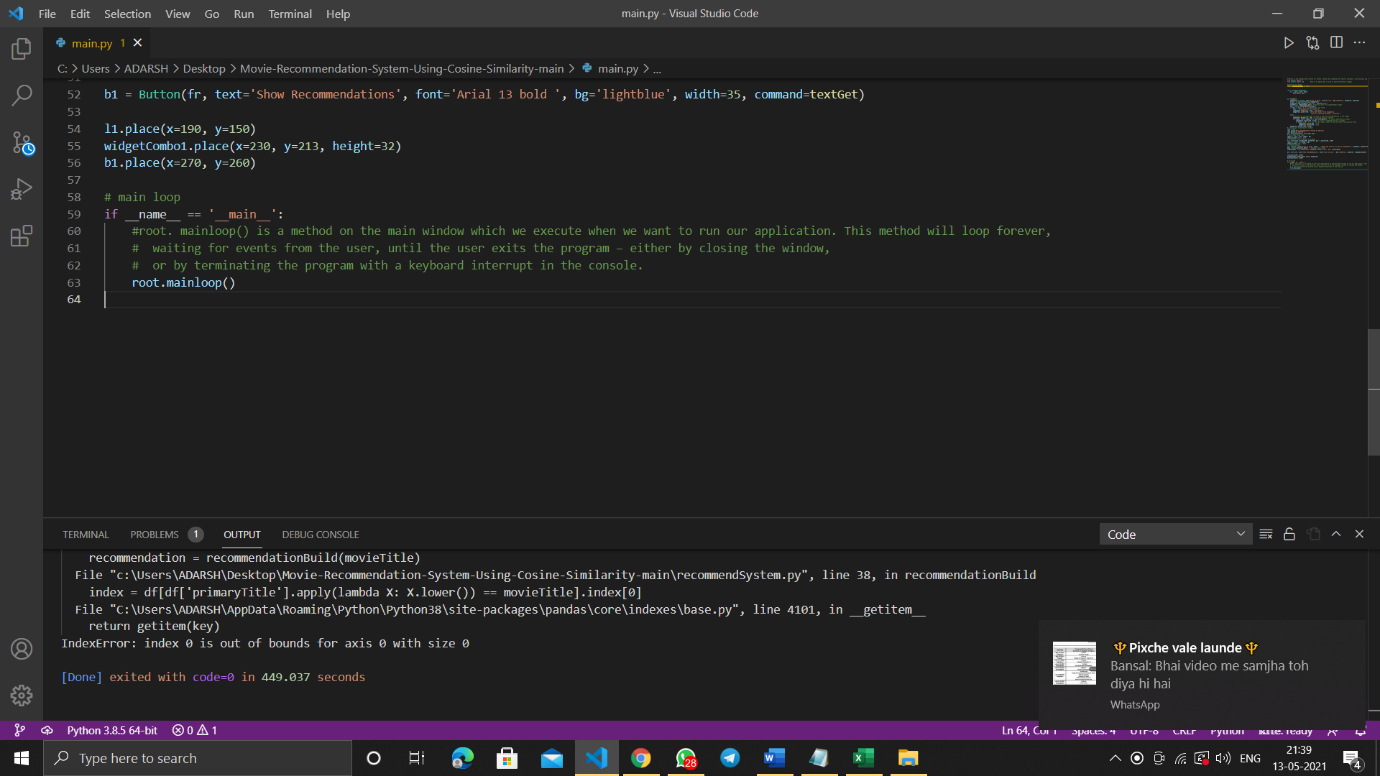
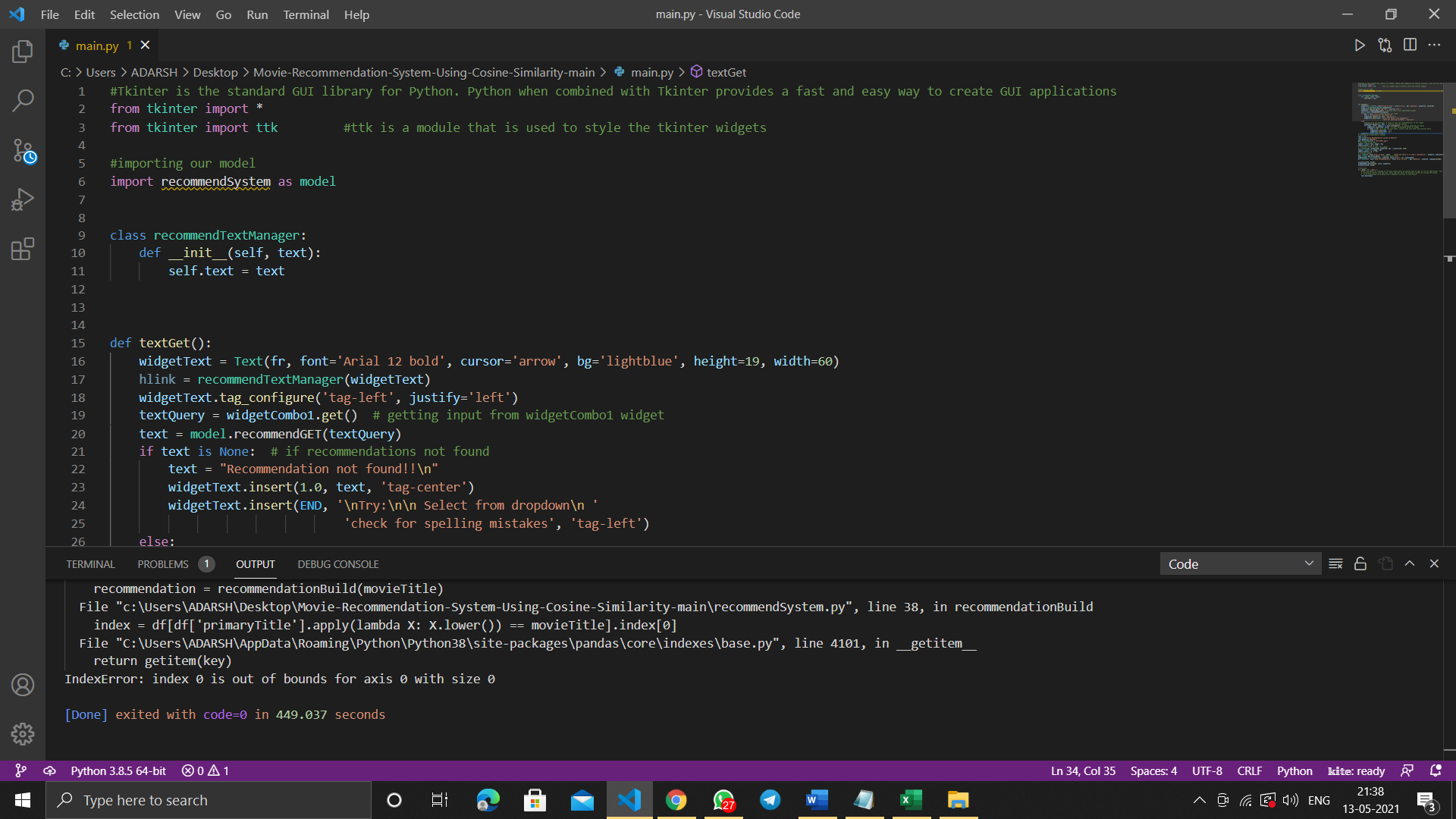
textGet function is defined to check that the user input is not empty and if not, then a suitable code is written to display the list of movies in the output widget.

**Final Output**

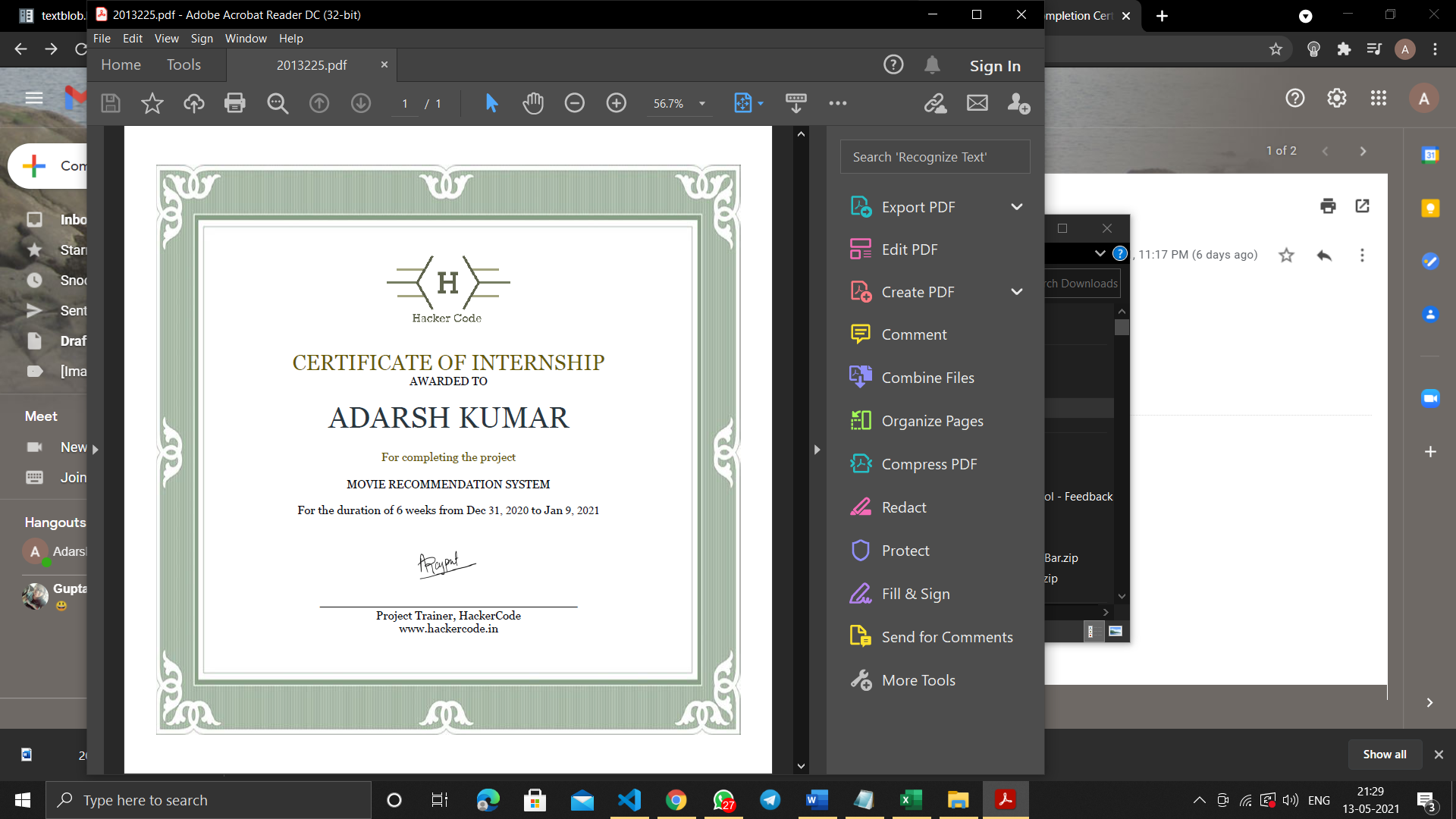




**Code snapshots**

****

**At last I would like to thanks Mr. Akshay Rajput sir for guidance in the following project during the winter bootcamp.**

****