**INSY 5336**

**Python Programming**

**Spring 2021**

**Final Project (100 points)**

**Due Date: December 9, 2020, 11:59 pm CST (no late submission)**

The following guidelines should be followed and will be used to grade your homework:

* All code to be implemented and submitted as a jupyter notebook (.ipynb) file. Submit a single ipynb file.
* This is an individual homework assignment, no group submissions will be accepted. If you discuss in groups, please write your code individually and submit.
* Sample runs shown in the question should be used as a guide for implementation. However extensive testing needs to be done on your code to deal with all test cases that might possibly be executed.
* The logic of how you are solving the problem should be documented in the cell preceding the code in markdown language. In the case that your code is incorrect, your logic counts as effort points.
* Every code segment in the jupyter notebook cells should be well documented with comments. Use # in the code to provide comments and they should explain the algorithm step and what the code segment is doing. Follow the example in the notebook files provided in the lectures.
* Error checking in your code is **very important** and differentiates a high quality programmer from a low quality one. Use try/except blocks, if statements and other python code constructs to deal with unexpected errors and deal with them gracefully. **The homework will be graded for robustness of your code. You will lose 50% of the points if your code contains error/does not run! You will lose 10% of the points if your code runs but produces wrong result. In the second situation, you will gain some points back if your logic is clear and correct.**

1. (100 points) Write a python program that fetches movie information for the top 500 most popular movies from Metacritics. On this websites, there is an option to show the top movies.

On Metacritics, it is called “Movies of All Time”

You will first write python script that collect the movie information for the top 500 movies from each website and store them in a comma separated file (called [your name]\_movies.csv).

In addition to the csv file, the data should also be stored in a SQLite database called MovieInfoDatabase in the directory that your Jupyter Notebook code will be executed from. The MovieInfoDatabase should have a table called MovieInfoTable.

Next, from the movie information you have collected, extract 2 pieces of information: The director, and the cast (actors/actresses). Build a dictionary of the movies that contain these information. Arrange them in any way you prefer but make sure we can access the information we need at any time.

Example:

Which movie do you want to check?

input: Saving Private Ryan

What information about this movie do you want to check? (Choose director or cast)

input: Cast

Output:

The cats of the movie Saving Private Ryan includes Matt Damon as Pvt. James Francis Ryan, Tom Hanks as Captain Miller, Adam Goldberg as Pvt. Stanley Mellish, Barry Pepper as Pvt. Daniel Jackson, Dennis Farina as Lt. Col. Anderson, Dylan Bruno as Toynbe, Edward Burns as Pvt. Richard Reiben, Giovanni Ribisi as T-5 Medic Irwin Wade, Jeremy Davies as Cpl. Timothy P. Upham, Joerg Stadler as Steamboat Willie, Max Martini as Cpl. Henderson, Paul Giamatti as Sgt. Hill, Ted Danson as Captain Hamill, Tom Sizemore as Sgt. Mike Horvath, Vin Diesel as Pvt. Adrian Caparzo

Then there are 3 tasks you need to complete:

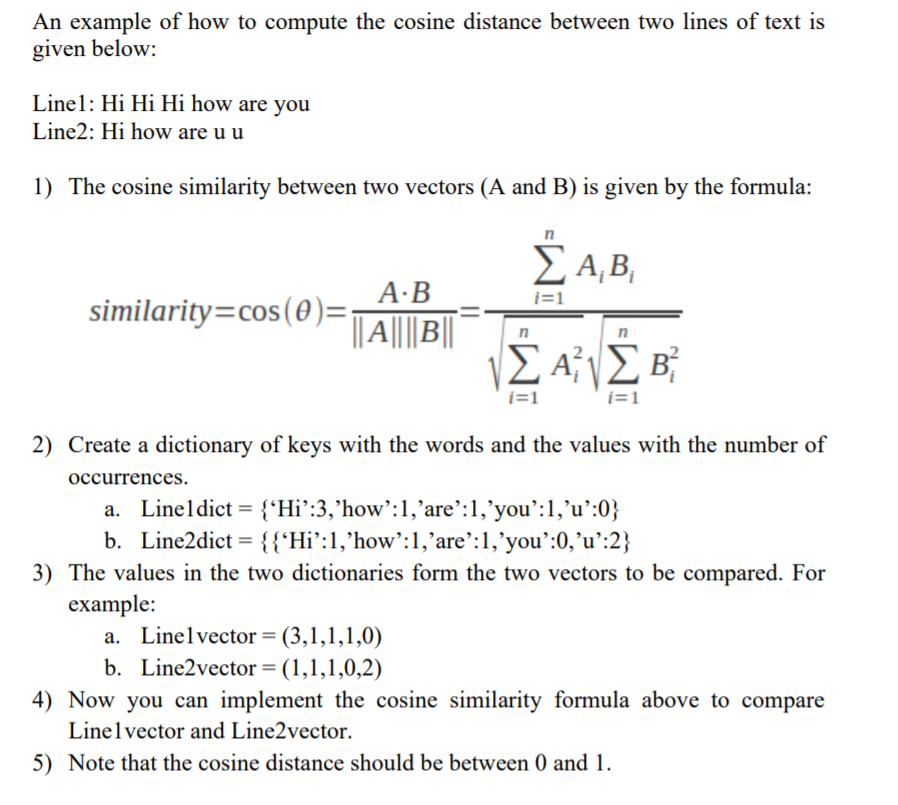
1. Analyze how many times has each actor/actress appeared in these top 500 movies, analyze how many times has each director appeared in these top 500 movies, what can that tell you about their career?

2. Create a dictionary of actors/actresses that the directors have worked together with in each movie, then calculate their cosine similarity, which directors work with similar groups of actors/actresses? Use director name as the dictionary name, actor/actress name as the key, and the times they have worked together in a movie as the value. For example: Michael Bay = {‘ Bruce Willis’: 50, ‘Ben Affleck’:20, ‘Liam Neeson’:10}, Steven Spielberg = {‘Liam Neeson’: 30, ‘Tom Hanks’:20, ‘Denzel Washington’:15}

Your program should show the similarity score between the directors. (An example is given below).

3. Pick 5 of your favorite actors/actresses from this list of top 500 movies. Then create a dictionary of all the actors/actresses that they have collaborated with in a movie. Following similar method as above in task 2, create the dictionaries, and compare these 5 actors, who is the most popular supporting actor/actress among them all?

Combine your finding with those in task 1 and 2, write a short report to observe how do directors and actors/actresses grow their career (Times new roman, 12 font size, no more than 1 page).



Example 2:

Michael Bay = Transformer: [Bruce Willis, Ben Affleck, Liam Neeson], Batman:[ Ben Affleck, Liam Neeson]

Steven Spielberg = Schindler's List:[ Liam Neeson, Tom Hanks], American Gangster:[ Denzel Washington]

Michael Bay = {‘Bruce Willis’: 1, ‘Ben Affleck’:2, ‘Liam Neeson’:2}

Steven Spielberg = {‘Liam Neeson’: 1, ‘Tome Hanks’:1, ‘Denzel Washington’:1}

Common vector = (Bruce Willis. Ben Affleck, Liam Neeson, Tom Hanks, Denzel Washington)

Michael Bay vector = (1,2,2,0,0)

Steven Spielberg vector = (0,0,1,1,1)

Then calculate the cosine similarity.

Your submission will include 4 files.

1) The ipynb file with your python code.

2) Your .csv file that stores the reviews.

3) Your .db file that stores the reviews.

4) Your short report in word document.