**INSY 5336**

**Python Programming**

**Final Project (100 points)**

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.**

(100 points) Write a python program that fetches movie information for the top 500 most popular movies from Metacritic. On this websites, there is an option to show the top movies called “Movies of all times”: <https://www.metacritic.com/browse/movies/score/metascore/all/filtered?sort=desc>

You will first write python script that collect the movie information for the top 500 movies from the website. Next, from the movie information you have collected, extract 2 pieces of information: The name of the director(s), and the genre information of the movie (drama, action, thriller, etc). 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. Store them in a comma separated file (called [your name]\_movies.csv). **(50 points)**

Your main program should ask users for three choices:

1. The 1st option is ‘movie’, when user chooses this one, then your program should ask for the name of a movie (your program should be able to handle all upper and lower cases), your program should find the movie and display the director information and genre information. **(10 points)**

2. The 2nd option is ‘director’, when user chooses this one, then your program should ask for the name of a director, then your program should find the person, and display: (a) all the movies that this person has directed, (b) a summary of the genres of the these movies **(15 points)**

3. The 3rd option is ‘comparison’, when user choose this one, then your program should ask for the names of 2 directors, then calculate their cosine similarity based on the genre of movies they have directed (An example is given below on how to calculate cosine similarity). **(15 points)**

Lastly, writing task:

Pick 3 of your favorite directors from this list of top 500 movies. Compare the career of these 5 directors (a vs b, b vs c, a vs c), do you see any similarity? Write a short report to summarize your finding (Times new roman, 12 font size, no more than 1 page).

**(10 points, attention: Do not write a description of your code, do not write a description of the result, I want to see your personal observation here)**

Your submission will include 3 files.

**1) The ipynb file with your python code.**

**2) Your .csv file that stores the reviews.**

**3) Your short report in word document.**

Example 1:

What do you want to check on Metacritic? (Please choose ‘movie’, ‘director’, or ‘comparison’)

input: Movie

What movie do you want to check?

input: savING private RYAN

Output:

The director of the movie is Steven Spielberg.

The genre of the movie is action, war, drama.

Example 2:

What do you want to check on IMDB? (Please choose ‘movie’, ‘director’, or ‘comparison’)

input: director

Who do you want to check?

input: Michael Bay

Output:

Michael Bay has directed Transformer 1, Transformer 2, Transformer 3, Transformer 4…… etc

His most directed genres are drama: 22, war: 6, action: 10, crime: 4, thriller: 9.

Example 3:

What do you want to check on IMDB? (Please choose ‘movie’, ‘director’, or ‘comparison’)

input: COMparison

Who do you want to compare?

input 1: Michael Bay

input 2: Steven Spielberg

Output:

Michael Bay has directed Transformer 1, Transformer 2, Transformer 3, Transformer 4…… etc

His most directed genres are drama: 22, war: 6, action: 10, crime: 4, thriller: 9.

Steven Spielberg has directed Saving Private Ryan, Shindler’s List….etc

His most often played genres are drama 17, crime 20, action 31.

Based on that, they have a cosine similarity score of 0.54.

Hint 1: Organize your data in dictionaries with multiple layers.

{‘Saving Private Ryan’:{( ‘Steven Spielberg’….): [‘action’, ‘war’, ‘drama’….]}}

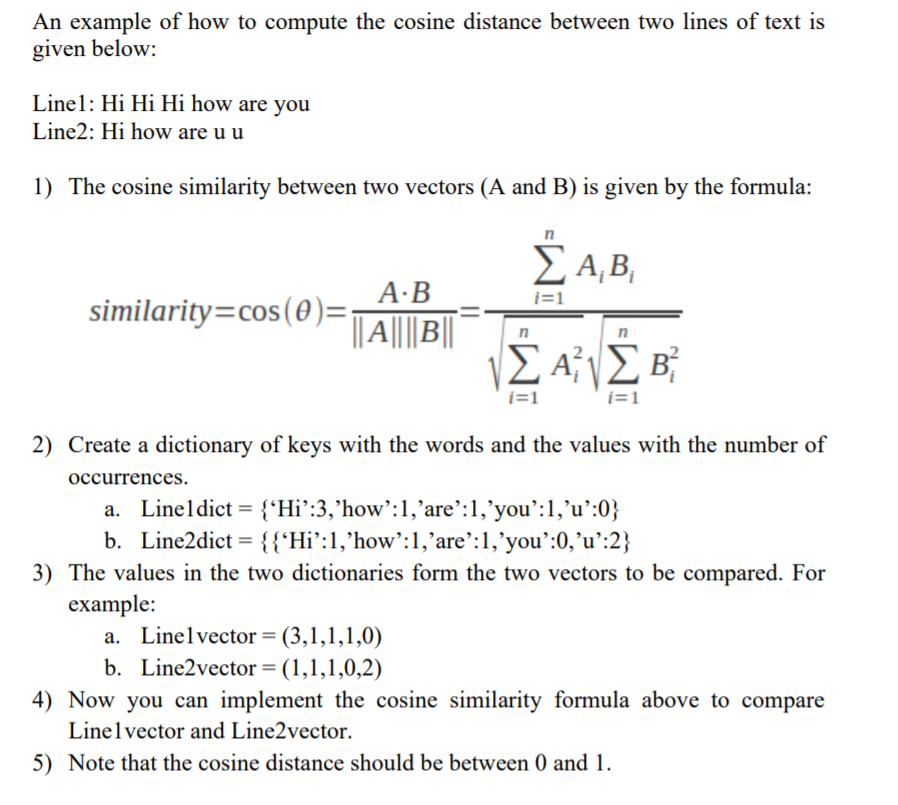
Movie name is the key on 1st level

The director and genre form a sub-dictionary that is the value on 1st level.

The director is the key on 2nd level (in the sub dictionary) (key cannot be mutable type, but some movies have multiple directors, so what do we do here?)

The genre is the value on the 2nd level (in the sub dictionary)

Hint 2: Some movies may be missing some information, your program will run into errors if you try to extract the same information for every movie. Think about how you deal with this.



Example 2:

Michael Bay = Transformer: [action, sci-fi, drama], Batman:[ action, drama]

Steven Spielberg = Schindler's List:[ drama, war], American Gangster:[ drama, crime, action]

Michael Bay = {‘action’: 2, ‘drama’: 2, ‘sci-fi’: 1}

Steven Spielberg = {‘drama’: 2, ‘war’:1, ‘crime’:1, ‘action’ : 1}

Common vector = (action, drama, sci-fi, war, crime)

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

Steven Spielberg = (1,2,0,1,1)

Then calculate the cosine similarity.