

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**🙞🕮🙜**



**BÁO CÁO TIẾN ĐỘ -TUẦN 05  
MÔN: ĐỒ ÁN 3**

**ĐỀ TÀI: TÌM HIỂU VỀ THUẬT TOÁN RECOMMENDATION**

**GVHD : Thầy Huỳnh Xuân Phụng**

**SVTH :**

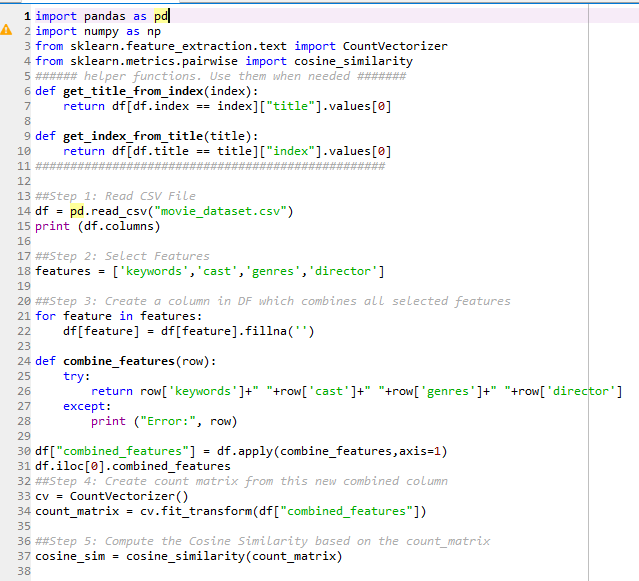
**Nguyễn Thành Như 17110202**

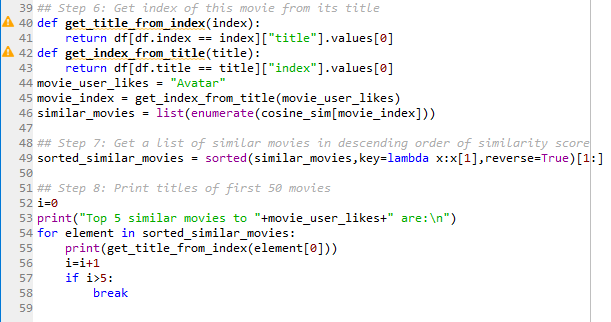
**Võ Ngọc Thuận 17110234**

**TP. Hồ Chí Minh, tháng 11 năm 2020**

## Demo Recommendation System with Content-based Filtering

### Code Demo





### Giải thích code:

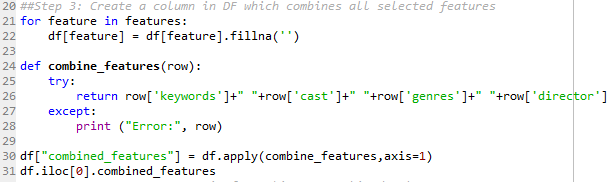
* **import pandas as pd** : chúng ta tiến hành import thư viện Pandas, nó là một thư viện Python cung cấp các cấu trúc dữ liệu nhanh, mạnh mẽ, linh hoạt và mang hàm ý. Chúng em dùng nó để đọc dữ liệu từ file csv.
* **from scipy import sparse** : scipy là một thư viện mã nguồn mở các thuật toán và các công cụ toán học cho Python, được xây dựng trên các đối tượng mảng NumPy tạo thành ngăn xếp NumPy bao gồm các công cụ như Pandas, SymPy và Matplotlib. SciPy cung cấp khá nhiều module tính toán từ đại số tuyến tính, tích phân, vi phân, nội suy đến xử lý ảnh, fourier transform... và cụ thể ở đây ta import gói ma trận sparse và các đoạn chương trình liên quan
* **from sklearn.metrics.pairwise import cosine\_similarity** : cosine\_similarity được lấy từ sklearn.metrics.pairwise để tính toán độ tương tự giữa 2 vectors
* **Step 1: df=pd.read\_csv("movie\_dataset.csv"):** gán biến df và đọc biến df với dữ liệu là file movie\_dataset.csv bằng việc đọc vào một file .csv bằng cách sử dụng hàm read\_csv và được trả về 1 dataframe.



* **Step 2: features = ['keywords','cast','genres','director']:** chọn 4 features ‘keywords’, ‘cast’, ‘genres’, ‘director’ làm bộ tính năng cho nội dung của movie\_dataset



* **Step 3:** Nhiệm vụ tiếp theo của chúng ta là tạo một hàm để kết hợp các giá trị của các cột này thành một chuỗi duy nhất. chúng ta cần gọi hàm này trên mỗi hàng của khung dữ liệu. Tuy nhiên, trước khi làm điều đó, chúng ta cần clean và xử lý trước dữ liệu để sử dụng. Điền tất cả các giá trị NaN bằng chuỗi trống trong khung dữ liệu.



* **Step 4: cv = CountVectorizer():** tạo object CountVectorizer().

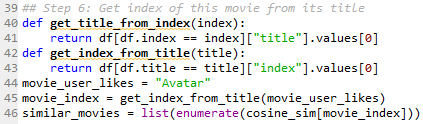
**count\_matrix = cv.fit\_transform(df["combined\_features"]):** combine chuỗi string(movie contents) với object CountVectorizer().



* **Step 5:** Chúng ta cần lấy ma trận tương tự cosine từ ma trận đếm.



* **Step 6:** Xác định hai hàm trợ giúp để lấy title phim từ chỉ mục phim và ngược lại.



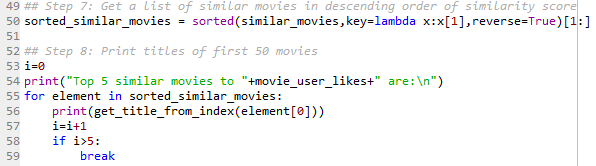
Bước tiếp theo là lấy tên phim mà người dùng đang thích. Sau đó, tìm chỉ số của bộ phim đó. Sau đó, truy cập vào hàng tương ứng với bộ phim này trong ma trận tương tự. Như vậy, chúng ta sẽ nhận được điểm tương đồng của tất cả các phim khác từ phim hiện tại. Sau đó, thống kê thông qua tất cả các điểm tương tự của bộ phim đó để tạo ra một bộ chỉ số phim và điểm tương tự. Điều này sẽ chuyển đổi một hàng điểm số tương tự như thế này- [1 0,5 0,2 0,9] thành hàng này- [(0, 1) (1, 0,5) (2, 0,2) (3, 0,9)].



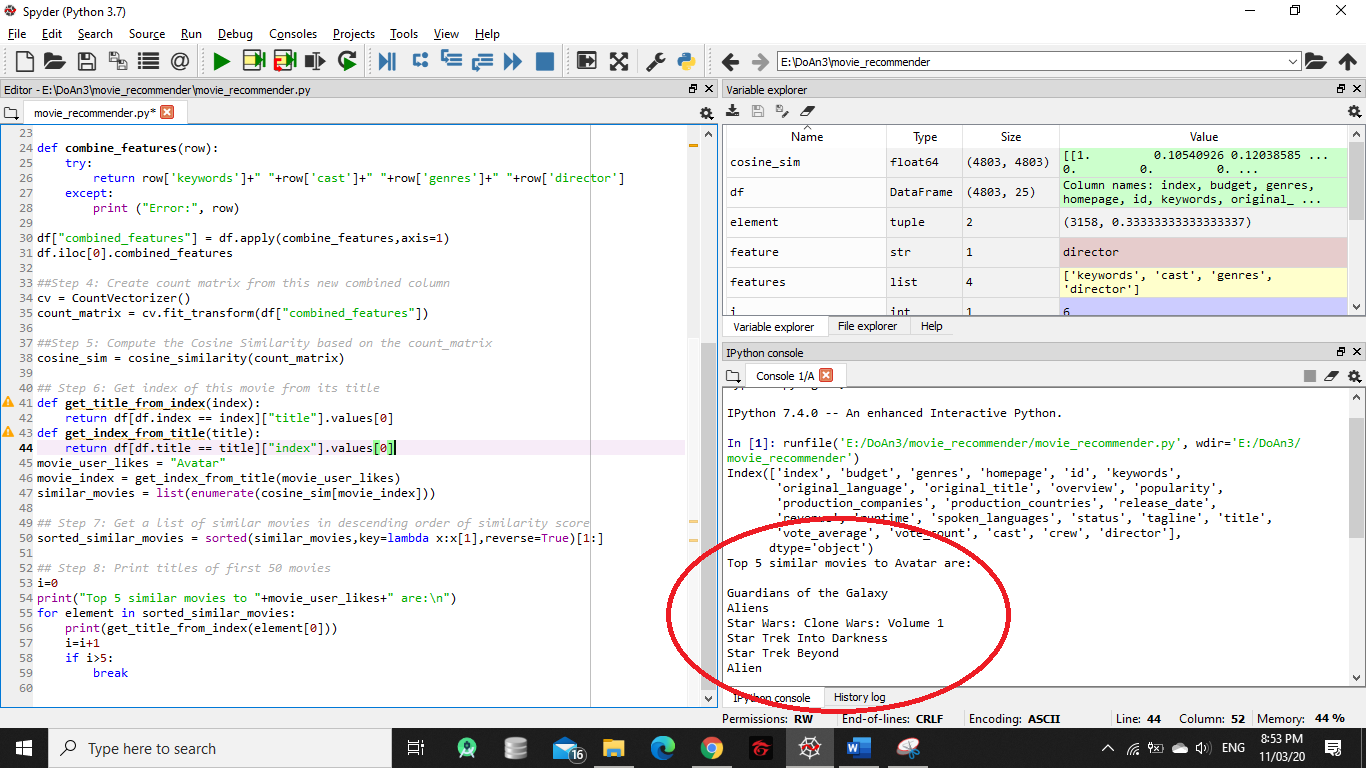
* **Step 7:** Sắp xếp danh sách similar\_movies theo điểm tương tự theo thứ tự giảm dần. Vì phim tương tự nhất với một phim nhất định sẽ là chính nó, loại bỏ phần tử đầu tiên sau khi sắp xếp các phim.



* **Step 8:** Chạy vòng lặp để in 5 entries đầu tiên từ danh sách sorted\_similar\_movies.



Kết quả: Top 5 bộ phim liên quan, gần với phim Avatar nhất là:



Top 5 bộ phim liên quan, gần với bộ phim Avatar nhất là:

Guardians of the Galaxy

Aliens

Star Wars: Clone Wars: Volume 1

Star Trek Into Darkness

Star Trek Beyond

Alien

# Công việc tuần sau:

* Demo dựa theo Clusting User.