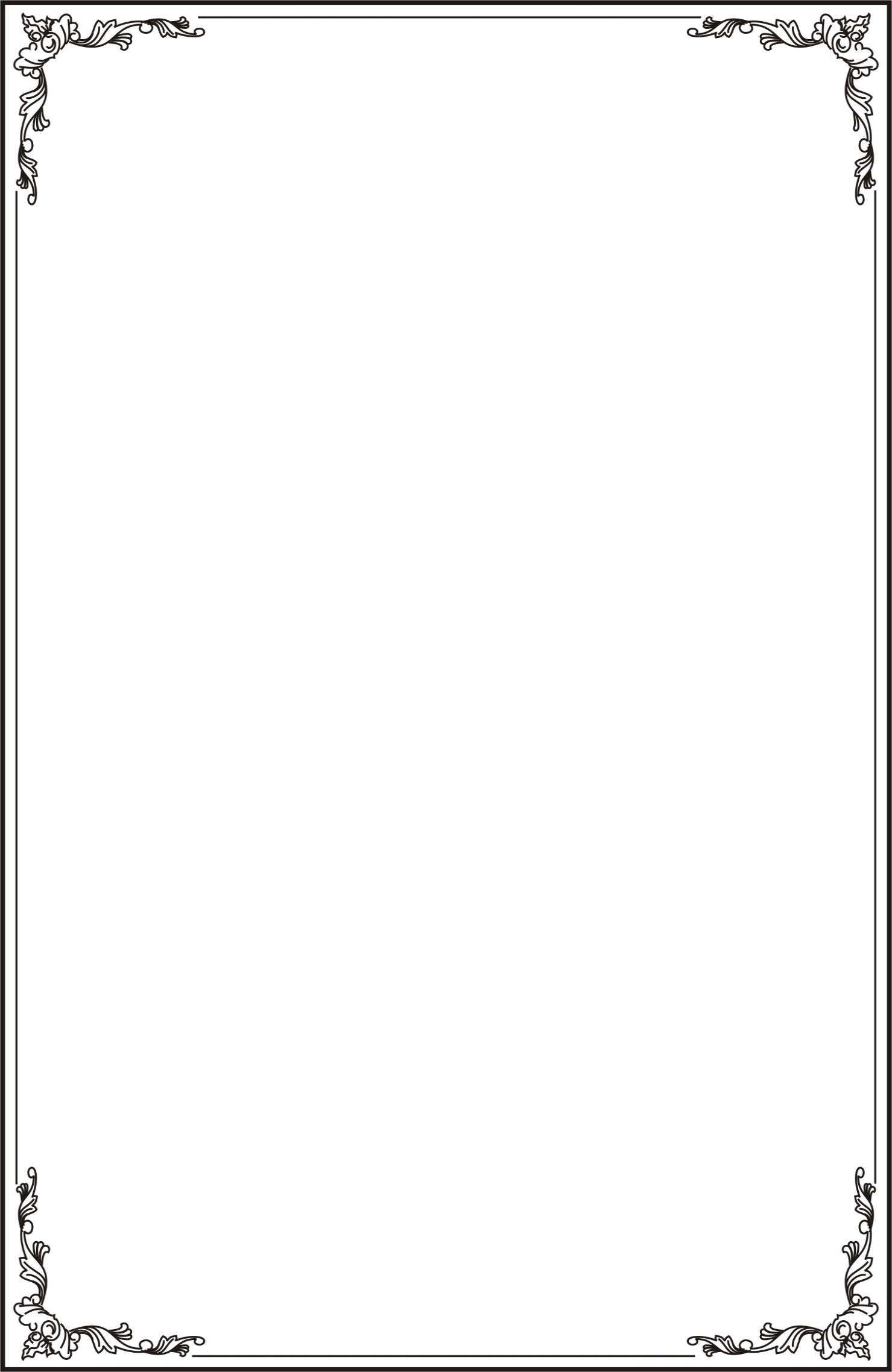
**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

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

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo

Description automatically generated

**BÁO CÁO**

**BIG DATA APPLICATION**

**ĐỀ TÀI: THE MOVIE RECOMMENDER SYSTEM**

**Nhóm: 11**

Sinh viên thực hiện:

Lý Quốc Dũng 19133015

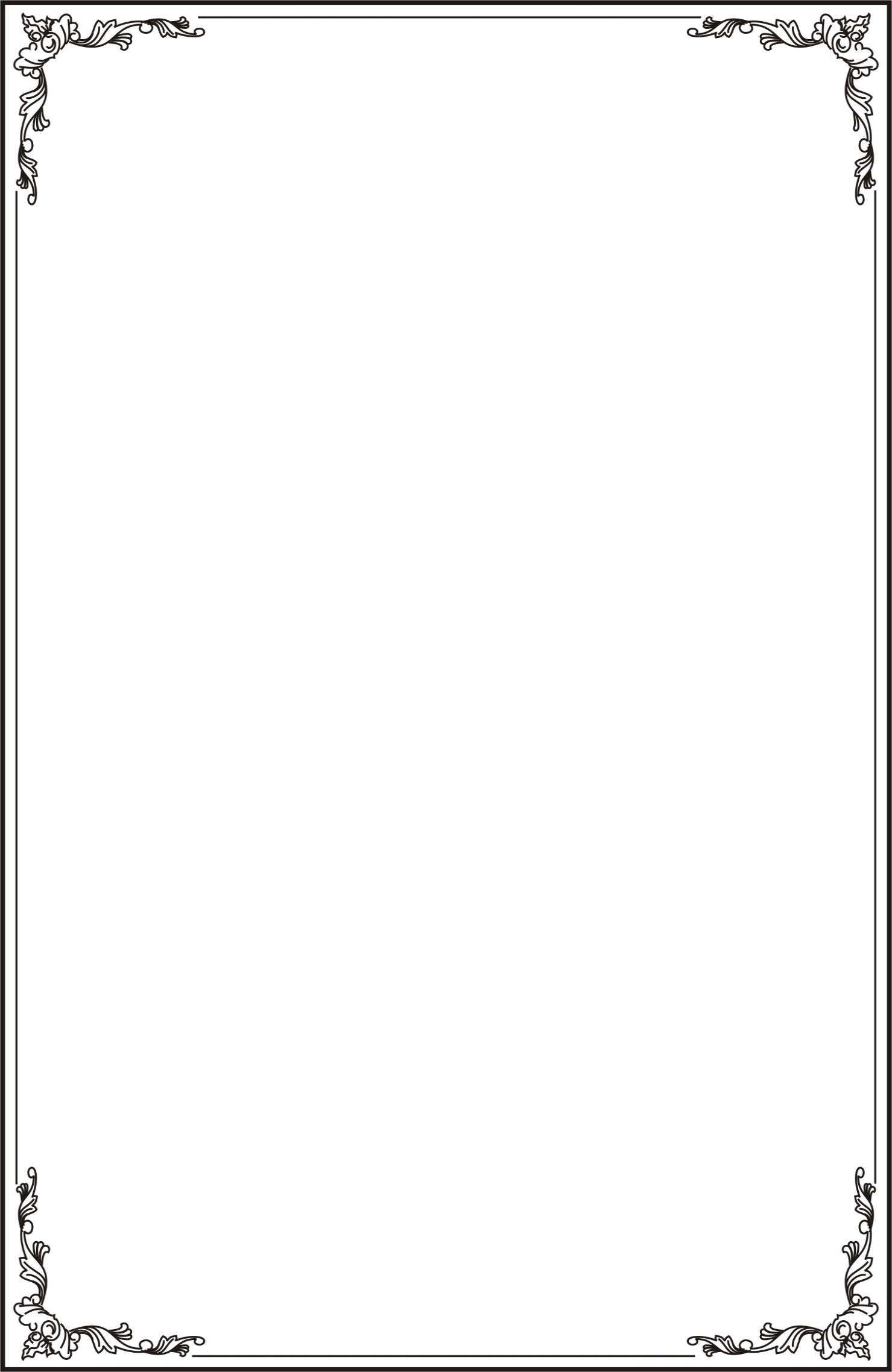
Đoàn Trần Đăng Khoa 19133028

Lê Thị Kim Ngân 19133038

Bùi Thị Ngân Tuyền 19133066

GVHD: Quách Đình Hoàng

*TP.Hồ Chí Minh, 16 tháng 12, năm 2022*

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

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

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo

Description automatically generated

**BÁO CÁO**

**BIG DATA APPLICATION**

**ĐỀ TÀI: THE MOVIE RECOMMENDER SYSTEM**

**Nhóm: 11**

Sinh viên thực hiện:

Lý Quốc Dũng 19133015

Đoàn Trần Đăng Khoa 19133028

Lê Thị Kim Ngân 19133038

Bùi Thị Ngân Tuyền 19133066

GVHD: Quách Đình Hoàng

*TP.Hồ Chí Minh, 16 tháng 12, năm 2022*

# **PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN**

Họ và tên sinh viên 1: Lý Quốc Dũng MSSV: 19133015

Họ và tên sinh viên 2: Đoàn Trần Đăng Khoa MSSV: 19133028

Họ và tên sinh viên 3: Lê Thị Kim Ngân MSSV: 19133038

Họ và tên sinh viên 4: Bùi Thị Ngân Tuyền MSSV: 19133066

Ngành: Kỹ thuật dữ liệu.

Giảng viên hướng dẫn: Quách Đình Hoàng

Tên đề tài: The Movie Recommender System

NHẬN XÉT:

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Điểm: ………………… (Bằng chữ……………………………………………………)

Tp. Hồ Chí Minh, ngày 16 tháng 12 năm 2022

GIẢNG VIÊN HƯỚNG DẪN

# **MỤC LỤC**

[**PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN** 3](#_Toc122466709)

[**MỤC LỤC** 4](#_Toc122466710)

[**1.Tóm tắt:** 5](#_Toc122466711)

[**2. Giới thiệu:** 5](#_Toc122466712)

[**3. Dữ liệu:** 5](#_Toc122466713)

[**4. Phương pháp:** 5](#_Toc122466714)

[**5. Thực nghiệm, kết quả, và thảo luận:** 6](#_Toc122466715)

[**6. Kết luận:** 12](#_Toc122466716)

[**7. Phụ lục:** 12](#_Toc122466717)

[**8. Đóng góp:** 12](#_Toc122466718)

[**9. Tham khảo:** 12](#_Toc122466719)

## 

## 

## 

## 

## **1.Tóm tắt:**

Trong các nền tảng hiện nay, để mang đến sự tối ưu kinh doanh và sự tiện lợi cho người dùng nên hệ thống đề xuất rất cần thiết với những lý do sau:

* Thu hút thêm nhiều người dùng trải nghiệm hệ thống
* Kéo dài thời gian người dùng ở tại hệ thống
* thúc đẩy doanh số cho việc kinh doanh
* Mang đến sự thoải mái và tiện lợi cho người dùng.
* …….
* Các thuật toán sử dụng:
* Content-Based
* Demographic Filtering
* Apriori Algorithiom
* Collaborative Filtering

⇒ hệ thống đề xuất Collaborative Filtering mang lại kết quả đề xuất tốt nhất trong các hệ thống nhóm làm.

## **2. Giới thiệu:**

Hệ thống đề xuất dự đoán khả năng người dùng sẽ ủng hộ một mặt hàng. Dựa trên tìm kiếm của người dùng và tương tác trước đây của người dùng với nguồn dữ liệu mà hệ thống học hỏi (ngoài dữ liệu từ những người dùng khác hoặc xu hướng lịch sử), hệ thống có khả năng đề xuất một mặt hàng cho người dùng. Ví dụ Netflix giới thiệu những bộ phim mà họ nghĩ rằng chúng tôi sẽ thích.

## **3. Dữ liệu:**

- Data được lấy từ nguồn Kaggle : <https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata?select=tmdb_5000_credits.csv>

- Data gồm có 4 csv : ratings.csv, movies.csv, tmdb\_5000\_credits.csv, tmdb\_5000\_movies.csv

\* ratings.csv: 100837 dòng ,4 cột :userId,movieId,rating,timestamp

\* movies.csv: 9743 dòng, 3 cột :movieId,title,genres

\* tmdb\_5000\_credits.csv : 4814 dòng, 4 cột: movie\_id,title,cast,crew

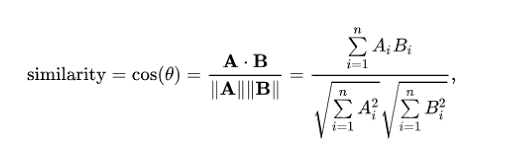
\* tmdb\_5000\_movies.csv: 4804 dòng, 20 cột (budget, genres, homepage, id, keywords, original\_language, original\_title, overview, popularity, production\_companies, production\_countries, release\_date, revenue, runtime, spoken\_languages, status, tagline, title, vote\_average, vote\_count)

## **4. Phương pháp:**

* **Content-based Filtering**: Gợi ý các movie dựa vào input từ khóa movie do người dùng nhập vào và dựa vào nội dung/thuộc tính (attributes) đó để đề xuất những movie tương tự dựa trên cosine similarity.
* **Demographic Filtering:** đề xuất các bộ phim nổi tiếng có độ phổ biến cao và được đánh giá cao hơn sẽ có xác xuất khán giả yêu thích cao hơn dựa trên thuật toán Weighted Rating (WR) và các giá trị độ phổ biến, số phiếu bầu chọn và xếp hạng trung bình bộ phim nhận được.
* **Collaborative Filtering (CF):** Collaborative filtering là thuật toán xây dựng mô hình từ những user đã đánh giá phim trước đó. Sử dụng mô hình này để tìm kiếm những phim có lượt đánh giá liên quan và đề xuất những bộ phim như đã chọn.
* **Apriori Algorithm Based Recommendation:** Thuật toán Apriori được sử dụng để hiểu rõ hơn về các mối quan hệ có cấu trúc giữa các mục khác nhau có liên quan. Ứng dụng thực tế của thuật toán là giới thiệu, đề xuất cho người dùng những movie dựa trên những movie đã xem.

## **5. Thực nghiệm, kết quả, và thảo luận:**

**5.1. Content-based**:

* **Ý tưởng thực hiện:** thuật toán sẽ dựa trên input từ khóa tìm kiếm do user Input vào, hệ thống sử dụng từ khóa input đó để tìm ra chính xác bộ phim có tên trùng với từ khóa đó, sau đó sử dụng cosin\_similarity dựa trên meta data để tính độ tương đồng của phim đó với các bộ phim khác và tiến hành đề xuất ra top 10 phim có độ tương đồng cao nhất.
* **cách tính Consin\_similarity:**
* ****
* **Tiến Hành 2 Thử Nghiệm**:
* **Thử nghiệm 1 :** 
  + Meta data: Overview
  + Kỹ thuật sử dụng: TF-IDF ( term frequency – inverse document frequency)
  + dựa trên độ consin\_similarity để tìm ra phim tương đồng
* **Thử nghiệm 2:** 
  + Meta data: Overview, Cast, Crew, Keyword, genres
  + Kỹ thuật sử dụng: CountVectorizer
  + dựa trên độ consin\_similarity để tìm ra phim tương đồng.

⇒ Kết quả thử nghiệm 1 và 2 đều cho ra các bộ phim chính xác với từ khóa input vào, tuy nhiên (theo ý kiến cá nhân) ở thử nghiệm 2 cho thấy đề xuất thực hiện tốt hơn trong việc thu thập nhiều thông tin hơn do có nhiều siêu dữ liệu hơn và đã đưa ra các đề xuất tốt hơn. Vì Nhiều khả năng những người dùng sẽ thích những bộ phim của cùng một nhà sản xuất/ diễn viên.

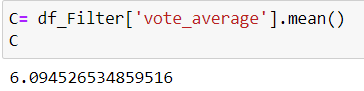
**5.2. Demographic Filtering:**

* MetaData: id, title, popularity, vote\_average, vote\_count
* Dựa trên hai giá trị vote\_count và vote\_average để trình ra điểm số cho bộ phim đó theo thuật toán Weighted Rating (WR)
* Sắp xếp điểm số và đề xuất bộ phim được đánh giá tốt nhất cho người xem.
* Dựa trên popularity để đề xuất các bộ phim có độ phổ biến cao cho người xem.

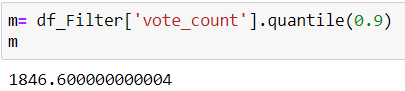


* Với: v : giá trị của cột vote\_count
* m : giá trị tối thiểu để được thống kê
* R : giá trị của cột vote\_average
* C : giá trị trung bình

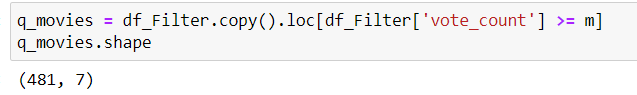
Tiến hành tính C và m



=> Tính ra C là 6 vậy xếp hạng trung bình cho tất cả các phim là xấp xỉ 6 trên thang điểm 10.

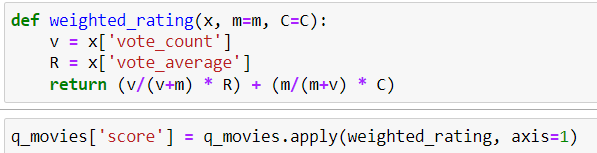


=> Chọn quantile(0.9) để một bộ phim có mặt trong bảng xếp hạng, nó phải có nhiều phiếu bầu hơn ít nhất 90% số phim trong danh sách từ đó tính ra m.

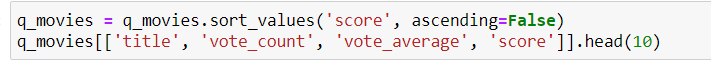


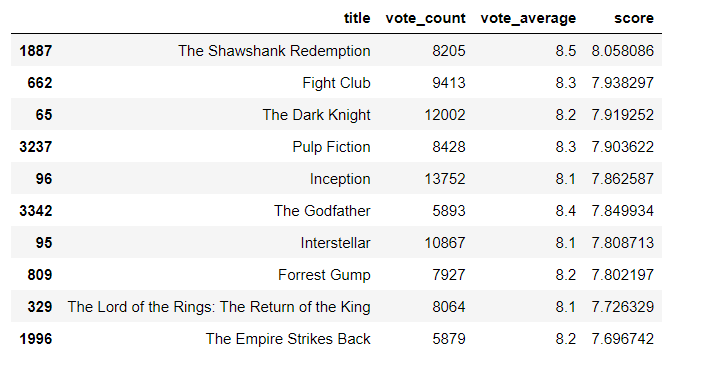
=>Tiến hành lọc thì nhận được có 481 bộ phim đạt điều kiện

Dùng weight\_rating để tìm giá trị cho cột score



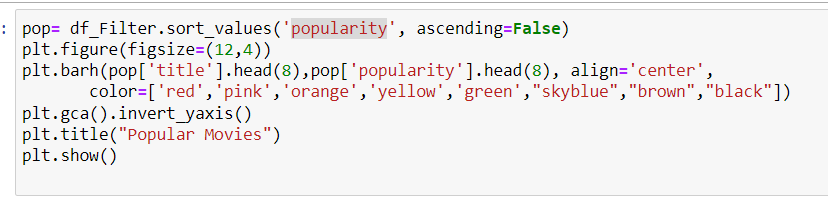
Tiến hành sắp xếp và in ra 10 bộ phim có giá trị số điểm cao nhất



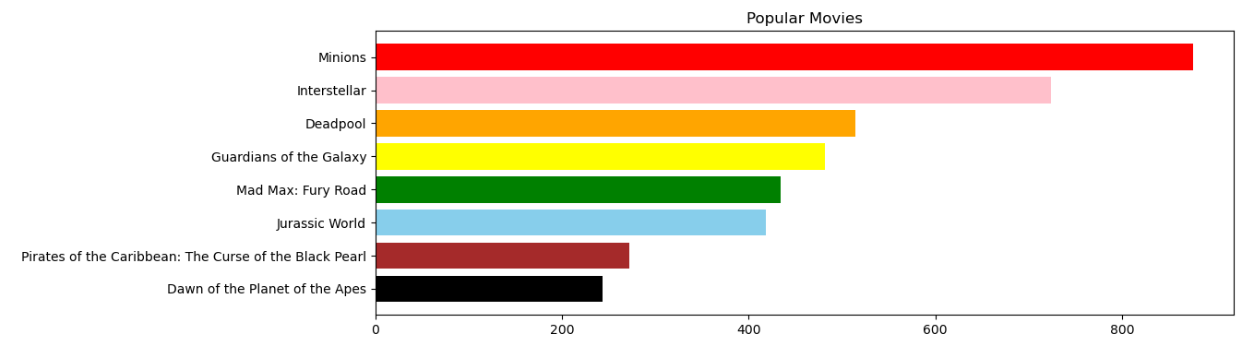


Đây là 10 bộ phim được đề xuất với số điểm cao nhất dựa trên lượt đánh giá và số điểm đánh giá của từng bộ phim đạt được.

Và để làm đa dạng hơn đề xuất các bộ phim cho người xem, em dùng giá trị popularity để đề xuất các bộ phim được yêu thích , phổ biến.



Kết quả đạt được:



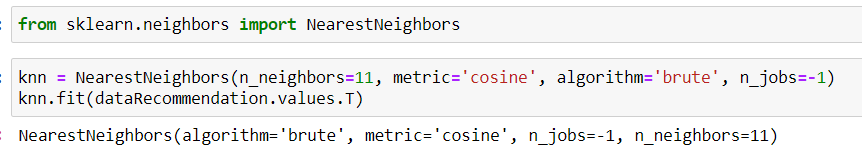
=> Đề xuất các bộ phim có độ phổ biến cao

Kết quả đạt được từ Demographic Filtering là danh sách đề xuất các bộ phim có điểm số đánh giá cao nhất, và các bộ phim có độ phổ biến cao.

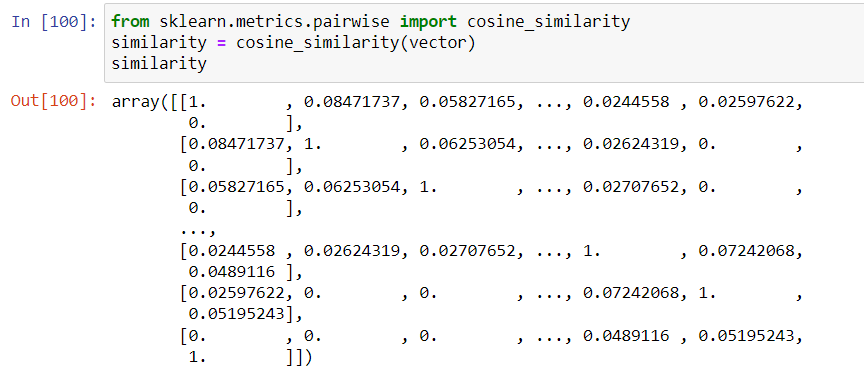
**5.3. Collaborative Filtering (CF):**

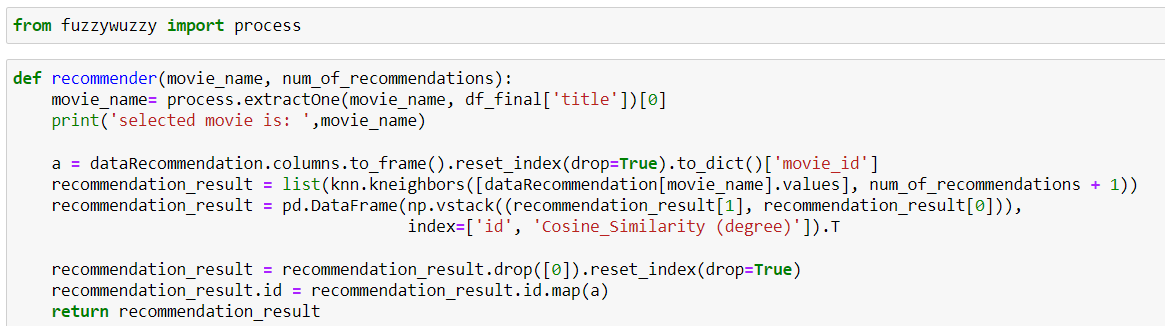
Trong phương pháp này, các mô hình được phát triển bằng cách sử dụng các thuật toán Machine Learning để khai thác dữ liệu khác nhau. Cụ thể ở bài này em dùng KNN (K-Nearest Neighbors) và recommender system để dự đoán xếp hạng của người dùng về các mục chưa được xác định.

Trong thuật toán KNN, em chọn algorithm là brute, metric là cosine và số note là 11 để lấy được danh mục phim.

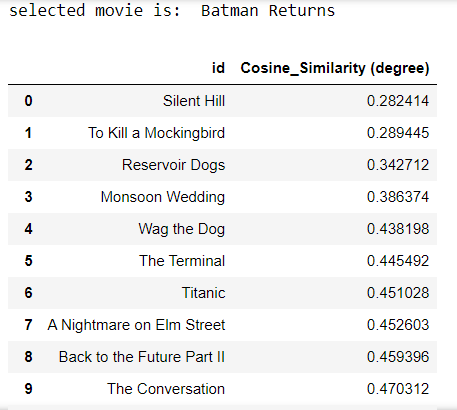


Trong thuật toán recommender system, em sử dụng kết quả similarity như đã tính ở trên kết hợp với thuật toán KNN để trả về danh sách phim đề xuất cho người xem.

****

****

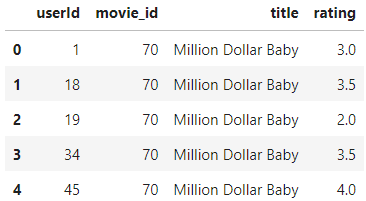
Kết quả đạt được:



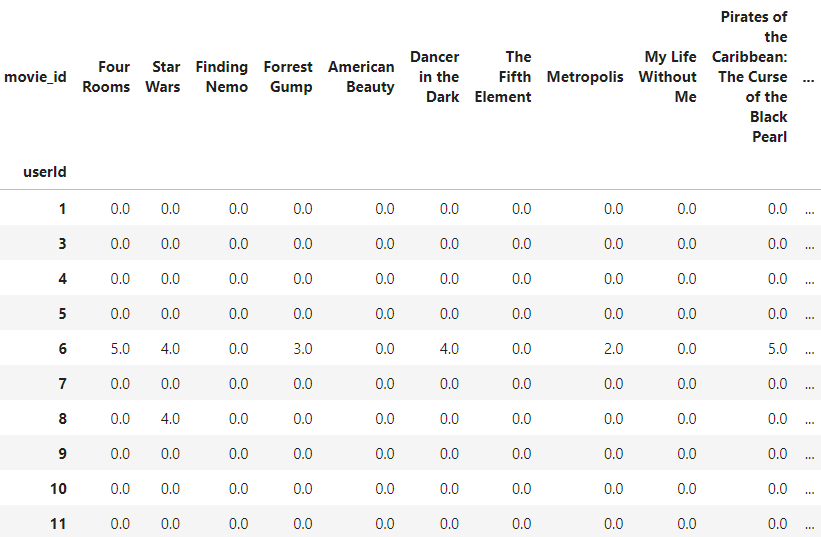
Kết quả đạt được từ thuật toán Collabretive filtering là danh sách phim được đề xuất dựa trên lịch sử đánh giá phim của người xem.

**5.4. Apriori Algorithm Based Recommendation:**

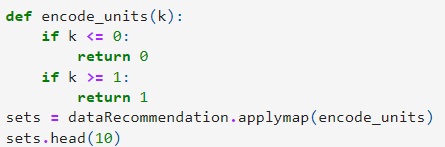
Dataframe thể hiện điểm đánh giá của user cho từng bộ phim

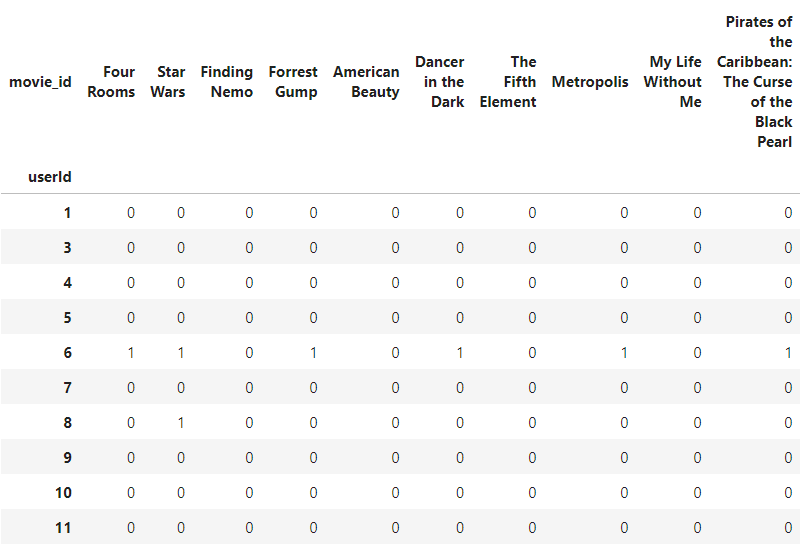
****

Tạo bảng từ data frame và thay thế movie\_id thành title

****

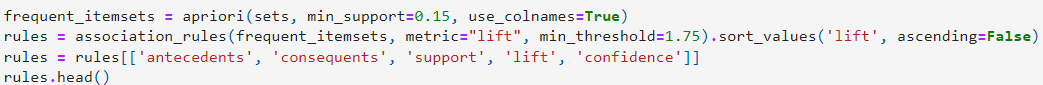
Dùng applymap để nhận và trả về cái đại lượng vô hướng cho mọi phần tử trong Dataframe để xác định user có quan tâm bộ phim đó hay không.

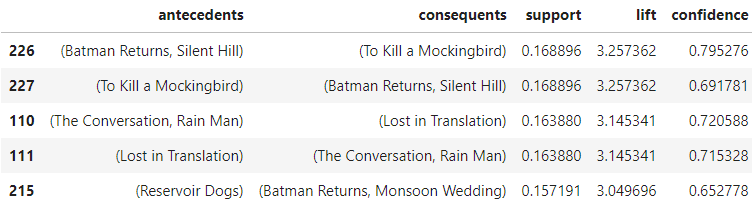
****

****

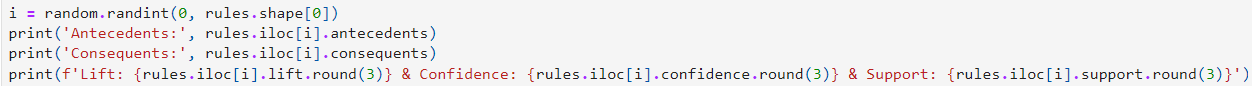
Xây dựng model với min\_support là 0.15, min\_threshold là 1.75 để quyết định quy tắc user có quan tâm movie như thế nào.

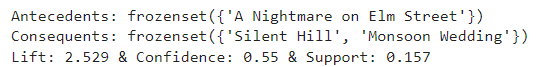
Kết hợp thu thập các quy tắc suy luận trong 1 khung dữ liệu với các chỉ số như support, lift, confidence.

****

****

Với cột antecedents và consequents lưu trữ các title kết hợp với support, lift, confidence sẽ tạo ra những gợi ý hợp lý cho người dùng.

****

****

## **6. Kết luận:**

Nhìn chung qua 3 thuật toán Content-based, Demographic Filtering và Collaborative Filtering (CF) ta nhận thấy Collaborative Filtering (CF) là phương pháp chính xác nhất do thuật toán này sử dụng siêu dữ liệu phim trước đây của người dùng. Dựa vào lịch sử đánh giá để đề xuất những phim khác.

## **7. Phụ lục:**

## **8. Đóng góp:**

* Lý Quốc Dũng : EDA, Content-Based, word
* Đoàn Trần Đăng Khoa : Clean Data, Demographic Filtering, word
* Lê Thị Kim Ngân :EDA, Collaborative Filtering, word
* Bùi Thị Ngân Tuyền : EDA , Apriori Algorithm, word

## **9. Tham khảo:**

**Library** : pandas, numpy, matplotlib, seaborn, mlxtend, NearestNeighbors, random, fuzzywuzzy, sklearn, matplotlib.

**Tài liệu tham khảo:**

[Xây dựng Collaborative Filtering - Recommender System cơ bản (viblo.asia)](https://viblo.asia/p/xay-dung-collaborative-filtering-rs-recommender-system-co-ban-phan-3-Az45bMqolxY)

[Build a Recommendation Engine With Collaborative Filtering – Real Python](https://realpython.com/build-recommendation-engine-collaborative-filtering/)

[Recommendation Systems Tutorial with Python using Collaborative Filtering | by Towards AI Editorial Team | Towards AI (medium0.com)](https://medium0.com/towards-artificial-intelligence/recommendation-system-in-depth-tutorial-with-python-for-netflix-using-collaborative-filtering-533ff8a0e444?source=friends_link&sk=802d36f89bb9a2827eb370eb3b3675a1)

[Implementing Apriori algorithm in Python](https://www.geeksforgeeks.org/implementing-apriori-algorithm-in-python/)