**TRƯỜNG ĐẠI HỌC PHENIKAA**

**KHOA CÔNG NGHỆ THÔNG TIN**

A logo of a university

Description automatically generated with low confidence

BÁO CÁO

**ĐỀ TÀI: Book – Recommendation system in python using ALS algorithm and Apache Spark**

**Sinh viên: Nguyễn Hoàng Anh MSV: 20010841**

**Đoàn Trung Kiên MSV: 20010905**

**Nguyễn Lý Khánh Linh MSV: 20010965**

**Giảng viên: Phạm Tiến Lâm**

**Học phần: Tích hợp và phân tích dữ liệu**

Năm học 2022 – 2023

**Mở đầu**

Trong bài báo cáo này, nhóm em trình bày cách sử dụng thuật toán ALS và framework Apache Spark để xây dựng một hệ thống gợi ý sách hiệu quả trong Python. Nội dung bài báo cáo trình bày những vấn đề sau:

* Đặt vấn đề
* Recommendation system (hệ thống khuyến nghị)
* Apache Spark
* Thuật toán ALS (Alternating Least Squares)
* Xây dựng hệ thống gợi ý sách sử dụng ALS và Apache Spark
* Thực hiện và đánh giá

**Nội dung**

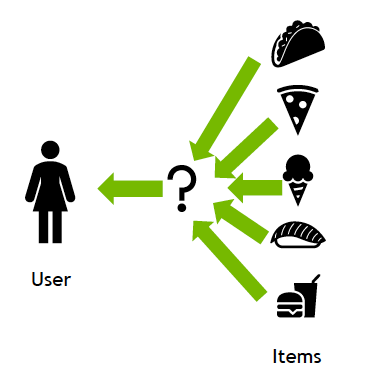
1. **Đặt vấn đề**

Ngày nay với sự phát triển mạnh mẽ của công nghệ thông tin và internet, việc tìm kiếm và khám phá thông tin, sở thích, học tập hoặc giải trí đã trở nên dễ dàng hơn bao giờ hết.

Các bạn có lẽ đã gặp những hiện tượng này nhiều lần:

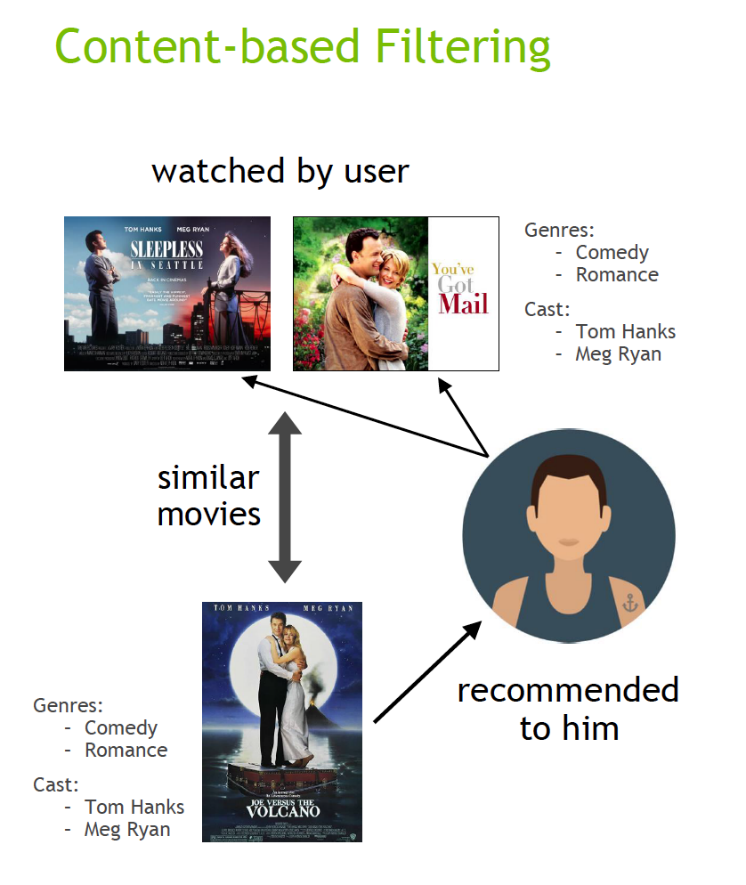
1. Youtube tự động chuyển các clip liên quan đến clip bạn đang xem.Youtube cũng tự gợi ý những clip mà có thể bạn sẽ thích.
2. Khi bạn mua một món hàng trên Amazon, hệ thống sẽ tự động gợi ý “Frequently bought together”, hoặc nó biết bạn có thể thích món hàng nào dựa trên lịch sử mua hàng của bạn.
3. Facebook hiển thị quảng cáo những sản phẩm có liên quan đến từ khoá bạn vừa tìm kiếm.
4. Facebook gợi ý kết bạn.
5. Netflix tự động gợi ý phim cho người dùng.

Và rất nhiều ví dụ khác mà hệ thống có khả năng tự động gợi ý cho người dùng những sản phẩm họ có thể thích. Bằng cách quảng cáo hướng đúng đối tượng như thế này, hiệu quả của việc marketing cũng sẽ tăng lên. Những thuật toán đằng sau những ứng dụng này là những thuật toán Machine Learning có tên gọi chung là *Recommender Systems* hoặc *Recommendation Systems*, tức *Hệ thống gợi ý*.

Recommendation Systems là một mảng khá rộng của Machine Learning và có tuổi đời ít hơn so với Classification vì internet mới chỉ thực sự bùng nổ khoảng 10-15 năm đổ lại đây. Có hai thực thể chính trong Recommendation Systems là users và items. **Users** là người dùng. **Items** là sản phẩm, ví dụ như các bộ phim, bài hát, cuốn sách, clip, hoặc cũng có thể là các users khác trong bài toán gợi ý kết bạn. Mục đích chính của các Recommender Systems là dự đoán mức độ quan tâm của một **user** tới một **item** nào đó, qua đó có chiến lược recommend phù hợp.

1. **Tổng quan** 
   1. **Recommendation Systems**

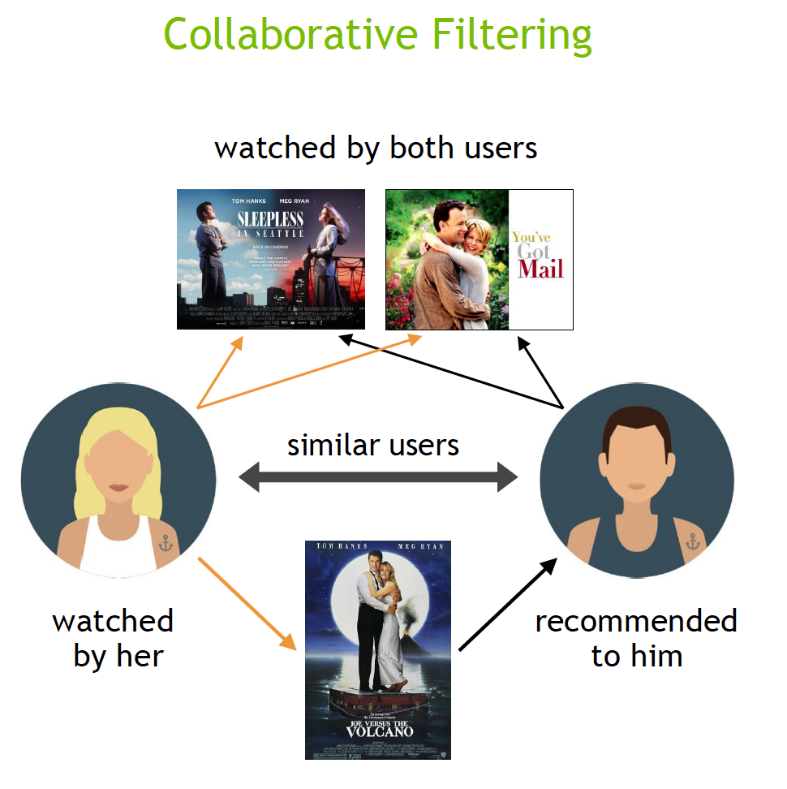
Các Recommendation Systems thường được chia thành hai nhóm lớn:

* 1. **Content-based systems:** đánh giá đặc tính của *items* được *recommended*.

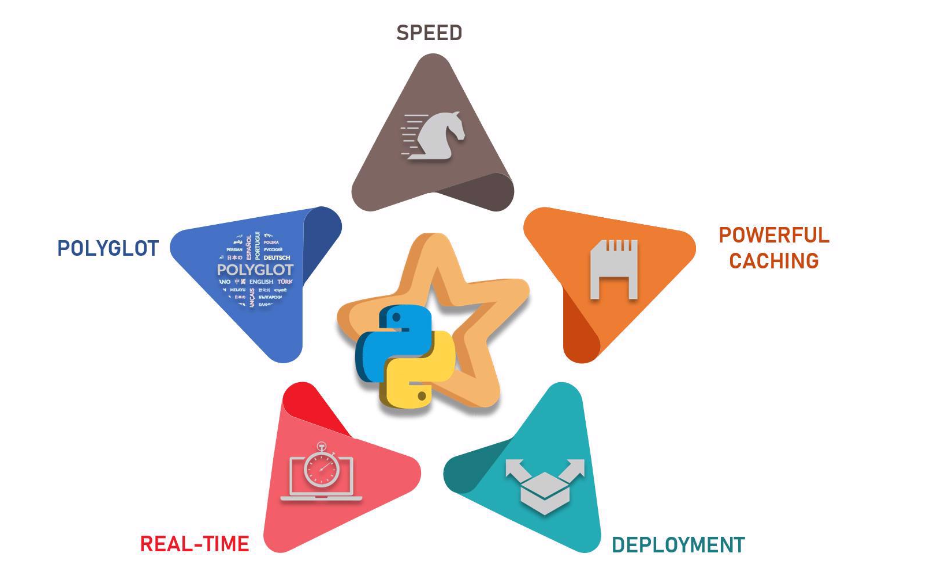
Ví dụ: một *user* xem rất nhiều các bộ phim về cảnh sát hình sự, vậy thì gợi ý một bộ phim trong cơ sở dữ liệu có chung đặc tính *hình sự* tới *user* này, ví dụ phim *Người phán xử*. Cách tiếp cận này yêu cầu việc sắp xếp các *items* vào từng nhóm hoặc đi tìm các đặc trưng của từng *item*. Tuy nhiên, có những *items* không có nhóm cụ thể và việc xác định nhóm hoặc đặc trưng của từng *item* đôi khi là bất khả thi.

* 1. **Collaborative filtering:** hệ thống gợi ý *items* dựa trên sự tương quan (similarity) giữa các *users* và/hoặc *items*. Có thể hiểu rằng ở nhóm này một item được recommended tới một user dựa trên những users có hành vi tương tự.

Ví dụ: users A, B, C đều thích các bài hát của Noo Phước Thịnh. Ngoài ra, hệ thống biết rằng users B, C cũng thích các bài hát của Bích Phương nhưng chưa có thông tin về việc liệu user A có thích Bích Phương hay không. Dựa trên thông tin của những users tương tự là B và C, hệ thống có thể dự đoán rằng A cũng thích Bích Phương và gợi ý các bài hát của ca sĩ này tới A.

1. **Apache Spark**

Apache Spark là một khung tính toán cụm nhanh được sử dụng để xử lý, truy vấn và phân tích dữ liệu lớn. Dựa trên tính toán trong bộ nhớ, nó có lợi thế hơn một số khung dữ liệu lớn khác.



Được viết bằng ngôn ngữ lập trình Scala, cộng đồng nguồn mở đã phát triển một công cụ tuyệt vời để hỗ trợ Python cho Apache Spark. PySpark giúp các nhà khoa học dữ liệu giao tiếp với RDD trong Apache Spark và Python thông qua thư viện Py4j. Có nhiều tính năng giúp PySpark trở thành một khung tốt hơn các tính năng khác:

* Tốc độ: Nó nhanh hơn 100 lần so với các khung xử lý dữ liệu quy mô lớn truyền thống.
* Bộ nhớ đệm mạnh mẽ: Lớp lập trình đơn giản cung cấp khả năng lưu trữ bộ nhớ cache và ổ đĩa mạnh mẽ.
* Triển khai: Có thể được triển khai thông qua Mesos, Hadoop thông qua Sợi hoặc trình quản lý cụm riêng của Spark.
* Thời gian thực: Tính toán thời gian thực và độ trễ thấp vì tính toán trong bộ nhớ.
* Polyglot: Hỗ trợ lập trình trong Scala, Java, Python và R.

Trong thời đại dữ liệu lớn, các nhà thực hành cần các công cụ nhanh và đáng tin cậy hơn bao giờ hết để xử lý luồng dữ liệu. Các công cụ trước đó như MapReduce được yêu thích nhưng rất chậm. Để khắc phục vấn đề này, Spark đưa ra một giải pháp vừa nhanh chóng vừa có mục đích chung. Sự khác biệt chính giữa Spark và MapReduce là nó chạy các tính toán trong bộ nhớ trong thời gian sau đó trên đĩa cứng. Nó cho phép truy cập và xử lý dữ liệu tốc độ cao, giảm thời gian từ hàng giờ xuống còn phút.

* 1. **Spark hoạt động như thế nào**

Spark dựa trên công cụ tính toán, có nghĩa là nó đảm nhiệm ứng dụng lập lịch, phân phối và giám sát. Mỗi tác vụ được thực hiện trên nhiều máy worker khác nhau được gọi là cụm máy tính. Một cụm máy tính đề cập đến việc phân chia các nhiệm vụ. Một máy thực hiện một nhiệm vụ, trong khi những máy khác đóng góp vào kết quả cuối cùng thông qua một nhiệm vụ khác. Cuối cùng, tất cả các nhiệm vụ được tổng hợp lại để tạo ra một đầu ra.

**Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, số

Mô tả được tạo tự động**

Spark được thiết kế để làm việc với:

* Python
* Java
* Scala
* SQL

Một tính năng quan trọng của Spark là có số lượng lớn thư viện tích hợp, bao gồm MLlib cho máy học. Nó cũng được thiết kế để hoạt động với các cụm Hadoop và có thể đọc nhiều loại tệp, bao gồm dữ liệu Hive, CSV, JSON, dữ liệu Casandra, v.v…

* 1. **Pyspark trong ngành công nghiệp**

Mọi ngành công nghiệp đều xoay quanh dữ liệu lớn và nơi có dữ liệu lớn, có phân tích liên quan. Vì vậy, hãy xem xét các ngành công nghiệp khác nhau nơi Apache Spark được sử dụng.

A picture containing LEGO

Description automatically generated with medium confidence

A picture containing clipart, design, LEGO

Description automatically generated Truyền thông là một trong những ngành công nghiệp lớn nhất phát triển theo hướng phát trực tuyến. Netflix sử dụng Apache Spark để xử lý luồng thời gian thực để cung cấp các đề xuất trực tuyến được cá nhân hóa cho khách hàng của mình. Nó xử lý 450 tỷ sự kiện mỗi ngày chảy vào các ứng dụng phía máy chủ.

Ảnh có chứa đồng hồ, thiết kế

Mô tả được tạo tự động Tài chính là một lĩnh vực khác mà việc xử lý thời gian thực của Apache Spark đóng vai trò quan trọng. Các ngân hàng đang sử dụng Spark để truy cập và phân tích hồ sơ truyền thông xã hội để hiểu rõ hơn có thể giúp họ đưa ra quyết định kinh doanh đúng đắn để đánh giá rủi ro tín dụng , quảng cáo được nhắm mục tiêu và phân khúc khách hàng. Khách hàng cũng giảm thời gian sử dụng Spark. Phát hiện gian lận là một trong những lĩnh vực được sử dụng rộng rãi nhất trong học máy mà Spark có liên quan.

Các nhà cung cấp dịch vụ chăm sóc sức khỏe đang sử dụng Apache Spark để phân tích hồ sơ bệnh nhân cùng với dữ liệu lâm sàng trong quá khứ để xác định bệnh nhân nào có khả năng phải đối mặt với các vấn đề sức khỏe sau khi được xuất viện. Apache Spark được sử dụng trong giải trình tự bộ gen để giảm thời gian cần thiết để xử lý dữ liệu bộ gen.

Ảnh có chứa văn bản, Phông chữ, hình mẫu, thiết kế

Mô tả được tạo tự động

Ảnh có chứa phim hoạt hình, Tác phẩm nghệ thuật của trẻ con, minh họa, đồ chơi

Mô tả được tạo tự động Bán lẻ và thương mại điện tử là một ngành mà người ta không thể tưởng tượng nó đang chạy mà không sử dụng phân tích và quảng cáo được nhắm mục tiêu. Một trong những nền tảng thương mại điện tử lớn nhất hiện nay, Alibaba chạy một số công việc Spark lớn nhất trên thế giới để phân tích petabyte dữ liệu. Alibaba thực hiện trích xuất tính năng trong dữ liệu hình ảnh. eBay sử dụng Apache Spark để cung cấp các ưu đãi được nhắm mục tiêu, nâng cao trải nghiệm của khách hàng và tối ưu hóa hiệu suất tổng thể.

Các ngành công nghiệp du lịch cũng sử dụng Apache Spark. TripAdvisor , một trang web du lịch hàng đầu giúp người dùng lên kế hoạch cho một chuyến đi hoàn hảo, đang sử dụng Apache Spark để tăng tốc các đề xuất tùy chỉnh được cá nhân hóa. TripAdvisor sử dụng Apache Spark để cung cấp lời khuyên cho hàng triệu khách du lịch bằng cách so sánh hàng trăm trang web để tìm giá khách sạn tốt nhất cho khách hàng của mình.

1. **Thuật toán ALS (Alternating Least Squares)**

Thuật toán ALS là một phương pháp lọc cộng tác (collaborative filtering) được sử dụng để xây dựng hệ thống gợi ý (recommendation systems). Nó là một phương pháp tiếp cận Matrix Factorization cho lọc cộng tác.

Ý tưởng chính của thuật toán ALS là xấp xỉ một ma trận lớn (thể hiện sự tương tác giữa người dùng và các mục tiêu) bằng cách phân tích ma trận thành hai ma trận con nhỏ hơn.

Thuật toán ALS hoạt động bằng cách lặp đi lặp lại hai bước:

1. Cố định ma trận người dùng: Trong bước này, ma trận người dùng được coi là cố định và ma trận mục tiêu được cập nhật. Sử dụng phương pháp bình phương tối thiểu (least squares), các giá trị trong ma trận mục tiêu được cập nhật để xấp xỉ ma trận ban đầu nhất.
2. Cố định ma trận mục tiêu: Trong bước này, ma trận mục tiêu được coi là cố định và ma trận người dùng được cập nhật. Tương tự như bước trước, các giá trị trong ma trận người dùng được cập nhật để xấp xỉ ma trận ban đầu nhất.

Quá trình lặp giữa hai bước trên được tiếp tục cho đến khi thuật toán hội tụ và ma trận xấp xỉ đạt được sự hài hòa giữa người dùng và mục tiêu.

Thuật toán ALS có thể được áp dụng cho các bài toán lọc cộng tác không xác định và lọc cộng tác đa dạng. Nó có thể xử lý hiệu quả các ma trận lớn và giúp xác định các mẫu ẩn trong dữ liệu người dùng và mục tiêu

**III. Xây dựng hệ thống gợi ý sách sử dụng ALS và Apache Spark**

1. **Load dữ liệu và chia mẫu trập train, test**

Dữ liệu sẽ được load dưới dạng rdd. Đây là định dạng dữ liệu chịu lỗi tốt và có khả năng phân tán linh hoạt trên nhiều cụm xử lý của spark. Vì những tiện ích này chúng thường được sử dụng trong các tính toán và biến đổi dữ liệu.

import pyspark as ps

from pyspark.sql import SQLContext

from pyspark.ml.evaluation import RegressionEvaluator

from pyspark.ml.tuning import ParamGridBuilder, CrossValidator

from pyspark.ml.recommendation import ALS

from pyspark.ml import Pipeline

from pyspark.sql import Row

from pyspark.sql.functions import col, udf, when

import numpy as np

from IPython.display import display, Image

spark = ps.sql.SparkSession.builder \

            .master("local") \

            .appName("book\_recommendation") \

            .getOrCreate()

sc = spark.sparkContext

sqlContext = SQLContext(sc)

# Đọc dữ liệu từ dat file

ratings\_df = spark.read.csv('/content/ratings.csv', header=True, inferSchema=True)

ratings\_df.printSchema()

books\_df = spark.read.csv('/content/books.csv', header=True, inferSchema=True)

books\_df.printSchema()

#Chia tỷ lệ tập train/test 0.8/0.2

traning\_df, test\_df = ratings\_df.randomSplit([0.8, 0.2])

**Hiển thị 10 dòng đầu tiên của rating.data**

+-------+-------+------+

|book\_id|user\_id|rating|

+-------+-------+------+

|      1|    314|     5|

|      1|    439|     3|

|      1|    588|     5|

|      1|   1169|     4|

|      1|   1185|     4|

|      1|   2077|     4|

|      1|   2487|     4|

|      1|   2900|     5|

|      1|   3662|     4|

|      1|   3922|     5|

+-------+-------+------+

1. **Xây dựng và đánh giá mô hình trên pyspark**

Sử dụng module pyspark.ml.recommendation ta có thể dễ dàng xây dựng một mô hình recommendation theo phương pháp ALS. Các tham số khi khởi tạo mô hình đó là:

* maxIter: Số lượng vòng lặp tối đa
* regParam: Hệ số kiểm soát regularization
* userCol: tên cột chứa id của user
* itemCol: tên cột chứa id của item
* ratingCol: tên cột chứa giá trị của rating

iterations = 10

regularization\_parameter = 0.1

ranks = 5

errors = []

err = 0

# Tạo mô hình ALS với các tham số cho trước

als = ALS(maxIter=iterations, regParam=regularization\_parameter, rank=ranks, userCol="user\_id", itemCol="book\_id", ratingCol="rating")

# Huấn luyện mô hình trên tập huấn luyện

model = als.fit(traning\_df)

Đánh giá mô hình thông qua chỉ số căn bậc 2 của trung bình sai số dự báo so với giá trị rating trên tập test, chỉ số RMSE.

# Tính toán độ đo RMSE trên tập kiểm tra

evaluator = RegressionEvaluator(metricName="rmse", labelCol="rating", predictionCol="prediction")

rmse = evaluator.evaluate(new\_predictions)

print("Root-mean-square error = " + str(rmse))

Root-mean-square error = 0.8993843347827746

1. **Khuyến nghị sản phẩm cho người dùng và khuyến nghị người dùng cho sản phẩm**

Thông thường các nền tảng recommendation sẽ tìm ra top 5 hoặc 10 sản phẩm để khuyến nghị cho người dùng vì đây là số lượng vừa đủ (không quá nhiều, cũng không quá ít) để người dùng lựa chọn. Trong pyspark, để tìm ra top 10 sản phẩm để khuyến nghị cho mỗi một user ta thực hiện lệnh recommendForAllUsers(10)

# Tạo ra top 10 bộ phim khuyến nghị cho mỗi một user

userRecs = model.recommendForAllUsers(10)

userRecs.select("user\_id", "recommendations.book\_id").show(10, False)

+-------+------------------------------+

|user\_id|book\_id                       |

+-------+------------------------------+

|1      |[1180, 7639, 7242, 5753, 4778]|

|2      |[4509, 4220, 9991, 7117, 3508]|

|3      |[4868, 1788, 9008, 19, 9049]  |

|4      |[4653, 4, 8703, 4706, 9912]   |

|5      |[8109, 5794, 4522, 6024, 9531]|

|6      |[5207, 6590, 9566, 8109, 3395]|

|7      |[5207, 6590, 9566, 6920, 3628]|

|8      |[4778, 9486, 8606, 7242, 9992]|

|9      |[9842, 9076, 7947, 5175, 9114]|

|10     |[8271, 3920, 267, 3753, 5794] |

+-------+------------------------------+

Trong thuật toán thì user và item là bình đẳng, chỉ cần hoán vị dòng cho cột tại ma trận tiện ích là ta thu được thuật toán recommend user cho item. Muốn tìm ra 10 cuốn sách để khuyến nghị cho một user ta sử dụng hàm recommendForAllItems(10).

# Tạo ra top 10 user khuyến nghị cho mỗi một bộ phim

bookRecs = model.recommendForAllItems(10)

bookRecs.select("book\_id", "recommendations.user\_id").show(10, False)

+-------+-----------------------------------+

|book\_id|user\_id                            |

+-------+-----------------------------------+

|1      |[26717, 51504, 21509, 4486, 41819] |

|2      |[4486, 21509, 26717, 46485, 17856] |

|3      |[48738, 26717, 7094, 41819, 11334] |

|4      |[4486, 34569, 11763, 628, 39207]   |

|5      |[4486, 17856, 52125, 21509, 11763] |

|6      |[21509, 50580, 46485, 38866, 21665]|

|7      |[51504, 21509, 17856, 11763, 24822]|

|8      |[4486, 38445, 17762, 39711, 21509] |

|9      |[48738, 7094, 26717, 27056, 46485] |

|10     |[21509, 4486, 17856, 11763, 51504] |

+-------+-----------------------------------+

**IV. Demo**

* Giao diện trang web

**HìnhA computer screen shot of a book

Description automatically generated with medium confidence 1. Trang home**

**A close-up of a book recommendation system

Description automatically generated with medium confidence**

**Hình 2. Trang Recommend**

1. **Đánh giá và kết luận**

* Khi người dùng tên một cuốn sách thì hệ thống sẽ đưa ra gợi ý 10 cuốn sách dựa trên những người dùng tương đồng về sở thích.

User to user or item to item

**Tài liệu tham khảo**