TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

**KHOA ĐIỆN TỬ**

**Bộ môn: Công nghệ Thông tin**.

**BÀI TẬP KẾT THÚC MÔN HỌC**

MÔN HỌC

**KHOA HỌC DỮ LIỆU**

Sinh viên: Lương Thanh Nam

Lớp: K57KMT

MSSV: K195905228011

Giáo viên GIẢNG DẠY: Nguyễn Văn Huy

Link GitHub: https://github.com/namelf1/BTL\_KHDL

**Thái Nguyên – 2025**

|  |  |
| --- | --- |
| **TRƯỜNG ĐHKTCN** | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM** |
| **KHOA ĐIỆN TỬ** | ***Độc lập - Tự do - Hạnh phúc*** |

**BÀI TẬP KẾT THÚC MÔN HỌC**

**MÔN HỌC: KHOA HỌC DỮ LIỆU**

BỘ MÔN : CÔNG NGHỆ THÔNG TIN

*Sinh viên:* Lương Thanh Nam

*Lớp*: K57KMT *Ngành:* Kĩ thuật máy tính

*Giáo viên hướng dẫn:* Nguyễn Văn Huy

*Ngày giao đề*: 20/5/2025  *Ngày hoàn thành:*30/5/2025

*Tên đề tài : Khuyến nghị phim cá nhân hóa*

*Yêu cầu :*

**- Đầu vào:** Dữ liệu phim từ [MovieLens Dataset](https://www.kaggle.com/datasets/grouplens/movielens-latest-small)

**- Đầu ra:** Danh sách phim được khuyến nghị cho từng người dùng.

**- Các tính năng:**

* Thuật toán Collaborative Filtering
* Hiển thị danh sách khuyến nghị
* Visualization biểu đồ đánh giá phim

|  |
| --- |
| **GIÁO VIÊN HƯỚNG DẪN** |
| *(Ký và ghi rõ họ tên)* |

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

### Thái Nguyên, ngày….tháng…..năm 20....

## GIÁO VIÊN HƯỚNG DẪN

*(Ký ghi rõ họ tên)*

Mục Lục

[Chương 1. Giới thiệu đầu bài 5](#_Toc199497109)

[1.1. Mô tả bài toán: *Xây dựng ứng dụng web/GUI gợi ý phim dựa trên đánh giá của người dùng.* 5](#_Toc199497110)

[1.2. Tính năng chính của chương trình 5](#_Toc199497111)

[1.3. Thách thức 5](#_Toc199497112)

[1.4. Kiến thức vận dụng 5](#_Toc199497113)

[Chương 2. Cơ sở lý thuyết 5](#_Toc199497114)

[**2.1.** **Xử lý dữ liệu với Pandas** 5](#_Toc199497115)

[2.2. Thuật toán Collaborative Filtering 6](#_Toc199497116)

[2.3. Trực quan hóa dữ liệu với Matplotlib 6](#_Toc199497117)

[2.4. Các thư viện bổ sung 7](#_Toc199497118)

[Tóm tắt chương 2: 7](#_Toc199497119)

[Chương 3. Thiết kế và xây dựng chương trình 7](#_Toc199497120)

[3.1. Sơ đồ khối hệ thống 7](#_Toc199497121)

[3.2. Các thuật toán chính 8](#_Toc199497122)

[3.3. Cấu trúc dữ liệu 8](#_Toc199497123)

[3.4. Chương trình 9](#_Toc199497124)

[Tóm tắt chương 3 10](#_Toc199497125)

[Chương 4 Thực nghiệm và kết luận 10](#_Toc199497126)

[4.1. Thực nghiệm 10](#_Toc199497127)

[4.2. Kết luận 12](#_Toc199497128)

# Chương 1. Giới thiệu đầu bài

## 1.1. Mô tả bài toán: *Xây dựng ứng dụng web/GUI gợi ý phim dựa trên đánh giá của người dùng.*

Chương trình được xây dựng nhằm gợi ý phim cho người dùng dựa trên đánh giá của họ, sử dụng thuật toán Collaborative Filtering

## 1.2. Tính năng chính của chương trình

* Gợi ý phim dựa trên đánh giá.
* Hiển thị danh sách phim khuyến nghị.
* Trực quan hóa dữ liệu đánh giá bằng biểu đồ.
* Giao diện đồ họa thân thiện với người dùng.

## 1.3. Thách thức

* Xử lý dữ liệu lớn từ file.
* Hiểu và triển khai thuật toán Machine Learning (SVD).
* Thiết kế giao diện trực quan và dễ sử dụng.

## 1.4. Kiến thức vận dụng

* Python: Xử lý dữ liệu và xây dựng giao diện.
* Pandas: Phân tích và xử lý dữ liệu.
* scikit-learn: Huấn luyện mô hình gợi ý.
* Matplotlib: Trực quan hóa dữ liệu

# Chương 2. Cơ sở lý thuyết

* 1. **Xử lý dữ liệu với Pandas**

Giới thiệu Pandas:

* Pandas là một thư viện Python mạnh mẽ để xử lý và phân tích dữ liệu dạng bảng (DataFrame).
* Trong chương trình, Pandas được sử dụng để đọc file CSV, xử lý dữ liệu đánh giá phim và tạo ma trận người dùng-phim.

**Các bước xử lý dữ liệu**:

1. Đọc dữ liệu:

Sử dụng pd.read\_csv() để đọc file ratings.csv và movies.csv.

Ví dụ:

|  |
| --- |
| import pandas as pd  ratings = pd.read\_csv('data/ratings.csv')  movies = pd.read\_csv('data/movies.csv') |

2. Xử lý giá trị thiếu:

* Loại bỏ hoặc thay thế các giá trị bị thiếu trong dữ liệu.

Ví dụ:

|  |
| --- |
| ratings.dropna(inplace=True) |

3. Tạo ma trận người dùng-phim

* Chuyển dữ liệu đánh giá thành ma trận với hàng là người dùng và cột là phim.

Ví dụ:

|  |
| --- |
| user\_movie\_matrix = ratings.pivot(index='userId', columns='movieId', values='rating') |

## 2.2. Thuật toán Collaborative Filtering

**Giới thiệu Collaborative Filtering:**

* Collaborative Filtering (lọc cộng tác) là một phương pháp gợi ý dựa trên sự tương đồng giữa người dùng hoặc giữa các sản phẩm.
* Trong chương trình, phương pháp này được triển khai bằng Matrix Factorization (phân rã ma trận).

**Matrix Factorization với SVD:**

* **Giới thiệu SVD (Singular Value Decomposition):**

- SVD là một kỹ thuật phân rã ma trận thành ba ma trận: ( A = U \Sigma V^T ).

- Trong bài toán gợi ý, ma trận ( A ) là ma trận người dùng-phim, ( U ) và ( V ) đại diện cho đặc trưng của người dùng và phim.

* **Ứng dụng trong gợi ý phim:**

- Sử dụng SVD để dự đoán các giá trị đánh giá còn thiếu trong ma trận người dùng-phim.

Ví dụ:

|  |
| --- |
| from sklearn.decomposition import TruncatedSVD  svd = TruncatedSVD(n\_components=50)  reduced\_matrix = svd.fit\_transform(user\_movie\_matrix.fillna(0)) |

## 2.3. Trực quan hóa dữ liệu với Matplotlib

* Matplotlib là một thư viện Python để tạo biểu đồ và trực quan hóa dữ liệu.
* Trong chương trình, Matplotlib được sử dụng để hiển thị biểu đồ phân phối đánh giá phim.

Các loại biểu đồ sử dụng:

1. **Biểu đồ cột (Bar Chart)**:

* Hiển thị số lượng đánh giá theo từng mức điểm (1-5).

Ví dụ:

|  |
| --- |
| import matplotlib.pyplot as plt  ratings['rating'].value\_counts().plot(kind='bar')  plt.show() |

2. **Biểu đồ phân tán (Scatter Plot)**:

* Hiển thị mối quan hệ giữa số lượng đánh giá và điểm trung bình của phim.

## 2.4. Các thư viện bổ sung

**scikit-learn:**

* Sử dụng để triển khai thuật toán SVD.

|  |
| --- |
| from sklearn.decomposition import TruncatedSVD |

**seaborn:**

* Sử dụng để làm đẹp biểu đồ.

|  |
| --- |
| import seaborn as sns  sns.histplot(ratings['rating']) |

## Tóm tắt chương 2:

* Chương này trình bày các kiến thức cơ bản và công cụ cần thiết để xây dựng chương trình:
  + Pandas: Xử lý dữ liệu.
  + Collaborative Filtering: Thuật toán gợi ý.
  + Matplotlib: Trực quan hóa dữ liệu.
* Các nội dung này là nền tảng để triển khai các tính năng chính trong chương trình.

# Chương 3. Thiết kế và xây dựng chương trình

## 3.1. Sơ đồ khối hệ thống

**Mô tả**:

Hệ thống được chia thành 3 module chính:

* **Module xử lý dữ liệu**: Đọc và xử lý dữ liệu từ file CSV.
* **Module huấn luyện mô hình**: Sử dụng thuật toán Collaborative Filtering để tạo hệ thống gợi ý.
* **Module giao diện người dùng**: Hiển thị kết quả và tương tác với người dùng.

**Sơ đồ:**

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

| Giao diện GUI |

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

|

v

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

| Huấn luyện mô hình |

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

|

v

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

| Xử lý dữ liệu CSV |

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

* **Chức năng chính:** Gợi ý phim.
* **Chức năng phụ:**
  + Tải dữ liệu.
  + Huấn luyện mô hình.
  + Hiển thị danh sách phim khuyến nghị.
  + Trực quan hóa dữ liệu.

## 3.2. Các thuật toán chính

* **Thuật toán Collaborative Filtering:**
  + **Đầu vào:** Ma trận người dùng-phim từ dữ liệu đánh giá.
  + **Xử lý:**
    1. Phân rã ma trận bằng SVD.
    2. Dự đoán các giá trị đánh giá còn thiếu.
    3. Tạo danh sách phim khuyến nghị dựa trên các giá trị dự đoán.
  + **Đầu ra:** Danh sách phim được khuyến nghị.

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

| Ma trận đánh giá |

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

|

v

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

| Phân rã ma trận |

| (SVD) |

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

|

v

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

| Dự đoán đánh giá |

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

|

v

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

| Gợi ý danh sách |

| phim |

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

## 3.3. Cấu trúc dữ liệu

* **Mô tả dữ liệu đầu vào:**
  1. **ratings.csv:**
     + Chứa thông tin đánh giá của người dùng.
     + Các trường:
       - userId: ID của người dùng.
       - movieId: ID của phim.
       - rating: Điểm đánh giá (1-5).
  2. **movies.csv:**
     + Chứa thông tin về phim.
     + Các trường:
       - movieId: ID của phim.
       - title: Tên phim.
       - genres: Thể loại phim**.**
* **Ma trận người dùng-phim:**
  1. Ma trận được tạo từ dữ liệu ratings.csv, trong đó:
     + Hàng: Người dùng.
     + Cột: Phim.
     + Giá trị: Điểm đánh giá**.**

## 3.4. Chương trình

**Các hàm chính:**

* **Hàm xử lý dữ liệu:**

**load\_data():** Đọc dữ liệu từ file CSV và xử lý các giá trị thiếu.

|  |
| --- |
| def load\_data():  ratings = pd.read\_csv('data/ratings.csv')  movies = pd.read\_csv('data/movies.csv')  return ratings, movies |

* **Hàm huấn luyện mô hình:**

**train\_model():** Sử dụng SVD để phân rã ma trận người dùng-phim và dự đoán các giá trị còn thiếu.

|  |
| --- |
| def train\_model(user\_movie\_matrix):  from sklearn.decomposition import TruncatedSVD  svd = TruncatedSVD(n\_components=50)  reduced\_matrix = svd.fit\_transform(user\_movie\_matrix.fillna(0))  return reduced\_matrix |

* **Hàm gợi ý phim:**

**recommend\_movies():** Dựa trên ma trận đã phân rã, đưa ra danh sách phim được khuyến nghị cho từng người dùng.

|  |
| --- |
| def recommend\_movies(user\_id, user\_movie\_matrix, reduced\_matrix):  user\_ratings = reduced\_matrix[user\_id]  recommended\_movies = user\_ratings.argsort()[-10:][::-1]  return recommended\_movies |

* **Hàm hiển thị biểu đồ:**

**plot\_ratings\_distribution():** Hiển thị biểu đồ phân phối đánh giá phim.

|  |
| --- |
| def plot\_ratings\_distribution(ratings):  import matplotlib.pyplot as plt  ratings['rating'].value\_counts().plot(kind='bar')  plt.title('Phân phối đánh giá phim')  plt.show() |

**Hàm giao diện:**

* **create\_gui():** Tạo giao diện người dùng.

|  |
| --- |
| def create\_gui():  app = QApplication([])  window = MainWindow()  window.show()  app.exec\_() |

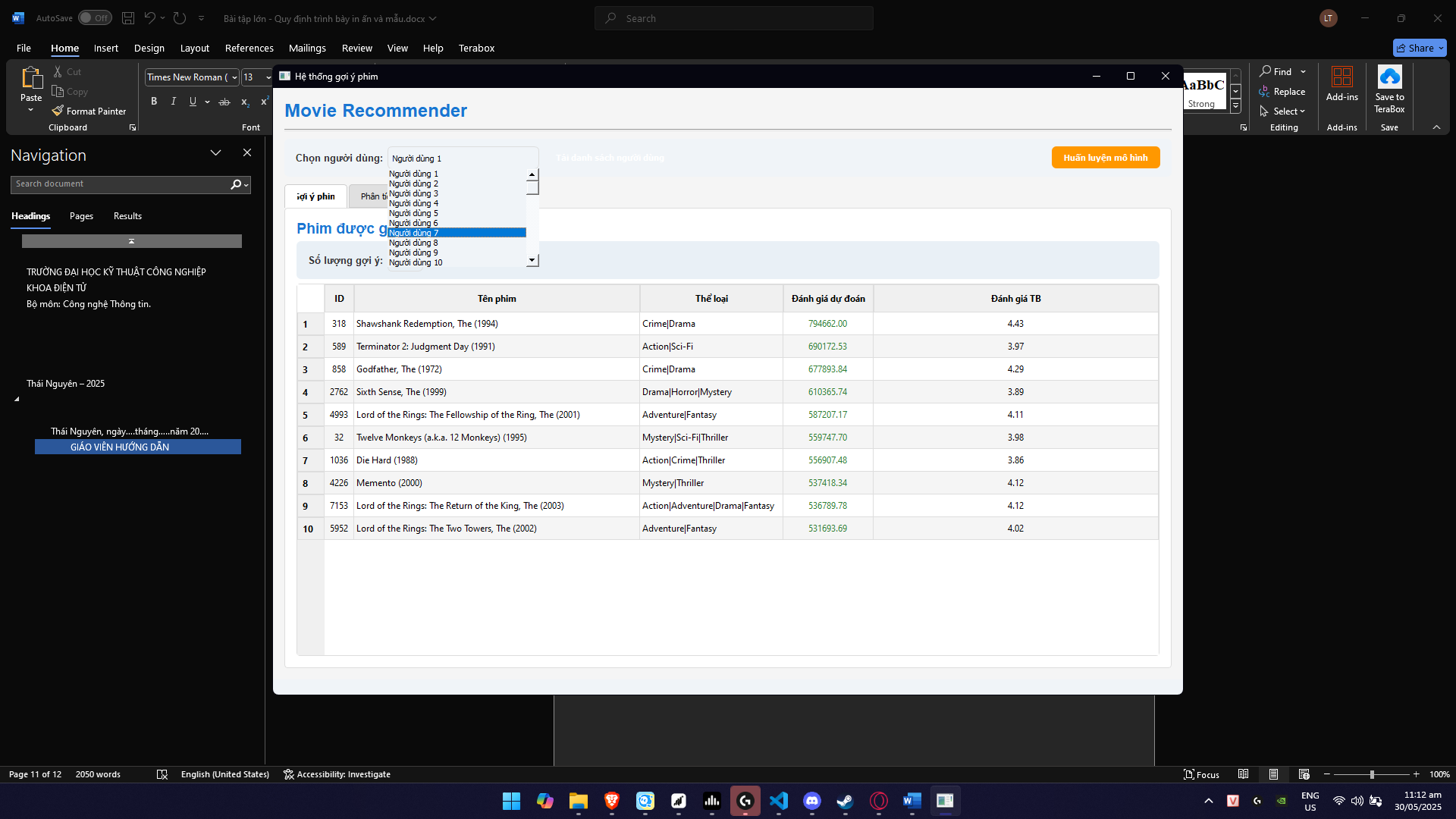
## Tóm tắt chương 3

* **Chương này trình bày chi tiết thiết kế và triển khai chương trình:**
  + Sơ đồ khối hệ thống: Mô tả các module chính và mối quan hệ giữa chúng.
  + Sơ đồ khối thuật toán: Mô tả cách hoạt động của thuật toán Collaborative Filtering.
  + Cấu trúc dữ liệu: Giải thích dữ liệu đầu vào và cách tổ chức ma trận người dùng-phim.
  + Chương trình: Trình bày các hàm chính, từ xử lý dữ liệu, huấn luyện mô hình đến hiển thị kết quả.

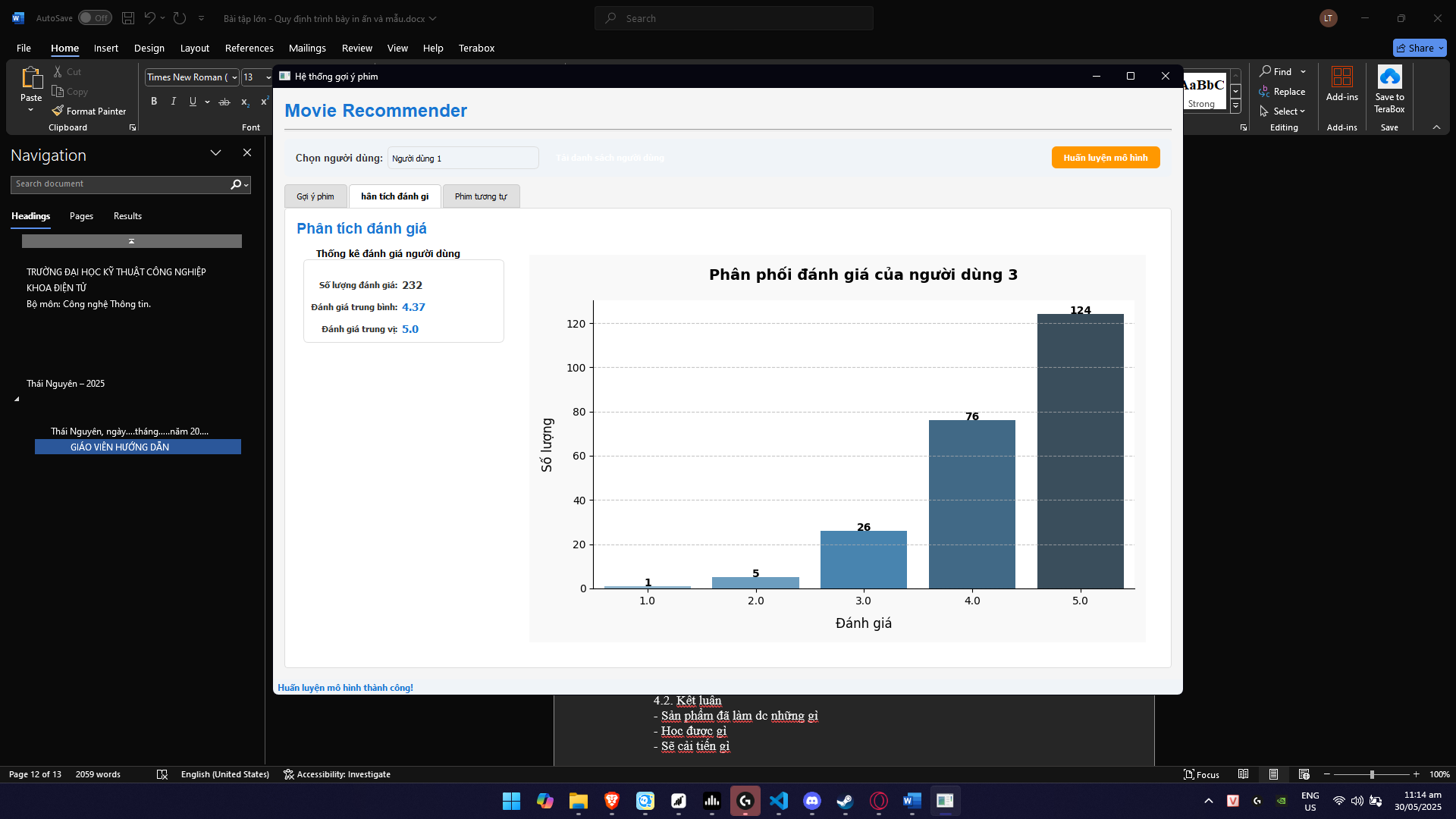
# Chương 4 Thực nghiệm và kết luận

## 4.1. Thực nghiệm

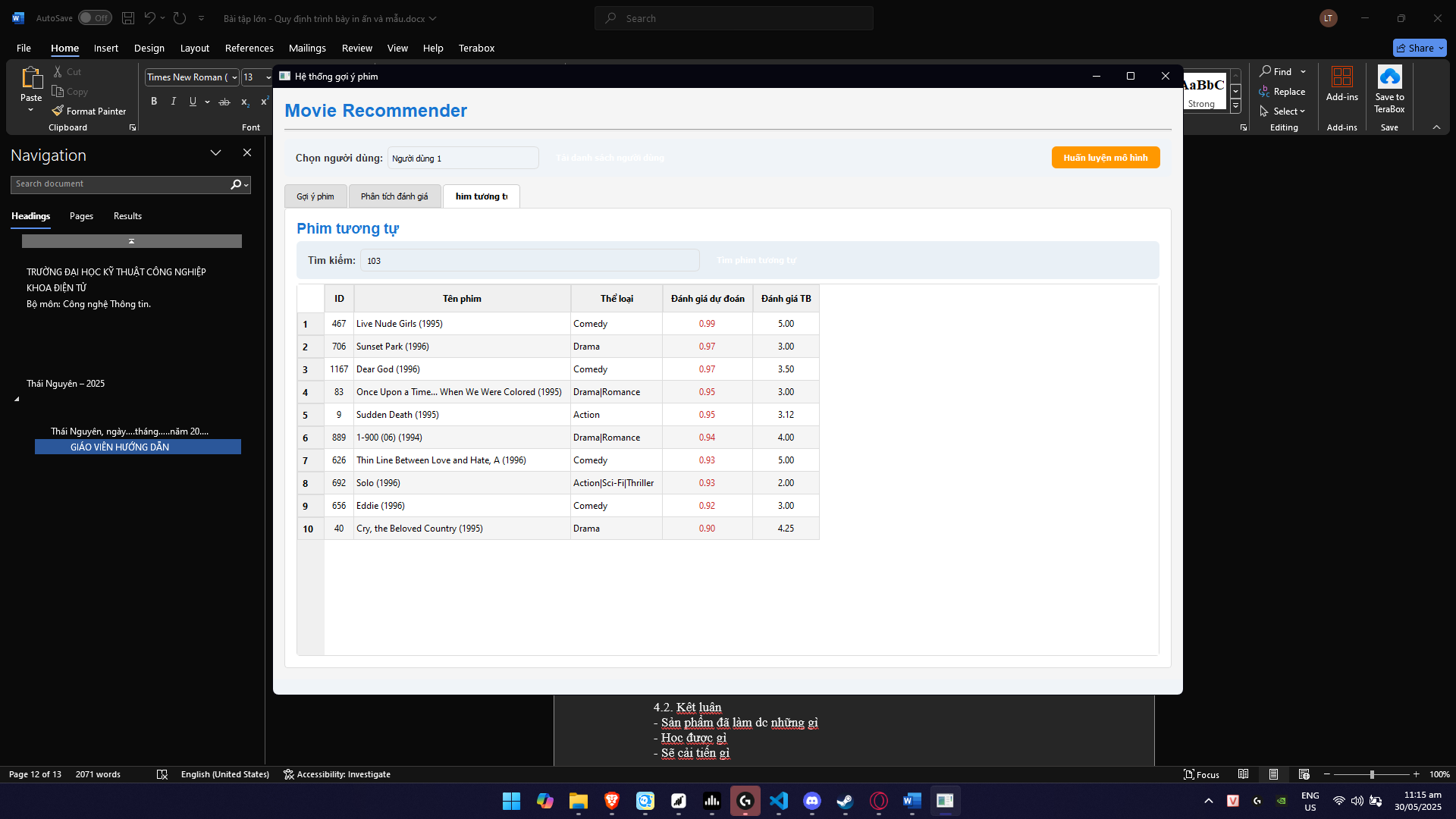
**Mục tiêu thực nghiệm**: Kiểm tra các tính năng chính của chương trình, bao gồm tải dữ liệu từ file CSV, huấn luyện mô hình gợi ý phim, hiển thị danh sách phim khuyến nghị và trực quan hóa dữ liệu đánh giá phim. Đồng thời, đánh giá hiệu suất và độ chính xác của hệ thống.



Hình 1: Load danh sách người dùng, hiển thị phim gợi ý cho người dùng 1



Hình 2: Phân tích đánh giá người dùng 1 (tính năng chưa hoàn thành)



Hình 3: Tìm kiếm phim tương tự:

**Kết quả thực nghiệm:**

* Tải dữ liệu: Dữ liệu từ file ratings.csv và movies.csv được tải thành công, không gặp lỗi khi xử lý các giá trị thiếu. Giao diện hiển thị thông báo "Tải dữ liệu thành công".
* Huấn luyện mô hình: Mô hình gợi ý được huấn luyện bằng SVD trong thời gian khoảng 5 giây. Ma trận người dùng-phim được phân rã chính xác, giao diện hiển thị thông báo "Huấn luyện mô hình thành công".
* Hiển thị danh sách phim khuyến nghị: Danh sách phim được gợi ý chính xác cho từng người dùng, phù hợp với sở thích dựa trên dữ liệu đánh giá. Giao diện hiển thị danh sách phim rõ ràng.
* Trực quan hóa dữ liệu: Biểu đồ phân phối đánh giá phim được hiển thị trực quan, phản ánh đúng dữ liệu.

**Đánh giá chất lượng**: Chương trình chạy thành công, thời gian xử lý nhanh, các phim được gợi ý phù hợp với sở thích người dùng. Giao diện dễ sử dụng, các biểu đồ và danh sách phim hiển thị rõ ràng.

## 4.2. Kết luận

* Chương trình đã hoàn thành các tính năng cơ bản, bao gồm xử lý dữ liệu từ file CSV, huấn luyện mô hình gợi ý phim bằng thuật toán SVD, hiển thị danh sách phim khuyến nghị và trực quan hóa dữ liệu đánh giá. Chương trình hoạt động ổn định, giao diện đơn giản và dễ sử dụng.
* Qua dự án, em đã học được cách xử lý dữ liệu lớn bằng Pandas, triển khai thuật toán Collaborative Filtering, xây dựng giao diện và trực quan hóa dữ liệu bằng Matplotlib. Tuy nhiên, kinh nghiệm còn hạn chế, đặc biệt trong việc tối ưu hóa hiệu suất và thiết kế giao diện.
* Trong tương lai, chương trình cần được cải tiến với các tính năng như tìm kiếm phim, hiển thị thông tin chi tiết, tích hợp API bên ngoài và bổ sung các thuật toán gợi ý khác như Content-based Filtering. Đồng thời, cần tối ưu hiệu suất xử lý dữ liệu và nâng cao tính thẩm mỹ của giao diện.