

Mini Project Report

Project Title: Personalized Recommendation System Web App using Streamlit

Author: [Your Name Here]

Platform: Python 3.x with Streamlit Web Framework

Tested Environment: Intel i3 Processor Laptop (Windows 10, 8GB RAM)

1. Introduction

Recommender systems have become an integral part of various applications in e-commerce, online streaming, and other user-centric platforms. This project aims to build a simple yet interactive web application that allows users to visualize personalized recommendations based on precomputed predicted ratings. The app is developed using **Streamlit**, a Python-based tool that converts scripts into shareable web apps.

2. Objective

To design and implement a web-based personalized recommendation system that:

- Accepts a CSV file with user-item prediction data
 - Allows user selection to view their top-N recommendations
 - Visualizes the predictions in an easy-to-understand bar chart
 - Enables filtered downloads and user feedback collection
-

3. Tools and Technologies

- **Programming Language:** Python 3.x
 - **Framework:** Streamlit
 - **Libraries:**
 - `pandas` – For data handling and manipulation
 - `matplotlib` and `seaborn` – For data visualization
 - `streamlit` – For UI and interactivity
-

4. Project Workflow

4.1 Input Data Format

The application expects a CSV file containing predictions with the following format:

```
user_id,item_id,predicted_rating
U1,I1,4.32
U1,I2,3.87
U2,I3,4.56
...
```

Each row represents a predicted rating for an item for a specific user.

4.2 Functional Modules

a. File Upload

- Users can upload a CSV file using Streamlit's file uploader widget.
- The uploaded file is read using `pandas.read_csv()`.

b. Data Preview

- The first few rows of the uploaded dataset are displayed for user confirmation.

c. User ID Selection

- A dropdown menu lists all unique `user_id` values.
- The user selects a specific ID to view personalized recommendations.

d. Top-N Recommendations

- The app filters rows for the selected user and sorts them by `predicted_rating` in descending order.
- Displays top 5 items by default.

e. Visualization

- A horizontal bar chart is generated using Seaborn to show the top-N item recommendations.
- X-axis: Predicted Rating, Y-axis: Item ID

f. Advanced Filtering and Download

- Additional slider widgets allow:
 - Selection of N items (Top-N)
 - Minimum rating threshold
- A downloadable CSV file with filtered recommendations is generated on button click.

g. User Feedback Section

- A free-text area to collect user thoughts on recommendation quality.
-

5. User Interface

The Streamlit app UI consists of:

- A page title and introduction
 - File uploader and preview table
 - User selection dropdown
 - Recommendation table
 - Seaborn bar chart
 - Sliders for top-N and rating threshold
 - Feedback text area and download button
-

6. Screenshots (*To be attached*)

- Main app layout with file uploader
 - User dropdown and recommendation preview
 - Bar chart of top-N predictions
 - Download and feedback section
-

7. Testing and Performance

- **Hardware:** Intel i3, 8GB RAM
 - **Software:** Python 3.11, Streamlit
 - **Browsers Tested:** Google Chrome, Microsoft Edge
 - **Performance:**
 - Able to load CSVs with up to 10,000 rows smoothly
 - Very minimal CPU/RAM usage observed
 - No GPU required
-

8. Output Example

user_id	item_id	predicted_rating
U1	I3	4.20
U1	I7	4.08
U1	I2	3.91

9. Future Enhancements

- Integrate live prediction models (e.g., Collaborative Filtering)
- Add user login functionality
- Host app on Streamlit Cloud or Heroku
- Use Plotly for interactive visualizations

- Build a REST API backend
-

10. Conclusion

This mini project successfully demonstrates the creation of a dynamic, interactive recommendation system dashboard using Streamlit. Despite running on basic hardware (Intel i3), the application delivers excellent performance for small-to-medium datasets. It is a great starting point for further enhancements involving real-time data and machine learning models.

11. References

- [Streamlit Documentation](#)
- [Pandas Library](#)
- [Seaborn Visualization](#)
- [Matplotlib](#)