**SUMMER TRAINING/INTERNSHIP**

**PROJECT REPORT**

(Term June-July 2025)

**Hybrid Collaborative Filtering for Personalized Product Recommendations**

**Submitted by:**

**Name:** Ponnuru Nithin

**Registration No:** 12303486

**Course Code: PETV86 – AI & ML for real world problem solving**

**Under the Guidance of:**

**Name:** Dr. Rosepreet Kaur Bhogal

# **School of Computer Science and Engineering**



A logo for a university

AI-generated content may be incorrect.

**Annexure-II: Student Declaration**

**To whom so ever it may concern**

I, **Ponnuru Nithin**, **12303486**, hereby declare that the work done by me on **“Hybrid Collaborative Filtering for Personalized Product Recommendations”** from **June,2025** to **July,2025** is a record of original work for the partial fulfillment of the requirements for the award of the degree, degree name.

**Name of the Student:** Ponnuru Nithin

**Registration Number:** 12303486

**Signature of the student:** P.Nithin

**Dated:** 24-08-2025

**Training Certification from organization**

****

**Acknowledgement**

First and foremost, I express my deepest gratitude for everyone who took part in the successful completion of the project, " Hybrid Collaborative Filtering for Personalized Product Recommendation," experiences that provide much enrichment and may not have been possible without the guiding force and backing from various individuals.

Foremost expressed upon embarking on the review of literature to the analysis of results, I extend heartfelt thanks to project guide and mentor Dr. Rosepreet Kaur Bhogal for support and encouragement. Their constructive criticism, domain expertise, and encouraging perspective contributed in advising mistakes or improvements that I am assorted to prepare an effective approach and make accurate judgment at every phase within the work:

At this stage, an invitation for thanks goes to the Your Institution Name faculty and administration for providing me with the resources yet needed and motivating me for undertaking the project. The academic ambience; availability of technical infrastructure; and close, cordial relationship with the competent faculty members made understanding and implementation of this project less cumbersome.

Thanks, dear friends and professional colleagues who were willing to spare their time for brainstorming through these tedious and lengthy discussions and group be together to interact. Our interaction provided for additional training and a deeper understanding in the topic areas of collaborative filtering and content-based recommendations.

A lot of warm thank-giving goes to my family. Their considerations and encouragement stuck with me in everything I had passed through for my continuous effort with the project. Their trust in my abilities simply fostered me to carry on and forge ahead to finish the project diligently and passionately.

Finally, my respect and acknowledgment go to the open-source community for their valuable contributions. I really need the liking of the people behind objects like Stack Overflow, GitHub, and other documentation sources that helped solve technical challenges and subsequent learning of this chapter.

Yet, the project has helped me in growing not only as a programmer but as a solver and a researcher. I have great pride in submitting this work as an emblem of my immersive experience, dedication, and curiosity.

**Chapter 1: Introduction**

* 1. **Company profile**

One of the giants in India, Lovely Professional University (LPU) is endowed with top-notch infrastructure, excellent academicism, and a very strong emphasis on research and innovation. The university is well located in Phagwara, Punjab, and over the years has evolved into a glaring star in higher education, offering more than 200 programs covering various disciplines such as engineering, technology, business, law, agriculture, and design.

The university is accredited A++ by NAAC and features among top-ranking universities in several national and international ratings. The university encourages collaboration between industry and academia so that the students work on actual-world problems through project work, internships, and startup incubations.

The project titled "Hybrid Collaborative Filtering for Personalized Product Recommendation" has been developed with guidance and support from the School of Computer Science and Engineering, LPU. It is an example of how the university is committed to practical learning experiences and fostering technological innovation among the students.

The project exploits the advanced techniques of data science, machine learning, and recommendation systems in line with the university's commitment to preparing future-ready professionals capable of designing meaningful solutions for the digital world.

* 1. **Overview of training domain**

The training domain was **AI/ML in the Real World**, where I explored how Artificial Intelligence and Machine Learning are used in day-to-day applications. It included both theoretical and practical implementations of various concepts such as:

**Types of AI**:

* + 1. **Narrow AI:** Narrow AI, with its narrow focus, has specific abilities yet lacks a broad, flexible intellect. Capable of visual recognition or answering queries, it remains enclosed within boundaries, understanding solely what it has been designed for.
    2. **General AI:** General AI, still theoretical, would possess reasoning akin to our own and aptitude to learn any task through its powers of comprehension. An AI achieving true generality would match the expansiveness of human cognition, though we have not reached such a point.
    3. **Super AI:** Future super AI, if ever attained, could surpass our talents altogether through self-augmenting progress. With problem-solving beyond what we can envision, it might push boundaries in unforeseen ways, for better or worse, demanding we reconsider what it means to be intelligent. Such potential must be approached prudently to guide its development safely, for unmatched gifts could come unmatched risks as well.

**Types of ML**:

* + 1. **Supervised Learning:** Labeled examples guide the development of algorithms capable of intricate prediction. Input-output pairs coupled with adaptive models, like linear regression and neural organization, enhance forecasts.

Briefly, decision trees categorise information to determine trends from training data. Outcomes result from step-by-step examinations reducing complexity.

* + 1. **Unsupervised Learning:** Unknown patterns emerge from unlabelled inputs as clustering unpacks convoluted associations. Dimensionality reduction simplifies overwhelming volumes to fundamental aspects. Additionally, anomaly detection isolates peculiarities diverging from commonalities.
    2. **Semi-Supervised Learning:** Semi-Supervised Learning enables leveraging unlabeled information to augment meager labeling efforts, benefiting tasks where annotation requires expertise or faces resource constraints. Through exploiting structure in unlabeled data, it complements limited hand-tagging for advancing speech decoding, diagnostic image classification, and other domains heavily reliant on examples.
    3. **Reinforcement Learning:** Reinforcement Learning cultivates agents via rewards and consequences to sequentially decide amid uncertainty, emulating how organisms learn from feedback to orchestrate behavior over time. Algorithms like Q-learning and Deep Reinforcement Learning underpin autonomous drones, robots, and digital assistants that interactively master their surroundings through trial and error.

**Real-world examples**:

* Netflix recommendations
* Google Translate
* Self-driving cars

**Important ML algorithms**:

* Logistic Regression
* Decision Trees and Random Forests
* Support Vector Machines (SVM)
* K-Nearest Neighbours (KNN)
* Naive Bayes
* Clustering (like K-Means)

Model evaluation techniques like Confusion Matrix, Accuracy, Precision, Recall, and FI Score.

* 1. **Objective of the project**

To create a recommendation system which integrates hybrid collaborative filtering to provide personalized product (movie) suggestions. This system has a few objectives, namely:

* Emulate the interactions among users and movies and create recommendations using collaborative filtering.
* Make use of content-based filtering through customization via genre features.
* Conduct exploratory data analysis using the movie data obtained.
* Automatically generate recommendations and EDA in PDF format for presenting.

In the end, this project demonstrates that a combination of multiple recommendation techniques significantly improves accuracy in a recommendation system as well as user satisfaction.

**Chapter 2: Training Overview**

* 1. **Tools & technologies**

To implement the recommendation system, the following tools and technologies were used:

#### **2.1** **Python**:

The primary programming language for the project due to its vast ecosystem and libraries that support data manipulation, visualization, and machine learning.

#### **2.2** **Pandas**:

Used for data loading, cleaning, and manipulation. It helps in handling tabular data efficiently.

**2.3** **NumPy**:

Essential for numerical computations, especially for generating simulated numerical data like ratings, durations, and similarity matrices.

#### **2.4 Seaborn & Matplotlib**:

These libraries were used to generate insightful visualizations during the EDA process.

#### **2.5 Scikit-learn**:

Provided the cosine\_similarity function to calculate similarity scores between users and items.

#### **2.6 ReportLab**:

A Python library used to generate PDF reports, making the output presentation-ready for stakeholders.

#### **2.7** **IDE**:

Used for writing and testing the code iteratively.The combination of these tools ensures a smooth workflow from data analysis to recommendation generation and report documentation.

* 1. **AREAS COVERED DURING TRAINING**
* Fundamentals of AI and ML
* Types of learning with examples and use cases
* Supervised and Unsupervised algorithms
* Deep learning and CNNs
* Model training, testing, and validation
* Real-time use of audio and video for building intelligent systems
* Voice recognition using NLP concepts
  1. **DAILY/WEEKLY WORK SUMMARY**

|  |  |
| --- | --- |
| **Week** | **Activities** |
| **Week 1** | I studied AI, ML types, real-life uses, and mathematics for ML. |
| **Week 2** | Python, Numpy, Pandas etc. Introduction to scikit library and Supervised learning techniques, hands-on with regression and classification models. |
| **Week 3** | Data preprocessing, train-test split. Model building, Random Forest, evaluation using accuracy & confusion matrix. |
| **Week 4** | Exploring other models like decision tree, SVM and application of models. |
| **Week 5** | Covered perception, neural networks and deep learning concepts. |
| **Final Days** | Preparing the project report, documentation, dataset organization, final deployment with some neural network concepts. |

**Chapter 3: Project Details**

* 1. **Title of the Project**

Hybrid Collaborative Filtering for Personalized Product Recommendation

* 1. **Problem Definition**

With the rapid growth of e-commerce products and digital content, it has become difficult for users to search for products they may like. Traditional recommendation systems either depend on historical user behaviour or the characteristics of the item, but both methods are limited. The limitations of collaborative filtering manifest as cold-start users; on the other hand, content-based filtering suffers from being too narrow. Thus, hybridizing a recommendation system is necessary to provide equilibrium that leads to personalized, relevant, and diverse recommendations.

* 1. **Scope and Objectives**

**The scope comprises of:**

* The use of simulated user ratings on IMDb-like movie data.
* Collaborative filtering via user-item matrices.
* Content-based filtering generation based on genre similarity.
* The combination of both methods to create hybrid recommendations.
* Automated visual and text report generation for analysis.

**Objectives:**

* Creating a recommendation system that is scalable and interpretable.
* Enhance accuracy using hybrid methods.
* Offer visual insights through EDA.
* Make the system applicable in various domains other than movies.
  1. **System Requirements**

**Software**: Python 3.13.3, IDE, ReportLab, Pandas, NumPy, Seaborn, Matplotlib, Scikit-learn.

**Hardware**: Any system with minimum 4GB RAM and an i3 processor or above.

**Dataset**: IMDb-like dataset containing titles, genres, years, and ratings.

* 1. **Architecture Diagram**

The architecture of the **Hybrid Collaborative Filtering Recommendation System** is modular and structured to simulate an end-to-end personalized recommendation workflow. The diagram can be divided into four primary layers:

**3.5.1 Data Ingestion Layer**

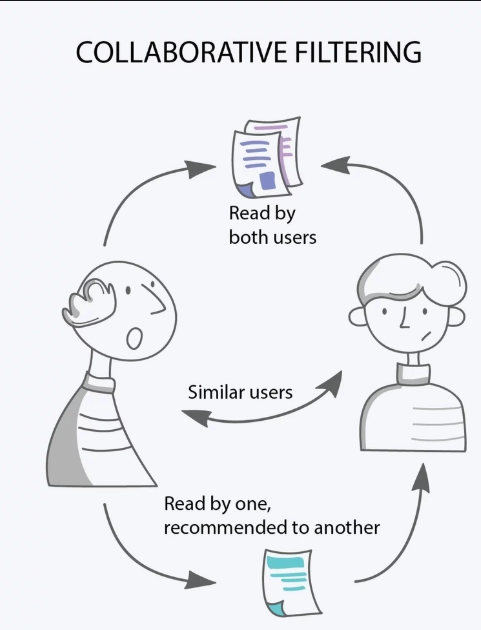
* **Source**: IMDb-style CSV file (imdb\_movie\_data\_2023.csv)
* **Loader**: Pandas is used to read and preprocess the dataset.

**3.5.2 Preprocessing Layer**

* Clean and normalize data (handle missing values, rename columns)
* Feature Engineering: Create duration, imdb\_rating, and decade
* Genre encoding for content-based filtering

**3.5.3 Recommendation Engine Layer**

* **Collaborative Filtering Module**
  + User-item matrix construction
  + Cosine similarity calculation
  + Score aggregation and ranking



**Figure 1 Collaborative Filtering**

* **Content-Based Filtering Module**
  + User genre profile creation
  + Cosine similarity with genre vectors
  + Recommendation based on content similarity

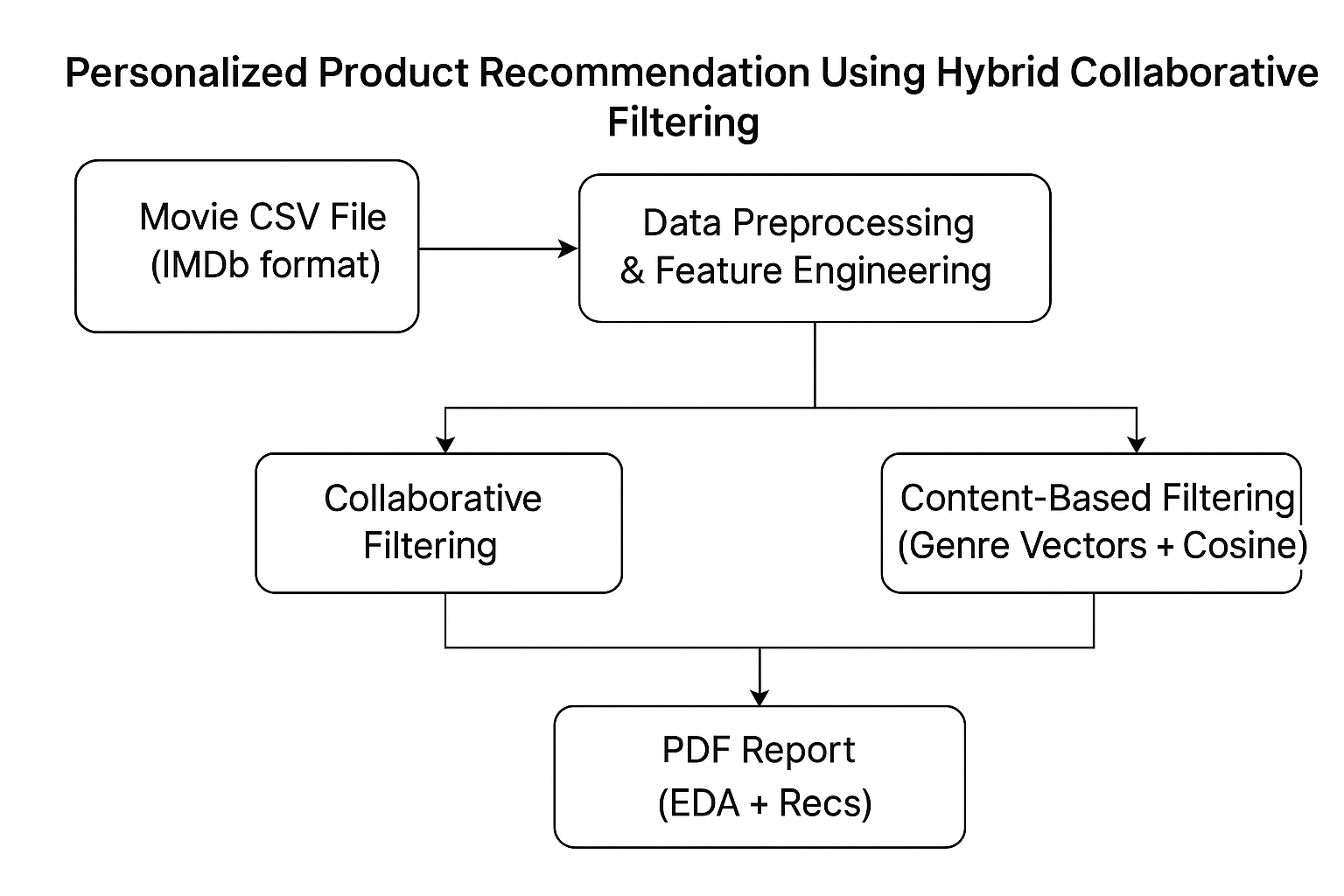
A cartoon of a person with a few articles

AI-generated content may be incorrect.

**Figure 2 Content-Based Filtering**

**3.5.4 Output & Report Generation Layer**

* Visualizations using matplotlib, seaborn (EDA)
* PDF Reports using ReportLab
  + EDA PDF
  + Recommendation report for each user
* CSV files for external analysis



**Figure 3: Diagram for Architecture**

* 1. **Data flow / UML Diagrams**

Data Flow Diagram (DFD – Level 1)

The DFD Level 1 diagram captures the interaction between different modules in a more process-oriented way.

Entities & Data Stores:

* External Entities:
  + User
  + Admin/Data Analyst
* Processes:
  + Load & Clean Movie Data
  + Generate User Ratings (Simulated)
  + Perform EDA
  + Compute Collaborative Recommendations
  + Compute Content-Based Recommendations
  + Generate PDF Reports
* Data Stores:
  + Movie Dataset
  + User Ratings Dataset
  + Processed Genre Features
  + PDF Reports

A diagram of a product recommendation

AI-generated content may be incorrect.

**Figure 4: DFD Diagram**

Diagram of a diagram of uml

AI-generated content may be incorrect.

**Figure 5: UML Diagram**

**Chapter 4: Implementation**

* 1. **Tools Used**

As discussed in Chapter 2, tools like Pandas, NumPy, Seaborn, and Scikit-learn were central to data analysis, visualization, and algorithm implementation.

* 1. **Methodology**

The implementation is divided into the following steps:

**4.2.1 Data Loading and Cleaning**

The movie data is loaded from a CSV file. Duplicate entries are removed, missing genres are filled with “Unknown”, and numerical features such as duration and imdb\_rating are generated for analysis.

**4.2.2 Exploratory Data Analysis (EDA)**

Various plots are generated to understand the distribution and relationships within the data. Bar plots, scatter plots, histograms, and heatmaps are included.

**4.2.3 Collaborative Filtering**

* A simulated set of users and movie ratings is created.
* A user-item matrix is built and filled with ratings.
* Cosine similarity is computed between items to find similar ones.
* Top N recommended movies are selected for each user.

**4.2.4 Content-Based Filtering**

* The genre of each movie is converted into binary feature vectors.
* For each user, movies with a rating ≥ 4 are used to compute a user profile.
* Cosine similarity is used to find movies that align with this profile.

**4.2.5 Hybrid Recommendation**

Both collaborative and content-based results are used to generate a recommendation report in PDF format using ReportLab.

* 1. **Modules / Screenshots:**

A diagram of a heatmap

AI-generated content may be incorrect.

**Figure 6 Heatmap of Duration & IMDB\_Rating**

A graph of a distribution of duration

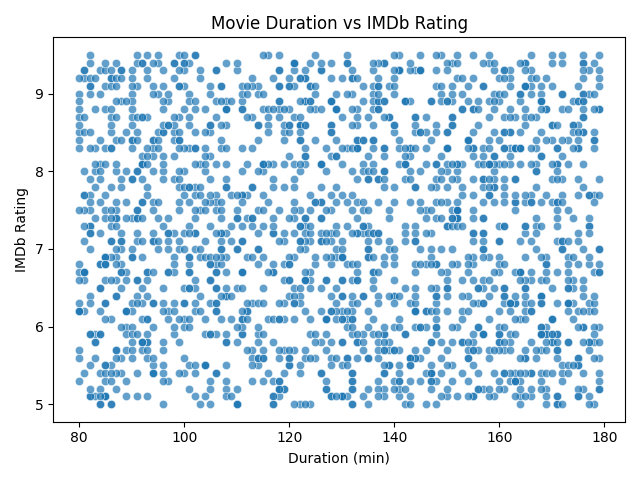
AI-generated content may be incorrect.

**Figure 7 Histogram of duration**

A graph of a distribution of imdb ratings

AI-generated content may be incorrect.

**Figure 8 Histogram of Rating**



**Figure 9 Scatterplot of Duration vs Rating**

A bar graph with different colored bars

AI-generated content may be incorrect.

**Figure 10 Bar plot of Genres**

A graph of a movie

AI-generated content may be incorrect.

**Figure 11 Movies Per Decade**

**4.4 Code Snippets:**

**4.4.1 Collaborative Filtering:**

def get\_collab\_recs(user\_id, n=5):

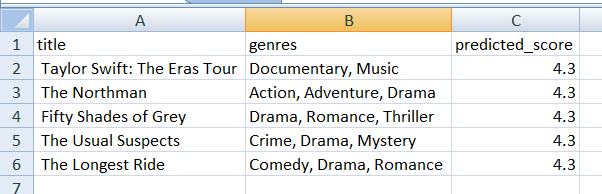
user\_ratings = user\_item\_matrix.loc[user\_id]

rated = user\_ratings[user\_ratings > 0]

scores = item\_similarity\_df[rated.index].dot(rated).div(item\_similarity\_df[rated.index].sum(axis=1))

recommendations = scores[~scores.index.isin(rated.index)].nlargest(n)

return df.loc[recommendations.index][['title', 'genres']]

****

**Figure 12 Collaborative Filtering**

**4.4.2 Content-Based Filtering:**

def get\_content\_recs(user\_id, n=5):

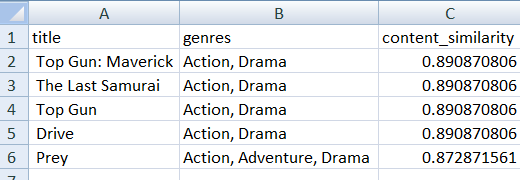
liked = ratings\_df[(ratings\_df['userId'] == user\_id) & (ratings\_df['rating'] >= 4.0)]

user\_profile = genre\_features.loc[liked['movieId']].mean().values.reshape(1, -1)

similarity\_scores = cosine\_similarity(user\_profile, genre\_features)[0]

df['content\_similarity'] = similarity\_scores

return df[~df['movieId'].isin(liked['movieId'])].nlargest(n, 'content\_similarity')



**Figure 12 Content-Based Filtering**

**4.4.3 EDA Report Generation:**

def generate\_eda\_pdf(filename="EDA\_Report.pdf"):

doc = SimpleDocTemplate(filename, pagesize=A4)

elements = [Paragraph("EDA Report", styles["Title"]), Spacer(1, 12)]

elements.append(Image("plots/top\_genres.png", width=400, height=250)

doc.build(elements)

A black text on a white background

AI-generated content may be incorrect.

**Figure 13 EDA Report**

**4.4.4 Recommendation Report Generation Function:**

def generate\_recommendation\_pdf(user\_id, collab\_df, content\_df, filename):

doc = SimpleDocTemplate(filename, pagesize=A4)

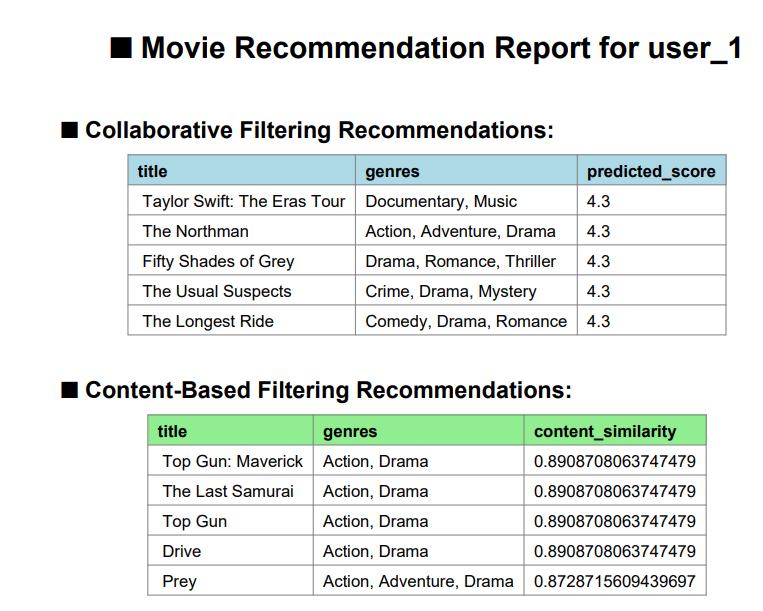
styles = getSampleStyleSheet()

elements = []

# Title

elements.append(Paragraph(f"🎬 Movie Recommendation Report for {user\_id}", styles["Title"]))

elements.append(Spacer(1, 12))



**Figure 14 Movie Recommendation Report**

**Chapter 5: Results and Discussion**

* 1. **Output / Report**

The outlined system provides:  
1. Collaborative filtering-based recommendations for each user (Top 5).

2. Content-Based filtering recommendations (Top 5).  
3. The consolidation of a formal report (PDF) for each user presenting the results in tabular format.  
4. Visualization through EDA stored as images and compiled into a separate EDA PDF.

* 1. **Challenges faced**

Cold Start Problem: New users have no data for collaborative filtering.

Limits on Data: The actual IMDb data was not employed and instead, simulated data must be produced to represent real behavior.

Balancing Two Methodologies: Careful tuning was required to ensure that the hybrid model did not too heavily weigh on one method.

* 1. **Learnings**

1. Abstract understanding of real-world recommendation systems.

2. Practical experiences through data visualization and documentation.

3. User preferences and feedback will improve recommendation accuracy.

4. Practical application of transforming the written output of code into formal documentation using ReportLab.

**Chapter 6: Conclusion**

This is an excellent project that has successfully implemented a Hybrid Collaborative Filtering Recommendation System. The system is an inclusive combination of both collaborative and content-based filtering to eliminate some of the most common recommendation problems such as cold start and narrow recommendations. Simulated user ratings and genre-based profiling provided a means for meaningful suggestions generated mimic the world's best e-commerce and entertainment platforms' personalization engines.

The following are what I learned from this project:

* A hybrid model is flexible and has higher accuracy.
* Visualizations give insights into the user preferences as well as the movie trends.
* Includes automatic pdf reports thus creating value for the system.

In the future, possible enhancements to the system will be incorporated with real-time feedback by users and more features like actors, directors, user demographics, and a combination of deep learning techniques for advanced personalization.

**References**

 Hybrid Recommender Systems: A Systematic Literature Review – A broad survey that categorizes hybrid methods (weighted, cascade, switching, etc.) and discusses challenges like cold-start and sparsity.

 AutoSVD++ (Hybrid Collaborative Filtering with Autoencoders) – Integrates deep learning (contractive auto-encoders) with matrix factorization for improved recommendation accuracy.

 Hybrid Collaborative Recommendation via Semi-AutoEncoder – Uses semi-autoencoder architecture to combine collaborative and content-based signals for better personalization.

 Hybrid CF with Double Neighbor Selection – Enhances accuracy by dynamically choosing neighbors based on user activity and trust, reducing cold-start issues.

 Hybrid MF + XGBoost Model (2024) – Combines matrix factorization with machine learning (XGBoost) using user/item features for higher-quality recommendations.