Indian Journal of Computer Science and Technology
https://www.doi.org/10.59256/indjcst.20250402029
Volume 4, Issue2 (May-August 2025), PP: 220-223.
www.indjcst.com
ISSN No: 2583-5300

AI-Powered Fashion Recommendation System for Smart Dress Rental Platforms

Pratiksha A. Chambhare[1], Monali N. Ninawe[2], Bhagyashree Kumbhare[3], Yamini Laxane[4]
[1,2]Students, MCA, Smt. Radhikatai Pandav College of Engineering, Nagpur, India.
[3]HOD, Department of MCA, Smt. Radhikatai Pandav College of Engineering, Nagpur, Maharashtra, India.
[4]Professor, Department of MCA, Smt. Radhikatai Pandav College of Engineering, Nagpur, Maharashtra, India.

To Cite this Article: *Pratiksha A. Chambhare[1], Monali N. Ninawe[2], Bhagyashree Kumbhare[3], Yamini Laxane[4], "AI-Powered Fashion Recommendation System for Smart Dress Rental Platforms", Indian Journal of Computer Science and Technology, Volume 04, Issue 02 (May-August 2025), PP: 220-223.*

Abstract: *This paper proposes a smart recommendation system for online dress rental platforms, leveraging Artificial Intelligence (AI) to enhance user personalization. It addresses the challenge user's face in selecting attire by providing intelligent, data-driven suggestions tailored to their preferences, body profile, and booking history. Using machine learning techniques such as collaborative and content-based filtering, the system suggests dresses that match user taste, improving the platform's usability, engagement, and conversion rate. The solution integrates seamlessly with existing web applications built using Java, JSP, and MySQL, and includes a micro service-based architecture for AI modules using Python. The outcome shows measurable improvements in user interaction and rental frequency.*

Key Words: *AI recommendation system, online dress rental, collaborative filtering, content-based filtering, Java, JSP, MySQL, user personalization, fashion tech.*

I.INTRODUCTION

Fashion rental services like Dazzle Wear provide users with a platform to rent attire for various occasions. However, users often struggle to find relevant dresses among hundreds of listings. Integrating AI into the rental system not only simplifies this process but also creates a personalized and engaging experience. Just like how Netflix or Amazon recommends content, fashion recommendations can streamline the user journey by offering styles aligned with user taste.

II.OBJECTIVES

- Develop an AI recommendation system to assist users in selecting outfits.
- Apply collaborative filtering and content-based filtering methods.
- Integrate AI logic with existing Java-based rental platform.
- Reduce browsing time and increase rental conversions.
- Provide a scalable and modular architecture for future improvements.

III.LITERATURE REVIEW

Recommendation systems are widely used in e-commerce. In fashion, platforms like Myntra and Rent the Runway implement AI to suggest dresses. Prior research emphasizes combining user-item interaction (collaborative filtering) with item properties (content-based filtering) to provide accurate suggestions. This paper adapts similar models to dress rental platforms.

IV.METHODOLOGY

4.1 Content-Based Filtering

- Compares dress features (color, type, tags) with the user's historical preferences.
- Uses TF-IDF encoding or one-hot encoding for categorical attributes.
- Computes similarity between dress vectors and user preference vector.

4.2 Collaborative Filtering

- Based on user-user or item-item interactions.
- Matrix Factorization (e.g., SVD – Singular Value Decomposition) is applied to reduce dimensionality and predict missing user-item ratings.
- Finds users with similar fashion behavior patterns and suggests dresses they liked.

4.3 Hybrid Filtering

- Combines content-based and collaborative filtering.
- Uses a weighted approach or model-level hybridization to balance both sources.
- Enhances cold-start problem handling and recommendation accuracy.

4.4 Demographic-Based Filtering (New)

- Incorporates demographic data (age, gender, body type) for recommendations.
- Rules are created based on common fashion preferences within user segments.
- Useful for new users without prior interactions (cold start).

4.5 Context-Aware Recommendation (New)

- Includes contextual information like:
- Event type (wedding, prom, cocktail party).
- Time of the year (seasonal fashion trends).
- Filters and re-ranks items that fit the specific context.

4.6 Popularity-Based Filtering (New)

- Uses booking counts, high ratings, and trending tags to promote widely liked dresses.
- Acts as a fallback mechanism when limited user data is available.

Similarity Metric

To compare users or items, the following similarity measures are used:

- **Cosine Similarity:**
  - Measures angle-based similarity between vectors (normalized).
  - Effective for high-dimensional data like tags and dress features.

- **Euclidean Distance:**
  - Computes the distance between two vectors.
  - Better suited for numeric attributes like ratings or size match scores.

- **Jaccard Similarity (New):**
  - Used for comparing binary tag sets or favorite item overlap.

Ranking and Recommendation Strategy

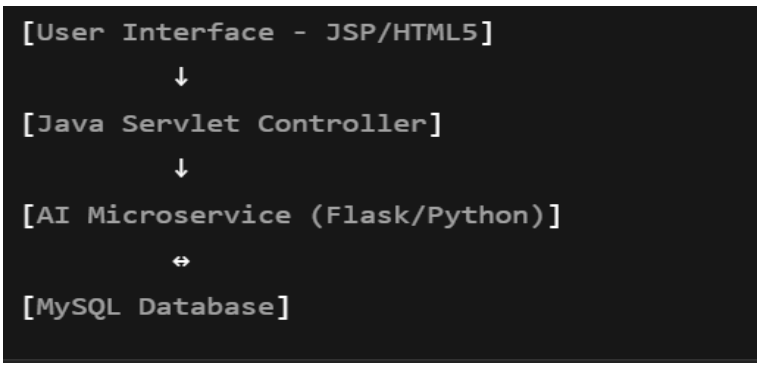
- **Top-N Ranking:**
  - Sorts items by predicted relevance score.
  - Recommends the top-N dresses based on user profile match.

- **Score Aggregation:**
  - Hybrid models average or weight scores from collaborative and content filters.

- **Diversity-Aware Re-Ranking (New):**
  - Avoids showing similar-looking dresses.
  - Encourages variety (e.g., different colors, styles) to increase engagement.

V.SYSTEM ARCHITECTURE

5.1 Architecture Diagram



[MySQL Database]

5.2 Component Details

- **Frontend**: HTML5, CSS3, Bootstrap, responsive layout.
- **Backend**: Java Servlets for session handling, requests.
- **AI Engine**: Flask-based Python app providing REST API for recommendations.
- **Database**: MySQL storing user profiles, dresses, and ratings.

VI.IMPLEMENTATION

6.2.1 Model Deployment via Flask API

- A recommendation model (built in Python) is trained offline using user interaction and dress metadata.
- The model is saved using joblib or pickle.
- A Flask API exposes an endpoint /recommend/<user_id> that returns a list of recommended dress IDs in JSON format.

Example Response:

json
CopyEdit

```
{
  "Recommendations": [101, 203, 145, 302]
}
```

6.2.2 API Communication from Java Servlet

- Java Servlets use HttpURLConnection or Apache HttpClient to make GET requests to the Flask API.
- JSON response is parsed using libraries like org.json or Gson.
- Recommended dress IDs are used to query dress details from the MySQL database.

6.2.3 JSP Page Rendering

- The recommendation list is passed to a JSP page using request.setAttribute().
- The page displays dress image, name, rating, and a “Rent Now” button.
- CSS3 and custom themes ensure consistency with DazzleWear’s branding.

6.2.4 User Feedback Collection

- Feedback is sent via AJAX to a Servlet endpoint (e.g., FeedbackServlet.java) and stored in a feedback table:
  - Fields: user_id, dress_id, feedback_type, timestamp

6.2.5 Feedback-Driven Re-ranking (New)

- User feedback is periodically analyzed to adjust recommendation scores.
- Positive feedback boosts relevance scores of similar items.
- Negative feedback reduces visibility of disliked dress types.

6.2.6 Scheduled Model Retraining (New)

- A Python script is scheduled (e.g., via cron) to retrain the model weekly or after significant interaction volume.
- The updated model is reloaded by the Flask API without restarting the service.

6.2.7 Caching Mechanism (New)

- Frequently accessed user recommendations are cached in memory (e.g., using Redis or in-JVM cache).
- Reduces API call load and speeds up page loading.

6.2.8 Error Handling and Logging (New)

- Flask API logs invalid user IDs or failed predictions.
- Java Servlet logs failed API calls or slow responses.
- JSP displays fallback content if recommendations are unavailable.

6.2.9 Security Measures (New)

- Only authenticated users can access /recommend/<user_id>.
- Flask API implements input validation to prevent attacks.
- API can be protected via tokens or rate-limiting

VII.FUTURE SCOPE

- Implement Deep Learning for outfit style prediction using CNNs on dress images.
- Use Augmented Reality (AR) for virtual try-on experience.
- Integrate Voice Search for accessibility.
- Enable Multilingual Support for broader reach.

- Deploy on cloud platforms like AWS for scalability.

VIII.CONCLUSION

The AI-powered fashion recommendation system significantly enhances user experience and rental efficiency. With reduced search time and better personalization, it transforms a traditional rental system into a smart, adaptive platform. The modular architecture ensures easy scalability and integration of future technologies like AR and deep learning.

References

1. *Ricci, F., Rokach, L., & Shapira, B. (2015). Recommender Systems Handbook. Springer.*
2. *Aggarwal, C. C. (2016). Recommender Systems: The Textbook. Springer.*
3. *Scikit-learn Documentation. (2024). https://scikit-learn.org*
4. *Flask Web Framework. (2024). https://flask.palletsprojects.com*
5. *Buyya, R. et al. (2013). Cloud Computing and Virtualization. Morgan Kaufmann.*