Innopolis University
Fall 2023
Practical Machine Learning and Deep Learning

Final Report

Movie Recommender System

December 2023

Author: Hamada Salhab
e-mail: h.salhab@innopolis.university
Link to GitHub repository: github.com/HamadaSalhab/movie-recommender-system

Introduction

This report details the development of a movie recommender system using the MovieLens 100K dataset. Our objective is to provide personalized movie suggestions to users, leveraging their demographic data and historical movie preferences. This document covers the data exploration phase, the construction and evaluation of a machine learning model, specifically LightFM, and the insights gathered during the training process.

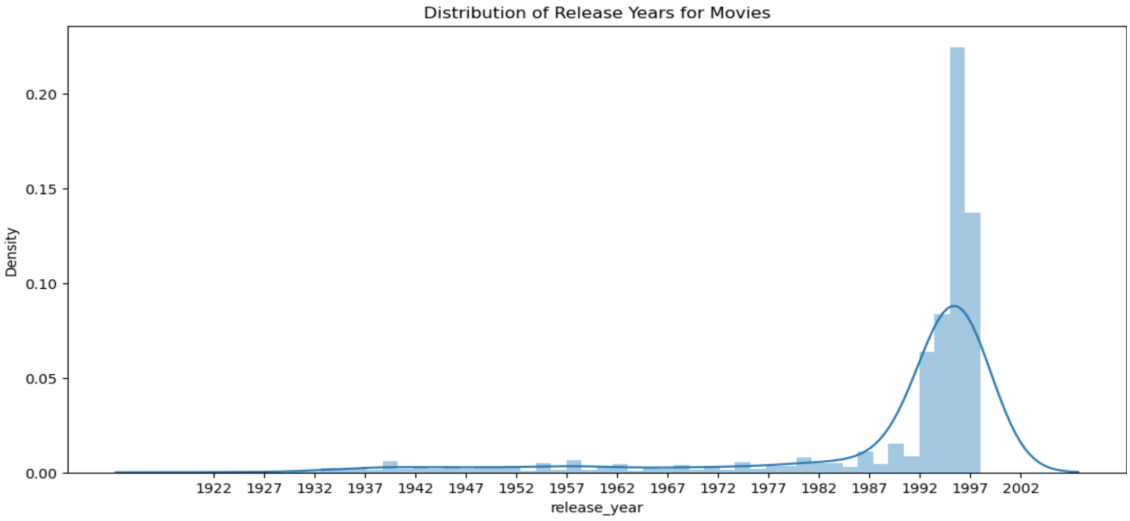
Contents

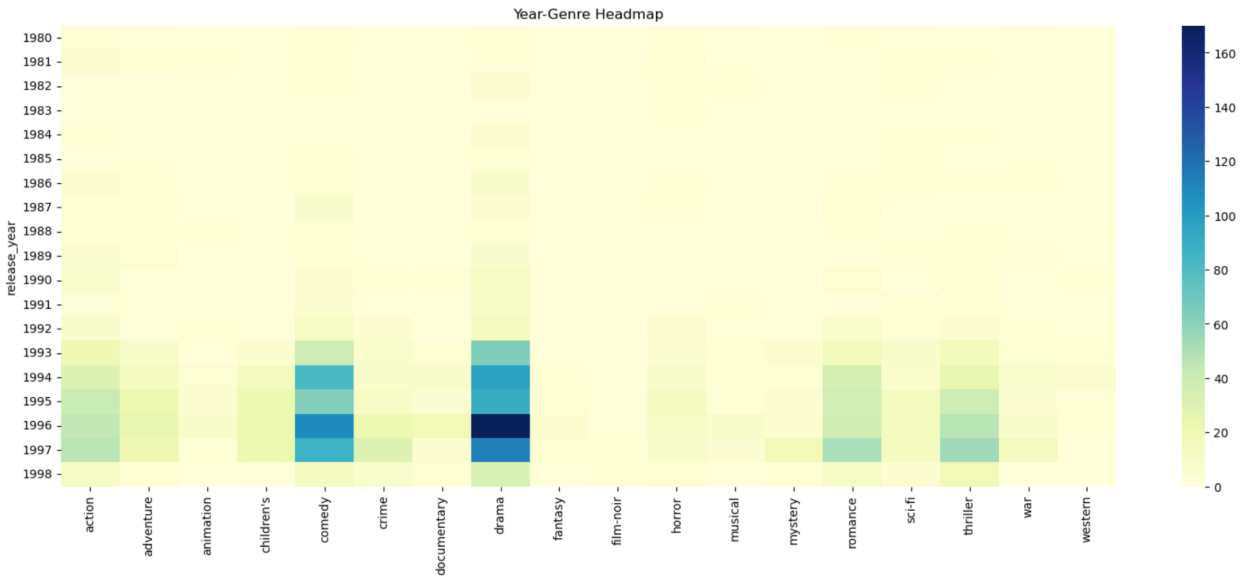
Data Exploration

The initial data exploration of the MovieLens 100K dataset provided multifaceted insights critical for the development of a robust movie recommender system. This analysis included an examination of the distribution of movie release years, genre prevalence across years, user demographics, occupation distribution, and the frequency and average of movie ratings.

1- Movie Release Years and Genre Trends

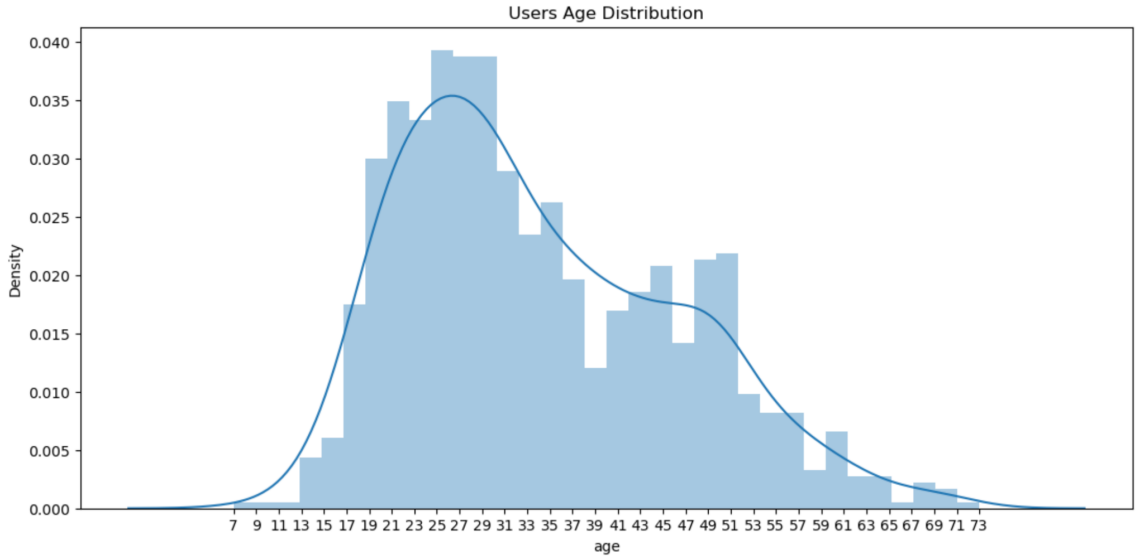
The dataset exhibited a considerable recency bias, with 78% of movies released between 1992 and 1998. This period dominance suggested a potential preference in data collection towards more contemporary titles. Further, a year-genre heatmap illuminated popular genres within this timeframe, indicating distinct shifts in genre popularity that could inform personalized genre recommendations.

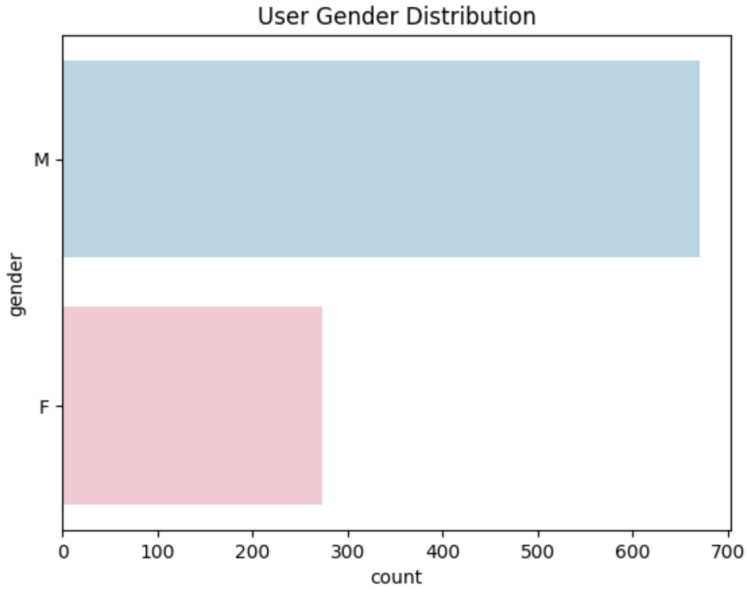




2- User Demographics

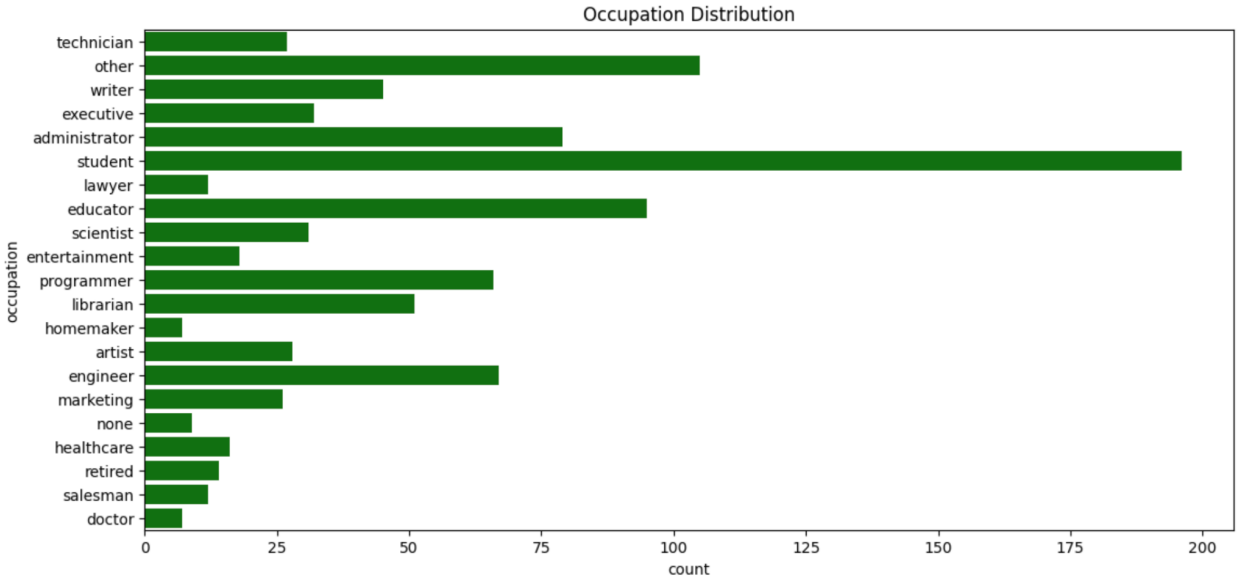
Analysis of user age revealed a median concentration in the 19-33 age bracket, encompassing 50% of the users, pointing towards a younger user base. The gender distribution skewed significantly towards male users, who constituted 71.05% of the user base, with a male-to-female ratio of approximately 2.45. This demographic slant could impact the recommendation model's predictions and necessitates a gender-sensitive approach to avoid bias.





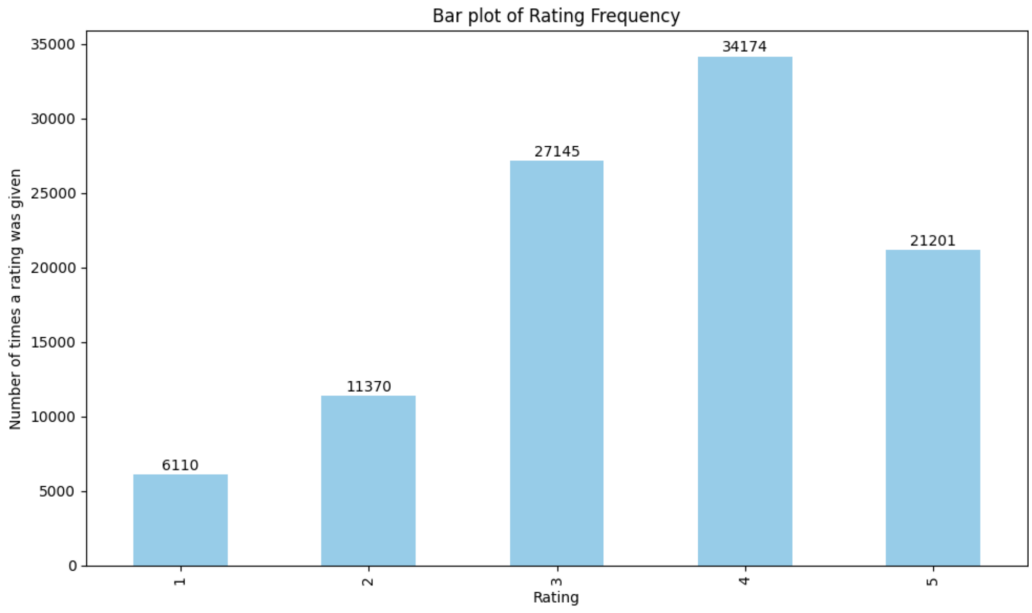
3- Occupation Distribution

The user occupation distribution provided a snapshot of the diverse professional backgrounds of the users, with students and educators being notably prevalent groups. Such diversity in occupation, coupled with the age and gender data, underscored the need for a recommendation system capable of understanding and catering to a wide range of user profiles.



4- Rating Patterns

A bar plot of rating frequency showed a higher occurrence of ratings at the upper end of the scale, with 4-star ratings being the most frequent. This suggests a user tendency to rate movies positively.



Moreover, an analysis of the most-rated movies, led by iconic titles like "Star Wars" and "Fargo," reflected the users' preference patterns, potentially guiding the popularity aspect of the recommendation algorithm.

Most rated movies:

movie_id	movie_name	rating
50	Star Wars	4.4
100	Fargo	4.2
181	Return of the Jedi	4.0
1	Toy Story	3.9
258	Contact	3.8
286	English Patient, The	3.7
300	Air Force One	3.6
121	Independence Day (ID4)	3.4
288	Scream	3.4
294	Liar Liar	3.2

Implications for the Recommender System

The insights derived from this exploratory analysis are crucial in shaping the feature selection and algorithmic strategy for the recommender system. The demographic leanings and rating tendencies present opportunities for personalized recommendation pathways, while the concentration of release years and genre trends could influence the system's predictive accuracy.

The recommender system's development, thus, incorporates a nuanced understanding of these initial findings, ensuring a tailored approach that reflects the rich tapestry of user preferences encapsulated in the dataset.

Solution Exploration

NOTE: If you're looking for the main technique that was used in this assignment (my main model for submission), please feel free to skip the first two bullet points.

In the quest to craft a movie recommender system that is both accurate and user-centric, we embarked on a journey exploring various recommendation techniques. Each method has its merits and potential applications, leading to a strategic decision-making process that culminated in the selection of a model that best fits our dataset's characteristics and our system's goals.

Initial Techniques

1- Collaborative Filtering (Item-Item) using Singular Value Decomposition (SVD):

NOTE: The relevant code for this technique can be found in "2.0-model-training-SVD.ipynb" notebook.

Our first approach utilized Collaborative Filtering with SVD, a popular technique for dimensionality reduction and latent feature extraction in recommendation systems. While SVD is effective in capturing underlying patterns in user-item matrices, it does not accommodate additional user or item metadata, which can enhance recommendation quality.

2- SVD++ using Surprise Library:

NOTE: The relevant code for this technique can be found in "2.1-model-training-SVD++.ipynb" notebook.

To leverage both explicit ratings and implicit feedback, we experimented with SVD++, an extension of SVD available in the Surprise library. SVD++ accounts for implicit interactions by incorporating them into the factorization process, potentially improving prediction accuracy.

3- LightFM:

NOTE: The relevant code for this technique can be found in "2.2-model-training-LightFM.ipynb" notebook.

After thorough experimentation, we settled on LightFM as our primary model. LightFM is a hybrid matrix factorization model that combines the principles of collaborative filtering and content-based recommendations. Its distinct advantages that influenced our decision are as follows:

- Hybrid Approach: LightFM can incorporate both user and item features into the recommendation process, enabling it to make predictions even when historical interaction data is sparse, a common scenario known as the cold start problem.

- Scalability and Flexibility: The model scales gracefully with the size of the dataset, managing a balance between performance and computational efficiency. This is particularly beneficial for extensive and growing datasets.

- Meta-Data Integration: Unlike traditional SVD, LightFM integrates metadata (such as user demographics or movie genres), which allows for richer user profiles and item descriptions, leading to more nuanced recommendations.

- Adaptability to Implicit Feedback: LightFM's ability to handle implicit feedback, like user clicks or views, alongside explicit ratings, allows for a fuller understanding of user preferences, an aspect crucial for personalized recommendations.

How LightFM Works

LightFM operates by learning latent embeddings for both users and items. For users, the embeddings encapsulate their preferences inferred from their features and interactions. For items, the embeddings represent their characteristics and how they relate to user preferences. The model then predicts a score for a user-item pair by calculating the dot product of their respective embeddings. These scores are used to rank items for each user, with higher scores indicating a stronger preference.

The model's training process involves optimizing these embeddings to maximize the accuracy of the predictions. Through a process of gradient descent and using loss functions like WARP (Weighted Approximate-Rank Pairwise), the model iteratively adjusts the embeddings to improve recommendation quality. The choice of loss function is pivotal, with WARP focusing on ranking the items correctly, which is particularly suitable for a recommender system where the order of recommendations is more important than the predicted ratings.

In conclusion, LightFM's hybrid nature, ability to leverage metadata, and adaptability to various types of feedback made it an ideal choice for our movie recommender system. It stands out as a versatile and powerful tool that aligns with our dataset's richness and our system's requirements to provide personalized movie suggestions.

Training Process

1- Preparing the Dataset

The initial step in our model's training process involved preparing the MovieLens 100K dataset to ensure compatibility with the LightFM model. This preparation included encoding categorical features, such as gender and occupation, using label encoding to convert these categories into a machine-readable format. Unique user and item identifiers were extracted to create a dataset specifically tailored for LightFM, which requires explicit user and item feature lists.

We then transformed these features into lists of strings, adhering to LightFM's input requirements. The dataset also included a rich set of movie metadata, such as genres and release years, which were crucial in adding context to the movies beyond mere ratings. Interaction data, representing the core user-movie interactions, was prepared by isolating user IDs, movie IDs, and corresponding ratings.

2- Model Training with Grid Search

To find the optimal set of hyperparameters for the LightFM model, we utilized a grid search strategy. This systematic approach involved iterating over a predefined range of values for the number of components (latent factors), learning rate, and epochs. The specific sets tried were:

- Number of Components: [5, 10, 20, 40] - representing the dimensionality of the feature embeddings.
- Learning Rate: [0.001, 0.01] - dictating the speed at which the model learns.
- Epochs: [5, 10, 50] - denoting the number of times the model would work through the entire training set.

Each combination of these hyperparameters was used to train a version of the LightFM model, which was then evaluated on a held-out test set derived from a 20% split of the interaction data.

Evaluation Metrics and Fine-Tuning

The model's performance was gauged using precision and recall at k (where k=10), alongside the F1-score, which balances the two. The grid search's objective was to maximize the F1-score, leading to the selection of the best hyperparameters. These best parameters provided a fine-tuned LightFM model that demonstrated superior performance on the test set. The final model was evaluated using the Area Under the Curve (AUC) score to assess its ranking capability.

Results and Discussion

The model training and hyperparameter tuning concluded with the identification of the optimal configuration for the LightFM model:

- **Best F1-Score:** 0.12 **-** This score, while modest, reflects a nuanced balance between precision and recall in the context of our dataset. Such a score could be attributed to the inherent challenges of the MovieLens 100K dataset, which, like many real-world datasets, is characterized by sparsity and a long-tail distribution of user interactions. This imbalance means that many items have very few ratings, making it challenging for the model to learn to recommend relevant items to users accurately.
- **Area Under The Curve Score:** 0.91 **-** This high score is indicative of the model's strong ability to rank relevant items higher than irrelevant ones. It suggests that while the precision and recall might be low, the model is still effective in distinguishing users' preferences.
- **Precision:** 0.13 **-** The relatively low precision could be due to several factors, including the model's tendency to recommend popular items that may not be as personalized. This can often happen when there is a bias towards more frequently rated movies, as is the case in many collaborative filtering systems.

- **Recall:** 0.11 - The recall score indicates that the model retrieves a small subset of all relevant recommendations. In a vast and diverse item space, capturing all relevant items is challenging, especially when recommendations are constrained to a top-10 list. This could be further impacted by the diversity of user interests and the limited amount of data on individual preferences.

- **Best Parameters:** {'n_components': 40, 'learning_rate': 0.01, 'epochs': 50} - These parameters reflect the most effective configuration after extensive experimentation. However, the choice of hyperparameters is a trade-off between model complexity and overfitting, and while these were the best within the search space, there is always the possibility that a more extensive search could yield better results.

The LightFM model, developed through a meticulous process, demonstrates an understanding of the MovieLens 100K dataset and is capable of delivering personalized recommendations. However, the modest precision, recall, and F1-scores highlight the complex nature of recommendation systems and the challenges inherent in predicting human preferences. These results underline the importance of continuous model iteration and the potential exploration of additional features or alternative modeling techniques to enhance the system's performance.

References

1. McKinney, W. et al. (2023). pandas. Retrieved from pandas.pydata.org.
2. Hunter, J. D. et al. (2023). Matplotlib. Retrieved from matplotlib.org.
3. Waskom, M. et al. (2021). seaborn. Retrieved from seaborn.pydata.org.
4. Harris, C.R. et al. (2022). NumPy. Retrieved from numpy.org.
5. Hug, N. (2023). Surprise: A Python scikit for recommender systems. Retrieved from surpriselib.com.
6. Kula, M. (2023). LightFM. Retrieved from LightFM documentation.