Movie Recommendation System Using Machine Learning

Taymaa Nasser 1222640, Ahmad Hamdan 1210241, Laith Nimer 1213046
Birzeit University
Faculty of Engineering & Technology Electrical
& Computer Engineering Department

Abstract*—This project focuses on the development of a movie recommendation system utilizing machine learning techniques implemented with scikit-learn. The system evaluates and compares the performance of three distinct classifiers: Decision Tree, Naïve Bayes, and K-Nearest Neighbors (KNN). A comprehensive design is used, encompassing sturdy data preprocessing steps via pandas. Each classifier undergoes a rigorous evaluation process, leveraging a well-defined test-train split of the dataset to measure performance. To assess and compare the models' effectiveness, performance metrics such as accuracy, precision, recall, and F1-score are calculated. The results highlight the strengths and weaknesses of each approach, offering valuable insights into their suitability for the recommendation task. This project demonstrates the practical application of machine learning in providing personalized recommendations while emphasizing the importance of systematic model evaluation.

I. Introduction

Movie recommendation systems have become integral to enhancing user experiences on platforms like Netflix, IMDb, and Amazon Prime Video. These systems provide personalized content, improving user satisfaction and platform engagement predicting what users might enjoy based on their viewing history, preferences, and other factors.

With the proliferation of digital content, developing efficient recommendation systems is critical to tackling challenges such as information overload and ensuring users are presented with relevant options. Traditional recommendation approaches relied on collaborative filtering or content-based filtering. However, these methods face limitations, including scalability issues and difficulty in capturing complex user-item interactions. This has paved the way for machine learning-based approaches, which offer improved performance and adaptability.

This study explores the use of machine learning algorithms, namely Naïve Bayes, Decision Trees, and K-Nearest Neighbors (KNN), to predict user preferences for movies. These algorithms were selected for their interpretability, computational efficiency, and suitability for various dataset characteristics. By leveraging a robust dataset and systematic preprocessing techniques, the study aims to evaluate the efficacy of these models in delivering accurate recommendations.

Furthermore, the project emphasizes the role of preprocessing, including encoding categorical features, normalizing numerical values, and splitting datasets into training and testing subsets. By simulating different user scenarios—such as preferences for similar or diverse movies—the study investigates how data characteristics influence model performance. The insights gained will inform the design of more robust and scalable recommendation systems in the future.

II. Methodology

II.I Dataset Description

The TMDB 5000 Movie Dataset from Kaggle was used. The dataset includes features such as genres, popularity, runtime, original language, production companies and user ratings. Preprocessing involved encoding categorical features with LabelEncoder, genre transformation with MultiLabelBinarizer, and numerical normalization using MinMaxScaler.

II.II Model Implementation

The movie recommendation system was implemented in Python, leveraging the scikit-learn library for its robust machine learning tools. Additionally, pandas was extensively used for preprocessing tasks such as handling missing data, transforming genres into lists, and merging encoded features. Three classifiers—Decision Tree, Naïve Bayes, and K-Nearest Neighbors (KNN)—were evaluated.

The dataset underwent comprehensive preprocessing, including genre encoding using MultiLabelBinarizer, normalization of numerical features with MinMaxScaler, and encoding of categorical variables using LabelEncoder. The data was split into features (X_data) and target labels (y_data) and further divided into training and testing sets with varying proportions.

II.II.1 Naïve Bayes: Assumes feature independence, using Gaussian distribution for numerical attributes. It evaluates the likelihood of features contributing to both classes and assigns the class with the highest posterior probability.

II.II.2 Decision Trees: Uses hierarchical splitting to maximize class separation, optimized with restricted depth and class balancing to prevent overfitting.

II.II.3 KNN: Predicts labels based on similarities in features using Euclidean distance. The number of neighbors (k) was tuned iteratively to balance overfitting and underfitting.

II.III Evaluation

To assess the system's performance, different dataset sizes (10,000, 50,000, and 100,000 samples) and test splits (20% and 30%) were experimented with. This resulted in six configurations, allowing for analysis of how dataset size and test proportions impacted model accuracy.

Each classifier underwent hyperparameter tuning for optimal performance. Performance metrics, including accuracy, precision, recall, and F1-score, were used for comparison. These metrics were calculated to evaluate the models' effectiveness in capturing patterns and generating accurate recommendations.

The assessment was conducted twice to evaluate how the correlation between a user's liked movies influenced performance:

- When users selected liked movies with significant similarities (e.g., shared genres, production companies, or language), the models performed better in predicting recommendations.
- When users' liked movies were diverse and less correlated, the models faced challenges, leading to reduced recommendation quality.

These experiments provided valuable insights into the impact of feature correlations, dataset size, and model design on recommendation system effectiveness. The results highlighted the strengths and limitations of each approach, informing the design of scalable and efficient systems.

III. Results

Table 1: Performance Metrics for Naive Bayes

Movies	Data Set	Test Split	Accuracy	Precision	Recall	F1
Non-Similar Movies	25,000	0.2	0.874	0.648	0.999	0.786
	50,000	0.2	0.731	0.464	1	0.633
	100,000	0.2	0.735	0.461	1	0.631
	25,000	0.3	0.873	0.644	0.995	0.783
	50,000	0.3	0.727	0.458	1	0.628
	100,000	0.3	0.737	0.465	1	0.635

Table 2: Performance Metrics for Decision Tree

Movies	Data Set	Test Split	Accuracy	Precision	Recall	F1
Similar Movies	25,000	0.2	0.989	0.889	1	0.941
	50,000	0.2	0.993	0.929	1	0.963
	100,000	0.2	0.995	0.949	1	0.973
	25,000	0.3	0.989	0.890	1	0.942
	50,000	0.3	0.992	0.926	0.999	0.961
	100,000	0.3	0.994	0.944	1	0.971
Non-Similar Movies	25,000	0.2	0.970	0.831	0.828	0.830
	50,000	0.2	0.973	0.881	0.816	0.847
	100,000	0.2	0.971	0.846	0.829	0.837
	25,000	0.3	0.971	0.843	0.823	0.833
	50,000	0.3	0.968	0.827	0.819	0.823

Table 3: Performance Metrics for KNN

Neighbors (k)	Data Set	Test Split	Accuracy	Precision	Recall	F1
3	25,000	0.2	0.927	0.831	0.861	0.846
	50,000	0.2	0.943	0.868	0.890	0.879
	100,000	0.2	0.955	0.895	0.912	0.903
	25,000	0.3	0.932	0.843	0.856	0.849
	50,000	0.3	0.941	0.866	0.881	0.874
	100,000	0.3	0.955	0.896	0.908	0.902
5	25,000	0.2	0.928	0.845	0.844	0.844
	50,000	0.2	0.942	0.877	0.872	0.875
	100,000	0.2	0.954	0.899	0.900	0.899
	25,000	0.3	0.926	0.839	0.837	0.838
	50,000	0.3	0.937	0.867	0.861	0.864
	100,000	0.3	0.952	0.898	0.896	0.897
7	25,000	0.2	0.922	0.833	0.832	0.833
	50,000	0.2	0.937	0.867	0.863	0.865
	100,000	0.2	0.953	0.899	0.892	0.896
	25,000	0.3	0.919	0.825	0.823	0.824
	50,000	0.3	0.933	0.855	0.853	0.854
	100,000	0.3	0.950	0.894	0.889	0.892

IV. Discussion

IV.I Naive Bayes

The Naïve Bayes classifier predicts whether a user will "like" a movie based on features like genres, runtime, popularity, and encoded categorical variables. Using Bayes' Theorem, it assumes feature independence, making calculations efficient. The Gaussian variant assumes numerical features follow a normal distribution, assigning the class with the highest posterior probability. While efficient and interpretable, this independence assumption limits performance with complex feature relationships. As shown in **Table 1**, Naïve Bayes performs well for similar movies, achieving near-perfect metrics due to strong feature correlations. However, for non-similar movies, high recall but low precision reflects over-prediction, reducing overall accuracy and F1 scores.

IV.II Decision Trees

The Decision Tree classifier predicts whether a user will "like" a movie based on features like genres, runtime, popularity, and language. By splitting data into subsets using metrics like Gini Impurity or Entropy, it creates a tree structure that is interpretable and efficient for both numerical and categorical data. To prevent overfitting, tree depth was restricted, and class balancing applied, ensuring accurate predictions and highlighting key features. As shown in **Table 2**, decision trees achieve high accuracy and recall for similar movies by capturing patterns in structured datasets. For non-similar movies, they maintain balanced performance, with strong F1 scores due to their ability to account for feature interactions. Decision trees outperform Naïve Bayes for diverse datasets, making them a reliable choice for mixed correlations.

IV.III KNN Clustering

The K-Nearest Neighbors (KNN) classifier predicts whether a user will "like" a movie by measuring feature similarities such as genres, runtime, and normalized attributes like popularity and vote averages. Using Euclidean distance, it identifies the k closest neighbors and classifies them based on their labels. As shown in **Table 3**, k values of 3, 5, and 7 were tested to balance overfitting and underfitting. KNN performs exceptionally well for similar movies, leveraging distance-based similarity to achieve high precision and recall. For non-similar movies, it slightly outperforms decision trees by maintaining a better precision-recall balance. While effective for diverse datasets, KNN's computational cost increases with dataset size due to distance calculations. Adjusting k fine-tunes performance, ensuring robust recommendations across different conditions.

IV.IV Comparing The Models

The results across the three models—Naïve Bayes, Decision Trees, and KNN—highlight their distinct strengths and limitations for movie recommendation tasks. Naïve Bayes performs exceptionally well for similar movies, achieving high accuracy, precision, and recall due to its reliance on feature independence. However, its performance declines for non-similar movies, with high recall but low precision indicating over-prediction, making it less suitable for datasets with interdependent features. In contrast, Decision Trees and KNN effectively handle non-similar movies, offering a balanced trade-off between precision and recall. Decision Trees leverage hierarchical splitting to capture feature relationships, resulting in strong F1 scores and balanced metrics for diverse datasets. KNN excels in distance-based similarity, achieving high precision and recall for similar movies while maintaining better performance than Naïve Bayes for non-similar movies. However, KNN's computational expense increases with larger datasets. Both Decision Trees and KNN provide robust predictions, but their need for computational resources and fine-tuning reflects the trade-offs between simplicity and adaptability in model selection.

V. Conclusion

This project explored the implementation and evaluation of three machine learning classifiers —Naïve Bayes, Decision Trees, and K-Nearest Neighbors (KNN)— for building a movie recommendation system. Each model exhibited distinct strengths and limitations, influenced by dataset characteristics, model design, and hyperparameter tuning. Naïve Bayes, while efficient and interpretable, struggled with feature interdependencies, performing best on datasets with highly correlated features but underperforming for diverse datasets. Decision Trees demonstrated versatility and reliability, excelling in both similar and non-similar movie scenarios due to their ability to capture feature relationships through hierarchical splitting. KNN emerged as the most effective for datasets with nuanced feature similarities, particularly in cases with similar user preferences, though it required higher computational resources for large datasets. The analysis highlighted that the dataset size, test split, and model hyperparameters significantly impacted performance metrics, including accuracy, precision, recall, and F1-score. Among the configurations, KNN with k=3 consistently delivered the highest performance, underscoring its adaptability to varying dataset complexities. This project emphasizes the importance of tailoring model selection and hyperparameter optimization to the specific characteristics of the recommendation task, providing a foundation for enhancing user experiences in personalized content systems.

References

[1]https://www.kaggle.com/datasets/asaniczka/tmdb-movies-dataset-2023-930k-movies?resource=download

[2 https://scikit-learn.org/dev/index.html

[3] https://www.ijnrd.org/papers/IJNRD2304674.pdf

[4]https://www.irjmets.com/uploadedfiles/paper/issue_12_december_2022/32414/final/fin_irjmets1671794536.pdf