Netflix Show Ratings Prediction Using Machine Learning Algorithms

A. Anandhi[1], M. Monica[2]
[1]MCA., M.Phil., Assistant professor, Department of Master of Computer Applications
[2]MCA., Christ College of Engineering and Technology Moolakulam, Oulgaret Municipality, Puducherry – 605010

***Abstract*—The exponential growth of digital streaming services has produced massive datasets, offering a unique opportunity to apply machine learning for analyzing viewer-centric metrics. Since IMDb ratings are widely regarded as a benchmark for content quality and audience reception, predicting these scores has become a significant objective in data analytics. This research introduces a web-based application built on a Flask architecture that classifies Netflix shows into IMDb rating categories using an integrated dataset. By utilizing a built-in repository of over one hundred records, the system operates independently of external data uploads. The methodology involves rigorous preprocessing including data cleaning and normalization followed by the implementation of Decision Tree, Random Forest, and Artificial Neural Network (ANN) models. These algorithms categorize content into four tiers: Excellent, Good, Moderate, and Poor. Beyond prediction, the application features secure role-based access, interactive performance dashboards, and automated PDF report generation for comprehensive analytical insights.**

***Index Terms*—Machine Learning, Netflix, IMDb Ratings, Flask, Random Forest, ANN, Web Application.**

I. INTRODUCTION

The digital landscape has been transformed by streaming giants like Netflix, which now host an overwhelming volume of multimedia content. This abundance often leads to "choice paralysis" for users struggling to identify high-quality programming among thousands of titles. Conventional methods of evaluating content, such as manual reviews or browsing external forums, are frequently inconsistent and time-consuming.

This project addresses these inefficiencies by providing an automated machine learning solution to predict IMDb rating categories based on historical metadata. By transforming raw attributes into actionable insights, the system assists in identifying potential content quality before it undergoes extensive public review. The resulting full-stack application utilizes a Python-based Flask backend, an SQLite database for secure storage, and a dynamic frontend built with HTML, CSS, and JavaScript.

1.1. PROBLEM STATEMENT

Most content evaluation remains reactive, occurring only after a show has garnered significant feedback. There is a distinct lack of proactive, automated tools capable of leveraging metadata such as cast, genre, and director to forecast a show's success. Furthermore, many existing analytical tools lack the enterprise-grade features required for professional study, such as role-based security and automated reporting.

1.2. OBJECTIVES

The primary goal of this research is to build an intelligent, data-driven platform that streamlines the rating prediction process.

Specific aims include:
Developing predictive models based on attributes like Genre, Duration, Year, and Country [14] .
Evaluating and comparing the performance of ANN, Random Forest, and Decision Tree algorithms.
Designing a user-centric interface featuring Chart.js for real-time data visualization and reporting.

II. LITERATURE REVIEW AND RELATED WORK

Predicting cinematic success has transitioned from basic statistical modeling to sophisticated machine learning frameworks.

2.1 STATISTICAL AND REGRESSION APPROACHES

Early studies by Dhir, Raj [1], and Mundra et al. [2]utilized Linear and Logistic Regression to forecast box office performance. While effective for financial forecasting, these linear models often fail to capture the complex, non-linear relationships inherent in categorical data like actor popularity or genre combinations.

2.2 MACHINE LEARNING IN CONTENT ANALYSIS

Recent academic shifts have favored classification algorithms. Latif and Afzal [7]emphasized the critical role of feature selection in predicting popularity. Research indicates that Random Forest models are particularly effective at managing high-dimensional data and mitigating the overfitting issues common in single decision trees [5], [15]. Additionally, deep learning via Artificial Neural Networks (ANN) has proven successful in identifying intricate patterns in viewer behavior for multiclass classification [10], [16].

2.3 GAP ANALYSIS

Despite the availability of these algorithms, they are rarely integrated into a cohesive, deployable web environment. Most research focuses on model accuracy in isolation rather than end-to-end solutions that offer secure authentication and professional reporting tools. This project bridges that gap by delivering a production-ready full-stack application.

III. METHODOLOGY

The system follows a logical pipeline: data collection, preprocessing, model training, and web integration.

3.1. DATA COLLECTION

The application uses a pre-integrated Netflix dataset of over 100 records, including features such as release year, country, and duration.

3.2. DATA PREPROCESSING

To ensure model integrity, the data undergoes several stages [12]:

Data Cleaning: Removal of null values and duplicates to eliminate bias.

Quartile Categorization: The 0-10 IMDb scale is binned into four classes: Excellent (7.4–10.0), Good (6.7–7.3), Moderate (5.8–6.6), and Poor (0.0–5.7).

Encoding and Normalization: Categorical strings are converted to numerical formats via Label Encoding, and features are scaled using StandardScaler to improve algorithmic convergence [3].

3.3. MACHINE LEARNING ALGORITHMS

Three specific models were implemented for comparison:

Artificial Neural Network (ANN): A Keras-based model featuring an input layer (36 neurons), two hidden layers (16 and 26 neurons), and a 4-neuron output layer (Figure 1). It utilizes ReLU and Softmax activations and is optimized over 100 epochs using the Adam optimizer [10]

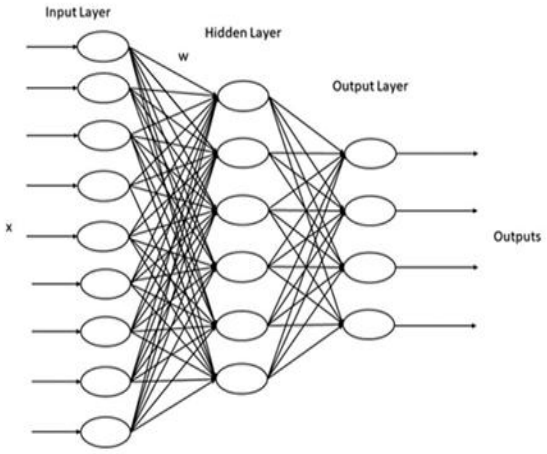

Figure 1. ANN

Random Forest: An ensemble method that aggregates multiple decision trees to enhance accuracy and reduce variance. [4], [5].

Decision Tree: Utilized for its interpretability and its ability to split data based on information gain.

In this study, the Decision Tree served as the performance benchmark.

IV. SYSTEM ARCHITECTURE AND IMPLEMENTATION

The application is built on a modular architecture to maintain scalability (Figure 2). The backend is powered by Python and Flask for API management and model serving, while the frontend utilizes HTML5, CSS3, and JavaScript. SQLite provides a lightweight solution for storing user data and prediction history.

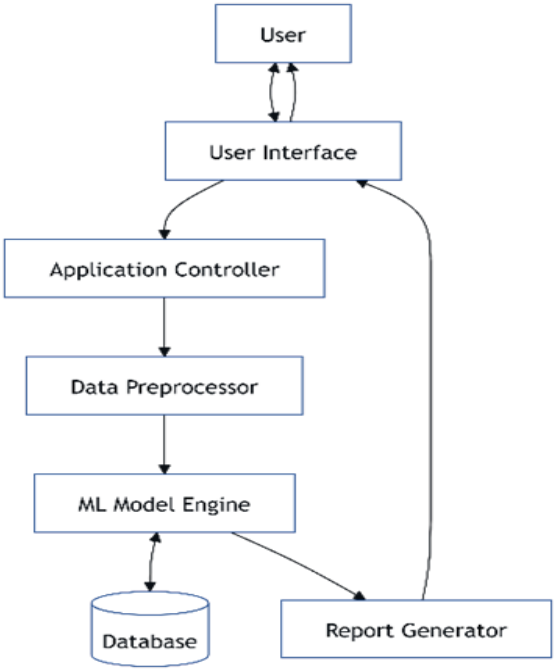

Figure 2. System Architecture

Key modules include:

Authentication: Features role-based access control (Admin vs. User) with secure password hashing.

Prediction: Processes user input through the model.pkl file to return instant results (Figure 3).

Reporting: Uses ReportLab to generate downloadable PDF summaries of analytics (Figure 4).

Title	Genre	Rating	Category
Breaking Bad	Crime	9.5	Excellent
When They See Us	Drama	8.8	Excellent

Figure 3. Sample Prediction

Total Shows	115
Genres	14
Countries	12
Avg Rating	7.43
Highest Rating	9.5
Lowest Rating	3.8

Figure 4. Generation Of Detailed PDF Reports Summarizing Analytical Results

V. RESULTS AND ANALYSIS

Models were tested using an 80/20 split. A dashboard was developed to allow users to visualize dataset distributions and compare model accuracies through dynamic charts. Users can input show details including Title, Genre, and Country to receive immediate classification [8] The results indicate that for this specific dataset, the Decision Tree model achieved the highest accuracy shown in (Figure 5).

Model	Accuracy (%)
Decision Tree	43.48%
Random Forest	30.43%
Artificial Neural Network	30.43%

Figure 5. Model Accuracy

VI. CONCLUSION AND FUTURE SCOPE

This research successfully demonstrates a full-stack application that integrates machine learning into a practical web interface for content evaluation. While the Decision Tree currently leads in performance, future iterations will focus on:
Real-time API Integration: Connecting directly to TMDB or Netflix for live data.
Sentiment Analysis: Using NLP to factor social media trends into the prediction logic [14]

Cloud Deployment: Utilizing Docker and AWS for global accessibility.

REFERENCES

[1] R. Dhir and A. Raj, "Movie Success Prediction using Machine Learning Algorithms and their Comparison," in 2018 First International Conference on Secure Cyber Computing and

Communications (ICSCCC), Jalandhar, India, 2018, pp. 385–390.
[2] S. Mundra, A. Dhingra, A. Kapur, and D. Joshi, "Prediction of a movie's success using data mining techniques," in Smart Innovation, Systems and Technologies, vol. 104, Springer, 2019, pp. 219–227.
[3] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd ed., O'Reilly Media, 2019.
[4] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[5] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
[6] Netflix Technology Blog, "How Netflix Uses Machine Learning and AI," Netflix TechBlog, 2024.
[7] M. H. Latif and H. Afzal, "Prediction of Movies popularity Using Machine Learning Techniques," in 2016 International Conference on ICE Cube, Quetta, Pakistan, 2016.
[8] S. Raschka and V. Mirjalili, Python Machine Learning, 3rd ed., Packt Publishing, 2019.
[9] A. Zanasi, N. F. F. Ebecken, and C. A. Brebbia, "A Data Mining Approach to Analysis and Prediction of Movie Ratings," WIT Transactions on ICT, vol. 33, p. 434, 2004.
[10] F. Chollet, Deep Learning with Python, 2nd ed., Manning Publications, 2021.
[11] K. Reitz and T. Schlusser, Flask Web Development, 2nd ed., O'Reilly Media, 2018.
[12] W. McKinney, Python for Data Analysis, 2nd ed., O'Reilly Media, 2017.
[13] J. Yan, Z. Zhang, and H. Dong, "AdaDT: An adaptive decision tree for addressing local class imbalance," Applied Intelligence, vol. 51, no. 7, 2021.
[14] W. R. Bristi et al., "Predicting IMDb Rating of Movies by Machine Learning Techniques," in 2019 ICCCNT, Kanpur, India, 2019.
[15] V. Gupta et al., "Predicting attributes-based movie success through ensemble machine learning," Multimed Tools Appl, vol. 82, 2023.
[16] A. El-Banbi et al., "Artificial Neural Network Models for PVT Properties," PVT Property Correlations, pp. 225-247, 2018.