Report

Movie Success Prediction and Sentiment Analysis App

Introduction

The movie industry generates massive amounts of data across genres, revenue, audience reviews, and performance. Predicting whether a movie will be a *Hit*, *Flop*, or *Average* can assist producers, marketers, and audiences in understanding potential outcomes even before release. This project focuses on developing an interactive web application that allows users to:

- Search for movies
- View detailed metadata
- Submit review comments for sentiment prediction
- Predict the commercial outcome of movies using trained ML models

Abstract

This application combines data scraping, natural language processing, machine learning, and database management to create an end-to-end movie prediction and analysis system. It integrates two key models:

1. A **sentiment classifier** to analyze user comments.
2. A **hit/flop/average predictor** that classifies movies based on popularity, revenue, and budget.

It uses real-time data fetched from **OMDb API** and **TMDb API**, then stores it in a structured **SQLite database**. All functionalities are tied together via an interactive **Streamlit web interface**.

Tools & Technologies Used

Category	Tool / Library
Programming Language	Python
Web Interface	Streamlit

Data APIs	OMDb API, TMDb API
ML Model	Random Forest Classifier ,Logistic regression
Data Preprocessing	Pandas, NumPy, ast
Sentiment Analysis	TextBlob / Custom Classifier
Visualization	Streamlit Image + Cards
Database	SQLite
File Storage	.pkl for models, .py for script, and .ipynb for model building

Steps Involved in Building the Project

1. **Data Collection**
    - Movie details and posters were fetched using OMDb and TMDb APIs.
    - Genre, ratings, budget, revenue, and director information were stored in a local SQLite DB.
    - To train the model, a 2-dataset was used from Kaggle, one for sentiment analysis and another to predict movie Success
2. **Preprocessing**
    - Genres were converted into list to display properly
    - Missing data was handled with defaults or dropped where necessary.
3. **Model Training (Offline)**
    - A Random Forest Classifier was trained using historical movie data for success prediction, and a logistic regression model was used to predict the sentiment of.
    - A logistic regression model was used to classify the sentiment..
4. **Sentiment Analysis**
    - Users can write a review; it’s classified as Positive or Negative using a sentiment model.
    - Sentiment scores are stored in a separate DB table (Score).
5. **Prediction & UI Display**
    - Genre-wise movie predictions (Hit/Flop/Average) are displayed with posters.

    - Reviews update score counters, and users get visual feedback.

6. **Integration & Testing**
    - Models were saved as .pkl and loaded into the app.
    - The complete system was tested for various genre-based predictions and review sentiment.

📌 Conclusion

This project successfully demonstrates how various technologies can be integrated to build an intelligent, interactive movie analytics platform. It enables users to:

- Access detailed movie info
- Predict box-office performance
- Submit feedback in natural language
- View sentiment-based top movie rankings

Such a system can be extended for real-time industry insights, recommendation systems, or production planning tools.
