Movie Rating Classification using Data Science Workflow

This document presents a complete data science workflow applied to analyze and classify movies based on their IMDb ratings using Python. The IMDb Most Popular Movies (2006–2016) dataset is used to understand key factors influencing movie ratings and to build a classification model.

Dataset Description

Dataset: IMDb Most Popular Movies (2006–2016)
Source: Kaggle (PromptCloudHQ)
The dataset contains information such as movie ratings, runtime, votes, revenue, metascore, and genres.

Step 1: Load & Explore Dataset

- The dataset is loaded using the pandas library.
- Initial rows are printed to understand the structure of the data.
- The number of rows and columns is checked.
- Data types of each column are examined.
- Missing values are identified to plan cleaning steps.

Step 2: Data Cleaning

- Irrelevant columns such as movie titles are removed.
- Rows with excessive missing values are dropped.
- Remaining missing values are handled using appropriate strategies such as removal or imputation.

Step 3: Target Label Creation

A new binary target column named **label** is created:
• High Rated: Rating ≥ 7
• Low Rated: Rating < 7
This column is used as the target variable for classification.

Step 4: Feature Engineering

- Categorical features such as Genre are encoded using one-hot encoding.
- Numerical features like Runtime, Votes, Revenue, and Metascore are selected.
- All features are converted into a model-ready numerical format.

Step 5: Exploratory Data Analysis (EDA)

- Rating vs Votes: Higher votes generally correlate with higher ratings.
- Rating vs Revenue: High-rated movies tend to generate more revenue.
- Rating vs Runtime: Moderate runtimes often receive better ratings.
- Genre distribution by label shows certain genres dominate high-rated movies.

Step 6: Train–Test Split

The dataset is split into training and testing sets:
• Training set: 80%
• Testing set: 20%

Step 7: Model Selection & Training

Logistic Regression is chosen as the classification model due to its simplicity and interpretability. The model is trained on the training dataset using the engineered features.

Step 8: Model Evaluation

- Accuracy is used to measure overall correctness.
- Precision, Recall, and F1-score evaluate classification quality.
- A confusion matrix is used to visualize prediction performance.

Step 9: Interpretation & Insights

- Votes and Revenue are strong predictors of movie ratings.
- Movies with higher metascores are more likely to be classified as high-rated.
- The model achieves reasonable accuracy, showing that ratings can be predicted using basic features.

Conclusion: This project demonstrates a complete data science pipeline from data exploration to model evaluation, highlighting how movie ratings can be effectively analyzed and classified using Python.