Netflix Movie Data Analysis & Machine Learning Implementation

Objective

This project explores Netflix movie data to uncover trends in genres, popularity, and viewer ratings. The analysis aims to answer key questions related to movie trends while setting the foundation for predictive modelling using machine learning.

Data Preprocessing

- **Dataset Loaded:** mymoviedb.csv
- **Missing Values Handling:** Dropped NaNs.
- **Date Processing:** Converted Release_Date from object type to datetime format, then extracted the year.
- **Feature Engineering:**
  - Dropped irrelevant columns: ['Overview', 'Original_Language', 'Poster_Url']
  - Transformed the Genre column from a string into multiple categories using .explode()
  - Categorical conversion of Vote_Average into four labels:
    - not_popular, below_average, average, popular

Exploratory Data Analysis (EDA)

1. Most Frequent Genre

- The **Drama** genre appears the most frequently, covering **14%** of all movies.
- Seaborn catplot used to visualize the genre distribution.

2. Highest-Voted Genres

- About **25.5%** of the dataset consists of movies categorized as **popular**.
- **Drama** is again the most highly rated genre (18.5% of all popular movies).

3. Highest & Lowest Popularity Movies

- **Most Popular Movie:** *Spider-Man: No Way Home* (Genres: Adventure, Science Fiction).
- **Least Popular Movie:** *The United States, Thread* (Genres: Music, Drama, War, Sci-Fi, History).

4. Year with Most Movies Released

- **Year 2020** had the highest number of movies released.

Machine Learning for Personalized Recommendations

Predictive Modeling:

To build an intelligent recommendation system, the following machine learning models can be implemented:

Supervised Learning for Popularity Prediction

- Predict a movie's success based on past trends using:
    - **Random Forest, Decision Trees, or XGBoost**
    - Features: Genre, Vote_Average, Release_Date, Popularity

Model Training & Evaluation:

- Used **Decision Tree, Random Forest, and XGBoost**.
- Evaluated models using **Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 Score**.

Explanation of Accuracy Evaluation:

In this project, we're predicting **movie popularity** as a **regression problem** (continuous value prediction). Since it's not a classification task, accuracy isn't the right metric. Instead, we evaluate model performance using the following metrics:

1. **Mean Absolute Error (MAE)**:
    - Measures the average absolute difference between actual and predicted values.
    - Lower MAE = Better predictions.
2. **Mean Squared Error (MSE)**:
    - Similar to MAE but squares the differences, penalizing larger errors more.
    - Lower MSE = Better model fit.
3. **R^2 Score (R-Squared)**:
    - Represents how well the model explains variance in the data (ranges from **0 to 1**).
    - **1.0** → Perfect predictions.
    - **0.5** → Model explains 50% of variance.
    - **0.0** → Model performs no better than predicting the mean.

How It Works:

- We **train the models** using X_train and y_train.
- We **predict** movie popularity using X_test.
- We compare predictions (y_pred) with actual values (y_test).