Delivering Personalized Movie Recommendations Using AI

GitHub link: https://github.com/Thulasimathi26/Data-Science.gitS

1. Problem Statement

In todays content-rich world, movie streaming platforms are often overwhelmed with large libraries of content, making it difficult for users to find movies they enjoy. Personalized movie recommendation systems aim to solve this problem by tailoring recommendations based on individual user preferences.

The goal of this project is to develop an AI-powered movie recommendation system that predicts and suggests movies a user is most likely to enjoy. The system will leverage user data, such as past viewing history, ratings, demographic information, and movie features (genre, director, actors, etc.), to make these predictions.

2. Project Objectives;

* The objective should be clear and precise.

* What exactly do you want to achieve in the project?

* The objective should be quantifiable or assessable in some way.

* How will you measure the success of the objective?

3. Flowchart of the Project Workflow

| Data Collection |

| Data Preprocessing & learning |

| Feature Engineering & Selection |

| Algorithm Selection |

| (Collaborative Filtering, SVD, |

| Content-Based, Hybrid) |

| Model Training & Evaluation |

| (Cross-validation, Hyperparameter |

```
|     Tuning, Metrics Calculation)          |
|    Model Interpretation & Analysis        |
| (Feature Importance, SHAP, LIME)          |
|       Deployment & Feedback               |
```

4. Data Description

Dataset Name: Student Performance Data Set

Source: UCI Machine Learning Repository

Type of Data: Structured tabular data

Records and Features: 395 student records and 33 features (numeric + categorical)

Target Variable: G3 (final grade, numeric)

Static or Dynamic: Static dataset

Attributes Covered: Demographics (age, address, parents education), academics (G1, G2, study time), and behavior (alcohol consumption, absences)

Dataset Link: Student Performance - UCI Machine Learning Repository

5. Data Preprocessing

1. Handling Missing Values Fill in, drop, or impute missing data using methods like mean, median, or interpolation.
2. Data Cleaning Remove duplicates, correct inconsistent formatting, and fix data entry errors.
3. Data Transformation Normalize or standardize numerical features to ensure uniform scales.
4. Encoding Categorical Variables Convert text-based categories into numerical values using one-hot or label encoding.
5. Outlier Detection and Removal Identify and handle outliers using statistical methods or visualization techniques.

6. Exploratory Data Analysis (EDA)

Univariate analysis

Histogram Shows the frequency distribution

Box Plot Highlights median, quartiles, and outliers

Summary Statistics Mean, median, mode, standard deviation

Bivariate and Multivariate Analysis

Scatter Plot: Shows the relationship between two continuous variables.

Visualizes the distribution of a numerical variable for different categories.

Key Insights

Correlation Identification: Bivariate analysis helps identify the strength and direction of the relationship between two variables (e.g., strong positive correlation between age and income).

Outliers: Visualizations like scatter plots or box plots can reveal outliers or extreme values that might distort the relationship between variables.

7. Feature Engineering

Feature Creation: Generating new features from existing ones that may enhance model performance. This can include domain-specific features or mathematical transformations (e.g., calculating age from a birthdate).

Handling Missing Values: Filling in missing data through imputation techniques (mean, median, mode) or creating an indicator variable to flag missing values.

Categorical Encoding: Converting categorical variables into numeric representations using methods like one-hot encoding, label encoding, or target encoding to make them usable in machine learning models.

8. Model Building

Algorithms used

Linear Regression: Simple linear relationship between input and output.

Ridge/Lasso Regression: Linear regression with L2 (Ridge) or L1 (Lasso) regularization to prevent overfitting.

Model selection

Problem Type and Data Characteristics

Model Complexity and Interpretability

Training Set split

Used to train the machine learning model. The model learns the patterns, relationships, and parameters from this subset of the data.

Used to evaluate the models performance on unseen data to check how well it generalizes to new, real-world data.

Residual plots: A common train-test split is 80% for training and 20% for testing, though other ratios like 70%-30% or 90%-10% can also be used based on the size of the dataset.

For smaller datasets, cross-validation may be used to maximize the training data.

9. Visualization of Results & Model Insights

Feature importance:

The Project Manager oversees the projects progress, sets timelines, and ensures communication among team members and stakeholders. They manage the scope and ensure that the project aligns with business objectives.

Model comparison

Define the project scope, goals, and timelines.

Allocate resources and manage team dynamics

Residual plot:

Communicate with stakeholders and ensure that the project meets business needs.

Ensure that deadlines and milestones are met.

User testing:

Integrated model into a Gradio interface to test predictions by inputting feature values

10. Tools and Technologies Used

Programming language: python3

Notebook Environment: Google Colab

Key Libraries:

pandas, numpy for data handling

matplotlib, seaborn, plotly for visualizations

scikit-learn for preprocessing and modeling

11. Team Members and Contributions

Data Cleaning: (Vignesh.P)

1. Handling Missing Values: Fill or remove missing data using techniques like mean imputation or deletion.

2. Removing Duplicates: Identify and remove duplicate records to ensure data consistency.

Exploratory Data Analysis (EDA): (Ragul.S)

1. Data Visualization: Using charts like histograms, box plots, and scatter plots to understand data distributions and relationships.

2. Summary Statistics: Calculating measures like mean, median, and standard deviation to summarize data features.

Feature Engineering: (Yashwanth.J)

1. Creating New Features: Deriving new variables from existing ones to capture hidden patterns (e.g., extracting year from a date).

2. Handling Categorical Variables: Converting categorical data into numerical format using techniques like one-hot encoding or label encoding.

Model Development: (Vishwa.V)

1. Choosing the Right Algorithm: Selecting a suitable machine learning model based on the problem type (e.g., classification, regression).

2. Training the Model: Using the training dataset to train the model and adjust its parameters.

Documentation and Reporting:

1. Summarizing the Process: Clearly documenting the methodology, data sources, and steps taken during data preprocessing, modeling, and evaluation.

2. Presenting Key Findings: Highlighting the main insights, model performance, and business implications in a clear and concise manner.