PREDICTING IMDb SCORES

DONE BY :

- Natheesh N
- Roshan Kumar A
- Sakthi Vihaas M
- Vigneshwar S N
- Swadithyan B

Introduction

In Phase 1, we laid the foundation for our IMDb Movie Score Prediction project by defining the problem, conducting design thinking, and charting the course for our journey. Now, as we step into Phase 2, we are poised to transform those designs into a practical, functioning solution.

This phase marks a critical juncture in the project's lifecycle. Here, we take our well-considered design and put it into innovation, turning concepts into reality. It's where the rubber meets the road, and our plans manifest into a robust IMDb movie score prediction system.

Step 1: Data Acquisition and Preparation

1.1 Data Collection

- **Data Sources:** Identify and collect data from various sources, including IMDb for movie scores, genre databases for genre information, premiere date databases, and language sources.
- **Data Integration:** Integrate data from multiple sources into a unified dataset.

1.2 Data Cleaning and Preprocessing:

- **Data Cleaning:** Remove duplicates, handle missing values, and address outliers.
- **Feature Engineering:** Create new features, if necessary, and transform existing ones.
- **Data Split:** Divide the data into training, validation, and testing sets.

Step 2: Model Development and Training

- Once the data has been collected, cleaned, and prepared, the next crucial step is training the machine learning model. This phase involves selecting an appropriate algorithm, preparing the data for training, and optimizing the model’s performance.

1. Algorithm Selection

- Choosing the right machine learning algorithm is fundamental to the success of the prediction model. For our IMDb movie score prediction system, regression algorithms are suitable since we are dealing with predicting continuous numerical values (IMDb scores). Several algorithms can be considered:

Linear Regression: A basic regression algorithm that assumes a linear relationship between features and target variable.

Random Forest: An ensemble learning method that combines multiple decision trees to improve accuracy and handle complex relationships.

Gradient Boosting: Another ensemble method that builds multiple weak learners sequentially, where each one corrects the errors of its predecessor, leading to higher accuracy. The choice of algorithm can significantly impact the model's accuracy, and it might be beneficial to experiment with multiple algorithms to find the best performer.

2. Data Preparation

- **2.1 Feature Scaling**

Features often have different scales, which can affect the performance of certain machine learning algorithms.Common techniques include Min-Max scaling (scaling features to a specific range) or Z-score normalization (scaling features to have a mean of 0 and a standard deviation of 1). Scaling ensures that all features contribute equally to the model’s predictions.

2.2 Data Splitting

The available data is typically divided into three subsets:

- **Training Data:** Used to train the model.
- **Validation Data:** Used to tune hyper parameters and evaluate different models during development.
- **Testing Data:** Kept separate and used only after the model is finalized to evaluate its real-world performance.
- A common split ratio might be 70% for training, 15% for validation, and 15% for testing.

3. Model Training

- After preparing the data, the selected algorithm is trained on the training dataset. During this process, the
- algorithm learns the patterns and relationships within the data.

3.1 Hyperparameter Tuning

- Most machine learning algorithms have hyper parameters that need to be tuned for optimal performance.
- Hyper parameters are configuration settings that are not learned from the data but have a significant impact on the algorithm’s behavior.
- Techniques like grid search or random search can be employed to find the best combination of hyper parameters.

3.2 Cross-Validation

- Cross-validation is a technique used to assess how well a model will generalize to an independent dataset.
- Common methods include k-fold cross-validation, where the dataset is divided into k subsets, and the model is trained and evaluated k times, each time using a different subset for evaluation and the remaining k-1 subsets for training.
- This helps in obtaining a more robust evaluation of the model's performance.

4. Model Evaluation

4.1 Performance Metrics

Several metrics can be used to evaluate the model's performance in regression tasks:

- **Mean Absolute Error (MAE):** Represents the average of the absolute errors between the predicted and actual values.
- **Root Mean Square Error (RMSE):** Measures the square root of the average of squared differences between predicted and actual values.
- **α R-squared (R^2) Score:** Indicates the proportion of the variance in the dependent variable that is predictable from the independent variables.

- The choice of metric depends on the specific requirements of the application. For our IMDb movie score prediction system, a low MAE and RMSE are desirable, indicating that the predicted scores are close to the actual scores.

5. Model Optimization and Iteration

- Based on the evaluation results, the model might need further optimization. This could involve revisiting feature selection, trying different algorithms, or collecting additional data for specific features. The process of optimization and iteration continues until the model meets the desired accuracy and performance goals.
- The Notebook for the above Process is given below:
- https://colab.research.google.com/drive/1xd4BpYmgKtsllrZjkWdRzzk-MRKtuxKj?usp=sharing

- This document outlines the detailed steps and tasks involved in transforming the design from the previous phase into a functional IMDb movie score prediction system. Each step is critical to the success of the project and should be executed meticulously. Regular feedback loops and continuous improvement are key to ensuring that the system meets its objectives and evolves with changing user needs and data.

THANKYOU !