PREDICTING IMDB SCORES - PHASE 5

PROBLEM STATEMENT:

The problem is to develop a machine learning model that predicts IMDb scores of movies available on Films based on features like genre, premiere date, runtime, and language. The objective is to create a model that accurately estimates the popularity of movies, helping users discover highly rated films that match their preferences. This project involves data preprocessing, feature engineering, model selection, training, and evaluation.

DESIGN THINKING PROCESS:

Design Thinking is a human-centered, iterative problem-solving approach that prioritizes understanding user needs, challenging assumptions, and redefining problems to identify alternative strategies and solutions.

Design Thinking is applied in predicting IMDb scores by leveraging a user-centric approach to understand, develop, and refine models. Here's how we have applied it:

DESIGN THINKING:

Data Source: Utilize a dataset containing information about movies, including features like genre, premiere date, runtime, language, and IMDb scores. We used Netflixoriginals.csv for our project which has features such as

Title	Genre	Premiere	Runtime	IMDB Score	Language

Data Preprocessing: Clean and preprocess the data, handle missing values, and convert categorical features into numerical representations.

Feature Engineering: Extract relevant features from the available data that could contribute to predicting IMDb scores.

Model Selection: Choose appropriate regression algorithms (e.g Linear Regression, Random Forest Regressor) for predicting IMDb scores.

Model Training: Train the selected model using the preprocessed data.

Evaluation: Evaluate the model's performance using regression metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared.

PHASES OF DEVELOPMENT:

PHASE 1:

In the initial phase of our project, the focus lies in the meticulous definition of the problem statement in conjunction with the implementation of a comprehensive design thinking process.

PHASE 2:

- In the second phase we put our design into innovation to solve the problem.
- We started exploring various regression models like gradient boosting and neural networks.
- We implemented the same for our project and witnessed the efficiency of both the models.

PHASE 3:

- In the third phase we started building our model.
- Before we did data cleaning and preprocessing to remove the outliers and null values from the dataset.

We performed univariate and bivariate analysis

Univariate analysis:

Univariate analysis is a statistical method used to describe and understand the distribution, characteristics, and patterns of a single variable in a dataset.

So, we used it to replace null values in each feature with NaN.

Also used functions like info(), desc() to display the information of the features.

Bivariate analysis:

The bivariate analysis involved the examination of the relationship between two distinct features, one of which was the IMDb score, to discern its characteristics in relation to the other feature

PHASE 4:

This is our final model building phase where we trained several regression models like,

- Linear regression
- Decision tree
- Random Forest Regression
- Gradient Boosting

Of which we selected the gradient boosting model which gave the least error than the other models.

PHASE 5:

Final phase which we are just now doing is this documentation which summarizes our project , model selection and evaluation.

ABOUT THE DATASET : NETFLIXORIGINALS.CSV

This dataset consists of all Netflix original films released as of June 1st, 2021. Additionally, it also includes all Netflix documentaries and specials. The data was web scraped off of this Wikipedia page, which was then integrated with a dataset consisting of all of their corresponding IMDB scores. IMDB scores are voted on by community members, and the majority of the films have 1,000+ reviews.
The features of this dataset are

Title	Genre	Premiere	Runtime	IMDB Score	Language

DATA PREPROCESSING :

- Various preprocessing techniques such as filling missing values, scaling and outlier detection have been done.
- Firstly, the false and missing values are replaced. In scaling, we transform the features of the dataset in such a way that the ranges of all values are similar.
- Outlier detection is where deviations of data points are detected in the dataset values.

MODEL TRAINING :

The various regression models used here are:

- Linear Regression is where change in the independent variable is associated with the constant change in the dependent variable.
- Random Forest Regression is used for regression tasks, where the goal is to predict a continuous numeric output rather than discrete classes.
- Decision Tree is similar to random forest for predicting the IMDB scores.
- Gradient Boosting Regression is a machine learning technique used for regression problems, particularly when predicting continuous numerical values

REGRESSION ALGORITHM:

Linear Regression:

- Firstly, we import the necessary libraries.
- We evaluate the variables X_train, X_temp, y_train, y_temp.
- Similarly, we also evaluate the variables X_val, X_test, y_val, y_test.
- Now, we create and train the model using linear regression.
- We make predictions on the validation set using the variable y_val_pred.
- Finally, we evaluate the model and print the Mean Squared Error and the R squared.

Random Forest Regression :

- Firstly, we import the necessary libraries.
- We evaluate the variables X_train, X_temp, y_train, y_temp.
- Similarly, we also evaluate the variables X_val, X_test, y_val, y_test.
- We create and train the random forest model.
- Now, we make predictions on the validation set using the variable y_val_pred.
- Finally, we evaluate the model and print the Mean Squared Error and the R squared.

Decision Tree:

- Firstly, we import the necessary libraries.
- We evaluate the variables X_train, X_temp, y_train, y_temp.
- Similarly, we also evaluate the variables X_val, X_test, y_val, y_test.
- We create and train the decision tree model.
- Now, we make predictions on the validation set using the variable y_val_pred.
- Finally, we evaluate the model and print the Mean Squared Error and the R squared.

Gradient Boosting:

- Firstly, we import the necessary libraries.
- We evaluate the variables X_train, X_temp, y_train, y_temp.
- Similarly, we also evaluate the variables X_val, X_test, y_val, y_test.
- We create and train the gradient boosting model.
- Now, we make predictions on the validation set using the variable y_val_pred.
- Finally, we evaluate the model and print the Mean Squared Error and the R squared.

***INFERENCE*:**

Of all the models, the least mse is given by **Gradient Boosting Model (comparatively)**. **So it is best to select the gradient boosting model.**

EVALUATION METRICS :

- We import the necessary libraries.
- We calculate the Root Mean Square Error.
- Then, we calculate the R^2 score using the variables ytest and Ypred1 and print the result.
- We calculate the best hyperparameters and train the model with it and make predictions on the test set.
- Finally, we calculate and print the Root Mean Squared Error, Mean Squared Error and the R squared.
- Finally, we create a bar chart to visualize the errors.
- **The evaluation is done for all the four models.**

Model Selected :

GRADIENT BOOSTING REGRESSION

****