PREDICTING IMDb SCORES - Phase 2

1. NEURAL NETWORKS
2. GRADIENT BOOSTING

PROBLEM:

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.

NEURAL NETWORKS:

- We import the required libraries first (Keras, Tensorflow).
- Then we separate the required columns with some variables x and y.
- We split the data into train and test data.
- The variables are standardized using StandardScalar().
- The variables are then fit into the train and test variables.
- To build the neural network we use the Keras library.
- The model is trained using the training data(train x, train y).
- Evaluate the model's performance on the test set using the evaluate() method.
- The trained model is used to make predictions on the test data(x_test) and store the predictions in the variable y_pred.

GRADIENT BOOSTING:

Import necessary libraries including Pandas, NumPy, and various modules from scikit-learn, such as train_test_split, GradientBoostingRegressor, mean_absolute_error, mean_squared_error, r2_score, LabelEncoder.
These libraries are essential for data manipulation, model building, and evaluation.

Preprocessing the Data:

- It selects the columns 'Genre', 'Runtime', and 'Language' as features (X) and 'IMDBScore' as the target variable (y).

Label Encoding:

- Categorical variables 'Genre' and 'Language' are encoded using label encoding. This step converts categorical values into numerical values, which can be used as features for the machine learning model.

Train-Test Split:

- The dataset is split into training and testing sets using train_test_split. 80% of the data is used for training, and 20% is reserved for testing. The random_state parameter is set to ensure reproducibility.

Standardization:

- Standardize the features using StandardScaler(). Standardization scales the features to have a mean of 0 and a standard deviation of 1, which can help certain machine learning algorithms, including gradient boosting.

Building and Training the Gradient Boosting Regressor:

- A Gradient Boosting Regressor model is created with specific hyperparameters, such as the number of estimators (trees), learning rate, and maximum depth. The model is then trained on the training data.

Making Predictions:

- The trained model is used to make predictions on the test data, resulting in 'y_pred,' which contains the predicted IMDb scores.

Model Evaluation:

- The code calculates and prints Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2) to evaluate the performance of the regression model.
    - MAE measures the average absolute difference between the actual and predicted values.
    - MSE measures the average squared difference between the actual and predicted values.
    - R^2 is a measure of how well the model fits the data, with values closer to 1 indicating a better fit.