PREDICTING IMDB SCORES- PHASE 4

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 data, runtime and language. The objective is to create a model that accurately estimates the popularity of movies, helping user discover highly rated movies that matches their preferences. This project involves data preprocessing, feature engineering, model selection, training and evaluation.

FEATURE ENGINEERING:

i) IMPUTATION:

- Firstly, we import the necessary libraries (SimpleImputer from sklearn.impute).
- We call the SimpleImputer() function an set the strategy attribute as "mean".
- Next, we create the imputed dataframe from the old dataframe(df2).
- Finally, we print the imputed dataframe.

ii) OUTLIERS:

- Firstly, we import the necessary libraries(numpy).
- We take the sample dataset and calculate the first quartile (Q1) and third quartile (Q3).
- We calculate the Interquartile range(IQR).
- We define the lower and upper bound to identify the outliers.
- Then we detect and handle the outliers.
- Finally, we print the outliers and the modified dataset.

iii) LOG TRANSFORMATION:

- Firstly, we import the necessary libraries(numpy).
- We apply log transformation as np.log(data).
- Then we print the original and the log transformed data.

iv) ONE HOT ENCODING:

- Firstly, we import the necessary libraries.
- We make a copy of the dataset.
- We apply one hot encoding on the copied dataframe.
- We print the dataframe after applying one hot encoding.

v) SCALING:

- Firstly, we import the necessary libraries.
- We convert the dataset to numpy format and we reshape it.
- We apply Min-Max scaling.
- Then we print the resulting dataset.

vi) NORMALIZATION:

- We take the sample data, convert it to numpy format and reshape it.
- We initialize the StandardScaler.
- We fit and transform the data and normalize it.
- We print the normalized data.

vii) STANDARDIZATION:

- We take the sample data and initialize StandardScaler.
- We fit and transform the data and standardize it.
- We print the standardized data.

viii) PLOTS FOR FEATURE ENGINEERING:

- Firstly, we import the necessary libraries.
- The plots that are to be plotted for feature engineering are,

1. Histogram
2. Scatter Plot
3. Box Plot
4. Bar Plot (for categorical data)
5. Time Series Plot (Assuming a time series dataset.)
6. Pair Plot (For a selection of features).
7. Scatterplot Matrix
8. Feature Density Plot
9. Correlation Matrix Plot
10. PCA Projection Plot

MODEL TRAINING:

i) 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 valiation set using the variable y_val_pred.
- Finally, we evaluate the model and print the Mean Squared Error and the R squared.

ii) 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.
- Then we perform feature scaling.
- We create an train the decision tree model with adjuste hyperparameters.
- Now, we make predictions on the validation set using te variable y_val_pred.
- Finally, we evaluate the model and print the Mean Squared Error and the R squared.

iii) RANDOM FOREST MODEL:

- 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 an 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.

iv) GRADIENT BOOSTING MODEL:

- Firstly, we import the necessary libraries.
- We evaluate the variables X_train, X_temp, y_train, y_temp.
- Standardizing the features is optional but it can help gradient boosting.
- We build and train the gradient boosting model.
- Now, we make predictions and evaluate the model.
- Finally, we print the Mean Absolute Error, Mean Squared Error and the R squared.

v) 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:

- 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 isualize the errors.