Predicting Future Amazon Movie Reviews

Project Objective:

The goal of this project was to predict star ratings for Amazon movie reviews by analyzing various attributes of each review.

DataSet Description:

The dataset included three CSV files: train.csv, test.csv, and sample.csv. The train.csv file contains detailed information about each Amazon movie review, with columns like Id, ProductId, UserId, HelpfulnessNumerator, HelpfulnessDenominator, Time, Summary, Text, and Score. The test.csv and sample.csv files contain only the Id and Score columns.

Data Loading and Processing:

The first step was to load all files into Google Colab for processing. To handle missing data, any null values in the Helpfulness columns were replaced with zero, and any null values in the Text column were filled with empty strings to ensure no missing values.

Feature Engineering:

For feature engineering, I transformed the Text data using TfidfVectorizer, converting it into a matrix of TF-IDF features with a maximum of 5,000 terms. This was followed by applying Principal Component Analysis (PCA), which reduced the dimensionality of the TF-IDF matrix to 100 principal components. This reduction maintained relevant information from the text data while minimizing computational complexity.

Data Transformation Modes:

I set up two modes for data transformation: Fit and Transform. In Fit mode (fit=True), I trained a new TfidfVectorizer on the Text column, created a TF-IDF matrix, and applied PCA to reduce it to 100 components. In Transform mode (fit=False), I reused the trained TfidfVectorizer and PCA models, ensuring that new data is transformed in the same feature space. After transformation, the PCA components were converted into a DataFrame with columns named pca_0 to pca_99, which was then concatenated with the original data (excluding the Text column). This final DataFrame contained all original columns, the new Helpfulness feature, and the PCA-transformed text features, making it ready for predictive modeling.

Data Preparation and Splitting:

Next, I checked if preprocessed versions of the data (X_train.csv and X_submission.csv) existed. If they did, they were loaded; otherwise, I generated these files. To do this, the add_features_to function was run on trainingSet (with fit=True), and a merged DataFrame was created from the transformed train and testingSet. After dropping duplicate columns, the X_train and X_submission data were saved to Google Drive. The data was then split into training and testing sets using train_test_split, with 75% of data used for training and 25% for testing.

Model Selection:

For model training, I considered two models: Random Forest and Logistic Regression. Random Forest is well-suited for handling a mix of feature types and reducing overfitting in large datasets, as it combines the outputs of multiple decision trees for improved accuracy. Logistic Regression, a simpler model, is often used as a baseline, especially when interpretability and understanding feature impact on predictions are important.

Hyperparameter Training:

A parameter grid (param_grid) was created for tuning the logistic regression model, setting values for C (inverse regularization strength) and max_iter (maximum iterations for model convergence). RandomizedSearchCV was then used to select the best hyperparameters for the logistic regression model through cross-validation. The best model was chosen and evaluated on the test set, achieving an accuracy score based on the percentage of correct predictions.

Model Evaluation:

Finally, a confusion matrix was created to assess the classifier's performance visually. This matrix shows the proportion of true and misclassified instances by normalizing counts by the actual class size, which helps highlight performance across all classes. The matrix was plotted as a heatmap, where each cell shows the normalized value of each class prediction.

This process resulted in a structured, streamlined workflow for predicting star ratings based on Amazon review attributes, leveraging both Random Forest and Logistic Regression models for optimal accuracy and interpretability.