Tarek Mourad (U30214686)
CS506 MIDTERM
03/27/2023

Introduction:

> **The goal of this competition is to predict the star rating associated with user reviews from Amazon Movie Reviews using the available features.**
>
> **You may use any methods except neural networks / deep learning on this assignment**

Feature Extraction:

Explanation:
This code performs feature extraction on review data for a machine learning model. The process(df) function takes a Pandas dataframe df as input and returns a modified version of the dataframe with additional columns representing extracted features.
The first step in the feature extraction process is dropping irrelevant columns such as 'ProductId' and 'UserId'. Next, a new column 'HelpfulnessRatio' is created which represents the ratio of helpfulness of a review.
The 'Time' column is converted to datetime format and new columns 'Year', 'Month', 'Day', and 'DayOfWeek' are created to represent time-based features.
Additional features are created such as 'ReviewLength', 'NumExclamation', 'NumCaps', and 'CapsRatio' by performing operations on the 'Text' column of the review data. Finally, a new column 'Review' is created by concatenating the 'Summary' and 'Text' columns.
A correlation matrix is calculated to identify relevant features for the target variable 'Score'. Columns with a correlation score greater than or equal to 0.08 are selected as relevant features. The relevant features, along with the 'Id', 'Score', and 'Review' columns, are included in the modified dataframe returned by the function.
The modified training and testing dataframes are then stored in CSV files named 'X_train.csv' and 'X_test.csv', respectively. The testing data is obtained by merging the modified training data with the original test data based on the 'Id' column.

Results:
Relevant columns with corr_matrix score >= 0.08 are: ['HelpfulnessDenominator', 'HelpfulnessRatio', 'Year']

Model Creation:

Explanation:
This code creates a machine learning model to predict scores for some type of reviews, and evaluates the model's performance using various metrics and visualizations.
The process can be broken down into the following steps:

1. Import necessary libraries: This code imports several libraries including pandas, numpy, scikit-learn, matplotlib, seaborn, and pickle. These libraries are used for data manipulation, machine learning modeling, visualization, and saving/loading models.
2. Load training set with new features into DataFrame: The code loads the training set data into a pandas DataFrame.
3. Replace missing values with an empty string: This code replaces any missing values in the 'Review' column with an empty string.
4. Split training set into training and testing set: This code splits the training set into a training set and a testing set using scikit-learn's train_test_split() function.
5. Feature selection: The code removes the 'Id' column from the training and testing set since it's not necessary for the model.
6. Convert Review data into numerical features using CountVectorizer and TfidfVectorizer: This code uses scikit-learn's CountVectorizer and TfidfVectorizer to convert the 'Review' column in the training set into numerical features that can be used by the model.
7. Learn the model: This code creates a Ridge regression model with an alpha of 1.0, and trains the model on the training set using the Tfidf-transformed 'Review' column as the input and the 'Score' column as the target.
8. Pickle the model: This code saves the trained model as a pickle file so that it can be loaded later.
9. Evaluate the model on the testing set: This code evaluates the trained model on the testing set by making predictions using the CountVectorizer and TfidfVectorizer-transformed 'Review' column in the testing set, and calculates R^2, MAE, and RMSE.
10. Save submission DataFrame to CSV file: This code saves the testing set DataFrame to a CSV file.
11. Plot the results: This code generates several visualizations of the model's performance on the testing set, including a scatter plot of the actual vs. predicted scores, a histogram of the residuals, a line plot of the actual and predicted scores, and a scatter plot of the predicted scores vs. residuals.
12. Create a DataFrame with actual and predicted values: This code creates a new pandas DataFrame with the actual and predicted scores from the testing set.
13. Create a heatmap: This code generates a heatmap of the correlation between the actual and predicted scores from the testing set.

Here are the results:
R^2 on testing set = 0.5276554272341696
MAE on testing set = 0.5988535391747627
RMSE on testing set = 0.6662103983736107