REPORT - HW1

ID	Vikramaditya
Accuracy	0.89
Rank	59 (as of February 10, 2023 at 5:47PM EST)

Problem Statement:
Sentiment Analysis of 18506 baby products using Logistic Regression.

Approach Used:

- Read Training File (reading Sentiments and Reviews)
- Data Cleaning
- Vectorizing Features
- Train the Model based on Features Extracted
- Read Test File (reading Reviews only)
- Data Cleaning Test Reviews
- Predict Sentiments
- Insert the predicted sentiments to output file

Description:

Read Training File
The read_table function is used to read data from a file and store it in a data structure called DataFrame. The first argument to the function is the file path, in this case I used a variable to store file location. The names argument specifies the column names for the data in the file, in this case sentiment and review.

The resulting DataFrame is stored in the variable k. The DataFrame can be used for a variety of data manipulation and analysis tasks, such as filtering, grouping etc.

Data Cleaning
The code contains several functions that perform various text cleaning(pre-processing) tasks such as removing punctuation, numbers, and lemmatizing words. The removePunctuationInitial and removeNumbers() functions perform the tasks of removing punctuation and numbers, respectively. The reviewLammatization() function uses the WordNetLemmatizer to lemmatize words to their base form.

The code also performs other text pre-processing tasks such as converting the text to lowercase, tokenizing the text into words, and removing any additional unwanted information such as symbols, quotes, etc. A function called modifyNot() that takes a string argument review and returns the same string with a modification. The function replaces every occurrence of the string "n't" with " not" in the input string and returns the modified string. And joinInSingleSentance() function to join all the tokens back to a single string.

Vectorizing Features
The code is using the TfidfVectorizer function from the TfidfVectorizer class in the scikit-learn library to vectorize the "review" column of the "k" pandas dataframe. The purpose of vectorization is to convert textual data into numerical data so that machine learning algorithms can process it. In this case, the text of the "review" column is transformed into a term frequency-inverse document frequency (TF-IDF) matrix, which measures the importance of each word in each document (review) in the corpus.

The "ngram_range" argument of the TfidfVectorizer function is set to (1,3), meaning that the vectorization process will consider individual, pairs, and triples of words when computing the TF-IDF scores. The "min_df" argument is set to .0010, meaning that words with a document frequency lower than .0010 of the total number of documents will not be considered.

Train the Model based on Features Extracted
The next step is to fit a Logistic Regression model to the training data. The LogisticRegression classifier is initialized with the multi_class argument set to 'multinomial' which specifies that the target is a multiple outcome problem(used as it has increased the correct predictions). The fit function is then called on the classifier with the training feature data (trainDimX) and target data (trainDimY) to train the model.

The train_test_split function is used to split the data into training and testing sets. The dimX array is the feature data, and dimY array is the target data. The test_size argument specifies the proportion of the data to be used as the testing set, in this case 0.01(99% data used for training as I am emphasizing more toward training of model). The random_state argument sets a seed value to ensure the same results are generated every time the code is run(by setting random_state to a specific integer value, the same random numbers(42) will be generated each time the code is run, which means that the same train-test split will be produced). The shuffle argument shuffles the data before splitting so that Model can be trained more rigorously.

Read Test File (reading Reviews only)
The code defined by function readTestFile(), reads a test file, stored at the file path variable defined on top, uses the open function in python. The file is opened in read mode ("r"). The contents of the file are then read into a list of strings using the readlines method. The list of strings is then converted into a Pandas dataframe with a single column named "review". Finally, the length of the dataframe is printed and the dataframe is returned.

Data Cleaning Test Reviews
The code performs all the data cleaning functions as similar to training data cleanup operations above.

Predict Sentiments
finalYPrediction=classifier.predict(testActualX), is using a previously trained Logistic Regression model (stored in the classifier object) to predict the target values for the transformed test data testActualX. The predicted target values are stored in the finalYPrediction object.

Insert the predicted sentiments to output file
The code is writing the sentiment prediction results to a file located at variable outputLocation.
For each sentiment prediction in the finalYPrediction array, the code checks if the sentiment is "1". If it is, it writes the string "+1" followed by a newline character ('\n') to the output file. If the sentiment is not "1", it writes the string representation of the sentiment followed by a newline character to the output file.