HOMEWORK 1

NAME : Preeti Bhattacharya
G number: G01302375
USERNAME IN MINER : Preeti
ACCURACY SCORE: 0.87
RANK: 219 (At time of Upload)

Introduction:

Logistic Regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.). In other words, the logistic regression model predicts P(Y=1) as a function of X.
In this assignment I implemented a basic Logistic regression to classify a dataset of 18,506 baby product reviews that was provided.

Approach:

I spent the initial days trying to learn about different libraries, what they do and how they are implemented, as I have very basic knowledge about data mining.
I divided my homework into parts -
1. Cleaning data and Preprocessing.
2. Vectorization
3. Implementing logistic regression.

1. Cleaning Data and Preprocessing:

A. Reading data from the Train and Test files:
I have converted test and train .dat files into .csv file. Once after opening the csv file all the lines are read and stored into the lines list. Each line is read from the lines list, and then split based on the (“\t”) parameter. The first element is stored in the rating list and the second element is stored in the review list.

B. Removing Punctuations and Numbers:
The review is then passed for cleaning where all the punctuations, special characters, html tags, whitespaces and numbers are removed. For this I have used Regular Expressions.
re.sub(re.compile('<.*,?(){}!@#$%^&*_+=[]|">|[^\w+]|[0-9] '), ' ', sentence). Also, the “#EOF” at the end of each review is removed.

C. Creating Dataframe:
I’ve created a Dataframe with Pandas and used dictionary that contains the ratings, reviews.

D. Remove Stopwords:
The Stop Words were then removed from the cleaned data set. I used the NLTK library to import stopwords for this. Then I made a set called stopwords to hold all of the stopwords because searching in a set is faster than searching in a list. I applied a function that removes stopwords on my removed_punctuations (Cleaned data).

E. Lemmatization and POS Tagging:
I conducted lemmatization and POS tagging to the cleaned sentences after removing Stopwords. I'm using nltk's WordNetLemmatizer and Pattern's tag for this. I made the decision to use lemmatization because it

keeps the word's meaning, unlike stemming. Which simply removes some of the characters in a sentence, in the end. I also used POS Tagging in conjunction with this. Basically, it returns a tuple with each word and its corresponding tag.

2. **Converting to Vectors**

For this I used – tfidfVectorizer, for TF-IDF I got an accuracy of 0.8707585908796196 (87%). So, I decided to move ahead with the TF-IDF Vectorizer. Here, it assesses how important a word is to a review in the context of the entire review set. Here, two metrics are multiplied: the number of times the word appears in the review and the word's inverse review frequency of the word across all reviews. This ensures that words or phrases that appear less frequently are given more weight, as they are more likely to be relevant to the reviews. Each word's TF-IDF was calculated, and instead of being allocated a count based on the bag of words model, each word was now awarded a "score," a floating-point value indicating the word's importance to the review in which it was discovered in relation to the overall dataset.

3. **Logistic Regression**

I used the logistic regression to analyze the sentiments of baby product. I got 0. 8707585908796196 accuracy and I got a score of 0.87 on miner. The data preparation for logistic regression is much like linear regression. Here we have used LogisticRegression() imported from sklearn and also used fit(x_train,y_train) method to fit the model according to the given training data.

The final list of predictions is written to an output file, using log from math and outer from numpy.

REDUCING RUN TIME :

Steps taken to reduce run time:
1. Removed punctuations and special characters.
2. Removed stop words.
3. Used lemmatization.
4. Used TF-IDF over Count Vectorizer.

CONCLUSION AND FUTURE WORK:

The final result of this assignment is a classifier with 87% accuracy. As the dataset was very large and with a large number of features there is still more room for improvement in terms of feature extraction. There are other methods that can be investigated such as using Spacy instead of NLTK for preprocessing as well as using Doc2Vec instead of TF-IDF for feature extraction. Other methods like PCA for dimensionality reduction can also be looked into.

REFERENCE:

- https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8
- https://machinelearningmastery.com/logistic-regression-for-machine-learning/#:~:text=Logistic%20regression%20uses%20an%20equation,an%20output%20value%20(y).