Homework 1 Report

Name : Uday Kumar Kamalapuram
GMU Id: G01340201
Miner UserId: uday
Public Score: 0.86
Rank: 249

APPROACH

Selecting the Notebook:
I have chosen Google colab to start the Assignment as I am familiar with Google colab and I can work on it from the cloud.

Data Uploading and Framing:
I have converted the given .dat file format to .csv and imported a files library from google.colab to upload test and train .csv files which I converted from .dat file. I have read all lines and splitted each line based on '\t', and stored them in a dictionary under 'rating' and 'review' keys and data framed with pandas.

Text Preprocessing:
I have imported nltk library and imported stopwords and WordNetLemmatizer from NLTK and string.punctuation from string import. I have excluded all those stop words and punctuation and lemmatize words and stored the cleaned text. Similarly I have done the text cleaning for the test.csv file.

Building Model:

Tf-Idf Transformation:
I have cleaned text data of test and train data and train data ratings. From sklearn.feature_extraction.text import TfidVectorizer. I have used the term frequency–inverse document frequency (tf-idf) for feature extraction, to map the most frequent words to feature indices and compute a word occurrence frequency. I have used the fit_transform method on train cleaned text(reviews) data (dfTest["lemmatised"]) to scale the train data and also to learn the scaling parameters of train rdata, fit_transform will calculate mean and variance of each feature present in the data and transform all features using respective mean and variance. I have used the transform method on test cleaned text(reviews) data(dfTest["lemmatised"]). It will just transform all features using respective mean and variance calculated from the train data.

Train-Test Split:
I have splitted the trained data using train_test_split from sklearn.model_selection. I have used this to get the accuracy of the model before uploading it to Miner. I have checked the accuracy on the train data by splitting the train data to x_test, x_train, y_train, y_test. I have reviews in x and ratings in y.

Logistic Regression:

I have imported LogisticRegression from sklearn.linear_model. I used the fit method to train the model with the x_train and y_train data which I generated from the train-test split. After that I predicted the output for x_test and compared with the y_test. I got 0.867 accuracy for the train-test Split data. Then proceeded to predict the ratings for actual test data and wrote the output to assignOut.text file and uploaded it to minor and got a 0.86 score.

References:

This is my POC project on NLP and LDA model:
https://github.com/uday44k/hello-world/blob/main/NLP_POC.ipynb

Sklearn.linear_model.logisticregression. scikit. (n.d.). Retrieved February 9, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Khanna, C. (2020, December 25). *What and why behind fit_transform() vs transform() in scikit-learn !* Medium. Retrieved February 9, 2022, from https://towardsdatascience.com/what-and-why-behind-fit-transform-vs-transform-in-scikit-learn-78f915cf96fe

Anjali, K. (n.d.). *TF IDF: TfidfVectorizer tutorial python with examples*. etutorialspoint. Retrieved February 9, 2022, from https://www.etutorialspoint.com/index.php/386-tf-idf-tfidfvectorizer-tutorial-with-examples