Assignment 2: Text Classification

MSBD6000H

Spring 2021

Prerequisites

1. Knowledge about Vector Space Model (VSM), Logistic Regression (LR), and Naive Bayes (NB).

2. You need to install the NLTK, Pandas, Numpy, Scipy, and scikit-learn packages:

```
pip3 install --upgrade nltk pandas numpy scipy scikit-learn
```

```
python
>>> import nltk
>>> nltk.download('punkt')
>>> nltk.download('stopwords')
```

3. Read the tutorial files `Classification.ipynb` and `Naive Bayes.ipynb`. You may find related ideas of the assignment there.

1 Assignment

You need to train a Logistic Regression classifier and a Naive Bayes classifier using the training data from `data/train.csv`, and predict the labels of test data from `data/test.csv`. The ground truth labels for the test set are presented in `data/answer.csv`. In this assignment, we use 3-gram words as features for classification. Follow the instructions below and complete the codes.

Q1 Finish the codes in the `to_numerical` function, to convert the n-gram words of training and testing sentences into numerical vectors. The resulting matrices should be stored in COO sparse matrix format (`https://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.coo_matrix.html`).

Q2 Finish the codes in the `classify_lr` function. Use Logistic Regression to train a classifier using the acquired `train_feats_matrix` from Q1. You may use the implementation from sklearn (`https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html`).

Q3 Finish the codes in the `compute_prior` function to calculate the prior array (i.e., $P(y)$ in the tutorial).

Q4 Finish the codes in the `compute_likelihood` function to calculate the likelihood matrix (i.e., $P(x|y)$ in the tutorial).

Q5 Finish the codes in the `classify_nb` function to calculate a prediction array given the input matrix, the prior array in Q3, and the likelihood matrix in Q4.

The expected test accuracy of both models should be greater than 0.46.

2 Submission

You need to submit two files, program output and your python script. After you finished the assignments, make sure you include the header information in the beginning of your code

```
# author: Your_name
# student_id: Your_student_ID
```

Copy all the program output in to text file named `StudentID_assignment2_output.txt`, and submit with your python script solution named `StudentID_assignment2.py` to canvas.