**NAME: XIANG LI, GWid: G47754486**

**Assignment #3**

# Instructions:

This assignment is due on **Friday November 8, 2019, by 11:59 pm**. Refer to course information about late submission policy. Late days are counted from 00:00 a.m. onwards.

Submit your solution in a zipped folder through blackboard.

All code must compile and run to receive full credit for coding parts.

Include citations for any online resources used or group discussions.

# Text Classification

In this assignment, you will use **scikit-­‐learn**, a machine learning toolkit in Python, to implement text classifiers for sentiment analysis. Please read all instructions below carefully.

# Datasets and evaluation:

You are given the following customer reviews dataset: CR.zip, which includes positive and negative reviews. CR is a small dataset that doesn’t have train/test divisions, so you are required to evaluate the performance using **10-­‐fold cross-­‐ validation**. Please use the following scikit-­‐learn modules in your implementation:

**Question 1:**

Using the scikit-­‐learn modules described above, Implement the following models and report the performance (accuracy and F1) for the CR dataset:

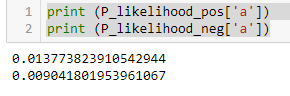
**Accuracy=** (tp+tn)/(tp+fp+tn+fn)

**Precision** = true positives / (true positives + false positives)

**Recall**=true positives / (true positives+ false negatives)

**F1** = 2PR/(P+R)

Some word as such “a” has a big impact on results,



And a big impact from STOP Words means we have to remove stop words

.

1. [10 points] A Naïve Bayes classifier with add-­‐1 smoothing using **binary** bag-­‐ of-­‐words features.

NLP\_HW3\_a\_naive\_bayes.py

1. [10 points] A Naïve Bayes classifier with add-­‐1 smoothing using binary bag-­‐ of-­‐ngrams features (with unigrams and bigrams).

NLP\_HW3\_b\_with\_bigram.py

1. [10 points] Logistic Regression classifier with L2 regularization (and default parameters) using binary bag-­‐of-­‐words features.

NLP\_HW3\_c\_L2.py

Sklearn default penalty is “L2”

1. [10 points] Logistic Regression classifier with L2 regularization using binary bag-­‐of-­‐ngrams features (with unigrams and bigrams).

NLP\_HW3\_L2\_bigrams.py

|  |  |  |
| --- | --- | --- |
|  | Accuracy | F1 |
| Method a) | 70.88% | 71.85% |
| Method b) | 79.30% | 80.28% |
| Method c) | 71.86% | 72.13% |
| Method d) | 70.43% | 71.59% |

Performance report [10 points].

Using 10-fold cross validation to divide all data.

For all the following scripts, to save your efforts to run them, I simply write the scripts without any input parameters.

IMPORTANT:

Please put all data and scripts in the same directory.

Python 3 was used and package numpy sklearn were imported.

*Reference:*

*1.* Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition Third Edition draft, and Second Edition

*2. Finite State Machine Designer (*http://madebyevan.com/fsm/)

*3.* Python, Documentation 3.7

*4. Numpy*

*5. Slides from Course*

*6.***scikit-­‐learn documentation:**

# Bag-­‐of-­‐words (or ngrams) feature extraction using CountVectorizer:

[http://scikit-­‐](http://scikit-/)learn.org/stable/modules/generated/sklearn.feature\_extraction.text.CountVectorizer.html

Use **binary features** (1/0 rather than counts).

# Naïve Bayes classifier:

[http://scikit-­‐](http://scikit-/)learn.org/stable/modules/generated/sklearn.naive\_bayes.MultinomialNB.html

# Logistic Regression classifier:

[http://scikit-­‐](http://scikit-/)learn.org/stable/modules/generated/sklearn.linear\_model.LogisticRegression.html

# Cross validation:

[http://scikit-­‐](http://scikit-/)learn.org/stable/modules/generated/sklearn.model\_selection.cross\_validate.html

# Classification report:

[http://scikit-­‐](http://scikit-/)learn.org/stable/modules/generated/sklearn.metrics.classification\_report.html