Name: Emmanuel Nimo

NLP Lab 4 Report

Q1. Explanation for libraries used

1. Sklearn: It has several classification algorithms which is designed to sync with numpy library which loads the given train and test data. It also allows you to implement a predictor and a target and also uses Laplace smoothing when you train a Naive Bayes classifier and it comes by default.
2. Numpy: This library has a high performance for reading or loading data from a file. It happens to be basis on which sklearn and panda was built, hence it is very supportive in terms of interoperation with Sklearn library packages.
3. Nltk: It has a library(nltk.corpus) which provides a normalization through the use of stopwords and this helps in training the data.
4. Argparse: It helps to process the command line argument when the program runs.

.

Q2. Evaluation of classifier and Result

Accuracy was used a metric for evaluating the classifiers.

Naive Bayes

mport pandas as pd

import numpy as np

import nltk

from nltk.corpus import stopwords

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.model\_selection import train\_test\_split

from sklearn import naive\_bayes as nb

from sklearn.metrics import roc\_auc\_score

import argparse

1. from sklearn.feature\_extraction.text import TfidfVectorizer

It is used for converting a collection of raw documents to a matrix of TF-IDF features.

from sklearn.model\_selection import train\_test\_split

It is used for split arrays or matrices into random train and test subsets

Result for Naïve Bayes Normalized

PS C:\Users\study\Desktop\Fall 2018\NLP\nlp-18-nimo\Lab 4> python lab4.py nb n nimo.txt

('Accuracy: ', 88.0)

('Score :', 0.9495408602379987)

1

0

1

Result for Naïve Bayes Unnormalized

PS C:\Users\study\Desktop\Fall 2018\NLP\nlp-18-nimo\Lab 4> python lab4.py nb u nimo.txt

('Accuracy: ', 80.4)

('Score :', 0.8906400153757448)

1

0

1

Logistic regression

import pandas as pd

import numpy as np

import nltk

from nltk.corpus import stopwords

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import roc\_auc\_score

import argparse

from sklearn.metrics import roc\_auc\_score

It is used to calculate the compute Area Under the Receiver Operating Characteristic Curve (ROC AUC) from prediction scores.

Result for logistic Regression with normalization

PS C:\Users\study\Desktop\Fall 2018\NLP\nlp-18-nimo\Lab 4> python lab4Log.py lr n nimo.txt

('Accuracy: ', 87.46666666666667)

1

0

1

Result for logistic regression without normalization

PS C:\Users\study\Desktop\Fall 2018\NLP\nlp-18-nimo\Lab 4> python lab4Log.py lr u nimo.txt

('Accuracy: ', 80.4)

0

0

1

Q3. Discussion of Results and classifiers

During the test, a file (nimo.txt) with three lines of sentences was passed to the classifiers.

Nimo.txt

I like the book

i dont like mangoes

I love her

From the result above, a multinomial feature, which helps in the implementation of discrete features (word counts for text classification), enables the Naïve Bayes to acquire a high accuracy when it is normalized than unnormalized. The predictions made by both normalized and unnormalized were excellent, accuracy for normalized was higher than unnormalized. This makes Naive Bayes have a good approximation in terms of developing a good classifier although it comes with a few errors when a large dataset is used.

Logistic regression also worked pretty well, just that the normalized had a high accuracy and better precision then the unnormalized. The unnormalized logistic regression was the only unique classifier among the outputs given, thus, [0 0 1].