CS 682 – Artificial Intelligence

Project 2 - Spam Filter

Monikrishna Roy

November 5, 2021

Purpose

The purpose of this assignment is to provide a multi-executable programming experience with a discriminatory algorithm that can make decisions using Artificial Intelligence. You should be able to utilize probability in order to classify using a Naive Bayes Classifier.

Task

Please write a program to parse an existing dataset on real-world SMS messages. (note: since these data come from real-world interactions, these messages may use language which I would never use in class and that violates professional conversational norms. If that is likely to trigger a negative reaction, please do not read the messages themselves. However, I feel that it is important to work with real-world data wherever possible).

You will need to write two programs:

1. *training -i <spam.csv file> -os <output spam probability file> -oh <output ham probability file>*

   Trains dataset from .csv file and save to new file. Each line of the .csv file has at least two fields, separated by a comma:

   ```
   1. <spam|ham> ham if it is a legitimate SMS, spam if not
   2. "..." the SMS message
   ```

   You will need to output two probability files (one for ham, one for spam):

   ```
   <count of the total number of words (n)>
   m lines, one for each word <word> <number of word occurrences>
   ```

2. *classify -i <testing dataset .csv file> -is <spam probability file> -ih <ham probability file> -o <classification output filename>*

   Classifies new data from training file and testing .csv file (same format as above, specified on the command-line)

   You will need to output one classification file:

   ```
   m lines, one for each SMS in the testing dataset
   (in the same order as the testing set is in <spam/ham>)
   (the classification of the SMS)
   ```

Graduate Student Extra Assignment

Please also write a program to add new data (in a .csv file) to the existing training database.

addtotraining -is <input spam probability file> -ih <input ham probability file> -s "<string>"

Solution/Algorithms

Naive Bayes Classifier is used to calculate the probability of the spam/hum messages.

Naive Bayes Classifier

A Naive Bayes classifier is a probabilistic machine learning model that's used for the classification task. The crux of the classifier is based on the Bayes theorem.

The multinomial Naive Bayes method is used here. This is mainly for the document classification problem, i.e., whether a document belongs to the category of sports, politics, technology, etc. The features/predictors used by the classifier are the frequency of the words present in the document.

The following formula is used to calculate the probability of a message being spam or not:

Equation

```
P(w|c) = (C(w, c) + 1) / (C(c) + |V|)

where,
P(w|c) = the probability of a word belongs to given class c (ham/spam)
C(w, c) =  the frequency of the word in the given class c
C(c) = total number of words in the given class
|V| = number of the unique vocabulary in the dataset.
```

Challenges

There are a few challenges for this project. To get good results, data processing and feature extraction are most important here. In this project, only text cleaning is done as data processing, but feature extraction has not been applied.

The following steps are done for text cleaning:

- convert all letters to lowercase
- clean punctuation
- clean numbers
- remove multiple spaces
- remove non-ascii characters
- remove not alphabetic characters
- remove single characters
- remove hyperlinks

Pre-requisites and Environment Settings

- Python >= 3.8
- Pandas

Run Command / Usage

```
<create a virtual environment>
$ python3 -m venv spamfilter

<activate the virtual environment>
$ source spamfilter/bin/activate

<install the requirement>
$ python3 -m pip install -r requirements.txt

<run the training code>
$ python3 code/training.py -i <spam.csv file> -os <output spam probability file>
    -oh <output ham probability file>

<classify the test set>
$ python3 code/classify.py -i <testing dataset .csv file> -is <spam probability
   file> -ih <ham probability file> -o <classification output filename>

<add new training to training set>
$ python3 code/addtotraining.py -is <input spam probability file> -ih <input ham
    probability file> -s <new training set file>
```

Some notes,

- Creating and activating virtual environment are optional steps.
- The first row of the input file is counted as a header. So every input file should have a header row; otherwise, the first row will be excluded from data.

Examples to Run

```
<to run train dataset>
$ python3 code/training.py -i data/spam.csv -os data/spam_probability.csv -oh
   data/ham_probability.csv

<to classify test data>
$ python3 code/classify.py -i data/test.csv -is data/spam_probability.csv -ih
   data/ham_probability.csv -o data/output.csv

<to add new dataset to trained data>
$ python3 code/addtotraining.py -is data/spam_probability.csv -ih data/
   ham_probability.csv -s data/new_training.csv
```