[Project Code: SFNB]
Email Spam Filtering Using Naive Bayes Classifier Learning Model

Project Duration : 22-Jan-2023 ~~ 11-Feb-2023
Submission Information: (via) CSE-Moodle

Objective:
Naive Bayes is a probabilistic algorithm that is based on the Bayes Theorem and is commonly used for email spam filtering in data analytics. If you have an email account, you may have noticed that emails are automatically categorized into different buckets and marked as important, spam, or promotions. It is amazing to see machines being so smart and doing this work for us.

In this assignment, your task is to design and implement a Naive Bayes classifier that predicts whether an email is spam or not spam (ham). In particular, you shall be doing the following tasks:

Your Tasks:

1. *Exploratory Data Analysis and Text-Preprocessing:*
    a. Clean the email text using appropriate text-preprocessing techniques, such as removing stop words and stemming. You can use the code-snippet below for this.
    b. Identify the 10 most frequently occurring words used in spam and ham emails.
    c. Plot the class distribution for the dataset using a bar chart.
    d. Plot Box plot and Kernel Density Estimation (KDE) plot of word length for both the classes. You can use the matplotlib and seaborn packages for parts (b) and (c). In your report, include observations and insights derived from your data analysis.

```
import re
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('omw-1.4')

def clean_text(text):
  # Lowercase the text
  text = text.lower()
  #Clean the text
  cleaned_text = re.sub("[\r\n]", "", text)
  # Tokenize the text
  tokens = word_tokenize(cleaned_text)
  # Remove stop words
  stop_words = set(stopwords.words('english'))
  tokens = [token for token in tokens if token not in stop_words]
  # Perform lemmatization
  lemmatizer = WordNetLemmatizer()
  tokens = [lemmatizer.lemmatize(token) for token in tokens]

  return tokens
```

2. *Building a Naive Bayes classifier from scratch:*
    a. Build a Naive Bayes classifier by randomly splitting the dataset as 80/20 split i.e., 80% for training and 20% for testing. Provide the accuracy by averaging over 10

random 80/20 splits. Select the model which you find to be the best (explain your choice in report)

   b. Train the Naive Bayes Classifier using Laplace correction on the train and test split selected in part (a). Print the final test accuracy.
   c. Now, train the model (with & without using Laplace correction) on the train and test split selected in part (a) but without cleaning the email text. Print the final test accuracy.

3. *Analyzing performance:*
   a. Create a classification report for both the classifiers in tabular form. (with and without Laplace correction).
   b. Analyze the impact of text cleaning on performance of the models. (*Note: You need to calculate accuracy, precision, recall, f1-score and support for both the classes (Ham and Spam) on the test set)*

4. *Report:*
   Write a brief (2-3 page) report/manual of your work including results, your observations and justifications on the same.

Dataset:
The dataset contains around 4600 emails labeled as spam or ham.

<u>*Files:*</u>
email_spam_dataset.csv – each line contains a label ("ham" for non-spam and "spam" for spam) and the text of an email sample.

<u>*Data Fields:*</u>

- **id** - an anonymous id unique to an email sample
- **Text -** a string containing email text
- **Label** - the spam indicator of an email sample (0: non-spam/ham; 1: spam)

Submission Details: (to be submitted in CSE-Moodle, **by one representative of the group**)

1. ZIPPED folder containing code (with comments) and the dataset files
2. Report (in pdf format)

Submission Guidelines:

1. You may use one of the following languages: C / C++ / Java / Python. No machine learning /data science /statistics package / library should be used for model creation.
2. Your Programs should run on a Linux Environment.
3. Your program should be standalone and should **not** use any special purpose library. **Numpy or Pandas may be used.** And, you can use libraries for other purposes, such as formatting and visualization of data.
4. You should submit the program file and README file and **not** the output/input file.
5. You should name your file as <GroupNo_ProjectCode.extension>.
   (e.g., *Group99_SFNB.zip* for code-distribution and *Group99_SFNB.pdf* for report)
6. The submitted program file *should* have the following header comments:
   # Group Number
   # Roll Numbers : Names of members (listed line wise)
   # Project Number
   # Project Title
7. Submit through CSE-MOODLE only.
   Link to our Course page: https://moodlecse.iitkgp.ac.in/moodle/course/view.php?id=508

You should not use any code available on the Web. Submissions found to be plagiarized or having used ML libraries (except for parts where specifically allowed) will be awarded zero marks.

For any questions about the assignment, contact the following TA:
Abhinav Bohra (Email: abhinavbohra09@gmail.com)