ARTIFICIAL INTELLIGENCE
CED16

Muskan Khandelwal (2017UCO1596)
Ishaan Rawat (2017UCO1644)
Shrey Jain (2017UCO1647)
Anish Jangra (2017UCO1654)

Submitted to : Dr. Shampa Chakraverty

Email Classification

In the first part of the project we have implemented the spam classifier using the ***Naive Bayes*** classifier from scratch. `Our objective was to understand the functioning of naive bayes classifier in the classification of emails throughly.`

In the second part of the project we have implemented the spam classifier using various classifiers from the scikit learn library and have compared their performance.

We have collected the data set from SpamAssassin public mail corpus containing around 1900 spam emails and 3900 legit emails.

Spam classification using Naive Bayes

Architecture of spam classifier

An Email body is divided into three parts header, subject line and body where header contains the information of the sender, subject contains the subject of the email and body contains the actual content of the email. Before the email can be classified by a filter, preprocessing of the email is required to get the desired features.

> Preprocessing the Emails

1. The body of the emails are extracted from the email.
2. A pandas data frame is created having the email bodies and category.
3. Morphological analysis of the messages in the data set is done. After the morphological analysis each message body is converted into the individual words. We have used the natural language tool kit (nltk) for this purpose. It involves:
   - Removing the HTML Tags from the body of the email.
   - Stemming of the words.
   - Removal of stop words.
4. A pandas series is created having all the words in all the emails.
5. A vocabulary is created with the most frequent 2500 words.
6. A feature matrix is created having entry for each word in the email.
7. Data is split into 70% training and 30% test set.
8. A sparse matrix is created for the training and the testing data.

Bayes Theorem

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

We assume that each word in an email is independent of every other word an hence the name *Naive Bayes*.

If a word is present in an email we can write the probability of the email being spam given that it contains a particular word as:

$$P(spam \mid word) = \frac{P(word \mid spam)\, P(spam)}{P(word)}$$

and the email being ham as:

$$P(ham \mid word) = \frac{P(word \mid ham)\, P(ham)}{P(word)}$$

For all the words in our vocabulary, we will find $P(word)$, the probability of a given word and $P(word \mid Spam)$, the probability of the word in spam emails.
We will also calculate the probability of spam emails in training data as as:

$$P(spam) = \frac{number\,of\,spam\,emails}{total\,number\,of\,emails}$$

> Training the Naive Bayes Classifier

1. Calculate $P(spam)$, the probability of spam emails.
2. For all the words in the dictionary calculate
    - $P(ham \mid word)$
    - $P(spam \mid word)$

Let’s say an email contains words X and Y.
The probability of the email being spam is:

$$P(spam \,|X,\, Y) = \frac{P(X \mid spam)}{P(X)} * \frac{P(Y \mid spam)}{P(Y)} * P(spam)$$

The probability of the email being ham is:

$$P(ham\,|X,\,Y) = \frac{P(X\mid ham)}{P(X)} * \frac{P(Y\mid ham)}{P(Y)} * P(ham)$$

For prediction we can remove the term $P(word)$ from the denominator of our expression because it is common in both cases.

If $P(spam\,|X,\,Y) > P(ham\,|X,\,Y)$ then the email will be classified as spam else it will be classified as non-spam or ham.

> Testing the Naive Bayes Classifier

1. Set the Priori as $P(spam)$.
2. Calculate the probability of an email being spam and non-spam by taking product of the test data and probabilities of individual words and multiply the priori to the result.
3. An email will be classified comparing the two probabilities.

> Results

Confusion Matrix for 1722 emails in test set

	Predicted Spam	**Predicted Ham**
Spam	True Positives = 557	False Negatives = 30
Ham	False Positives = 15	True Negatives = 1120

- $Accuracy$ = $\frac{emails\ classified\ correctly}{total\ number\ of\ emails} = 97.39\%$
- $Recall score$ = $\frac{True\ Positives}{True\ Posotives + False\ Negatives} = 94.89\%$
- $Precision score$ = $\frac{True\ Positives}{True\ Posotives + False\ Positives} = 97.37\%$
- $Fscore$ = $2 * \frac{Precision * Recall}{Precision + Recall} = 96\%$

Spam classification using scikit learn

We made classification models using *Naive Bayes, Decision Tree and Support vector machine* using the scikit learn library in python.
The vocabulary was generated using *CountVectorizer* from the sklearn library instead of generating the features manually.

> Results

	Naive Bayes	Decision Tree	SVM
Accuracy	93.61	96.20	83.20
Recall	80.68	94.94	47.65
Precision	99.11	93.26	99.25
F1 score	88.95	94.09	64.39