1 Abstract

Spam-Ham email classification is a binary classification problem and hence Spam email classifier model is created using Naive Bayes classification method.

2 Algorithm

The steps involved in developing the model is as follows:

1. Collection of Training and Test Dataset
2. Model Selection
3. Feature Extraction and Data Preprocessing
4. Features selection
5. Model training
6. Prediction of labels of Test dataset
7. Accuracy calculations
8. goto step 4

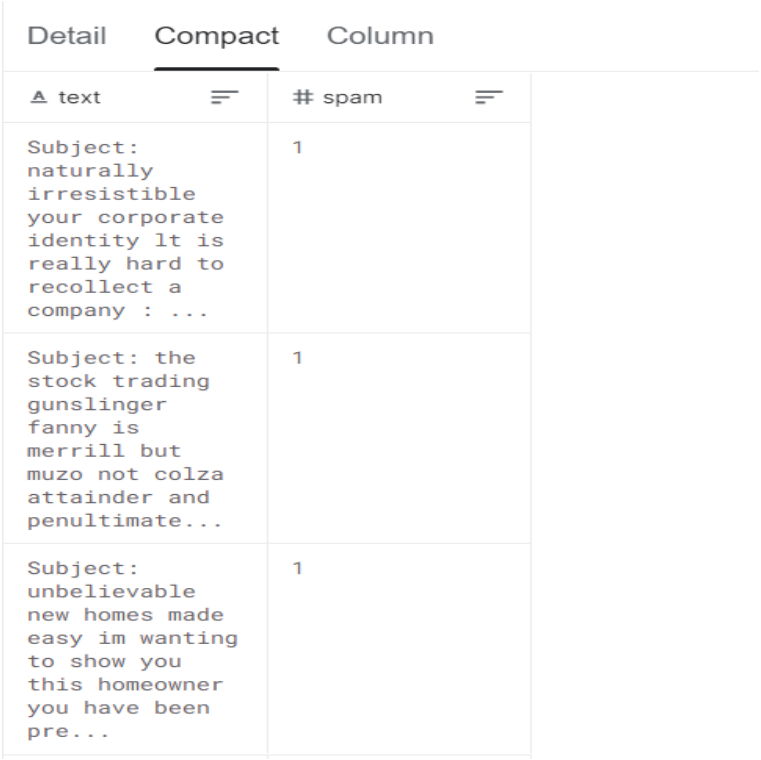
3 Training and Test Dataset

The training dataset is taken from kaggle.
which is available at, https://www.kaggle.com/venky73/spam-mails-dataset
It contains 29% spam emails and 71% ham emails
This contains 4 columns,shown below

#	label	text	label_num
605	ham	Subject: enron methanol ; meter # : 988291 this is a follow up to the note i gave you on monday , 4...	0
2349	ham	Subject: hpl nom for january 9 , 2001 (see attached file : hplnol 09 . xls) - hplnol 09 . xls	0
3624	ham	Subject: neon retreat ho ho ho , we ' re around to that most wonderful time of the year - - - neon ...	0

The Test dataset is also taken from kaggle.
which is available at,

https://www.kaggle.com/balakishan77/spam-or-ham-email-classification
it contains only two columns as shown below:


Detail Compact Column

text	spam
Subject: naturally irresistible your corporate identity lt is really hard to recollect a company : ...	1
Subject: the stock trading gunslinger fanny is merrill but muzo not colza attainder and penultimate...	1
Subject: unbelievable new homes made easy im wanting to show you this homeowner you have been pre...	1

4 Model selection

Initially we used three models,

- Naive Bayes
- Random Forest
- Support Vector Machine

Due to lack of resources Random Forest is eliminated.SVM and Naive Bayes were giving equivalent results hence Naive Bayes is finalised. The algorithm of Naive Bayes classification method is exactly the same as discussed in class.

5 Feature Extraction

Each email is a string of text, as shown in the dataset image of training data.For Naive Bayes classification a binary training dataset is required. The features are extracted by using CountVectorizer library from sklearn.feature_extraction.text

Where each feature is a unique word in all the text emails of training dataset. Then finally a 2D numpy array is generated where each row corresponds to an email,each column

corresponds to unique word(features) and each cell has a binary value. Where the (i,j) cell has value '1' if i_{th} email contains j_{th} word otherwise it contain value '0'. Then the total features obtained so far are more than 37000.
All the features are enlisted in the file, **"AllFeatures.txt"**.

6 Feature Selection

To reduce the number of features, information gain of all the features is calculated.
The file **"topFeatures.txt"** contains indices of all the features in descending order of information gain. Top 500 features with maximum information gain has been used to train the model. The file name **"final_features.txt"** contains all the 500 features that the model uses.

7 Training model

Naive bayes classification method has been used to train the model.
The filename **"naiveBayes.py"** cantains the source code of data preprocessing and model training.
while training the data the number of features used in the model plays an important role.Different number of features gives different accuracy.

s.no.	no. of features	accuracy %
1	300	77.76
2	400	88.7
3	500	90.8
4	800	88.4
5	1000	87.6
6	2000	84.3

Finally the model is trained using top 500 features.

8 Labels Prediction

The trained model is saved in a program file name **"trainedModelNaiveBayes.py"**.
All the required parameters are saved in a text file with their respective name as the name of file.

- P(i/ham) is probability of each feature i for all ham emails, represented by "predictors_ham" saved in text file **"naiveBayesFeature0.txt"**.

- P(i/apam) is probability of each feature i for all spam emails, represented by "predictors_spam" saved in text file **"naiveBayesFeature1.txt"**.
- p(spam) is the probability of spam emails,represented by p_spam and saved in file **"naiveBayesPspam.txt"**.

To predict labels the file **"trainedModelNaiveBayes.py"** is executed which uses all the above mentioned text files along with a file **"final_features.txt"** . The output is a numpy binary array named "predicted_label"where 1 states spam email and 0 as ham emails.

9 References

1.https://www.computer.org/csdl/pds/api/csdl/proceedings/download-article/12OmNC0PGLp/pdf
2.https://ieeexplore.ieee.org/abstract/document/8474778/
3.https://ieeexplore.ieee.org/abstract/document/7065547