**PYTHON-MACHINE LEARNING**

**PROJECT: E-mail Categorization Spam / not Spam**

**Prathmesh Prakash Magdum**

**Computer Science and Engineering**

**magdumprathmesh1247@gmail.com**

**ABSTRACT:**

Spam email is one of the biggest issues in the world of internet. Spam emails not only influence the organisations financially but also exasperate the individual email user.

This documentation aims to recognize, optimal machine learning algorithm in email categorization (i.e. spam or not spam) using following two machine learning algorithms:

1. Naive Bayes classifier method
2. Support Vector Machines classifier method

In this process, dataset is divided into different sets and given as input to each algorithm.

Total three experiments are performed and the result is obtained in terms of accuracy.

**INTRODUCTION:**

Email system is one of the most effective and commonly used sources of communication. The reason of the popularity of email system lies in its cost effective and faster communication nature. Unfortunately, email system is getting threatened by spam emails. Spam emails are the uninvited emails sent by some unwanted users also known as spammers with the motive of making money. The email users spend most of their valuable time in sorting these spam mails. Multiple copies of same message are sent many times which not only affect an organisation financially but also irritates the receiving user. Spam emails are not only intruding the user’s emails but they are also producing large amount of unwanted data and thus affecting the network’s capacity and usage. In this paper, a Spam Mail Detection (SMD) system is proposed which will classify email data into spam and ham emails. The process of spam filtering focuses on three main levels: the email address, subject and content of the message. All mails have a common structure i.e., subject of the email and the body of the email. A typical spam mail can be classified by filtering its content.

The process of spam email detection can be broadly categorized into two approaches: knowledge engineering and machine learning approach. Knowledge engineering is a network-based approach in which IP (internet protocol) address, network address along with some set of defined rules are considered for the email classification. The approach has shown promising results but it is very time consuming. The maintenance and task of updating rules is not convenient for all users. On the other hand, machine learning approach does not involve any set of rules and is efficient than knowledge engineering approach. The classification algorithm classifies the email based on the content and other attributes. For most of the classification problems the process of feature extraction and selection is very important.

The proposed spam mail detection system is inspired from the effectiveness of machine learning approach. In spam mail detection system, initially email data is collected. The email data collected is raw and unstructured in nature. In order to reduce the computations and to obtain accurate results, email data needs to be pre-processed. The data is pre-processed by removing stop words, stemming and word tokenization is also performed to acquire valuable information.

**MACHINE LEARNING IN E-MAIL CLASSIFICATION Algorithms :**

Machine learning field is a subfield from the broad field of artificial intelligence, this aims to make machines able to learn like human. Learning here means understood, observe and represent information about some statistical phenomenon. In unsupervised learning one tries to uncover hidden regularities (clusters) or to detect anomalies in the data like spam messages or network intrusion. In e-mail filtering task some features could be the bag of words or the subject line analysis. Thus, the input to e-mail classification task can be viewed as a two dimensional matrix, whose axes are the messages and the features. E-mail classification tasks are often divided into several sub-tasks. First, Data collection and representation are mostly problemspecific (i.e. e-mail messages), second, e-mail feature selection and feature reduction attempt to reduce the dimensionality (i.e. the number of features) for the remaining steps of the task.

1. **Naive Bayes classifier method:**

Bayesian classifier is working on the dependent events and the probability of an event occurring in the future that can be detected from the previous occurring of the same event.

FORMULA :

P(A|B) = P(B|A) P(A)

P(B)

Similarly, this technique can be used to classify spam e-mails. If some words occur often in spam but not in ham, then this incoming e-mail is probably spam. Naïve bayes classifier technique has become a very popular method in mail filtering software. Bayesian filter should be trained to work effectively.

S [T] = \_\_\_\_\_C Spam(T)\_\_\_\_\_

C Spam(T) + C Ham(T)

Where CSpam(T) and CHam(T) are the number of spam or ham messages containing token T, respectively.

Now, lets consider probability for,

M - Message

S[M] – Spam Message

H[M] – Ham Message

The message is considered spam if the overall spamminess product S[M] is larger than the hamminess product H[M]. The above description is used in the following algorithm:

Stage1:

Training Parse each email into its constituent tokens Generate a probability for each token W S[W] = Cspam(W) / (Cham(W) + Cspam(W))

store spamminess values to a database.

Stage2:

Filtering For each message M while (M not end) do scan message for the next token Ti query the database for spamminess S(Ti) calculate accumulated message probabilities S[M] and H[M] Calculate the overall message filtering indication by:

I[M] = f(S[M] , H[M])

f is a filter dependent function, such as

S [T] = \_\_\_\_\_C Spam(T)\_\_\_\_\_

C Spam(T) + C Ham(T)

I [M] = (1+S[M]-H[M]) / 2

{

if I[M] > threshold

message is marked as spam

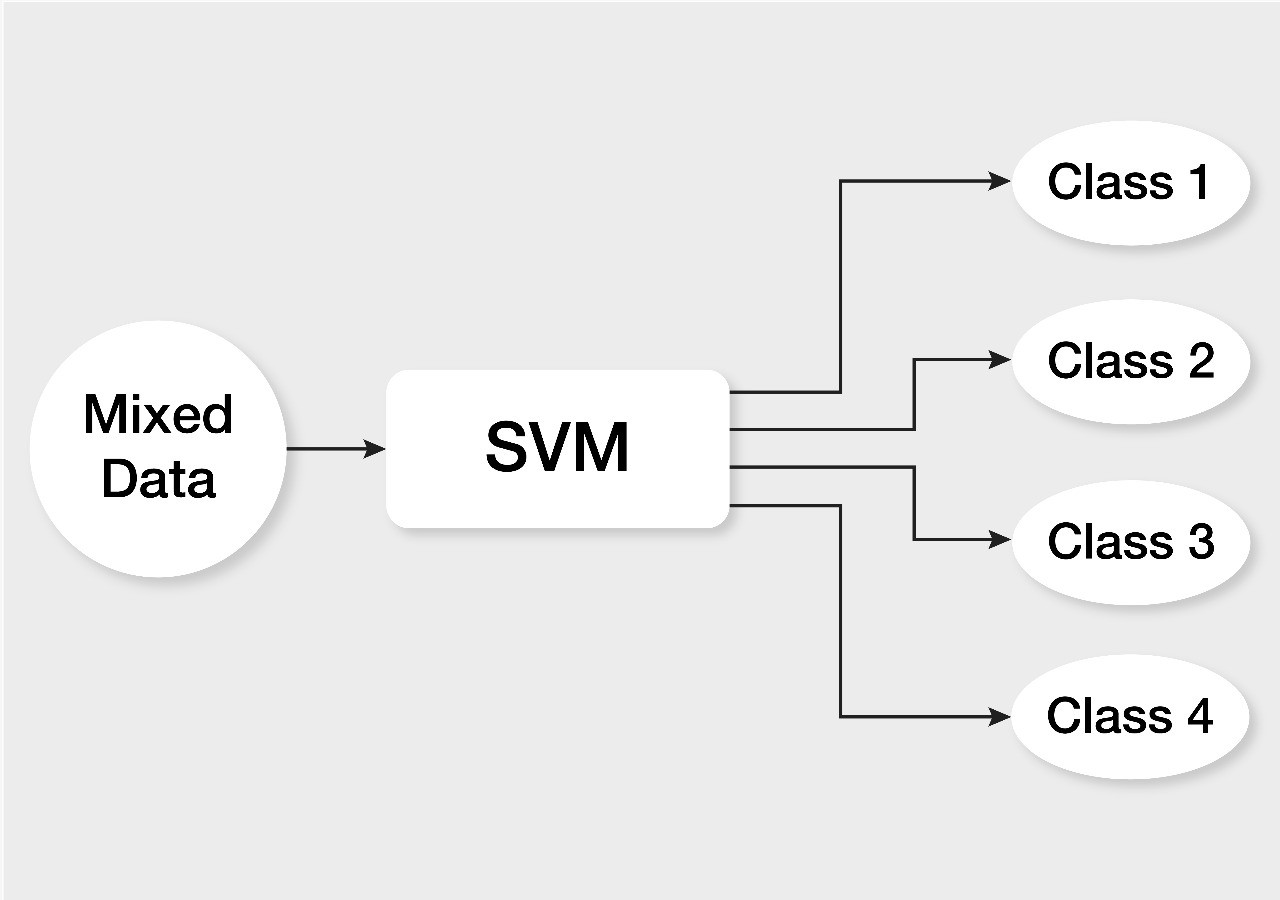
else

message is marked as non-spam

}

1. **Support Vector Machines classifier method:**

Support Vector Machines are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships, the SVM modelling algorithm finds an optimal hyperplane with the maximal margin to separate two classes.





**MACHINE LEARNING ALGORITHM IMPLEMENTATION AND OPTIMAL SOLUTION:**

Experiment Implementation In order to test the performance of above mentioned six methods, some corpora of spam and legitimate emails had to be compiled; there are several collections of email publicly available to be used by researchers. Spam-Assassin is used in this experiment, which contains 6000 emails with the spam rate 37.04%.Thus we have divided the corpora into training and testing sets keeping, in each such set, the same proportions of ham (legitimate) and spam messages as in the original example set. Each training set produced contained 62.96% of the original set; while each test set contain 37.04%.

|  |  |  |
| --- | --- | --- |
| MESSAGE COLLECTION | TRAINING SET | TESTING SET |
| SPAM MESSAGE | 2378 | 1400 |
| HAM MESSAGE | 1398 | 824 |
| TOTAL | 3776 | 2224 |

To get optimal solution for recognizing email spam classification we used two algorithm and performed experiment, after that we compared them by using Accuracy criteria.

Accuracy (A) is the percentage of all emails that are correctly categorized,

A = # of e-mails correctly categorized / Total # of e-mails

= **(Nham→ham + Nspam→spam) / Nham + Nspam**

|  |  |
| --- | --- |
| Algorithm | Accuracy |
| Naïve Bayes | **98.77%** |
| Support Vector Machine | **97.97%** |

**CONCLUSION:**

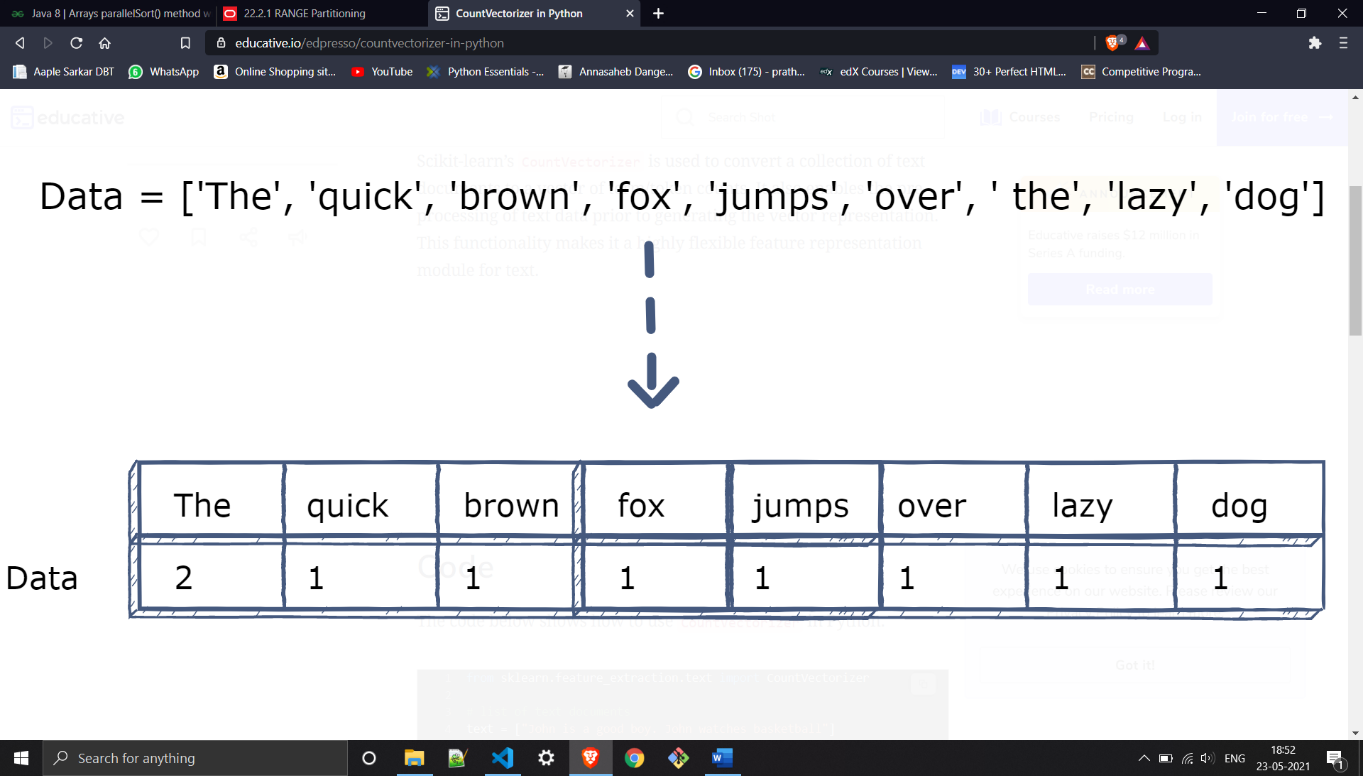
* **In this Mini Project we review two machine learning methods and their applicability to the problem of spam e-mail classification.**
* **After Performing Experiments and analysis from all point of views we got the optimal result by considering accuracy criteria.**
* **Therefore, optimal solution i.e., optimal classification algorithm is Naïve Bayes Algorithm which gives more accuracy and less time as compared to SVM Algorithm.**

**Presentation Questions and their answers:**

**1] What is CountVectorizer() ?**

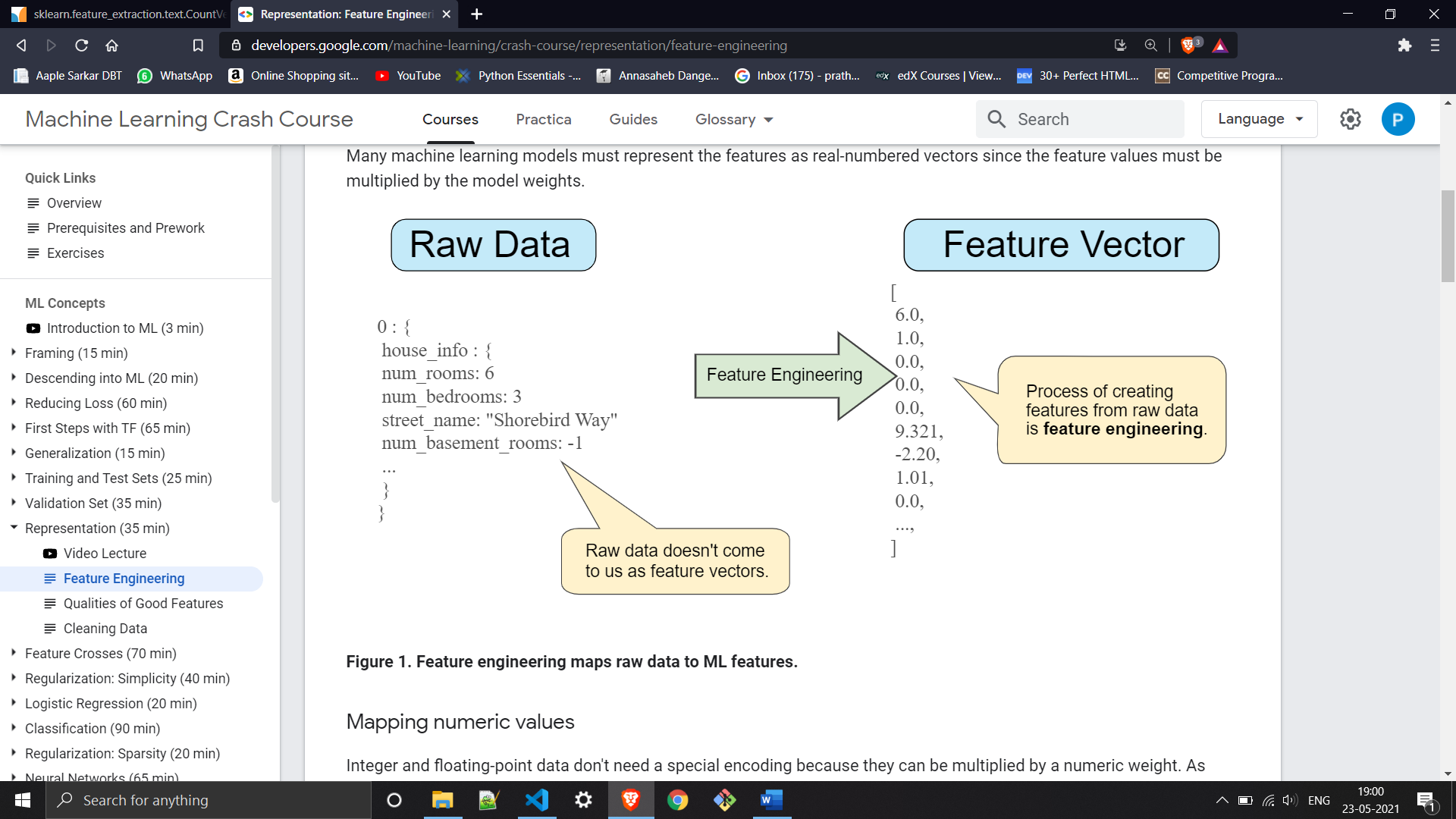
**CountVectorizer** is a great tool provided by the scikit-learn library in Python. It is used to transform a given text into a vector on the basis of the frequency (count) of each word that occurs in the entire text.

It also enables the ​pre-processing of text data prior to generating the vector representation. This functionality makes it a highly flexible feature representation module for text.



**2] What is the integer value stored in the features?**

Numerical data is used to mean anything represented by numbers (floating point or integer). Data generally needs to be put into numeric form for machine learning algorithms to use the data to make predictions. In machine learning guides categorical string data is usually one-hot-encoded



3] Reduce the written content in Introduction part of presentation.