VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM, KARNATAKA

MINOR-PROJECT-1 REPORT

ON

"UBE IDENTIFICATION IN EMAIL USING MACHINE LEARNING"

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted by

USN	NAME
2SD18CS105	SMITA S HEGDE
2SD18CS106	SMRUTI DESHPANDE
2SD18CS129	PRABHA H B
2SD18CS135	T BHARGAVI

Under the Guidance of

Prof. / Dr. RAMACHANDRANAYAK N YADAWAD
Dept. of CSE, SDMCET, Dharwad

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
S.D.M. COLLEGE OF ENGINEERING & TECHNOLOGY,
DHARWAD-580002

2020-2021

S.D.M COLLEGE OF ENGINEERING & TECHNOLOGY,

DHARWAD –580002

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the Minor-Project-1 work and presentation entitled **"UBE IDENTIFICATION IN EMAIL USING MACHINE LEARNING"** *is a bonafide work carried out by* **SMITA S HEGDE (2SD18CS105)**, **SMRUTI DESHPANDE (2SD18CS106)**, **PRABHA H B (2SD18CS129)**, and **T BHARGAVI (2SD18CS135),** *students of S. D. M. College of Engineering & Technology, Dharwad, in partial fulfillment for the award of* **Bachelor of Engineering in Computer Science and Engineering of *Visvesvaraya Technological University, Belgaum***, *during the year 2020-2021. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the department library. The Minor-Project-1 has been approved, as it satisfies the academic requirements in respect of project report prescribed for the said degree.*

---------- ----------

Dr. / Prof. RAMACHANDRANAYAK N YADAWAD **Dr. U P Kulkarni**

Project Guide HOD-CSE

ABSTRACT

E-mail spam has become a major problem now a days. The increasing volume of unsolicited bulk e-mail (also known as spam) has generated a need for reliable anti-spam filters. If a device is being attacked via spam mails then the problems which can harm a device are :It can fill up storage unknowingly, the attacker can be able to access user's data.

To overcome these issues ,a spam classifier is designed and it has made user to get aware of spam mails. Spam Classifier will identify spam emails by using machine learning algorithms and its technique in which SVM(Support Vector Machine) algorithm works at very successful rate.

Working and Methodology of Spam classifier:

The data is acquired and collected. The collected data is further gone through word cloud, stemming and lemmatization to eliminate common prefixes and suffixes of word in order to convert the words to its base form. The resultant is divided into training and test sets. Then the data pre-processing is performed on the train data, which takes care of missing and erroneous values in the dataset.

Now, in the feature extraction procedure the features are selected and modified. Using SVM algorithm technique, model is built on the train data set. The constructed model is tested on the test data set. Finally based on the accuracy, the model can be used for the spam identification.

Table of Contents

PROBLEM STATEMENT

Spamming is one of the major attacks that accumulate the large number of compromised machines by sending unwanted messages, viruses and phishing through emails

Nowadays there are lot of people trying to fool you just by sending you false e-mails like you have won 1000 dollars, this much amount is deposited in your account once you open this link then they will try to hack your information.

Sometimes relevant e-mails are considered as spam e-mails.

Problem faced due to spam:

- Unwanted email irritating Internet consumers.
- Critical email messages are missed and/or delayed.
- Loss of Internet performance and bandwidth.
- Billions of dollars lost worldwide.
- Increase in Worm and Trojan Horses.
- Spam can crash mail servers and fill up hard drives.

CHAPTER 1: INTRODUCTION

Recently unsolicited bulk e-mail which is also known as spam, has become a big trouble over internet. In recent statistics 45% of all e-mails are spam which is about 15.4 billion emails per day. Usually most of the spam are dealt by blocking e-mails coming from certain addresses or filtering messages with certain subject lines. Spammers began to use several tricky methods to overcome the filtering methods like using random sender addresses or append random characters to the beginning or the end of the message subject line. Knowledge engineering and machine learning are two general approaches used in e-mail filtering to identify spam e-mails. In knowledge engineering approach a set of rules has to be specified according to which e-mails are classified as spam or ham. A set of such rules should be created either by the user of the filter, or by some other authority (e.g. the software company that provides a particular rule-based spam-filtering tool). By applying this method, no promising results shows because the rules must be constantly updated and maintained, which is a waste of time and it is not convenient for most users. Machine learning approach is more efficient than knowledge engineering approach; it does not require specifying any rules. Instead, a set of training samples, these samples is a set of pre-classified e-mail messages. A specific algorithm is then used to learn the classification rules from these e-mail messages. Machine learnig approach has been widely studied and there are lots of algorithms can be used in e-mail filtering.

CHAPTER 2: LITERATURE SURVEY

Priti Kulkarni, Dr. Haridas Acharya, *Comparative analysis of classifiers for header based emails classification using supervised learning*, International Research Journal of Engineering and Technology (IRJET). Volume: 03 Issue: 03 | Mar-2016

The author of this paper have used Decision tree, Bayes network, K-Nearest Neighbor, Random Forest and Bagging algorithms to test spam classification using email header fields. Result shows that decision tree (J48) is very simple and performs better than all classifiers. K-nearest neighbor also performs good but bagging and random forest does not show promising result.

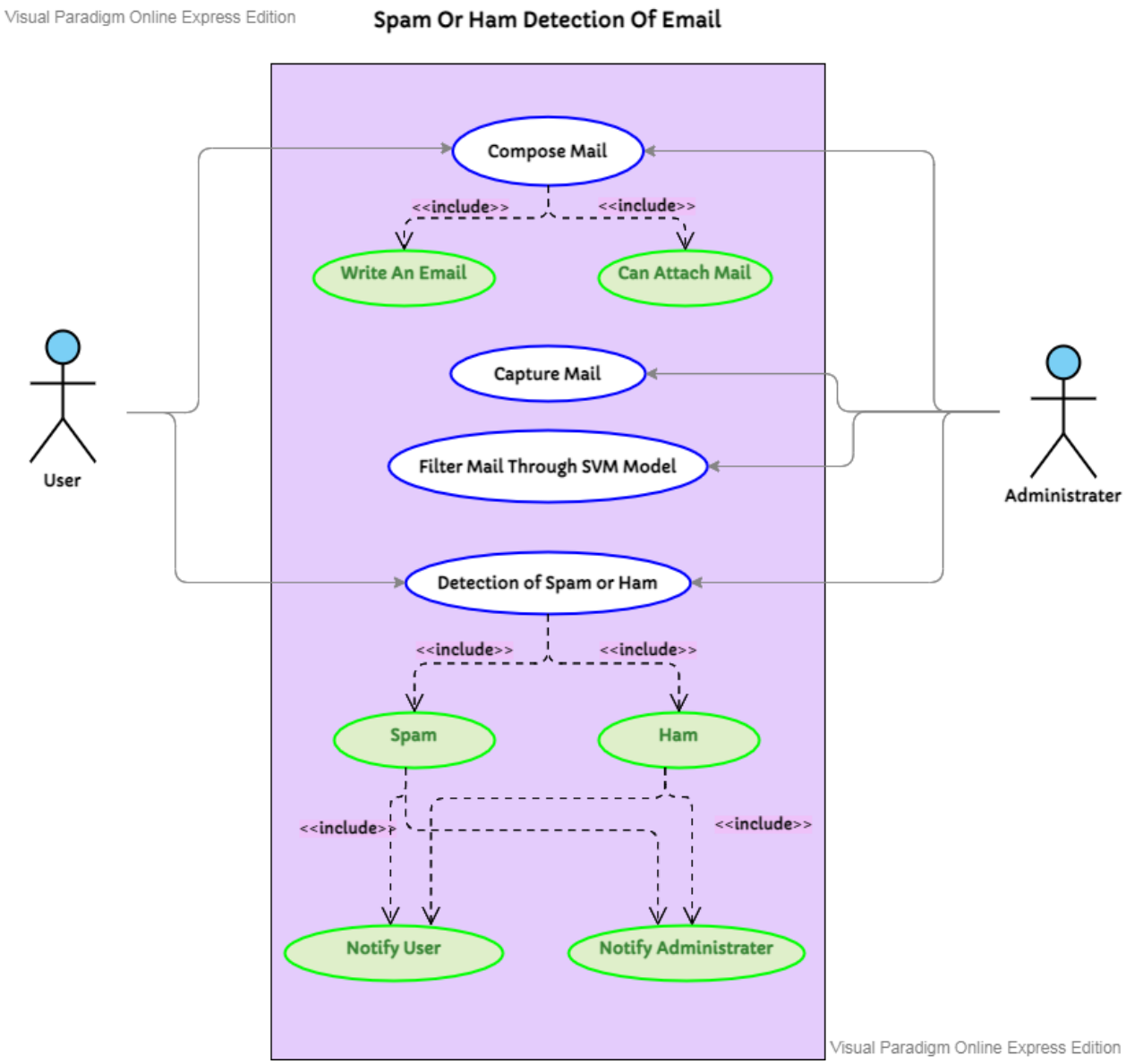
Shripriya Dongre, Prof. Kamlesh Patidar, *A Survey: E-Mail Spam Classification using Machine Learning Techniques*, International Journal of Science, Engineering and Technology

This survey paper elaborates different Existing Spam Filtering system through Machine learning techniques by exploring several methods, concluding the overview of several Spam Filtering techniques and summarizing the accuracy of different proposed approach regarding several parameters

V.Christina, S.Karpagavalli, G.Suganya,*Email Spam Filtering using Supervised Machine Learning Techniques*, International Journal on Computer Science and Engineering,Vol. 02, No. 09, 2010, 3126-3129

In our work, we generated spam and legitimate message corpus from the latest mails and employed machine learning techniques to build the model. The performance of the model is evaluated using 10-fold cross validation and observed that Multilayer Perceptron classifier out performs other classifiers and the false positive rate also very low compared to other alogorithms.

CHAPTER 3: DETAILED DESIGN



CHAPTER 4: PROJECT SPECIFIC REQUIREMENTS

- Email dataset as an input for the training model.
- Processor : Intel Core i3 or higher
- RAM: 4gb
- Hard disk: 16GB or more
- Operating System: Windows 10
- IDE: PyCharm / Jupiter Notebook
- Programming language: Python

CHAPTER 5: IMPLEMENTATION

```
import os
from collections import Counter

folder='email/'

files=os.listdir(folder)
len(files)

%config IPCompleter.greedy=True

emails=[folder + file for file in files]
emails

del emails[0]
emails

words=[]

for email in emails:
    f=open(email , encoding='latin-1')
    blob=f.read()
    words+=blob.split(" ")
    #words=words+blob.split()

for i in range(len(words)):
    if not words[i].isalpha():
        words[i]=""

word_dict=Counter(words)
len(word_dict)
```

```
del word_dict[""]

word_dict=word_dict.most_common(900)

len(word_dict)
for i in word_dict:
   print(i[0])

features=[]
labels=[]
for email in emails:
      f=open(email, encoding='latin-1')
      blob=f.read().split(" ")
      data=[]
      for i in  word_dict:
            data.append(blob.count(i[0]))
      features.append(data)

      if 'spam' in email:
            labels.append(1)
      if 'ham' in  email:
            labels.append(0)

len(features)

len(labels)

import numpy as np

features=np.array(features)
features.shape

labels=np.array(labels)
labels.shape
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(features , labels , test_size=0.2 , random_state=3)

X_train.reshape(-1,1)
y_train.reshape(-1,1)

from sklearn.naive_bayes import MultinomialNB
classifier=MultinomialNB()

classifier.fit(X_train,y_train)

new_email="""Your mobile no has won a cash prize of rupees 1 lakh and a gift of cost rupees
5000.click here to continue"""

sample=[]
for i in word_dict:
   sample.append(new_email.split(" ").count(i[0]))

sample=np.array(sample)

classifier.predict(sample.reshape(1,900))

new_email="""Hi the team saw your presentation on marketting and they are really happy.When will
you be for a coffee?"""

sample=[]
for i in word_dict:
   sample.append(new_email.split(" ").count(i[0]))

sample=np.array(sample)

classifier.predict(sample.reshape(1,900))

y_pred=classifier.predict(X_test)
y_pred
```

```
from sklearn.metrics import accuracy_score
accuracy_score(y_pred,y_test)

from sklearn.svm import SVC
classifier=SVC()

classifier.fit(X_train,y_train)

new_email="""Your mobile no has won a cash prize of rupees 1 lakh and a gift of cost rupees
5000.click here to continue"""

sample=[]
for i in word_dict:
   sample.append(new_email.split(" ").count(i[0]))

sample=np.array(sample)

classifier.predict(sample.reshape(1,900))

new_email="""Hi the team saw your presentation on marketting and they are really happy.When will
you be for a coffee?"""

sample=[]
for i in word_dict:
   sample.append(new_email.split(" ").count(i[0]))

sample=np.array(sample)

classifier.predict(sample.reshape(1,900))

y_pred=classifier.predict(X_test)
y_pred

from sklearn.metrics import accuracy_score
accuracy_score(y_pred,y_test)
```

CHAPTER 6: RESULTS

```
In [530]: import os
          from collections import Counter

In [531]: folder='email/'

In [532]: files=os.listdir(folder)
          len(files)

Out[532]: 3055
```

```
In [533]: emails=[folder + file for file in files]
          emails

Out[533]: ['email/.ipynb_checkpoints',
           'email/0001.1999-12-10.farmer.ham.txt',
           'email/0002.1999-12-13.farmer.ham.txt',
           'email/0003.1999-12-14.farmer.ham.txt',
           'email/0004.1999-12-14.farmer.ham.txt',
           'email/0005.1999-12-14.farmer.ham.txt',
           'email/0006.2003-12-18.GP.spam.txt',
           'email/0007.1999-12-14.farmer.ham.txt',
           'email/0008.2003-12-18.GP.spam.txt',
           'email/0009.1999-12-14.farmer.ham.txt',
           'email/0010.1999-12-14.farmer.ham.txt',
           'email/0011.1999-12-14.farmer.ham.txt',
           'email/0012.1999-12-14.farmer.ham.txt',
           'email/0013.1999-12-14.farmer.ham.txt',
           'email/0014.1999-12-15.farmer.ham.txt',
           'email/0015.1999-12-15.farmer.ham.txt',
           'email/0016.1999-12-15.farmer.ham.txt',
           'email/0017.2003-12-18.GP.spam.txt',
           'email/0018.2003-12-18.GP.spam.txt',
```

```
In [534]: del emails[0]
          emails

Out[534]: ['email/0001.1999-12-10.farmer.ham.txt',
           'email/0002.1999-12-13.farmer.ham.txt',
           'email/0003.1999-12-14.farmer.ham.txt',
           'email/0004.1999-12-14.farmer.ham.txt',
           'email/0005.1999-12-14.farmer.ham.txt',
           'email/0006.2003-12-18.GP.spam.txt',
           'email/0007.1999-12-14.farmer.ham.txt',
           'email/0008.2003-12-18.GP.spam.txt',
           'email/0009.1999-12-14.farmer.ham.txt',
           'email/0010.1999-12-14.farmer.ham.txt',
           'email/0011.1999-12-14.farmer.ham.txt',
           'email/0012.1999-12-14.farmer.ham.txt',
           'email/0013.1999-12-14.farmer.ham.txt',
           'email/0014.1999-12-15.farmer.ham.txt',
           'email/0015.1999-12-15.farmer.ham.txt',
           'email/0016.1999-12-15.farmer.ham.txt',
           'email/0017.2003-12-18.GP.spam.txt',
           'email/0018.2003-12-18.GP.spam.txt',
           'email/0019.1999-12-15.farmer.ham.txt',
```

In [535]:
```
words=[]

for email in emails:
    f=open(email , encoding='latin-1')
    blob=f.read()
    words+=blob.split(" ")
    #words=words+blob.split()
```

In [536]:
```
for i in range(len(words)):
    if not words[i].isalpha():
        words[i]=""
```

In [537]:
```
word_dict=Counter(words)
len(word_dict)
```

Out[537]: 32993

In [538]:
```
del word_dict[""]
```

In [539]:
```
word_dict=word_dict.most_common(900)
```

In [540]:
```
len(word_dict)
for i in word_dict:
    print(i[0])
```

```
the
to
ect
and
for
of
a
hou
you
on
is
in
this
enron
i
be
that
will
have
```

In [541]:
```
features=[]
labels=[]
for email in emails:
        f=open(email, encoding='latin-1')
        blob=f.read().split(" ")
        data=[]
        for i in  word_dict:
                data.append(blob.count(i[0]))
        features.append(data)

        if 'spam' in email:
                labels.append(1)
        if 'ham' in  email:
                labels.append(0)
```

In [542]:
```
len(features)
```
Out[542]: 3054

In [543]:
```
len(labels)
```
Out[543]: 3054

In [544]:
```
import numpy as np
```

In [545]:
```
features=np.array(features)
features.shape
```
Out[545]: (3054, 900)

In [546]:
```
labels=np.array(labels)
labels.shape
```
Out[546]: (3054,)

In [547]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(features , labels , test_size=0.2 , random_state=3)
```

In [548]:
```
X_train.reshape(-1,1)
y_train.reshape(-1,1)
```
Out[548]:
```
array([[0],
       [1],
       [0],
       ...,
       [1],
       [0],
       [1]])
```

In [549]:
```
from sklearn.naive_bayes import MultinomialNB
classifier=MultinomialNB()
```

In [550]:
```
classifier.fit(X_train,y_train)
```
Out[550]: MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)

In [551]:
```
new_email="""Your mobile no has won a cash prize of rupees 1 lakh and a gift of cost rupees 5000.click here to continue"""
```

In [552]:
```
sample=[]
for i in word_dict:
    sample.append(new_email.split(" ").count(i[0]))
```

In [553]:
```
sample=np.array(sample)
```

In [554]:
```
classifier.predict(sample.reshape(1,900))
```
Out[554]: array([1])

In [555]:
```
new_email="""Hi the team saw your presentation on marketting and they are really happy.When will you be for a coffee?"""
```

In [556]:
```
sample=[]
for i in word_dict:
    sample.append(new_email.split(" ").count(i[0]))
```

In [557]:
```
sample=np.array(sample)
```

In [558]:
```
classifier.predict(sample.reshape(1,900))
```
Out[558]: array([0])

In [559]:
```
y_pred=classifier.predict(X_test)
y_pred
```

Out[559]:
```
array([0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0,
       1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0,
       0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0,
       0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0,
       0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0,
       1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0,
       1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1,
       0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0,
       1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1,
       1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1,
       0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0,
       1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0,
       0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1,
       1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0])
```

In [560]:
```
from sklearn.metrics import accuracy_score
accuracy_score(y_pred,y_test)
```

Out[560]:
```
0.9099836333878887
```

In [561]:
```
from sklearn.svm import SVC
classifier=SVC()
```

In [562]:
```
classifier.fit(X_train,y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\base.py:193: FutureWarning: The default value of gamma will change from
'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explicitly to 'auto' or 'scale' to avoid t
his warning.
  "avoid this warning.", FutureWarning)
```

Out[562]:
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='rbf', max_iter=-1, probability=False, random_state=None,
    shrinking=True, tol=0.001, verbose=False)
```

In [563]:
```
new_email="""Your mobile no has won a cash prize of rupees 1 lakh and a gift of cost rupees 5000.click here to continue"""
```

In [564]:
```
sample=[]
for i in word_dict:
    sample.append(new_email.split(" ").count(i[0]))
```

In [565]:
```
sample=np.array(sample)
```

In [566]:
```
classifier.predict(sample.reshape(1,900))
```

Out[566]:
```
array([1])
```

In [567]:
```
new_email="""Hi the team saw your presentation on marketting and they are really happy.When will you be for a coffee?"""
```

In [568]:
```
sample=[]
for i in word_dict:
    sample.append(new_email.split(" ").count(i[0]))
```

In [569]:
```
sample=np.array(sample)
```

In [570]:
```
classifier.predict(sample.reshape(1,900))
```

Out[570]:
```
array([0])
```

In [571]:
```
y_pred=classifier.predict(x_test)
y_pred
```

Out[571]:
```
array([0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0,
       1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0,
       0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0,
       1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1,
       0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1,
       0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0,
       1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1,
       0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0,
       1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1,
       1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0])
```

In [572]:
```
from sklearn.metrics import accuracy_score
accuracy_score(y_pred,y_test)
```

Out[572]:
```
0.9279869067103109
```

CHAPTER 7: CONCLUSION and FUTURE SCOPE

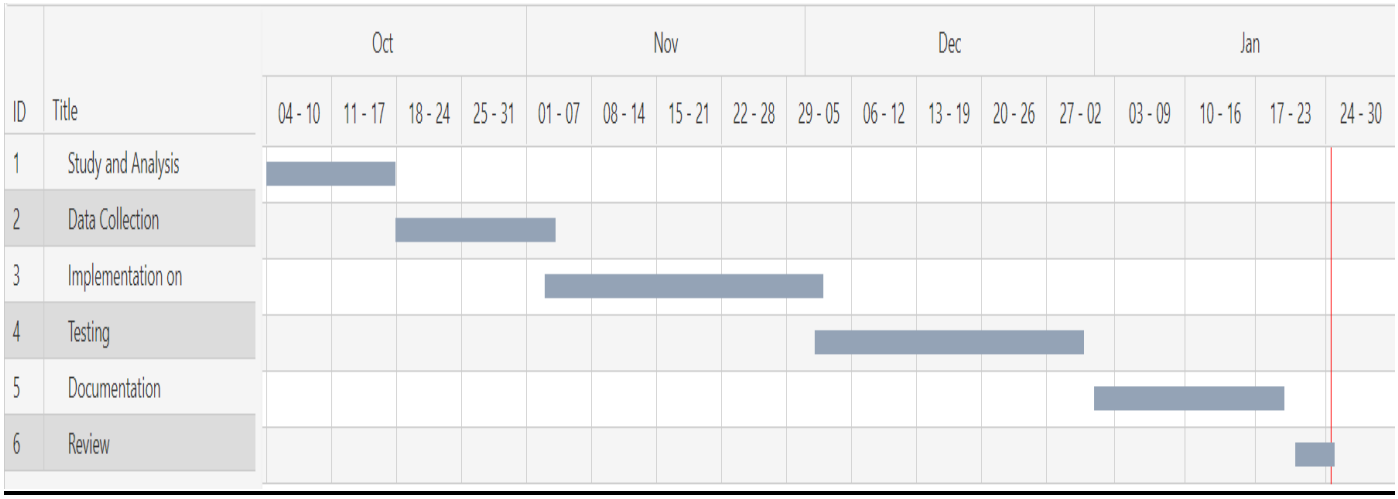
Conclusion

- E-mail spam filtering is an important issue in the network security and machine learning techniques.
- Machine learning algorithms have been applied in the field of spam filtering.
- In this study, we reviewed machine learning approaches and their application to the field of spam filtering.
- The basic design of email spam filter and the processes involved in filtering spam emails were looked into.
- Further research to enhance the effectiveness of spam filters need to be done.

Future Scope

The current proposed system is for English language mails but as future scope we can design the system for multiple languages.

Modules and Requirements completed are



REFERENCES

1] logsat.com[online] in 2009, “Spam Filter ISP System Requirement”

2] emregedikoglu, jahk 1, fstasel in 20 Nov 2019 , “CankayaUniversity / ceng- 407-408-2019-Spam-SMS-Detection”

3] shradhanjali, Toran verma, “Email spam Detection and classification using SVM and Feature extraction” , International journal of Advance research , ideas and innovations in technology volume 3 issue 2

4] Nabin jamkatel, Rajiv gupta, Rakesh chetri, sabina lamichhanne in 2019 , Asian school of management and technology, “Spam Email Identification”