A

Project Report

On

EMAIL SPAM DETECTION USING MACHINE LEARNING ALGORITHMS

Submitted to

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES,RK VALLEY

in partial fulfillment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY IN
ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

M.Mani Varsha	**(R170425)**
K.Ramani	**(R170692)**
R.Reddy Rani	**(R170434)**

Under the Guidance of

P.Janardhana Reddy
Assistant professor, Department of ECE.

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

Idupulapaya, VEMPALLI – 516330

2019-2023

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

Idupulapaya, VEMPALLI – 516330

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the project report entitled **"EMAIL SPAM DETECTION USING MACHINE LEARNING ALGORITHMS,"** a bonafide record of the project work done and submitted by

M.Mani Varsha (R170425)
R.Reddy Rani (R170434)
K.Ramani (R170692)

for the partial fulfillment of the requirements for the award of B.Tech. Degree in **ELECTRONICS AND COMMUNICATION ENGINEERING, Rajiv Gandhi University of Knowlegde Technologies,Rk Valley.**

GUIDE
P.JANARDHANA REDDY
Asst.Professor

Head of the Department
B.MADAN MOHAN
Asst.Professor

External Viva-Voce Exam Held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We hereby declare that the project report entitled **“Email Spam Detection using Machine Learning Algorithms”** submitted to the Department of **ELECTRONICS AND COMMUNICATION ENGINEERING** in partial fulfillment of requirements for the award of the degree of **BACHELOR OF TECHNOLOGY**. This project is the result of our own effort and that it has not been submitted to any other University or Institution for the award of any degree or diploma other than specified above.

M.Mani Varsha	**R170425**
R.Reddy Rani	**R170434**
K.Ramani	**R170692**

ACKNOWLEDGEMENT

We are thankful to our guide **Mr.P.Janardhana Reddy** for his valuable guidance and encouragement. His helping attitude and suggestions have helped us in the successful completion of the project.

We would like to express our gratefulness and sincere thanks to **Mr.B.Madhan Mohan**, Head of the Department of ELECTRONICS AND COMMUNICATION ENGINEERING, for his kind help and encouragement during the course of our study and in the successful completion of the project work.

We have great pleasure in expressing out hearty thanks to our beloved Director **K.Sandhya Rani** for spending valuable time with us to complete the project

Successful completion of any project cannot be done without proper support and encouragement. We sincerely thanks to the **Management** for providing all the necessary facilities during the course of study.

We would like to thank our parents and friends, who have the greatest contributions in all our achievements, for the great care and blessings in making us successful in all our endeavors.

M.Mani Varsha **(R170425)**
R.Reddy Rani **(R170434)**
K.Ramani **(R170692)**

I. ABSTRACT

Every day, the rate of spam emails and spam messages is increasing. Such spam emails are mostly sent by people to earn income or for any advertisement for their benefit. This increasing amount of spam mail causes traffic congestion and waste of time for those who are receiving that spam mail. The real cost of spam emails is very much higher than one can imagine. Sometimes, the spam emails also have some links which have malware. And also, some people will get irritated once they see their inbox which is having more spam mails. Sometimes, the users easily get trapped into financial fraud actions, by seeing the spam mails such as job alert mails and commercial mails and offer emails. It may also cause the person to have some mental stress. To reduce all these risks, the system has proposed a machine learning model which will detect spam mail and non-spam emails.The aim of this project is to develop a system which can perform early detection of spam mails with a higher accuracy by combining the results of different machine learning techniques. The algorithms like Neural Networks,Naive Bayes,Support vector classifier,Logistic Reggression are used. The accuracy of the model using each of the algorithms is calculated. Then the one with a good accuracy is taken as the model for Email Spam Detection.

Keywords:
• Neural Networks
• Naive Bayes
• Support vector classifier
• Logistic Reggression
• Spam email

II. TABLE OF CONTENTS

III. LIST OF FIGURES

1.INTRODUCTION

Technology has become a vital part of life in today's time.With each passing day, the use of the internet increases exponentially, and with it, the use of email for the purpose of exchanging information and communicating has also increased, it has become second nature to most people.

While e-mails are necessary for everyone, they also come with unnecessary, undesirable bulk mails, which are also called Spam Mails. Anyone with access to the internet can receive spam on their devices. Most spam emails divert people's attention away from genuine and important emails and direct them towards detrimental situations. Spam emails are capable of filling up inboxes or storage capacities, deteriorating the speed of the internet to a great extent. These emails have the capability of corrupting one's system by smuggling viruses into it, or steal useful information and scam gullible people. The identification of spam emails is a very tedious task and can get frustrating sometimes.

While spam detection can be done manually, filtering out a large number of spam emails can take very long and waste a lot of time. Hence, the need for spam detection softwares has become the need of the hour. To solve this problem,various spam detection techniques are used now. The most common technique for spam detection is the utilization of Naive Bayesian method and feature sets that assess the presence of spam keywords. The development of spam detection with the help of Naive Bayesian method, resulting in almost 98.8% accuracy.

2.OBJECTIVE

The aim of this project is to develop a system which can perform Detection of Spam mails.To give knowledge to the user about fake emails and relevant emails.To classify that mail is spam or not.

3.LITERATURE REVIEW

An approach using random forest algorithm approach is proposed by Akinyelu and Adewumi in order to identify the phishing or spam emails. It used 200 emails. The main motto of research was to reduce features and increase efficiency/accuracy. Accuracy of up to 99.7% with a minimal amount of 0.06% false positives is achieved by the proposed algorithm.

The research only covered the classification aspect without considering vital information which can affect the results, especially, in case of limited text in the email.

Yüksel et al aimed to resolve the problem of spam by inhibiting the spam emails from being spread within the email systems. To achieve this, they propose a cloud base system, which involves the identification of spam emails using analytics and machine learning algorithms like support vector machines and decision trees. The results of the tests show that the SVM leads to a higher accuracy of up to 97.6% and a false-positive rate of 2.33%. The decision tree attains a lower accuracy of 82.6% and a false-positive rate of 17.3%. Results reveal that the increase in spam emails is affected by the no. of received emails. Lee et al proposed an optimal technique for spam detection.

4.Existing Methods

It is important to find an algorithm that gives the best possible outcome for any particular metric for correct classification of emails and spam or ham.The present systems of spam detection are reliant on three major methods:-

4.1 Linguistic Based Methods:

Unlike humans, who can grasp linguistic constructs along with their exposition, machines cannot and hence it is necessary to teach machines some languages to help them understand these constructs. This is the technique that is used in places like search engines in order to ascertain the next terms for suggestions to the user while they are typing their search. Sentences are divided into two Unigrams (words taken are one by one) and two Bigrams (words that are taken two at a time). Since this technique requires that every expression be remembered, this method is not feasible and also time-intensive.

4.2 Behavior-Based Methods:

This technique is Metadata-based. This approach requires that users generate a set of rules, and the users must have a thorough understanding of these rules. Since the attributes of spam change over time so the rules also need to be reformed from time to time. As a result, it still requires a human to scrutinise the details and is majorly user-dependent.

4.3. Graph-Based Methods:

This technique uses a single graphical representation by incorporating numerous, heterogeneous particulars. Graph-based anomaly recognition algorithms are executed which detect abnormal forms in the data showing behaviours of spammers. This method is not dependable, so it is taxing to recognise faulty opinions. Feature Engineering mostly depends on the commercial appeal of terms and is absolutely content-oriented, and does not depend on statistics. All these attributes lead to a noteworthy decline of this structure.

5.Proposed method

The dataset is taken from SpamAssassin , 2500 non-spam messages belong to easy_ham and they should be easily differentiated from spam. Instead of using sophisticated and hybrid models, this study relies on relatively simple classification algorithms to solve this problem like Logistic Regression, Naive Bayes, and Support Vector Machine. The concept of Neural Networks is also used to select the best activation function for spam detection. The dataset is in the form of HTML files which are converted into plaintext during text preprocessing.

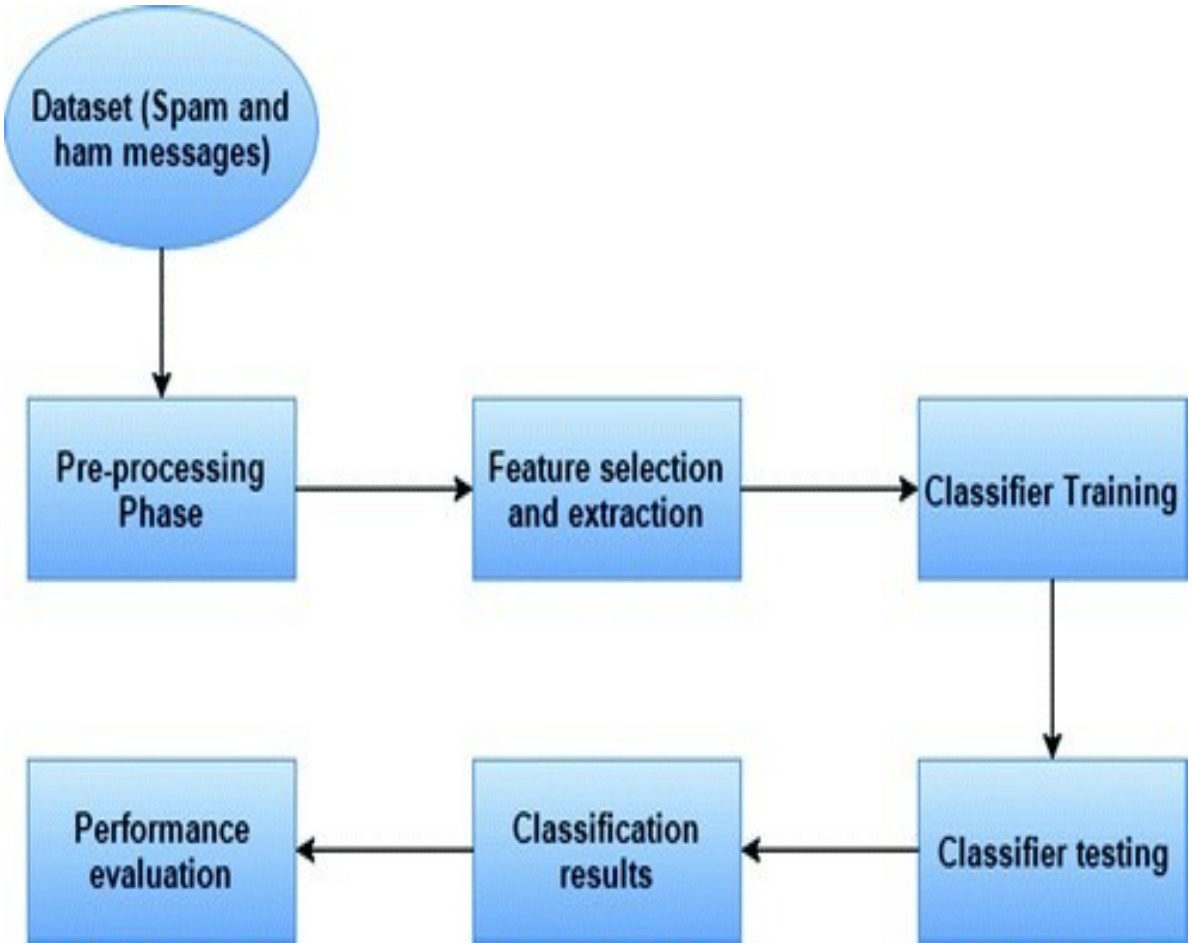


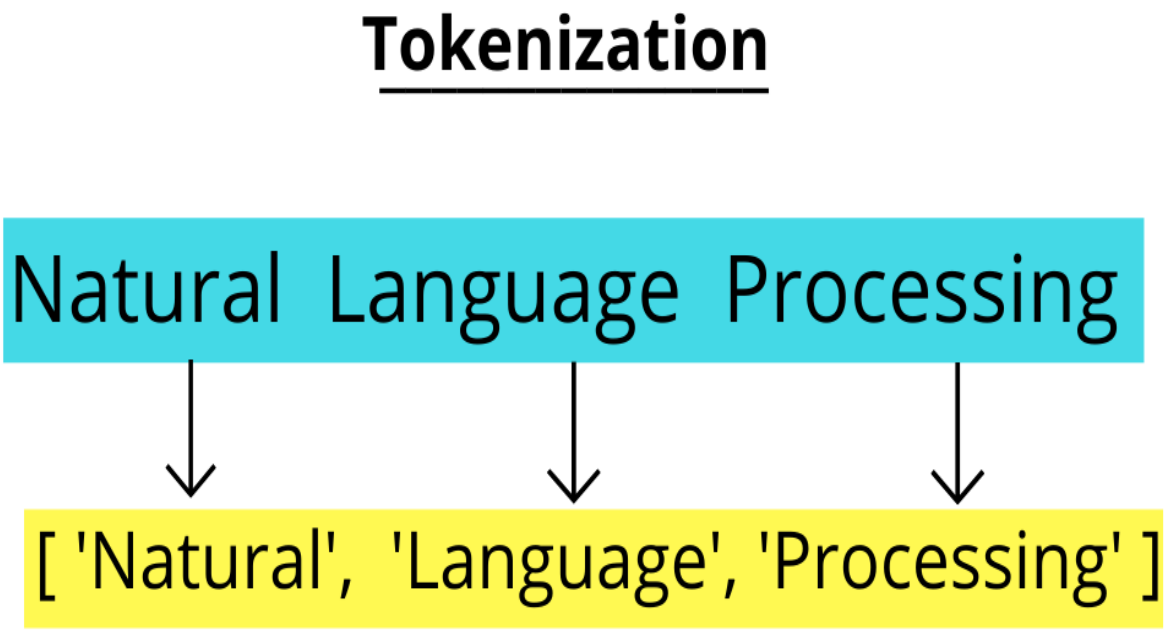
fig **5.1. Flow of working**

1.Data Preprocessing:-

- Lower Case
- Tokenization
- Removing Special Characters
- Removing Stop Words and Punctuation
- Lemmatization and Stemming

A.Tokenization:

Tokenization is the process of breaking a stream of text up into words,phrases,symbols,or other meaningful elements called Tokens. The list of Tokens becomes input for further processing such as parsing or text mining.



B.Removal of Stop Word:

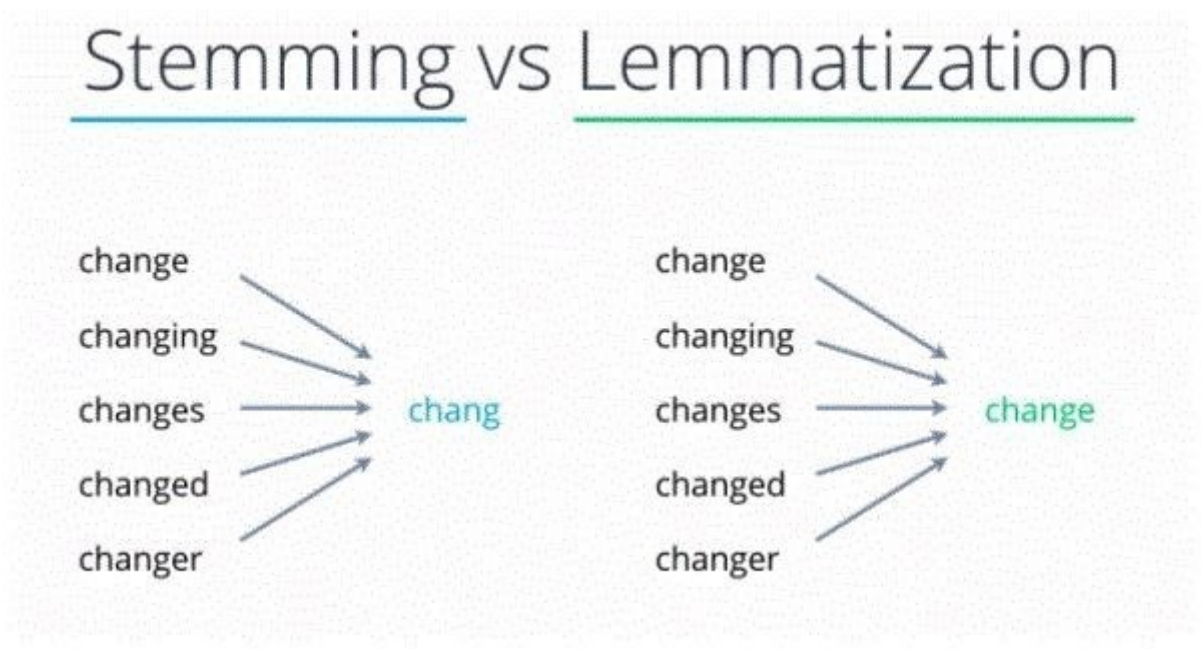
Sometimes,the extremely common word which would appear to be of very littel value in helping select documents matching user need are excluded from the vocabulary entirely.

Sample text with Stop Words	Without Stop Words
GeeksforGeeks – A Computer Science Portal for Geeks	GeeksforGeeks , Computer Science, Portal ,Geeks
Can listening be exhausting?	Listening, Exhausting
I like reading, so I read	Like, Reading, read

C. Stemming and Lemmatization:-

Stemming is the process of removing the last few characters of a given word,to obtain a shorter form,even if that form doesn't have any meaning.

Lemmatization in linguistics,is the process of grouping together the different inflected forms of a word so they can be analysed as a single item.



2. Feature Selection and Extraction:

● Number of Capitilized words.

● Sum of all the character length of words.

● Number of words containing letters and numbers.

●Max of ratio of digital characters to all the characters of each word.

●Sum of all the character lenghts of words,etc.....

3. Classifier Training:

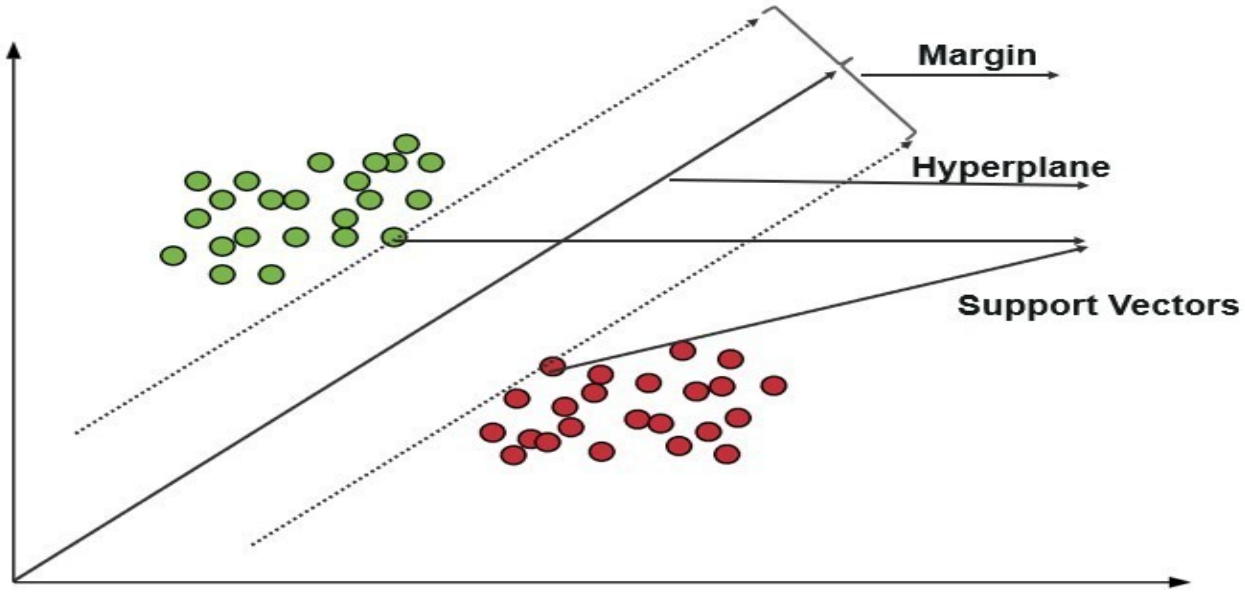
A.Support Vector Machine.

B.Naive Bayes.

C.Logistic Regression.

5.1.Support vector Machine:-

In machine learning support-vector machines(SVMs, also vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model linear that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). A SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.



Steps to implement:

1.Data Pre-processing step

2.Fitting Support Vector Machine to the Training set

3.Predicting the test result

4.Test accuracy of the result(Creation of Confusion matrix)

5.Visualizing the test set result

5.2.NAIVE BAYES:-

Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

Why is it called Naïve Bayes?

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

Naïve: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.

Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem.

Bayes' Theorem:

Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

The formula for Bayes' theorem is given as:

$$P(A|B)= \frac{P(B|A)P(A)}{P(B)}$$

Where,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

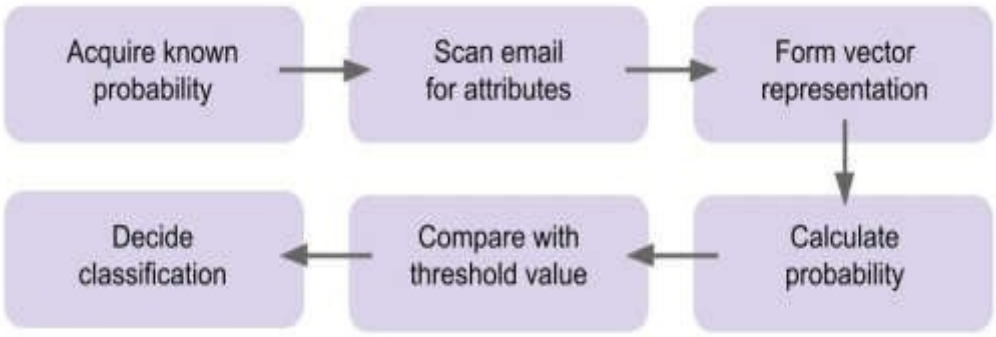
P(B) is Marginal Probability: Probability of Evidence.

Working of Naïve Bayes' Classifier:

1. Convert the given dataset into frequency tables.

2.Generate Likelihood table by finding the probabilities of given features.

3.Now, use Bayes theorem to calculate the posterior probability.



Advantages of Naïve Bayes Classifier:

1.Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.

2.It can be used for Binary as well as Multi-class Classifications.

3.It performs well in Multi-class predictions as compared to the other Algorithms.

4.It is the most popular choice for text classification problem.

Disadvantages of Naïve Bayes Classifier:

1.Naive Bayes assumes that all features are independent or unrelated, so it cannot learn the relationship between features.

Applications of Naïve Bayes Classifier:

1.It is used for Credit Scoring.

2.It is used in medical data classification.

3.It can be used in real-time predictions because Naïve Bayes Classifier is an eager learner.

4.It is used in Text classification such as Spam filtering and Sentiment analysis.

Types of Naïve Bayes Model:

There are three types of Naive Bayes Model, which are given below:

Gaussian: The Gaussian model assumes that features follow a normal distribution. This means if predictors take continuous values instead of discrete, then the model assumes that these values are sampled from the Gaussian distribution.

Multinomial: The Multinomial Naïve Bayes classifier is used when the data is multinomial distributed. It is primarily used for document classification problems, it means a particular document belongs to which category such as Sports, Politics, education, etc.The classifier uses the frequency of words for the predictors.

Bernoulli: The Bernoulli classifier works similar to the Multinomial classifier, but the predictor variables are the independent Booleans variables. Such as if a particular word is present or not in a document. This model is also famous for document classification tasks.

Steps to implement:

1.Data Pre-processing step

2.Fitting Naive Bayes to the Training set

3.Predicting the test result

4.Test accuracy of the result(Creation of Confusion matrix)

5.Visualizing the test set result

5.3.C.LOGISTIC REGRESSION:-

Logistic regression predicts the output of a categorical dependentvariable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1. Logistic Regression is much similar to the LinearRegression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems. In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1). Logistic regression is

one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc. Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:

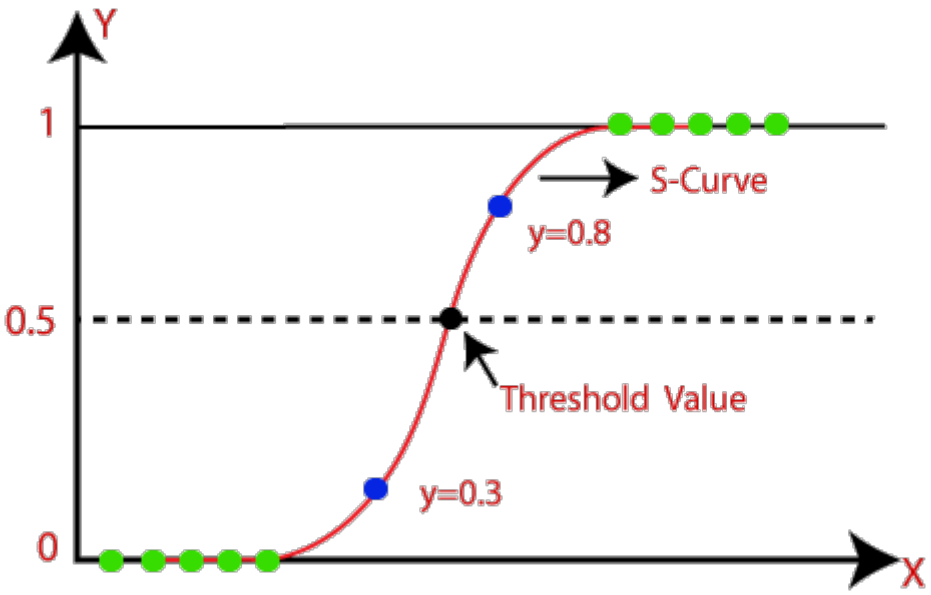


fig **5.3.C.** Sigmoid Function

Logistic Function (Sigmoid Function):

The sigmoid function is a mathematical function used to map the predicted values to probabilities. It maps any real value into another value within a range of 0 and 1. The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function. In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

Assumptions for Logistic Regression:

1.The dependent variable must be categorical in nature.

2.The independent variable should not have multi-collinearity.

Type of Logistic Regression:

On the basis of the categories, Logistic Regression can be classified into three types:

Binomial: In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.

Multinomial: In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep" .

Ordinal: In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

Steps to implement:

1.Data Pre-processing step

2.Fitting Logistic regression to the Training set

3.Predicting the test result

4.Test accuracy of the result(Creation of Confusion matrix)

5.Visualizing the test set result

6.ACCURACY

6.1.CODE

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
df = pd.read_csv('mail_data(1).csv')
print(df.sample(5))
print(df.shape)
print(df.info)
print(df.sample(5))
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
df['Category'] = encoder.fit_transform(df['Category'])
print(df.head())
print(df['Category'].value_counts())
import matplotlib.pyplot as plt
plt.pie(df['Category'].value_counts(),
labels=['ham','spam'],autopct='%0.2f')
plt.show()
import nltk
df['num_characters'] = df['Message'].apply(len)
print(df.head())
df['num_words']
=
df['Message'].apply(lambda
x:len(nltk.word_tokenize(x)))
print(df.head())
df['num_sentences']
=
df['Message'].apply(lambda
x:len(nltk.sent_tokenize(x)))
print(df.head())
```

```
df[['num_characters','num_words','num_sentences']].describe()
df[df['Category'] ==0]
[['num_characters','num_words','num_sentences']].describe()
df[df['Category']== [['num_characters','num_words','num_sentences']].describe()
import seaborn as sns
plt.figure(figsize=(12,6))
sns.histplot(df[df['Category'] == 0]['num_characters'])
sns.histplot(df[df['Category'] == 1]['num_characters'],color='red')
plt.figure(figsize=(12,6))
sns.histplot(df[df['Category'] == 0]['num_words'])
sns.histplot(df[df['Category'] == 1]['num_words'],color='red')
sns.heatmap(df.corr(),annot=True)
from nltk.corpus import stopwords
import string
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize
ps = PorterStemmer()
def transform_text(Message):
Message= Message.lower()
Message= nltk.word_tokenize(Message)
y = []
for i in Message:
if i.isalnum():
y.append(i)
Message= y[:]
y.clear()
for i in Message:
if i not in stopwords.words('english') and i not in string.punctuation:
y.append(i)
Message= y[:]14
y.clear()
for i in Message:
y.append(ps.stem(i))
return " ".join(y)
```

```
transform_text("I'm gonna be home soon and i don't want to talk about this
stuff anymore tonight, k? I've cried enough today.")
df['Message'][10]
df['transformed_text'] = df['Message'].apply(transform_text)
print(df.head())
from wordcloud import WordCloud
wc=WordCloud(width=500,height=500,min_font_size=10,background_color=' white')
spam_wc =wc.generate(df[df['Category']
['transformed_text'].str.cat(sep=" "))
plt.figure(figsize=(15,6))==1]
plt.imshow(spam_wc)
ham_wc
=
wc.generate(df[df['Category']
['transformed_text'].str.cat(sep=" "))
==
0]
plt.figure(figsize=(15,6))
plt.imshow(ham_wc)
print(df.head())
from
sklearn.feature_extraction.text
CountVectorizer,TfidfVectorizer
cv = CountVectorizer()
tfidf = TfidfVectorizer()
import
X = tfidf.fit_transform(df['transformed_text']).toarray()
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
X=scaler.fit_transform(X)
print(X.shape)
y= df['Category'].values
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test= train_test_split(X,y,test_size=0.2,random_state=2)
```

```
from sklearn.naive_bayes import GaussianNB,MultinomialNB,BernoulliNB
from import sklearn.metrics
accuracy_score,confusion_matrix,precision_score
gnb = GaussianNB()
mnb = MultinomialNB()
bnb = BernoulliNB()
gnb.fit(X_train,y_train)
y_pred1 = gnb.predict(X_test)
print(accuracy_score(y_test,y_pred1))
print(confusion_matrix(y_test,y_pred1))
print(precision_score(y_test,y_pred1))
mnb.fit(X_train,y_train)
y_pred2 = mnb.predict(X_test)
print(accuracy_score(y_test,y_pred2))
print(confusion_matrix(y_test,y_pred2))
print(precision_score(y_test,y_pred2))
bnb.fit(X_train,y_train)
y_pred3 = bnb.predict(X_test)
print(accuracy_score(y_test,y_pred3))
print(confusion_matrix(y_test,y_pred3))
print(precision_score(y_test,y_pred3))
from sklearn.linear_model import LogisticRegression
LR = LogisticRegression()
LR.fit(X_train,y_train)
y_pred = LR.predict(X_test)
print(accuracy_score(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))
print(precision_score(y_test,y_pred))
from sklearn.svm import SVC
SV=SVC()
SV.fit(X_train,y_train)
y_pred = SV.predict(X_test)
print(accuracy_score(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))
```

```
print(precision_score(y_test,y_pred))
# building a predective syste,
input="I've been searching for the right words to thank you for this
breather. I promise i wont take your help for granted and will fulfil my
promise. You have been wonderful and a blessing at all times"
transformed_msg=transform_text(input)
vector_input=tfidf.transform([transformed_msg])
result=mnb.predict(vector_input)[0]
if result==1:
print("spam")
else:
print("Ham")
```

TESTING DATA:

```
input="congratulations you won 1000 on this number you win prize"
transformed_msg=transform_text(input)
vector_input=tfidf.transform([transformed_msg])
result=mnb.predict(vector_input)[0]
if result==1:
print("spam")
else:
print("Ham")
```

Result Analysis:

	Accuracy score	***Precision score***
Support vector machine	0.96	0.94
Navie bayes	0.97	0.96
Logistic regression	0.96	1.0

7.Future Scope:

In the future, we plan to deal with more challenging problems such as the analysis and management of report in spam Emails filters storing. Solution for this problem is another focus of work in the future.

8.Conclusion:

The detection of spam at a place close to the sending server is an important issue in the network security and machine learning techniques have a very important role in this topic. In this paper, we reviewed some machine learning techniques used in spam filters and presented challenges faced by these techniques. We also presented an empirical evaluation in terms of various metrics of three machine learning algorithms namely Naïve Bayes, Support Vector Machine,Logistic Regression.The evaluation was based on data set of e-mail messages collected from different e-mail accounts located on different e-mail servers. Although all learning classifiers showed ability to learn but the NB classifier based filters showed better performance in terms of all measuring parameters. However, none of these classification techniques showed 100% predicative accuracy. The dynamic structure of spam and the reaction of spammers towards spam filters makes spam filtering an active area for research and thus there exists a wide scope for development of new spam filters and improvements in the existing ones.

References

[1]AKINYELU, A. A., & ADEWUMI, A. O. (2014). “Classification of phishing email using random forest machine learning technique”. Journal of Applied Mathematics.

[2] Vinodhini. M, Prithvi. D, Balaji. S “Spam Detection Framework using ML Algorithm” in IJRTE ISSN: 2277- 3878, Vol.8 Issue.6, March 2020.

[3] YUs KSEL, A. S., CANKAYA, S. F., & Us NCUs , It. S. (2017).
“Design of a Machine Learning Based Predictive Analytics System for Spam Problem.” Acta Physica Polonica.

[4] Javatpoint, “Machine Learning Tutorial” 2017 https://www.javatpoint.com/machine- learning.

[5]SpamAssassin, “Spam and Ham Dataset", Kaggle, 2018. https://www.kaggle.com/veleon/ham-and-spam-dataset

[6]SpamAssassin, “Spam Classification Kernel”, 2018 https://www.kaggle.com/veleon/spam-classification