EMAIL SPAM DETECTION USING MACHINE LEARNING

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology/Master of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted by

Raghu Sai K - AP20110010311

Sai Sri Latha K - AP20110010713

Pavan Charan P - AP20110010288

Puneeth T - AP20110010344

Under the Guidance of

Dr. Krishna Prasad

SRM University–AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

December, 2022

1. Certificate

Date: 12-Dec-22

This is to certify that the work present in this Project entitled "**Email Spam Detection Using Machine Learning**" has been carried out by **SAI SRI LATHA,** under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in the **School of Engineering and Sciences**.

Supervisor

(Signature)

Dr. Krishna prasad.

Assistant Professor,

Department of CSE,

SRM University.

2. Acknowledgments

First and foremost, I want to thank our mentor, Dr. Krishna Prasad, and all the people who helped with this initiative. I am grateful to him for introducing me to so many new things.

Secondly, I would like to thank each and every member of my group. The group has taught me much about scientific research and life in general, and each member has given me a broad spectrum of professional and personal advice.

Group Members:

Raghu Sai K

Sai Sri Latha K

Pavan Charan P

Puneeth T

3. Table of Contents

4. Abstract

Email is one of the most popular and widely used legitimate communication methods. Worldwide accessibility, relatively fast message sending Low sending cost. Inadequate Email Logs and Rising Email Volume Business and financial transactions are direct contributors to the rise of email-based threats. Email spam is one of the major problems of today's Internet. Spam detection and filtering is an important and huge problem for email and today's IoT service providers. Multiple machine learning and deep learning techniques were used for this purpose; The Naïve Bayes, Decision Trees, Neural Networks, and Random Forests. Spam can be detected with natural language Processing and machine learning methods. Machine learning methods are Commonly used in spam filtering.

5. List of Tables

6. List of Figures

7. 1. Introduction

Email or email spam is defined as" the use of email for unsolicited transmission." Emails to recipient groups or promotional emails. spam meaning the recipient has not been given permission to receive these emails. "Over the past decade, the popularity of using spam emails has increased. disaster on the internet. Spam consumes disk space, time, and message speed. Automated email filtering is the most effective method It detects spam, but modern spammers can easily bypass all these spam filtering applications. Years ago, most spam emails could be blocked manually from a specific email address. A machine learning approach is used for spam detection.

Filtering is one of the most popular approaches developed to stop spam. important technique. Much spam filter research focuses on more. A challenging classifier-related problem. Machine learning for spam classification has become a major research topic in recent days. Effectiveness of the proposed work in investigating and identifying the use of various learning algorithms for classifying spam messages from email. These methods are used to classify emails as spam (valid messages) or spam (unwanted messages) using machine learning classifiers. The most common spam detection techniques use Naive Bayes and a set of functions that evaluate the presence of spam keywords.

1.1 Spam And Ham

" Ham" This term was coined by Spam Bayes Around 2001, defined as" generally unwanted email." It is not considered spam."

Spam is an economically viable type of commercial advertising as email can be a very cheap medium for senders. Spam is usually caused by sharing our email addresses on unauthorized or malicious websites. Spam has many effects. A bunch of silly emails fills up your inbox. Your internet speed will drop significantly. It steals useful information such as contact list details. Change search results in any computer program. Spam is very time-consuming and can quickly become very frustrating when received in bulk.

8. 2. Methodology

We have taken a dataset from the Kaggle website and done some Data Analysis like text transformation, Feature Extraction. As it is a text dataset, in-text transformation, we removed stop words, punctuations, and after done stemming.

After that, uses the NLTK package to create machine-readable text from text sources. readable style. Utilize tokenization for analyzing data. phase. Stem to the last word recovered and remove the stop words, record. Essentially, this is a word processor.

We have downloaded the CSV file of the spam-message dataset. Let us have a look at the dataset we took from Kaggle,

	Category	Message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

Figure 1. Dataset Taken

In total, there are 5572 rows, each defining the type of message['Category'] and the message's description. Of these, 4825 messages are classified as ham communications, and 747 as spam messages.

Here is a report of the above description,

```
df['Category'].value_counts()
```

```
ham     4825
spam     747
Name: Category, dtype: int64
```

Figure 2. Categorial View of Dataset

1. 2.1 Data Preprocessing

When I look at the data, it's always a huge dataset with many rows and columns, of news. However, this is not always the case. The data can be in various formats. For example, photos, Audio, video files Structured tables, etc. Machines don't understand images, videos, or text data, only machines understand the 1's and 0's.

Data pre-processing steps:

- **Data clean-up:** This step includes tasks such as entering 'missing values', 'smoothing noisy data', 'identifying or removing outliers', and 'resolving discrepancies.
- **Data conversion**: Aggregation and normalization are executed to scale to a specific value.
- **Data reduction:** This section contains an overview of a minimal data set, but the same analysis results so far.

2. 2.2 Stop Words Removal

An English word that adds little is referred to as a stop word, giving a sentence meaning. You can disregard it without risk by sacrificing the sentence's intended meaning.

For instance, when looking for a search term such as "how to cook a vegetarian and cheese sandwich," the search engine will attempt to browse internet pages that mention "how," "people," and "make." veggies, cheese, and a sandwich. are, 'to,' and 'a' a phrase that is frequently used in English to remove or cease using these three words to find pages with the keyword "veg," and "sandwich" to concentrate on what you do.

```
In [24]: import string
         import nltk
         nltk.download('stopwords')
         string.punctuation

         [nltk_data] Downloading package stopwords to C:\Users\RAGHU
         [nltk_data]     SAI\AppData\Roaming\nltk_data...
         [nltk_data]   Package stopwords is already up-to-date!

Out[24]: '!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'

In [25]: from nltk.corpus import stopwords
         print(stopwords.words('english'),end=" ")

         ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "y
         ourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's",
         'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who
         'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has',
         n', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while',
         etween', 'into', 'through', 'during', 'before', 'after', 'above', 'belo
         f', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there
         'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'r
         's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've",
         en', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn'
         n't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't",
         n't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn',
```

Figure 3. Stop Words Removal

3. 2.3 Stemming

Reducing words to their stems is a process called stemming (prefixes, suffixes, or lemma-like stems) Stemming is crucial for natural language comprehension (NLU) and processing language naturally (NLP).

```
6]: from nltk.stem.porter import PorterStemmer
    ps = PorterStemmer()
    ps.stem('loving')   # example of stemming

6]: 'love'
```

Figure 4. Example of Stemming

4. 2.4 Tokenization

Breaking the flow of texts is what tokenization entails into sentences, glyphs, words, or other expressive components known as tokens. Additionally, the token rundown is utilized to contribute to further processing, including parsing and content mining. Tokenization is advantageous for semantics (like content) and for software engineering lexical research and splitting.

as well as growth. What is what can be difficult to describe at times The definition of "word." Due to tokenization, this is on a word-by-word basis. Tokens frequently rely on traditional heuristics. Tokens are separated by spaces, for instance, punctuation, including "new lines" and "spaces." Each A token consists of an alphabetical string that is repeated. equal to numbers. Punctuation and spaces may or may not be used. the token list that results.

Here are some of the results of using tokenization,

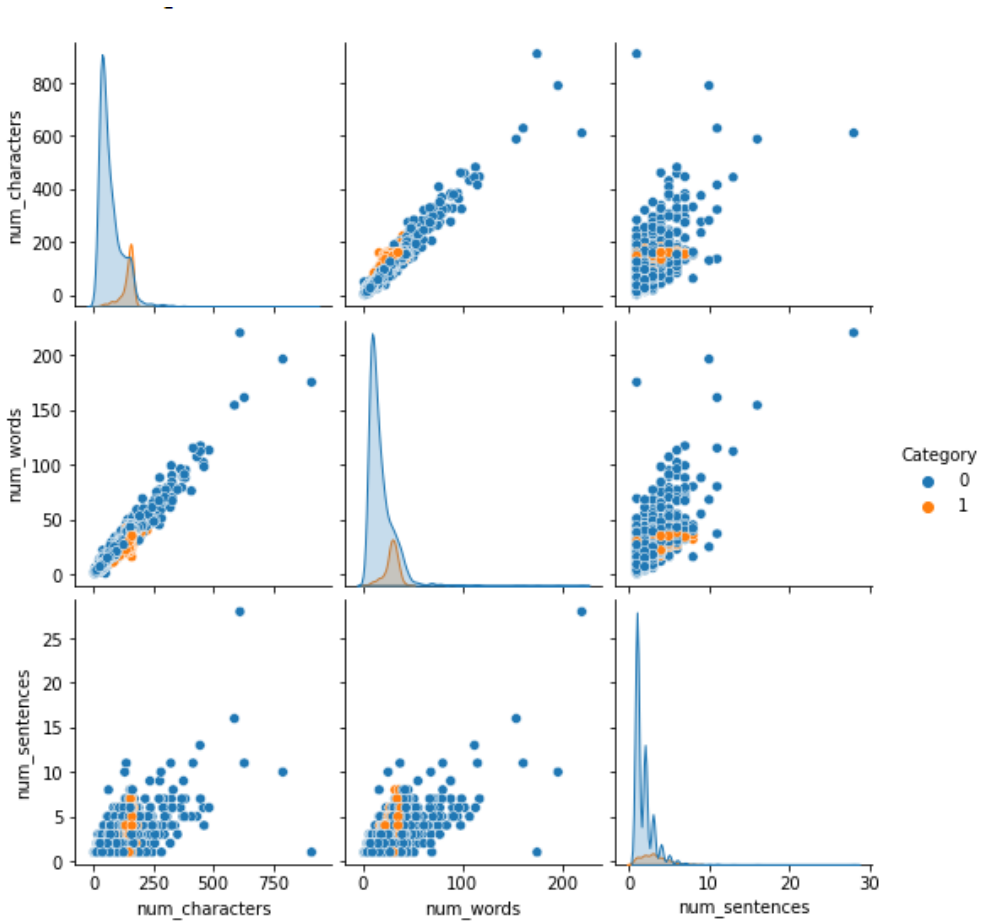

Figure 5. The plot of Word & Sentence tokenization

The above figure depicts the occurrence of each character, word, and sentence in the spam and ham categories.

5. 2.5 TF-IDF Vectorizer

TFIDF works by proportionally increasing the frequency with which a word appears in a document, which is then balanced by the frequency with which the word appears in the total number of documents. Calculate

Tf -IDF (term frequency-inverse document frequency).

- Term Frequency (TF) tracks how frequently a term appears in a document.
- TF(t) = (Num. of occurrences of t in a document) / (Total terms in the document).
- IDF: Inverse Document Frequency, which gauges a term's importance.

IDF(t) is equal to log(Total documents/documents containing term t).

6. 2.6 Splitting the Dataset

First, why is it necessary to divide the data? Because we must first train our model before testing it on the divided data. In order to enhance the effectiveness and accuracy of our model, the majority of the dataset is often used as training data.

We'll employ the train test split() function, which divides the data into two sets—the training set and the testing set—by taking the message, the category, and the split factor as parameters.

9. 3. Naive Bayes Approach

We choose the GaussianNB classifier for the Naive Bayes, Bernoulli, and MultinomialNB models/algorithms. As previously stated, we will perform data analysis using the NLTK package.

The dataset has been cleaned and converted to text. even though there are several no.of approaches How to determine whether an email is spam. I employed a naive Bayes method. The components of a naïve Bayes Classifier are using Bayesian theorem-based classification methods. This is a family of algorithms rather than a

single algorithm, and they all share universal principles Each set of characteristics is a classification. separate from one another.

1. 3.1 Bayes Therom

The Bayes theorem can be used to determine the likelihood that A will take place now that B has happened. where B is proof and A is a supposition. Here, it is assumed that The predictors and characteristics are separate. Meaning, the existence of Other attributes is unaffected by a particular trait. Thus, it is known as naïve.

$$\mathbf{P(A \mid B) = P(B \mid A)*P(A) / P(B)}$$

P(A | B) is a **posterior probability**, which is the likelihood that Given the observed event B, hypothesis A.

P(B | A) stands for **Likelihood Probability**, The likelihood that evidence provided for a hypothesis' likelihood of being correct.

Priority Probability (P(A)) is the likelihood of an earlier hypothesis examining the proof.

P(B) stands for the probability of a marginal event.

2. 3.2 Gaussian NB

When it is assumed that all continuous variables connected to each feature are distributed randomly, Gaussian Naive Bayes is utilized. A normal distribution is another name for a gaussian distribution.

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$$

Figure 6. Gaussian NB Formulation

1. 3.2.1 Implementation of Gaussian NB

Gaussian Naive Bayes implementation on the training set. The outcome of applying the Gaussian NB classifier to the transformed text is

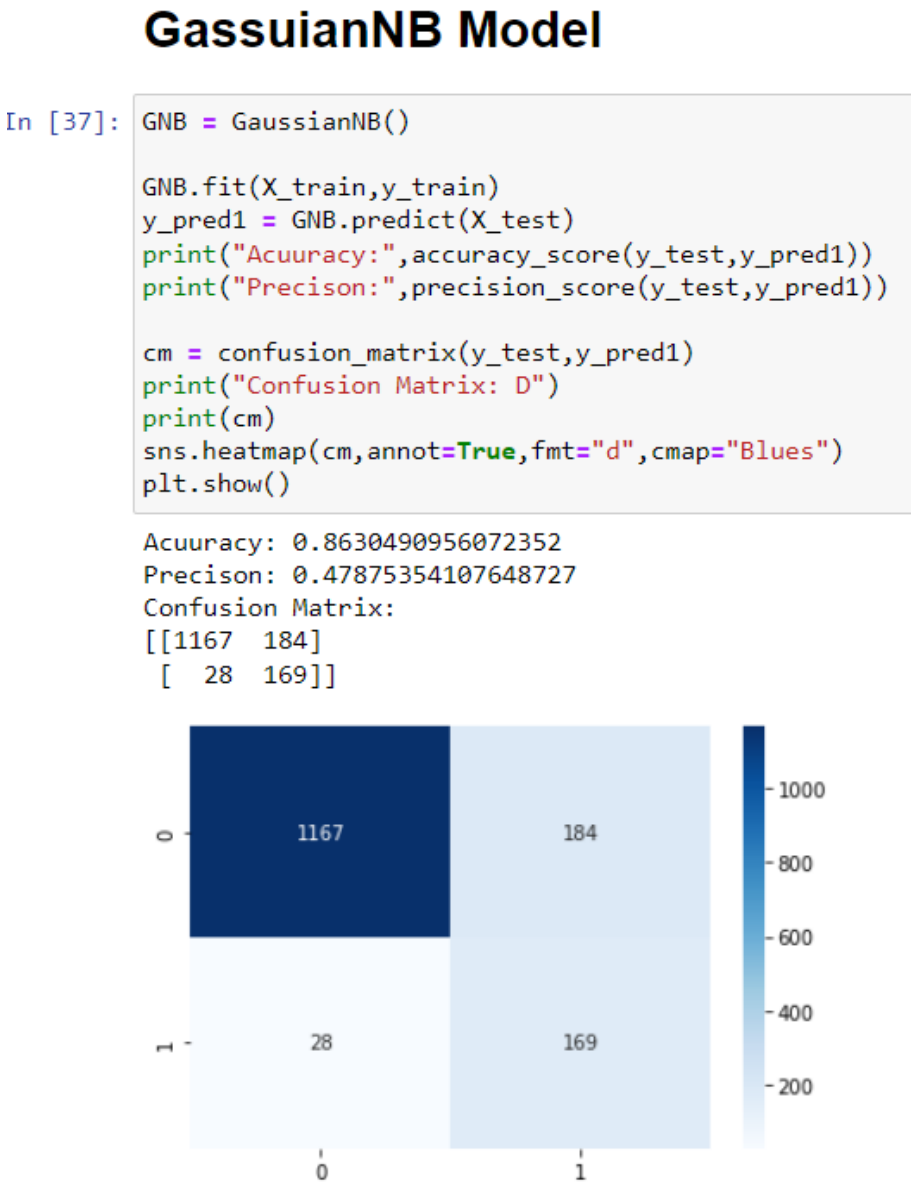

Figure 7. Confusion Matrix of Gaussian NB

We obtained a precision of 0.47 and an accuracy of 0.86. We can see that there are 184 True Negatives and 28 False Positives from the confusion matrix above.

Confusion Matrix:

For a specific set of test data, the performance of the classification models is assessed using the confusion matrix. It can only be established if the real test data values are known.

Classification Report:

In machine learning, a classification report is a performance evaluation indicator. It is used to demonstrate the following properties of our trained model: precision, recall, F1 Score, and support.

	Precision	Recall	F1 score	support
Ham	0.98	0.86	0.92	1351
Spam	0.48	0.86	0.61	197
accuracy			0.86	1548
macro avg	0.73	0.86	0.77	1548
weighted avg	0.91	0.86	0.88	1548

Table 1. Classification Report of Gaussian NB

Here we have taken four performance measuring metrics to evaluate the Gaussian NB model - precision, recall, F1 score, and accuracy.

3. 3.3 Multinomial NB

Multinomial Naive Bayes, For classification based on separate features (such as word counts for text classification), Bayes classifiers work well. In general, integer numbers of features are needed for multinomial distributions. However, in actuality, decimals like TF-IDF also function.

1. 3.3.1 Implementation of Multinomial NB

Multinomail Naive Bayes implementation on the training set. The outcome of applying the MNB classifier to the tranformed text is:

MultinomialNB Model

```
In [39]: MNB = MultinomialNB()

MNB.fit(X_train,y_train)
y_pred2 = MNB.predict(X_test)
print("Accuracy:",accuracy_score(y_test,y_pred2))
print("Precision:",precision_score(y_test,y_pred2))

cm = confusion_matrix(y_test,y_pred2)
print("Confusion Matrix:")
print(cm)
sns.heatmap(cm,annot=True,fmt="d",cmap="Blues")
plt.show()
```

```
Accuracy: 0.9715762273901809
Precision: 1.0
Confusion Matrix:
[[1351    0]
 [  44  153]]
```

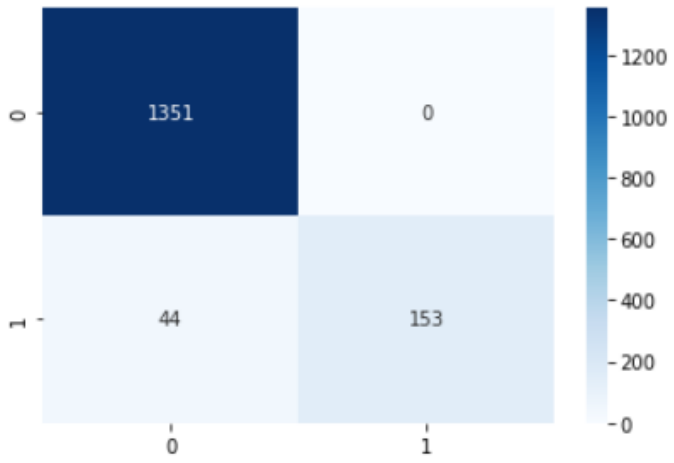

Figure 8. Confusion Matrix of Multinomial NB

2.

We obtained a precision of 1 and an accuracy of 0.97. We can see that there are 0 True Negatives and 44 False Positives from the confusion matrix above.

	Precision	Recall	F1 score	support
Ham	0.97	1.00	0.98	1351
Spam	1.00	0.78	197	197
accuracy			0.97	1548
macro avg	0.98	0.89	0.93	1548
weighted avg	0.97	0.97	0.97	1548

Table 2. Classification Report of Multinomial NB

Here we have taken four performance measuring metrics to evaluate the Multinomial NB model - precision, recall, F1 score, and accuracy.

3.

4. 3.4 Bernoulli NB

Bernoulli Naive Bayes is part of the Naive Bayes family, Accepts only binary values. Checking to see if each value is a word that appears in a document is the most typical example. This is a fairly straightforward model. If word frequency counts are less crucial, Bernoulli can produce superior results. In other words, we must count every aspect of binary term occurrences, such as whether a word appears in a document. Instead of determining the word frequency in a document, these functions are employed.

In other words, the Bernoulli distribution has two outcomes that are mutually exclusive. P(X=0)=1-p or P(X=1)=p Multiple aspects of the Bernoulli NB theorem are possible, but they are always regarded as binary variables. H. Let's assume it has a boolean value. As a result, this class demands that the samples be represented as a feature vector with binary values. Instances of BernoulliNB that are given various sorts of data can binarize the input.

$$P(x_i \mid y) = P(i \mid y)x_i + (1 - P(i \mid y))(1 - x_i)$$

Figure 9. Bernoulli NB Formulation

1. 3.4.1 Implementation of Bernoulli NB

Bernoulli Naive Bayes implementation on the training set. The outcome of applying the BNB classifier to the transformed text is

BernoulliNB Model

```
In [41]: BNB = BernoulliNB()

         BNB.fit(X_train,y_train)
         y_pred3 = BNB.predict(X_test)
         print("Accuracy:",accuracy_score(y_test,y_pred3))
         print("Precision:",precision_score(y_test,y_pred3))

         cm = confusion_matrix(y_test,y_pred3)
         print("Confusion Matrix:")
         print(cm)
         sns.heatmap(cm,annot=True,fmt="d",cmap="Blues_r")
         plt.show()
```

```
Accuracy: 0.9806201550387597
Precision: 0.9826589595375722
Confusion Matrix:
[[1348    3]
 [  27  170]]
```

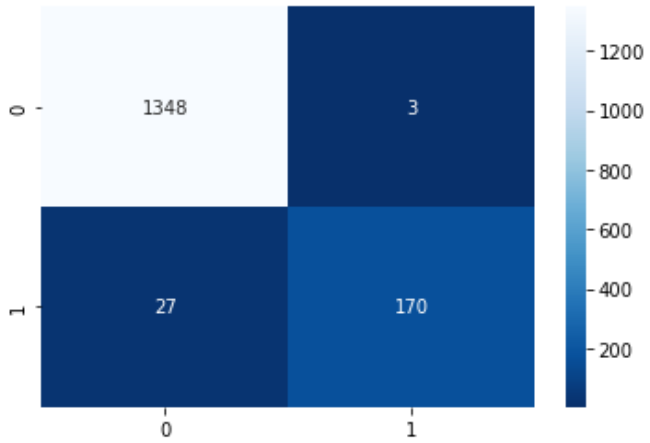

Figure 10. Confusion Matrix of Bernoulli NB

We obtained a precision of 0.98 and an accuracy of 0.98. We can see that there are 3 True Negatives and 27 False Positives from the confusion matrix above.

	Precision	Recall	F1 score	support
Ham	0.98	1.00	0.99	1351
Spam	0.98	0.86	0.92	197
accuracy			0.98	1548
macro avg	0.98	0.93	0.95	1548
weighted avg	0.98	0.98	0.98	1548

Table 3. Classification Report of Bernoulli NB

Here we have taken four performance measuring metrics to evaluate the Bernoulli NB model - precision, recall, F1 score, and accuracy.

10.

11. 4. Logistic Regression Approach

Logistic Regression is a subset of the Generalized Linear Model algorithm class. Logistic regression is a statistical technique for predicting binary outcomes. Based on previous observations of the dataset, yes or no.

By analyzing the relationships between one or more existing independent variables, logistic regression models predict dependent data variables. Logistic regression, for example, can be used to predict whether a political candidate will win or lose an election, or whether a student will be admitted to a specific university. These binary results allow us to choose between two options.

1. 4.1 Implementation of Logistic Regression

Implementation on the training set. The outcome of applying the Logistic Regression Linear classifier to the transformed text is

Logistic Regression

```
[43]: from sklearn.linear_model import LogisticRegression
log_reg_classifier=LogisticRegression(solver='liblinear')
log_reg_classifier.fit(X_train, y_train)
y_pred_log=log_reg_classifier.predict(X_test)
print("Accuracy:",accuracy_score(y_test,y_pred_log))
print("Precision:",precision_score(y_test,y_pred_log))


cm = confusion_matrix(y_test,y_pred_log)
print("Confusion Matrix:")
print(cm)
sns.heatmap(cm,annot=True,fmt="d",cmap="Blues_r")
plt.show()
```

```
Accuracy: 0.9541343669250646
Precision: 0.9772727272727273
Confusion Matrix:
[[1348    3]
 [  68  129]]
```

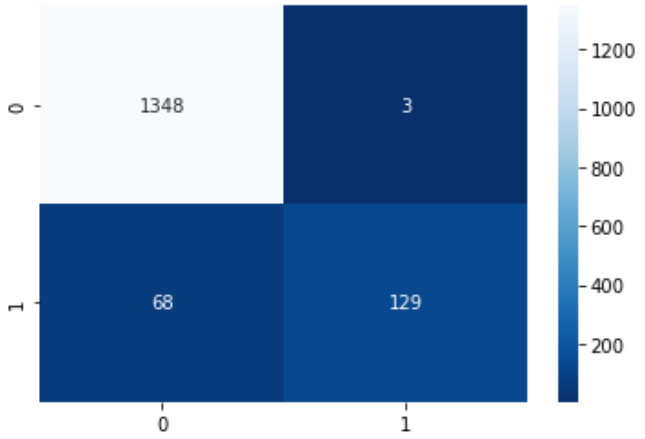

Figure 11. Confusion Matric of Logistic Regression Model

We obtained a precision of 0.97 and an accuracy of 0.95. We can see that there are 3 True Negatives and 68 False Positives from the confusion matrix above.

	Precision	Recall	F1 score	support
Ham	0.95	1.00	0.97	1351
Spam	0.98	0.65	0.78	197
accuracy			0.95	1548
macro avg	0.96	0.83	0.88	1548
weighted avg	0.96	0.95	0.95	1548

Table 4. Classification Report of Logistic Regression Model

Here we have taken four performance measuring metrics to evaluate the Logistic Regression model - precision, recall, F1 score, and accuracy.

12. 5. Convolutional Neural Networks [CNN]

Convolutional neural networks (CNN/ConvNet) are the most widely employed class of deep neural networks in deep learning for the analysis of visual images. Matrix multiplication comes to mind when we think of neural networks.

We will construct a CNN model and train it using the train set.

13.

1. 5.1 Implementation of CNN Model

2.

CNN implementation on the training set. The outcome of applying the CNN model to the transformed text is:

3.

```
In [51]: from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.optimizers import Adam
model = Sequential()
model.add(Dense(512,activation='relu',input_shape
                =(X_train.shape[1],)))
model.add(Dropout(0.3))
model.add(Dense(256, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(128, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy',
              optimizer=Adam(),
              metrics=['accuracy'])

from keras.callbacks import EarlyStopping
es = EarlyStopping(monitor='val_accuracy',
                   mode='max',
                   patience=10,
                   restore_best_weights=True)

NN_Classifier = model.fit(X_train, y_train,
                    batch_size=100,
                    epochs=1000,
                    callbacks =[es],
                    verbose=0,
                    validation_data=(X_test, y_test))

y_pred_NN=(model.predict(X_test) > 0.5).astype(int)



print("Accuracy:",accuracy_score(y_test,y_pred_NN))
print("Precision:",precision_score(y_test,y_pred_NN))
print('\nNeural Networks Confusion Matrix : ')
cm = confusion_matrix(y_test, y_pred_NN)
print(cm)
sns.heatmap(cm,annot=True,fmt="d",cmap="Greys")
plt.show()
```

```
49/49 [==============================] - 0s 3ms/step
Accuracy: 0.9812661498708011
Precision: 0.9941176470588236

Neural Networks Confusion Matrix :
[[1350    1]
 [  28  169]]
```

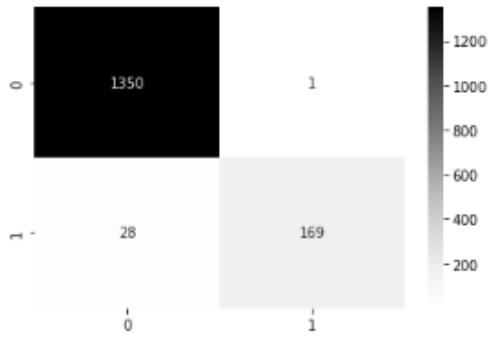

Figure 12. Confusion Matrix of CNN Model

We obtained a precision of 0.98 and an accuracy of 0.98. We can see that there are 1 True Negatives and 28 False Positives from the confusion matrix above.

	Precision	Recall	F1 score	support
Ham	0.98	1.00	0.99	1351
Spam	0.99	0.86	0.92	197
accuracy			0.98	1548
macro avg	0.99	0.93	0.96	1548
weighted avg	0.98	0.98	0.98	1548

Table 5. Classification Report of CNN Model

Here we have taken four performance measuring metrics to evaluate the Logistic Regression model - precision, recall, F1 score, and accuracy.

14. 6. Comparing Results Of Naive Bayes Classifiers

On the training set, we have thus far utilised Gaussian NB, Multinomial NB, and Bernoulli NB classifiers, and the results are shown as a confusion matrix. Let's examine the results to determine which Naive Bayes classifier is more reliable at detecting spam.

We have used pandas to construct a data frame that details the precision and accuracy of various models. Later, the seaborn library is used to represent graphs.

	Model	Accuracy	Precision
0	Gaussian NB	0.86	0.47
1	Multinomial NB	0.97	1.0
2	Bernoulli NB	0.98	0.98

Table 6. Results of NB Classifiers

The same analysis is represented in the graph below, where models are taken on the x-axis, and their accuracy is scaled on the y-axis.

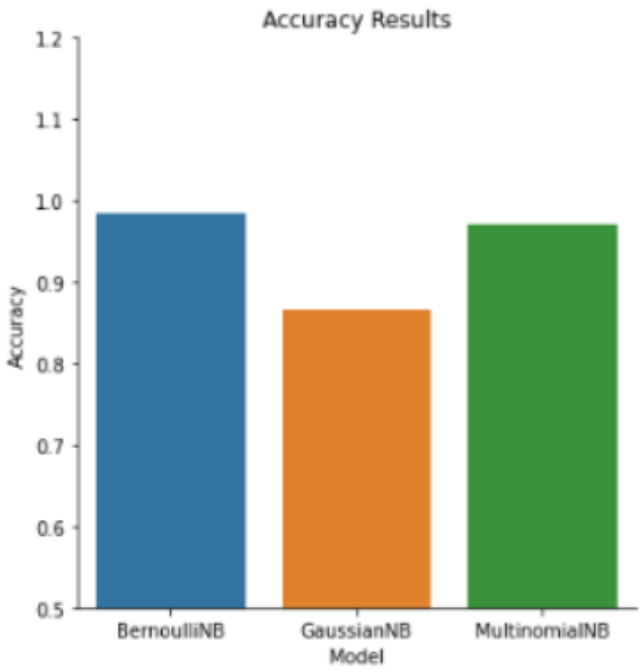

Figure 13. Comparison of Results of Naive Bayes

15. 7. Model Results Comparison

16.

So far we have used Naive Bayes classifiers - Gaussian NB, Multinomial NB, and Bernoulli NB; Generalized Linear Classifier - Logistic Regression; and Convolutional Neural Networks - CNN.

Now let us take a step further and compare the results of all models.

	Model	Accuracy	Precision
0	Gaussian NB	0.86	0.47
1	Multinomial NB	0.97	1.0
2	Bernoulli NB	0.98	0.98
3	Logistic regression	0.95	0.96
4	CNN	0.98	0.98

Table 7. Model Comparison

The graph below illustrates the same by using models as the x-axis and accuracy scales as the y-axis.

17.

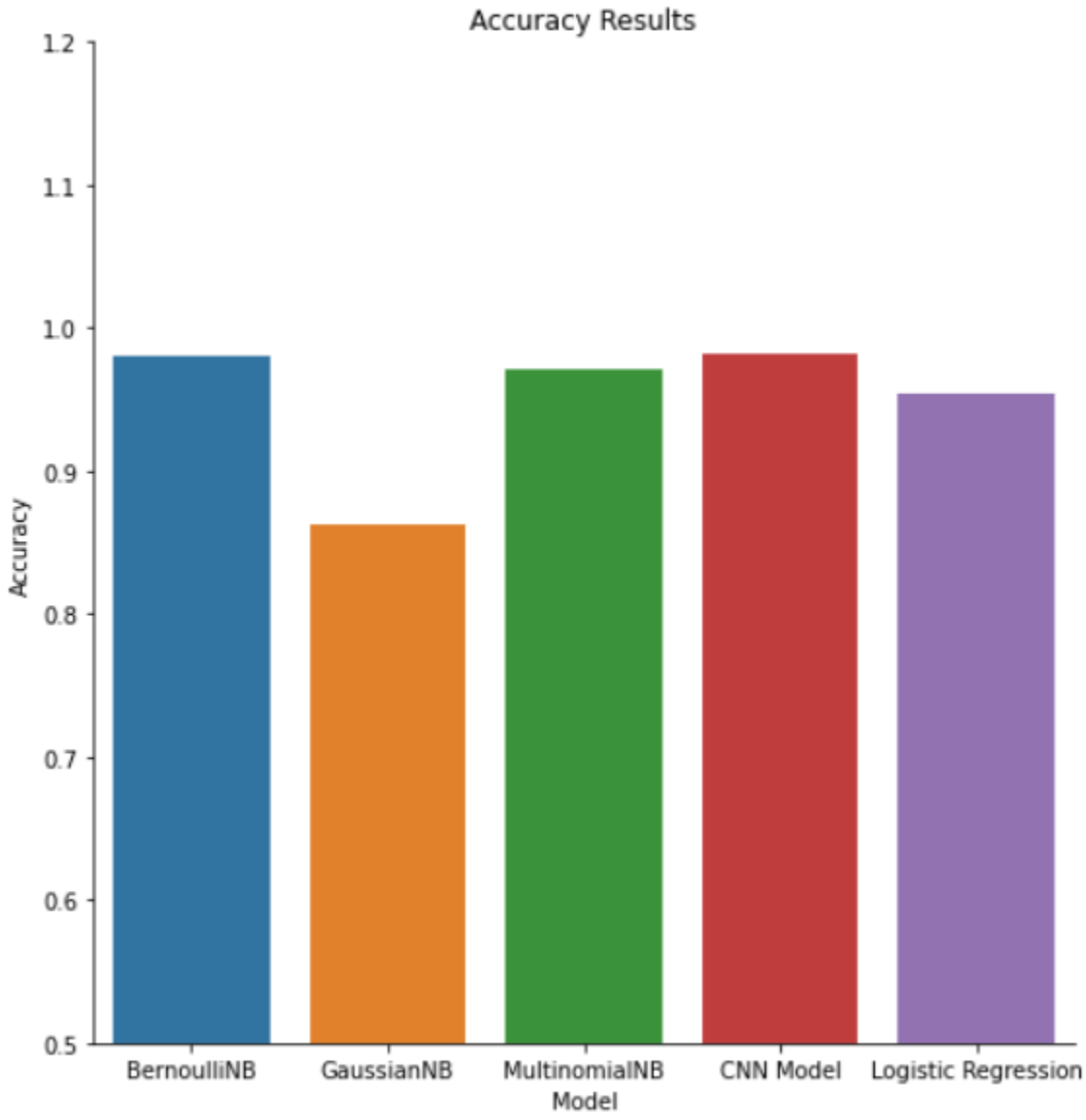

Figure 14. Accuracy Results

18.

19. 8. Concluding Remarks

In this study, we learned what spam email actually is as well as how to filter it. Additionally, we are aware of phishing, which attempts to obtain sensitive information including passwords, credit card numbers, bank account information, and personally identifying data.

We recognize the significance of preventing spam emails and their importance. Machine learning approaches can be quite useful in preventing them. Naive Bayes, which uses probability, and Logistic Regression, which looks for linear patterns in character sequences, are two methods used to detect spam.

20. References

1. Ghulam Mujtaba, Email classification research trends: review and open issues. IEEE Access, 5:9044–9064, 2017.
2. Sunil B Rathod, Content-based spam detection in email using a bayesian classifier, pages 1257–1261. IEEE, 2015.
3. Kriti Agarwal and Tarun Kumar. Email spam detection using an integrated approach of na¨ıve Bayes and particle swarm optimization.
4. In 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), pages 685–690. IEEE, 2018
5. Cihan Varol, “Comparison of String Matching Algorithms on Spam Email Detection”, Dec 2018.
6. Rathod, Sunil B., "Content-based spam detection in email using Bayesian classifier." International Conference on. IEEE, 2015.
7. Sahın, Fatih Orhan. "Spam/ham e-mail classification using machine learning methods based on bag of words technique." 2018 26th Signal Processing and Communications Applications Conference (SIU). IEEE, 2018.