**Email Spam Detection System - A Natural Learning Processing Approach**

## **Overview:**

Email spam detection is the process of filtering out unwanted or unsolicited emails, commonly known as spam, from legitimate emails (ham). This task is typically achieved using machine learning algorithms, rule-based methods, or a combination of both. The process involves analysing the content, headers, and metadata of emails to identify patterns or features indicative of spam, such as certain keywords, suspicious links, or unusual sender behaviour. Techniques like Naive Bayes, Support Vector Machines, and deep learning are widely used for building spam classifiers. Spam detection systems aim to enhance user experience by reducing inbox clutter and protecting against potential threats like phishing attacks and malware, while striving to minimize false positives to avoid misclassifying legitimate emails.

In this project, we aim to use two natural language techniques names as Count Vectorizer and Tfidfvectorizer for vectorising the text into numbers and then applying various machine learning algorithms to detect whether the email is spam or not.

**Objective:**

The main motto of this project is to detect whether the email is spam or ham.

## **Problem Statement:**

Email spam detection is the process of filtering out unwanted or unsolicited emails, commonly known as spam, from legitimate emails (ham). Detecting the spam messages from ham at an early stage can serve as potential indicators of saving our emails, which is crucial in our fight against the hackers. Therefore, to detect whether the email is spam or not, we aim to use two natural language techniques names as Count Vectorizer and Tfidfvectorizer for vectorising the text into numbers and then applying various machine learning algorithms to detect whether the email is spam or not.

## **Dataset:**

There are total 5572 records of the email texts in the dataset. From this records, there are total one input features ('Text') and one Output feature ('Target')

1. Text ---------> Message Text
2. Target ---------> Type of the Message Text

## **Algorithm:**

1. First, the email dataset is obtained.
2. Then the dataset is cleaned and pre-processed.
3. Then Exploratory Data Analysis (EDA) is performed on the dataset and we have formed three new columns named as 'number\_of\_characters', 'number\_of\_words','number\_of\_sentences'. Then we have performed statistical analysis on it.
4. Text Data is pre-processed where it is converted into lower case and all the special characters has been removed from it. Stop words, punctuations has been removed and the stemming has been done to bring out the correct root words.
5. Count Vectorizer and tfidf vectorizer is performed on the pre-processed data where the text is converted into the numbers.
6. Various Machine-Learning algorithms has been applied on the vectorised data, so that the models can be easily be trained on the data and we can easily get the accuracy and the precision scores.
7. For improving the accuracy of the model, voting classifier and Stack Classifier has been applied on the dataset so that the accuracy of the model can be improved.
8. Files has been created for the objects of tfidf and mnb so that we can use these files while developing the website to predict the type of text message.
9. Then we have developed the web application for determining whether the email is spam or not.
10. Here from this project, Multinomial Naive Bayes is considered for this project as it has the highest score.

## **Algorithms Used:**

1. Gaussian Naive Bayes
2. Multinomial Naive Bayes
3. Bernoulli Naive Bayes
4. Logistic Regression
5. Support Vector Machines
6. Multinomial Naive Bayes
7. Decision Tree Classifier
8. K Neighbors Classifier
9. Random Forest Classifier
10. Ada Boost Classifier
11. Bagging Classifier
12. Extra Trees Classifier
13. Gradient Boosting Classifier
14. XGB Classifier

## **Hyper Parameters Used:**

1. kernel
2. gamma
3. max\_depth
4. solver
5. penalty
6. n\_estimators
7. random\_state
8. probability
9. voting
10. estimators
11. final\_estimator

**Implementation :**

1. **GET THE DATA:**
   1. The spam classifier dataset is obtained from the kaggle website.
2. **DATA CLEANING :**
   1. The data set is cleaned, pre-processed and unnecessary columns are removed.
   2. Label Encoding is done on the Output (‘Target’) column to convert the text into numeric values.
3. **EXPLORATORY DATA ANALYSIS:** 
   1. Here EDA is performed on the dataset to check whether the dataset is balanced or not.
   2. Then three new columns ‘**num\_of\_characters**’,**’num\_of\_words’** and **‘num\_of\_sentences’** are formed in the dataset.
   3. Statistical Information and correlation is obtained for the new columns that has been added in the dataset.
4. **DATA PREPROCESSING :** 
   1. All the texts are converted in lower case.
   2. The texts are tokenized into words.
   3. Special Characters are removed from the tokenized texts.
   4. Stop words and punctuations are removed from the tokenized texts.
   5. Stemming is done on every word of the tokenized texts, to bring every word to its root word.
5. **WORD CLOUD:**
   1. Word Cloud is used to generate most frequent words for ‘spam’ and ‘ham’ from the given list of all the words.
   2. Then plotting of all the frequently used words for ‘spam’ and ‘ham is done using histplot.
6. **MODEL BUILDING:**
   1. Tfidf Vectorizer is applied on the pre-processed data to convert the text into vectors.
   2. The entire input data is in the form of vectors, the data can be trained easily using ML Algorithms.
   3. Naive Bayes, Gaussian Naive Bayes, Bernoulli Naive Bayes are the algorithms used.
   4. Models are created for these three algorithms and the dataset is trained easily on these models.
   5. Output is predicted for these three models using the test data.
   6. Accuracy Score, Confusion Matrix and Precision Score is computed for these three models used.
   7. Various algorithms has been used like decision tree, KNN, Random forest etc. and accuracy and precision scores has been calculated for every algorithms.
   8. However, the accuracy score remains the same for all the three models used.
   9. So Voting Classifier and Stacking Classifier has been used to check the accuracy and precision of the model. However, the accuracy and the precision level remains the same.
7. **WEBSITE BUILDING:**
   1. Here a Website has been built to display whether the text is ‘spam’ or ‘ham’.

## **Output:**

1. Gaussian Naive Bayes

a. accuracy\_score = 86.94390715667312

b. precision\_score = 0.5068493150684932

1. Multinomial Naive Bayes

a. accuracy\_score = 97.09864603481626

b. precision\_score = 1.0

1. Binomial Naive Bayes

a. accuracy\_score = 98.35589941972921

b. precision\_score = 0.991869918699187

1. Voting Classifier

a. accuracy\_score = 98.16247582205028

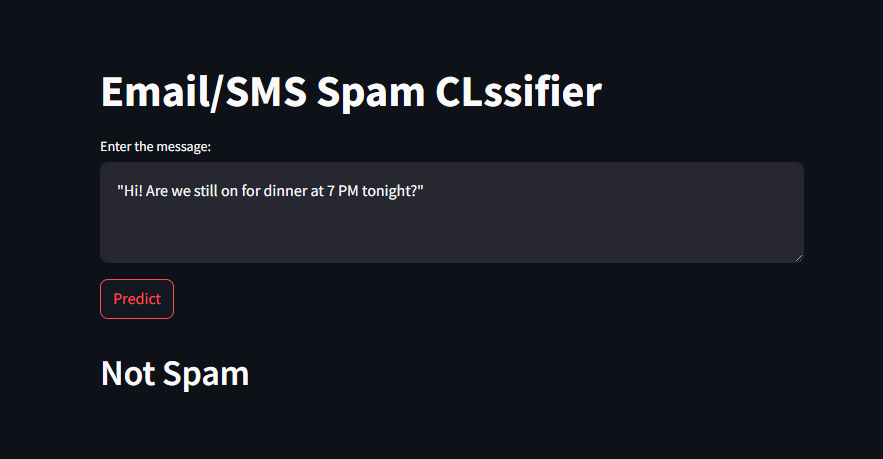
b. precision\_score = 0.9917355371900827

1. Stacking Classifier

a. accuracy\_score = 97.87234042553192

b. precision\_score = 0.9393939393939394

**Screenshot:**



**Observations:**

* Here a message is given in the text box and it has predicted that the message is **‘not spam’**