PROJECT TITLE: Building a Smarter AI-Powered Spam Classifier.

DEFINITION:

For the majority of internet users, email has become the most often utilized formal communication channel. In recent years, there has been a surge in email usage, which has exacerbated the problems presented by spam emails. Spam, often known as junk email, is the act of sending unsolicited mass messages to a large number of people. 'Ham' refers to emails that are meaningful but of a different type. Every day, the average email user receives roughly 40-50 emails. Spammers earn roughly 3.5 million dollars per year from spam, resulting in financial damages on both a personal and institutional level.

DESIGN PROCESS:

There are many different ways to design a spam filtering system, but one common approach is to use machine learning algorithms. These algorithms can be trained on data sets of known and non-spam emails and then used to classify new emails.

Many machine-learning algorithms can be used for this task, including support vector machines, naive Bayes classifiers, and decision trees. Each algorithm has its strengths and weaknesses, so choosing the right one for your particular data set and application is essential. Once you have chosen an algorithm, you must train it on your data set. This process involves providing the algorithm with training examples, which it will use to learn the characteristics of spam and non-spam emails. Once the algorithm has been trained, it can be used to classify new emails.If you are unsure which algorithm to use or how to train it on your data set, many online resources can help you.

INNOVATION STEPS:

As a beginner, I built an SMS spam classifier but did a ton of research to know where to start.

1. Load and simplify the dataset.

Our SMS text messages dataset has 5 columns if you read it in pandas: v1 (containing the class labels ham/spam for each text message), v2 (containing the text messages themselves), and three Unnamed columns which have no use.

2. Explore the dataset: Bar Chart.

It's a good idea to carry out some Exploratory Data Analysis (EDA) in a classification problem to visualize, get some information out of, or find any issues with your data before you start working with it.

3. Explore the dataset: Word Clouds.

For my project, I generated word clouds of the most frequently occurring words in my spam messages.First, we'll filter out all the spam messages from our dataset. df_spam is a DataFrame that contains only spam messages.

4.Handle imbalanced datasets.

To handle imbalanced data, you have a variety of options. I got a pretty good f-measure in my project even with unsampled data, but if you want to resample.

5.Split the dataset.

In Machine Learning, we usually split our data into two subsets — train and test. We feed the train set along with the known output values for it (in this case, 0 or 1 corresponding to spam or ham) to our model so that it learns the patterns in our data. Then we use the test set to get the model's predicted labels on this subset. Let's see how to split our data.

6.Apply Tf-IDF Vectorizer for feature extraction.

Our Naïve Bayes model requires data to be in either Tf-IDF vectors or word vector count. The latter is achieved using Count Vectorizer, but we'll obtain the former through using Tf-IDF Vectorizer.

7.Train our Naive Bayes Model.

We fit our Naïve Bayes model, aka MultinomialNB, to our Tf-IDF vector version of x_train, and the true output labels stored in y_train.

8.Check out the accuracy, and f-measure.

It's time to pass in our Tf-IDF matrix corresponding to x_test, along with the true output labels (y_test), to find out how well our model did!

9.View the confusion matrix and classification report.

Let's now look at our confusion matrix and f-measure scores to *confirm* if our model is doing OK or not:

10.Heatmap for our Confusion Matrix (Optional).

You can create a heatmap using the seaborn library to visualize your confusion matrix.

OVERVIEW OF DATASET:

The SMS Spam Collection is a set of SMS tagged messages that have been collected for SMS Spam research. It contains one set of SMS messages in English of 5,574 messages, tagged according to being ham (legitimate) or spam.

Data processing

- Import the required packages
- Loading the Dataset
- Remove the unwanted data columns
- Preprocessing and Exploring the Dataset
- Build word cloud to see which message is spam and which is not.
- Remove the stop words and punctuations
- Convert the text data into vectors

Building a sms spam classification model

- Split the data into train and test sets
- Use Sklearn built-in classifiers to build the models
- Train the data on the model
- Make predictions on new data

Import the required packages

```
%matplotlib inline

import matplotlib.pyplot as plt

import csv

import sklearn

import pickle

from wordcloud import WordCloud
```

```
import pandas as pd

import numpy as np

import nltk

from nltk.corpus import stopwords

from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer

from sklearn.tree import DecisionTreeClassifier.
```

Loading the dataset

```
data = pd.read_csv('dataset/spam.csv', encoding='latin-1')

data.head()
```

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

Preprocessing and exploring dataset

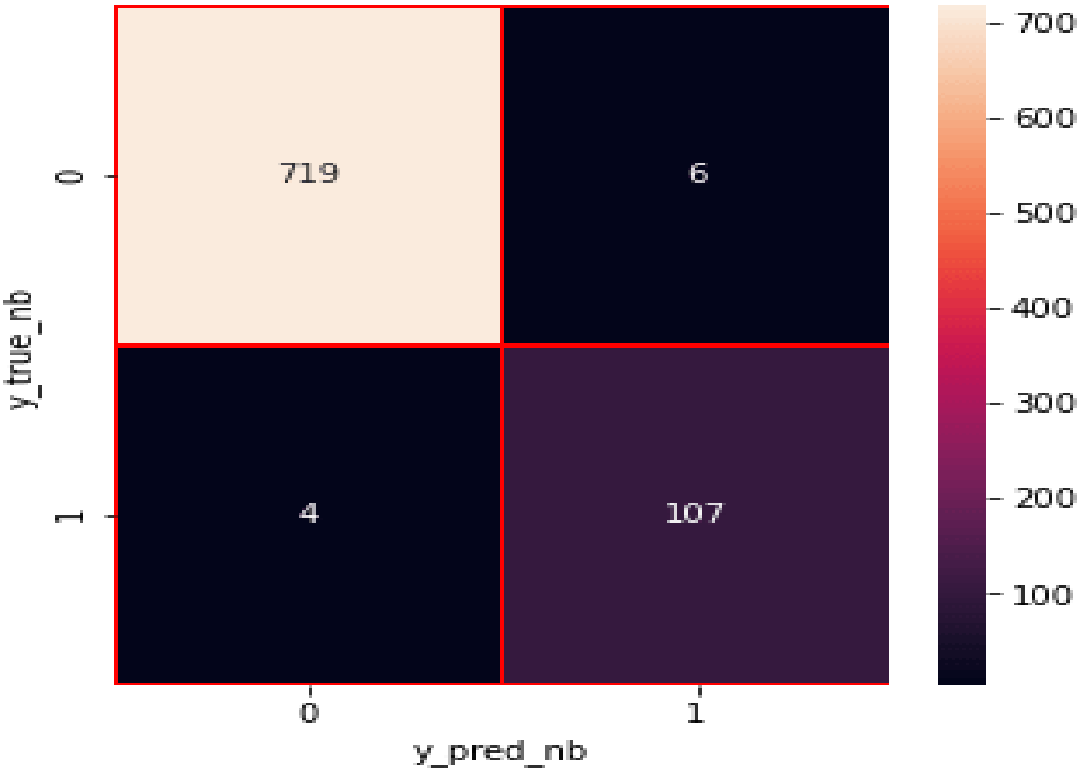
```
# Import nltk packages and Punkt Tokenizer Models
import nltk
nltk.download("punkt")
import warnings
warnings.filterwarnings('ignore')
```

	label	text
1990	ham	HI DARLIN IVE JUST GOT BACK AND I HAD A REALLY...
1991	ham	No other Valentines huh? The proof is on your ...
1992	spam	Free tones Hope you enjoyed your new content. ...
1993	ham	Eh den sat u book e kb liao huh...
1994	ham	Have you been practising your curtsey?
1995	ham	Shall i come to get pickle
1996	ham	Lol boo I was hoping for a laugh
1997	ham	\YEH I AM DEF UP4 SOMETHING SAT
1998	ham	Well, I have to leave for my class babe ... Yo...
1999	ham	LMAO where's your fish memory when I need it?

```
ham_words = ''
```

```
spam_words = ''

# Creating a corpus of spam messages

for val in data[data['label'] == 'spam'].text:

    text = val.lower()

    tokens = nltk.word_tokenize(text)

    for words in tokens:

        spam_words = spam_words + words + ' '

# Creating a corpus of ham messages

for val in data[data['label'] == 'ham'].text:

    text = text.lower()

    tokens = nltk.word_tokenize(text)

    for words in tokens:

        ham_words = ham_words + words + ' '


spam_wordcloud = WordCloud(width=500,
height=300).generate(spam_words)

ham_wordcloud = WordCloud(width=500,
height=300).generate(ham_words)
```

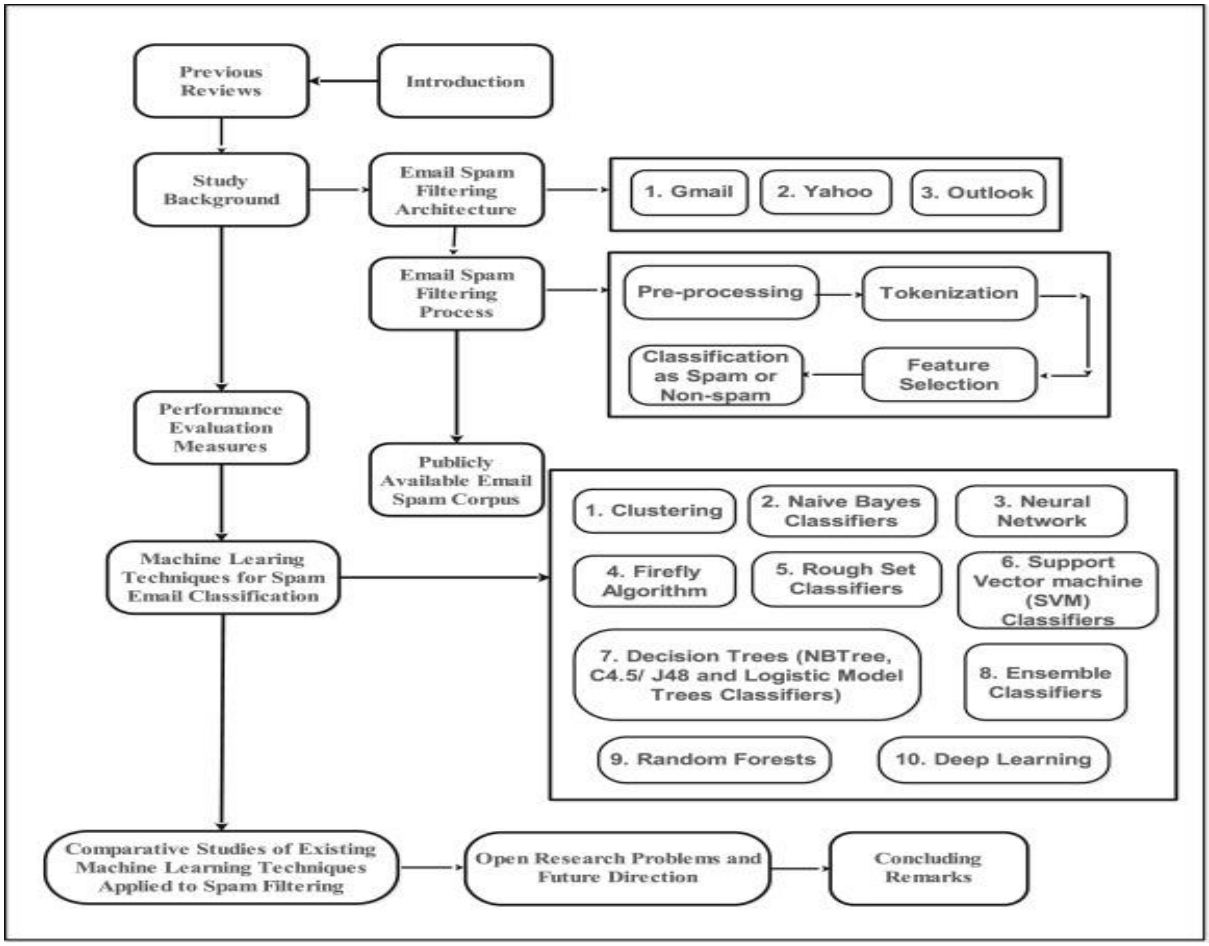
```
#Spam Word cloud

plt.figure( figsize=(10,8), facecolor='w')

plt.imshow(spam_wordcloud)

plt.axis("off")

plt.tight_layout(pad=0)

plt.show()

#Creating Ham wordcloud

plt.figure( figsize=(10,8), facecolor='g')

plt.imshow(ham_wordcloud)

plt.axis("off")

plt.tight_layout(pad=0)

plt.show()
```



from the confusion matrix, we can see that the Naive Bayes model is balanced. That's it !! we have successfully created a spam classifier.

Selecting a machine learning algorithm:

Machine learning approach have proved to be more efficient than knowledge engineering approach. No rule is required to be specified, rather a set of training samples which are pre-classified email messages are provided. A particular machine learning algorithm is then used to learn the classification rules from these email messages . Several studies have been carried out on machine learning techniques and many of these algorithms are being applied in the field of email spam filtering. Examples of such algorithms include Deep Learning, Naïve Bayes, Support Vector Machines, Neural Networks, K-Nearest Neighbour, Rough sets, and Random Forests.



Training the model:

To classify spam messages, the BERT model is best for this task as it contains many versions.

In this work, the researchers used the second approach where the finetuning process is made by using linear layers, dropout layers , batch normalization layers , Rectified Linear Unit (ReLU) activations, and log softmax activation with Xavier initialization of weights added at the end in the classifier part of the pre-trained model. The main reason for adding the dropout layer is to avoid overfitting, batch normalization is used to reduce internal covariate shift, and Xavier initialization will help converge the designed model faster.

Evaluating its performance:

Machine learning algorithms have been extensively applied in the field of spam filtering. Substantial work have been done to improve the effectiveness of spam filters for classifying emails as either ham (valid messages) or spam (unwanted messages) by means of ML classifiers. They have the ability to recognise distinctive characteristics of the contents of emails. Many significant work have been done in the field of spam filtering using techniques that does not possess the ability to adapt to different conditions; and on problems that are exclusive to some fields e.g. identifying messages that are hidden inside a stego image.

Most of the machine learning algorithms used for classification of tasks were designed to learn about inactive objective groups. The authors in posited that when these algorithms are trained on data that has some data that have been poisoned by an enemy, it makes the algorithms susceptible to a number of different attacks on the reliability and accessibility of the data. As a matter of fact, manipulating as minute as 1% of the training data is enough in certain instances . Though it might be strange to hear that the data supplied by an enemy is used to train a system, it does happen in some real world systems.

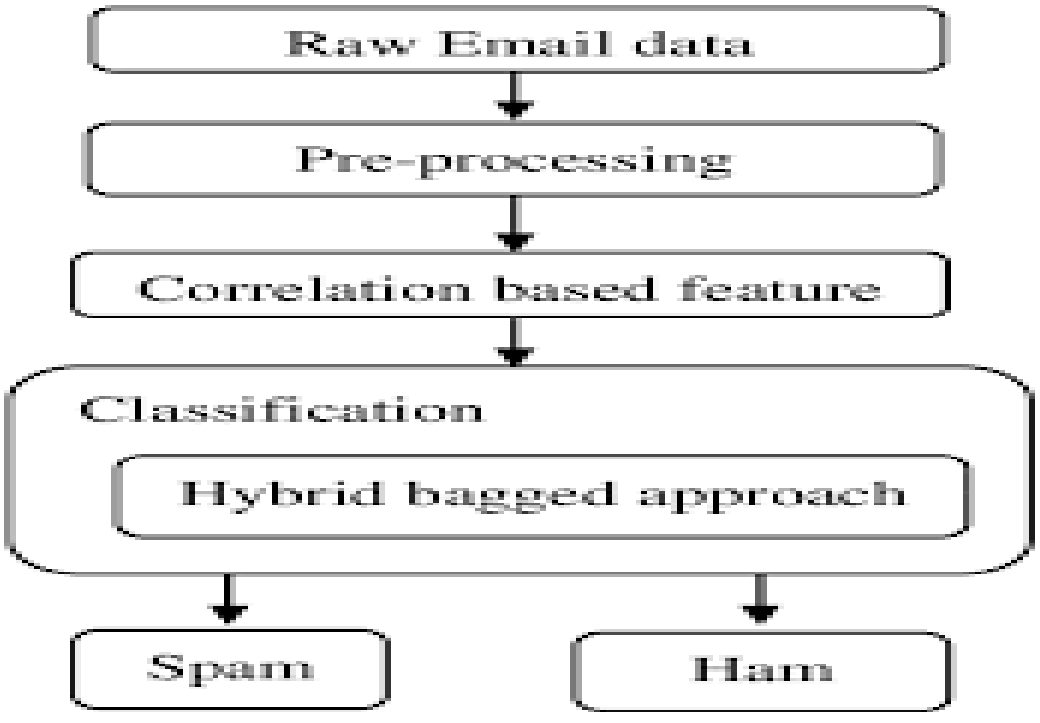


Figure. 1 Basic process for email filtering

CONCLUSION:

By accurately identifying and filtering spam, individuals and organizations can focus on important emails and mitigate potential risks associated with malicious content. In conclusion, email spam detection using machine learning offers a promising solution to the pervasive problem of unwanted and harmful emails.