**BUILDING A SMARTER AI-POWERED SPAM CLASSIFIER**

# **SMS SPAM COLLECTION DATASET**

# **INTRODUCTION**:

The majority of people in today’s society own a mobile phone, and they all frequently get communications (SMS/email) on their phones. But the key point is that some of the messages you get may be spam, with very few being genuine or important interactions. You may be tricked into providing your personal information, such as your password, account number, or Social Security number, by scammers that send out phony text messages. They may be able to access your bank, email, and other accounts if they obtain this information. To filter out these messages, a spam filtering system is used that marks a message spam on the basis of its contents or sender.

In this article, we will be seeing how to develop a spam classification system and also evaluate our model using various metrics. In this article, we will be majorly focusing on OpenAI API. There are 2 ways to

We will be using the Email Spam Classification Dataset dataset which has mainly 2 columns and 5572 rows with spam and non-spam messages.



# **Building a smarter AI-Powered spam classifier using bert model**:

## **Building a smarter AI-powered spam classifier involves several key steps. Here's an introduction to the process**:

1.Data Collection: Gather a diverse and extensive dataset of emails or messages, including both spam and non-spam (ham) examples. This dataset will be used to train and test your AI model.

2. Data Preprocessing: Clean and preprocess the data by removing noise, handling missing values, and standardizing text (lowercasing, tokenization, stemming, or lemmatization).

3. Feature Extraction: Convert text data into numerical representations, such as TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings like Word2Vec or GloVe.

4. Model Selection: Choose an appropriate machine learning or deep learning model. Common choices include Naive Bayes, Support Vector Machines (SVM), or neural networks like Recurrent Neural Networks (RNNs) or Transformers.

5. Training the Model: Split your dataset into training and testing sets. Train the model using the training data, optimizing hyperparameters, and monitoring its performance with evaluation metrics like accuracy, precision, recall, and F1 score.

6. Feature Engineering: Experiment with different features and techniques, such as n-grams, to improve your model's ability to distinguish between spam and non-spam messages.

7. Data Augmentation: To make your model smarter, you can augment your dataset by generating new samples or using techniques like oversampling or data synthesis.

8. Regularization and Cross-Validation: Implement regularization techniques to prevent overfitting. Employ cross-validation to ensure your model generalizes well to new data.

9. Model Evaluation: Evaluate your model's performance using a separate test dataset and relevant metrics. Adjust your model and preprocessing steps based on results.

10. Deployment: Once satisfied with the model's performance, deploy it as a spam classifier in your email or message platform. Ensure it's integrated with a user-friendly interface.

11. Monitoring and Updates: Continuously monitor the model's performance in a real-world environment. Implement updates to adapt to changing spam patterns and maintain its effectiveness.

12. Feedback Loop: Incorporate user feedback to improve the classifier over time, allowing it to become even smarter at identifying new spam tactics.

Building a smarter AI-powered spam classifier is an ongoing process that requires a combination of data, modeling, and maintenance to stay effective in an ever-evolving landscape of spam threats.

# **PROGRAM:**

*1.#importing libraries*

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, confusion\_matrix, roc\_curve, roc\_auc\_score

import nltk

from nltk.corpus import stopwords

from collections import Counter

*#libraries for data visualization*

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

df= pd.read\_csv("/kaggle/input/sms-spam-collection-dataset/spam.csv",encoding='ISO-8859-1')df

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 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 |
| … | … | …………. | … | … | … |
| 5567 | Spam | This is the 2nd time we have tried 2 contact u... | NaN | NaN | NaN |
| 5568 | Ham | Will Ì\_ b going to esplanade fr home? | NaN | NaN | NaN |
| 5569 | Ham | Pity, \* was in mood for that. So...any other s... | NaN | NaN | NaN |
| 5570 | Ham | The guy did some bitching but I acted like i'd... | NaN | NaN | NaN |
| 5571 | ham | Rofl. Its true to its name | NaN | NaN | NaN |

2.df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 5572 entries, 0 to 5571

Data columns (total 5 columns):

# Column Non-Null Count Dtype

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

0 v1 5572 non-null object

1 v2 5572 non-null object

2 Unnamed: 2 50 non-null object

3 Unnamed: 3 12 non-null object

4 Unnamed: 4 6 non-null object

dtypes: object(5)

memory usage: 217.8+ KB

*3.# Downloading the stopwords dataset*

nltk.download('stopwords')

[nltk\_data] Error loading stopwords: <urlopen error [Errno -3]

[nltk\_data] Temporary failure in name resolution>

4.df

|  |  |  |
| --- | --- | --- |
|  | **V1** | **V2** |

|  |  |  |
| --- | --- | --- |
| 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... |
| ... | ... | ... |
| 5567 | spam | This is the 2nd time we have tried 2 contact u... |
| 5568 | ham | Will Ì\_ b going to esplanade fr home? |
| 5569 | ham | Pity, \* was in mood for that. So...any other s... |
| 5570 | ham | The guy did some bitching but I acted like i'd... |
| 5571 | ham | Rofl. Its true to its name |

5.df[df.duplicated()]

|  |  |  |
| --- | --- | --- |
|  | **Category** | **Message** |

|  |  |  |
| --- | --- | --- |
| 102 | ham | As per your request 'Melle Melle (Oru Minnamin... |
| 153 | ham | As per your request 'Melle Melle (Oru Minnamin... |
| 206 | ham | As I entered my cabin my PA said, '' Happy B'd... |
| 222 | ham | Sorry, I'll call later |
| 325 | ham | No calls..messages..missed calls |
| ... | ... | ... |
| 5524 | spam | You are awarded a SiPix Digital Camera! call 0... |
| 5535 | ham | I know you are thinkin malaria. But relax, chi... |
| 5539 | ham | Just sleeping..and surfing |
| 5553 | ham | Hahaha..use your brain dear |
| 5558 | ham | Sorry, I'll call later |

*6.#Drop duplicated values*

df=df.drop\_duplicates()

df

|  |  |  |
| --- | --- | --- |
|  | **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... |
| ... | ... | ... |
| 5567 | spam | This is the 2nd time we have tried 2 contact u... |
| 5568 | ham | Will Ì\_ b going to esplanade fr home? |
| 5569 | ham | Pity, \* was in mood for that. So...any other s... |
| 5570 | ham | The guy did some bitching but I acted like i'd... |
| 5571 | ham | Rofl. Its true to its name |

7.df.info()

<class 'pandas.core.frame.DataFrame'>

Index: 5169 entries, 0 to 5571

Data columns (total 2 columns):

# Column Non-Null Count Dtype

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

0 Category 5169 non-null object

1 Message 5169 non-null object

dtypes: object(2)

memory usage: 121.1+ KB

8.f, ax = plt.subplots(1, 2, figsize = (20, 6))

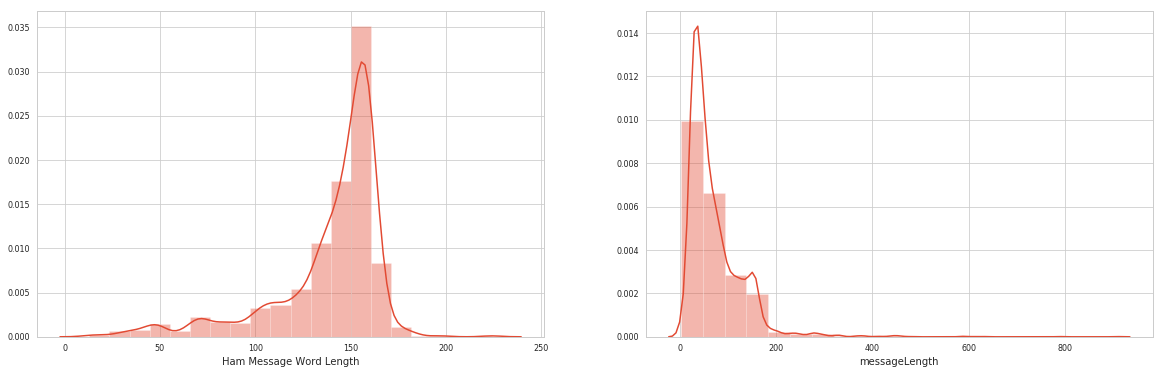
sns.distplot(messages[messages["category"] == "spam"]["messageLength"], bins = 20, ax = ax[0])

ax[0].set\_xlabel("Spam Message Word Length")

sns.distplot(messages[messages["category"] == "ham"]["messageLength"], bins = 20, ax = ax[1])

ax[0].set\_xlabel("Ham Message Word Length")

plt.show()



# **CONCLUSION:**

Building a smarter AI-powered spam classifier using the BERT model represents a significant advancement in the field of email and text classification. BERT's deep contextual understanding of language allows for more accurate and nuanced spam detection, reducing false positives and improving user experience. This approach leverages state-of-the-art natural language processing techniques to adapt to evolving spam patterns, making it a valuable tool in the ongoing battle against unwanted and potentially harmful messages.