Spam Classifier

Team Member : SANJAY S P - 71772114136

PHASE 1 : DOCUMENT SUBMISSION

**PROBLEM DEFINITION:**

The primary goal of the project is to build a system capable of classifying messages into two categories:

* Spam messages:
  + Any kind of unwanted, unsolicited digital communication that gets sent out in bulk.
* Non-spam messages:
  + These are legitimate messages.

The system should be designed in such a way that it can classify the messages in Emails and Text messages.

The system should be designed using Natural Language processing(NLP) and Machine learning.

* Natural Language Processing:

This technique will process and analyze the textual content of messages.

* Machine learning:

This is the fundamental component of a spam classifier, as it enables the system to learn and identify patterns and characteristics of spam and non spam messages.

**DESIGN THINKING:**

1. Create an empty list that you will use to store the preprocessed messages.
2. Create a loop to process every message in the MESSAGE column of the dataset.
   1. Remove all the non-alphanumeric characters.
   2. Convert the message to lowercase.
   3. Split the text into words.
   4. Remove the stopwords and lemmatize the words.
   5. Convert the words back into sentences.
   6. Append the preprocessed message into the corpus list.

**Feature engineering using the TF-IDF technique:**

* Feature engineering is the process of converting raw data features into new features suited for machine learning models.
* The Term Frequency-Inverse Document Frequency(TF-IDF) works by assigning weights to words in a document based on how frequently they appear. TF-IDF gives words that appear frequently in a document but are rare in the corpus higher weight. This allows machine learning algorithms to better understand the meaning of the text.

1. Creating and Training the Model:
   1. Start by creating and initializing a naive bayes model using the scikit-learn multinomial NB class.
   2. Fit the training data, allowing the model to train on the training set
   3. Then make predictions on the training and testing sets using the predict method.
   4. These predictions will help you evaluate your model.
2. Model evaluation:
   1. Evaluate the performance of the model using the classification report function from scikit-learn.
   2. Pass the training set prediction and the actual training set labels as input.
   3. Do the same for the test set.