# **Spam Detection - Exploratory Data Analysis and Model Building**

# This code includes general steps for data preprocessing, exploratory data analysis (EDA), and model building. Additionally, fine-tuning steps are introduced to enhance the model's performance.

# **General Steps**

### **1. Data Loading and Cleaning**

* The SMS dataset is loaded into a Pandas DataFrame from a CSV file.
* Columns with missing values (Unnamed: 2, Unnamed: 3, Unnamed: 4) are dropped.
* Columns 'v1' and 'v2' are renamed to 'target' and 'text', respectively.

### **2. Exploratory Data Analysis (EDA)**

* The number of words and sentences in each text message is calculated to provide insights into the length of messages.

### **3. Data Preprocessing**

* Text data is preprocessed by converting it to lowercase, tokenizing, removing special characters, stopwords, and applying stemming. This step helps in standardizing the text for better model performance.

### **4. Word Cloud Visualization**

* Word clouds are generated for both spam and non-spam messages to visualize the most frequent words.

### 

### **5. Model Building**

* In the model building phase, we employ an ensemble approach to enhance the spam detection model's performance. The primary classifier used is the Multinomial Naive Bayes, which has shown promising results in terms of accuracy and precision. Additionally, we explore the power of combining multiple models through ensemble techniques.

### **Fine-Tuning Steps**

#### 1. Change the max\_features parameter of TfIdf

* The max\_features parameter of the TF-IDF vectorizer is experimented with different values to observe its impact on model performance. TF-IDF is adjusted to consider a varying number of features.

#### 2. Scaling Features

* Scaling features involves adjusting the range of values of different features to make them comparable. This fine-tuning step ensures that the model is not sensitive to the scale of input features.

#### 3. Consider Features Related to Number of Characters

* Features related to the number of characters in text messages are introduced. This fine-tuning step enables the model to capture patterns associated with the length of messages, potentially improving its predictive capabilities.

#### 4. Ensemble Techniques - Voting Classifier and Stacking Classifier

* Voting Classifier:
  + Predictions from multiple models (SVM, Naive Bayes, and Extra Trees) are combined by considering the weighted sum of their predicted probabilities. This ensemble method aims to improve overall prediction accuracy.
* Stacking Classifier:
  + Predictions from multiple models (SVM, Naive Bayes, and Extra Trees) are combined using another model (Random Forest) as a final layer. This stacking approach leverages the diverse strengths of individual models to enhance overall predictive performance.

### **Conclusion**

This code provides a comprehensive approach to spam detection, starting from data loading and cleaning to exploratory data analysis, data preprocessing, and model building. The inclusion of fine-tuning steps allows for experimentation and optimization to achieve the best possible model performance.