SMS SPAM DETECTION

**SMS Spam detection is a process of identifying the messages as the spam (unwanted promotional or malicious) or the ham messages (legitimate). It can be applied on the Emails too to detect the spam emails.**

**Problem Statement:** There are many problems that occur with spam Emails or messages as we receive so many in a single day. Hackers or Outsiders may send promotional or offer mails and messages to attract users which is a kind of phishing attack that may be a big issue for people who are not aware of these actions. A software or application that detects spam messages/emails will be a helpful tool which will prevent users from these kinds of attacks (other kind of threats).

Spam detection, in essence is a binary text classification problem but there are many methods to detect spam messages with different machine learning algorithms like linear regression classifier, Support Vector Machine(SVM), Decision Tree Classifier, Naive Bayes, KNeighbors classifier, Random Forest, Ada Boost Classifier, Bagging classifier, Extra Trees Classifier, Gradient Boost Classifier, XGBoost Classifier etc.

I have collected a dataset containing 5000 spam & ham messages (approximately) and performed various steps to detect the spam SMSs with high rate of efficiency.

**The basic steps I have performed in this project are:**

* Data Cleaning **– In this step, I started with checking the data available. Found the null values and duplicate data and removed them. Then applied the basic steps of renaming the columns to easily understand the data.**
* EDA (Exploratory Data Analysis) **– In this step, I tried analyzing the data by p plotting various graphs and reading the min and max values from the data. And found that data is highly imbalanced**.
* Text Processing **– To transform the data according to the requirements I have followed some steps like converting the entire text in lower case, applied tokenization to count the total number of characters, words and sentences in each row using ‘punkt’ library from nltk. And then removed special characters, stop words and punctuations from the text and applied stemming. With this created a new column as transformed text which has the text ready for the further process. Performed few steps using word cloud and plotting graphs to know the most common used words in both the categories of messages**
* Model Building **– After getting a better understanding of data, I started building the model using various modules and used 80% of data for training and 20% for testing. Since naive bayes works well on textual data I started with this and the tried with KNeighbors Classifier and Random Forest algorithms to see the performances of different models on the data. (Random forest gave best results with high precision score and accuracy)**
* Improvements **– Various other techniques can be implemented to improve the performance of the models like hyperparameter tuning, trying with alternate features, applying scaling and stacking methods.**