**BUILDING AI POWERED SPAM CLASSIFIER**

Phase – 3

We started building our project by importing the needed header and then loading the dataset which we want to perform our project operations and finally the coding.

**Import the headers:**

%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

from sklearn.model\_selection import GridSearchCV,train\_test\_split,StratifiedKFold,cross\_val\_score,learning\_curve

**Loading the dataset:**

data = pd.read\_csv('dataset/spam.csv', encoding='latin-1')

data.head()

**Coding:**

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()

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

plt.imshow(ham\_wordcloud)

plt.axis("off")

plt.tight\_layout(pad=0)

plt.show()

data = data.replace(['ham','spam'],[0, 1])

data.head(10)