

In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
data=pd.read_csv("spam.csv", encoding="latin-1")
```

In [3]:

```
data.head()
```

Out[3]:

	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

In [4]:

```
data.columns
```

Out[4]:

Index(['v1', 'v2', 'Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], dtype='object')

In [5]:

```
data.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], axis=1, inplace=True)
```

In [6]:

```
data.head()
```

Out[6]:

	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...

In [9]:

```
data['v1']=data['v1'].map({'ham':0, 'spam':1})
```

In [10]:

```
data.head()
```

Out[10]:

	v1	v2
0	0	Go until jurong point, crazy.. Available only ...
1	0	Ok lar... Joking wif u oni...
2	1	Free entry in 2 a wkly comp to win FA Cup fina...
3	0	U dun say so early hor... U c already then say...
4	0	Nah I don't think he goes to usf, he lives aro...

In [11]:

```
from sklearn.feature_extraction.text import CountVectorizer  
from sklearn.model_selection import train_test_split
```

In [15]:

```
X=data['v2']  
y=data['v1']
```

In [16]:

```
X.shape
```

Out[16]:

```
(5572,)
```

In [17]:

```
y.shape
```

Out[17]:

```
(5572,)
```

In [18]:

```
data.isnull().sum()
```

Out[18]:

```
v1      0  
v2      0  
dtype: int64
```

In [19]:

```
cv=CountVectorizer()
```

In [20]:

```
X=cv.fit_transform(X)
```

In [21]:

```
x_train, x_test,y_train, y_test=train_test_split(X,y, test_size=0.2, random_state=42)  
x_train.shape
```

Out[21]:

```
(4457, 8672)
```

In [22]:

```
x_test.shape
```

Out[22]:

```
(1115, 8672)
```

In [23]:

```
from sklearn.naive_bayes import MultinomialNB
```

In [24]:

```
model=MultinomialNB()
```

In [25]:

```
model.fit(x_train, y_train)
```

Out[25]:

```
MultinomialNB()
```

In [26]:

```
model.score(x_test, y_test)
```

Out[26]:

```
0.97847533632287
```

In [27]:

```
msg="You Won 500$"  
data = [msg]  
vect = cv.transform(data).toarray()  
my_prediction = model.predict(vect)
```

In [28]:

```
vect
```

Out[28]:

```
array([[0, 0, 0, ..., 0, 0, 0]], dtype=int64)
```

In [29]:

```
import pickle
pickle.dump(model, open('spam.pkl', 'wb'))
model1 = pickle.load(open('spam.pkl', 'rb'))
```

In [30]:

```
from win32com.client import Dispatch
```

In [31]:

```
def speak(text):
    speak=Dispatch("SAPI.SpVoice")
    speak.Speak(text)
```

In [34]:

```
def result(msg):
    data = [msg]
    vect = cv.transform(data).toarray()
    my_prediction = model1.predict(vect)
    if my_prediction[0]==1:
        speak("This is a Spam mail")
        print("This is a Spam mail")
    else:
        speak("This is not a Spam mail")
        print("This is not a Spam mail")
```

In [36]:

```
import tkinter as tk
```

In [*]:

```
root=tk.Tk()
root.geometry("200x200")
l2=tk.Label(root, text="Email Spam Classification Application")
l2.pack()
l1=tk.Label(root, text="Enter Your Message:")
l1.pack()
text=tk.Entry(root)
text.pack()
def result():
    data = [text.get()]
    vect = cv.transform(data).toarray()
    my_prediction = model1.predict(vect)
    if my_prediction[0]==1:
        speak("This is a Spam mail")
        print("This is a Spam mail")
    else:
        speak("This is not a Spam mail")
        print("This is not a Spam mail")
B=tk.Button(root, text="Click", command=result)
B.pack()

root.mainloop()
```

In []: