Importing libs and dataset

In [33]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report
```

In [23]:
```
df = pd.read_csv('spam.csv', encoding = "ISO-8859-1")
df.head()
```

Out[23]:

	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

Data exploration

In [24]:
```
df.isna().sum()
```

Out[24]:
```
v1              0
v2              0
Unnamed: 2    5522
Unnamed: 3    5560
Unnamed: 4    5566
dtype: int64
```

In [25]:
```
# checking the frequency of ham and spam
df['v1'].value_counts()
```

Out[25]:
```
ham     4825
spam     747
Name: v1, dtype: int64
```

In [26]:
```
df['spam'] = df['v1'].apply(lambda x: 1 if x=='spam' else 0)
df.head()
```

Out[26]:

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

Model creation

In [29]:
```
x_train, x_test, y_train, y_test = train_test_split(df['v2'], df['spam'], test_size=.20)

print("X_train size: ", x_train.shape)
print("Y_train size: ", y_train.shape)
print("X_test size: ", x_test.shape)
print("Y_test size: ", y_test.shape)
```

```
X_train size:  (4457,)
Y_train size:  (4457,)
X_test size:  (1115,)
Y_test size:  (1115,)
```

In [31]:
```
v = CountVectorizer()
x_train_updated = v.fit_transform(x_train.values)
# print(v.get_feature_names())
x_train_updated.toarray()[:2]
```

Out[31]:
```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]], dtype=int64)
```

In [32]:
```
# model creation
model = MultinomialNB()
model.fit(x_train_updated, y_train)
```

Out[32]:
```
MultinomialNB()
```

In [37]:
```
# checking its accuracy
x_test_vectorized = v.transform(x_test)

y_pred = model.predict(x_test_vectorized)
acc = accuracy_score(y_test, y_pred)
print(f"Accuracy: {acc:.2f}")
```

```
Accuracy: 0.99
```

In [39]:
```
report = classification_report(y_test, y_pred)
print(report)
```

```
              precision    recall  f1-score   support

           0       0.99      1.00      0.99       980
           1       0.99      0.92      0.95       135

    accuracy                           0.99      1115
   macro avg       0.99      0.96      0.97      1115
weighted avg       0.99      0.99      0.99      1115
```

Testing

In [40]:
```
text1 = input("Enter a sms: ")
text2 = input("Enter another text: ")
data = [text1, text2]

data = v.transform(data)
model.predict(data)
```

```
Enter a sms: Hello how are you?
Enter another text: You have won a 1 cr. jackpot
```

Out[40]:
```
array([0, 1], dtype=int64)
```