##Importing libraries

```
import numpy  as np
import pandas as pd
import matplotlib.pyplot as plt
import os
for dirname, _,filenames in os.walk('/kaggle/input'):
  for filename in filenames:
    print(os.path.join(dirname,filename))

data = pd.read_csv('spam.csv')
data
```

```
     Category                                            Message
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...
...       ...                                                ...
5567     spam  This is the 2nd time we have tried 2 contact u...
5568      ham              Will ü b going to esplanade fr home?
5569      ham  Pity, * was in mood for that. So...any other s...
5570      ham  The guy did some bitching but I acted like i'd...
5571      ham                        Rofl. Its true to its name

[5572 rows x 2 columns]
```

```
data.columns
```

```
Index(['Category', 'Message'], dtype='object')
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Category  5572 non-null   object
 1   Message   5572 non-null   object
dtypes: object(2)
memory usage: 87.2+ KB
```

##Dropped the columns unnamed:0

```
data.isna().sum()
```

```
Category    0
Message     0
dtype: int64
```

```
data['Spam']=data['Category'].apply(lambda x:1 if x=='spam' else 0 )
data.head(5)
```

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

```
<google.colab._quickchart_helpers.SectionTitle at 0x7ba9ab948430>
```

```
from matplotlib import pyplot as plt
_df_16['index'].plot(kind='hist', bins=20, title='index')
plt.gca().spines[['top', 'right',]].set_visible(False)
plt.tight_layout()
```

```
from matplotlib import pyplot as plt
_df_17['Spam'].plot(kind='hist', bins=20, title='Spam')
plt.gca().spines[['top', 'right',]].set_visible(False)
plt.tight_layout()
```

```
<google.colab._quickchart_helpers.SectionTitle at 0x7ba9ab94aa70>
```

```
from matplotlib import pyplot as plt
import seaborn as sns
_df_18.groupby('Category').size().plot(kind='barh',
color=sns.palettes.mpl_palette('Dark2'))
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
from matplotlib import pyplot as plt
import seaborn as sns
_df_19.groupby('Message').size().plot(kind='barh',
color=sns.palettes.mpl_palette('Dark2'))
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
<google.colab._quickchart_helpers.SectionTitle at 0x7ba9ab94bb50>
```

```
from matplotlib import pyplot as plt
plt.figure(figsize=(6, 6))
_df_20.plot(kind='scatter', x='index', y='Spam', s=32, alpha=.8)
plt.gca().spines[['top', 'right',]].set_visible(False)
plt.tight_layout()
```

```
<google.colab._quickchart_helpers.SectionTitle at 0x7ba9ab948520>
```

```
from matplotlib import pyplot as plt
import seaborn as sns
```

```
def _plot_series(series, series_name, series_index=0):
  from matplotlib import pyplot as plt
  import seaborn as sns
  palette = list(sns.palettes.mpl_palette('Dark2'))
  xs = series['index']
  ys = series['Spam']

  plt.plot(xs, ys, label=series_name, color=palette[series_index %
len(palette)])

fig, ax = plt.subplots(figsize=(10, 5.2), layout='constrained')
df_sorted = _df_21.sort_values('index', ascending=True)
for i, (series_name, series) in
enumerate(df_sorted.groupby('Category')):
  _plot_series(series, series_name, i)
  fig.legend(title='Category', bbox_to_anchor=(1, 1), loc='upper
left')
sns.despine(fig=fig, ax=ax)
plt.xlabel('index')
plt.ylabel('Spam')
```

```
from matplotlib import pyplot as plt
import seaborn as sns
def _plot_series(series, series_name, series_index=0):
  from matplotlib import pyplot as plt
  import seaborn as sns
  palette = list(sns.palettes.mpl_palette('Dark2'))
  xs = series['index']
  ys = series['Spam']

  plt.plot(xs, ys, label=series_name, color=palette[series_index %
len(palette)])

fig, ax = plt.subplots(figsize=(10, 5.2), layout='constrained')
df_sorted = _df_22.sort_values('index', ascending=True)
for i, (series_name, series) in
enumerate(df_sorted.groupby('Message')):
  _plot_series(series, series_name, i)
  fig.legend(title='Message', bbox_to_anchor=(1, 1), loc='upper left')
sns.despine(fig=fig, ax=ax)
plt.xlabel('index')
plt.ylabel('Spam')
```

```
from matplotlib import pyplot as plt
import seaborn as sns
def _plot_series(series, series_name, series_index=0):
  from matplotlib import pyplot as plt
  import seaborn as sns
  palette = list(sns.palettes.mpl_palette('Dark2'))
  counted = (series['index']
```

```
                .value_counts()
              .reset_index(name='counts')
              .rename({'index': 'index'}, axis=1)
              .sort_values('index', ascending=True))
  xs = counted['index']
  ys = counted['counts']
  plt.plot(xs, ys, label=series_name, color=palette[series_index %
len(palette)])

fig, ax = plt.subplots(figsize=(10, 5.2), layout='constrained')
df_sorted = _df_23.sort_values('index', ascending=True)
for i, (series_name, series) in
enumerate(df_sorted.groupby('Category')):
  _plot_series(series, series_name, i)
  fig.legend(title='Category', bbox_to_anchor=(1, 1), loc='upper
left')
sns.despine(fig=fig, ax=ax)
plt.xlabel('index')
plt.ylabel('count()')
```

```
from matplotlib import pyplot as plt
import seaborn as sns
def _plot_series(series, series_name, series_index=0):
  from matplotlib import pyplot as plt
  import seaborn as sns
  palette = list(sns.palettes.mpl_palette('Dark2'))
  counted = (series['index']
                .value_counts()
              .reset_index(name='counts')
              .rename({'index': 'index'}, axis=1)
              .sort_values('index', ascending=True))
  xs = counted['index']
  ys = counted['counts']
  plt.plot(xs, ys, label=series_name, color=palette[series_index %
len(palette)])

fig, ax = plt.subplots(figsize=(10, 5.2), layout='constrained')
df_sorted = _df_24.sort_values('index', ascending=True)
for i, (series_name, series) in
enumerate(df_sorted.groupby('Message')):
  _plot_series(series, series_name, i)
  fig.legend(title='Message', bbox_to_anchor=(1, 1), loc='upper left')
sns.despine(fig=fig, ax=ax)
plt.xlabel('index')
plt.ylabel('count()')
```

```
<google.colab._quickchart_helpers.SectionTitle at 0x7ba9ab94a4d0>
```

```
from matplotlib import pyplot as plt
_df_25['index'].plot(kind='line', figsize=(8, 4), title='index')
```

```
plt.gca().spines[['top', 'right']].set_visible(False)
plt.tight_layout()
```

```
from matplotlib import pyplot as plt
_df_26['Spam'].plot(kind='line', figsize=(8, 4), title='Spam')
plt.gca().spines[['top', 'right']].set_visible(False)
plt.tight_layout()
```

```
<google.colab._quickchart_helpers.SectionTitle at 0x7ba9ab94ba00>
```

```
from matplotlib import pyplot as plt
import seaborn as sns
import pandas as pd
plt.subplots(figsize=(8, 8))
df_2dhist = pd.DataFrame({
    x_label: grp['Message'].value_counts()
    for x_label, grp in _df_27.groupby('Category')
})
sns.heatmap(df_2dhist, cmap='viridis')
plt.xlabel('Category')
plt.ylabel('Message')
```

```
<google.colab._quickchart_helpers.SectionTitle at 0x7ba9ab94afb0>
```

```
from matplotlib import pyplot as plt
import seaborn as sns
figsize = (12, 1.2 * len(_df_28['Category'].unique()))
plt.figure(figsize=figsize)
sns.violinplot(_df_28, x='index', y='Category', inner='stick',
palette='Dark2')
sns.despine(top=True, right=True, bottom=True, left=True)
```

```
from matplotlib import pyplot as plt
import seaborn as sns
figsize = (12, 1.2 * len(_df_29['Message'].unique()))
plt.figure(figsize=figsize)
sns.violinplot(_df_29, x='index', y='Message', inner='stick',
palette='Dark2')
sns.despine(top=True, right=True, bottom=True, left=True)
```

```
from matplotlib import pyplot as plt
import seaborn as sns
figsize = (12, 1.2 * len(_df_30['Category'].unique()))
plt.figure(figsize=figsize)
sns.violinplot(_df_30, x='Spam', y='Category', inner='stick',
palette='Dark2')
sns.despine(top=True, right=True, bottom=True, left=True)
```

```
from matplotlib import pyplot as plt
import seaborn as sns
figsize = (12, 1.2 * len(_df_31['Message'].unique()))
```

```
plt.figure(figsize=figsize)
sns.violinplot(_df_31, x='Spam', y='Message', inner='stick',
palette='Dark2')
sns.despine(top=True, right=True, bottom=True, left=True)
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(data.Message,data.Spam,
test_size=0.20)

from sklearn.feature_extraction.text import CountVectorizer
```

##Naive Bayes Have three Classifier(Bernouli,Multinominal,Gaussian) Here I use Multinominal Bayes Because here data in a discrete form discrete data(e.g movie ratings ranging 1 to 5 as each rating will have certain frequency to represent)

```
from sklearn.naive_bayes import MultinomialNB

from sklearn.pipeline import Pipeline
clf=Pipeline([
    ('vectorizer',CountVectorizer()),
    ('nb',MultinomialNB())
])
```

##Traning The Model

```
clf.fit(X_train,y_train)
```

```
Pipeline(steps=[('vectorizer', CountVectorizer()), ('nb',
MultinomialNB())])
```

##Here I given Two email Two detect 1st One is looking good and the other one looking spam

```


emails=[
    'Sounds great! Are you home now?',
    'Will u meet ur dream partner soon? Is ur career off 2 a flyng
start? 2 find out free, txt HORO followed by ur star sign, e. g. HORO
ARIES'
]
```

###Predict Email

```
clf.predict(emails)
```

```
array([0, 1])
```

##Prediction of model

```
clf.score(X_test,y_test)
```

```
0.9883408071748879
```