

```
In [1]: ▶ import pandas as pd
import numpy as np
import seaborn as sns
```

```
In [2]: ▶ data=pd.read_csv("test.csv")
```

```
In [3]: ▶ data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype
---  -
0   id                    1000 non-null   int64
1   battery_power        1000 non-null   int64
2   blue                 1000 non-null   int64
3   clock_speed          1000 non-null   float64
4   dual_sim             1000 non-null   int64
5   fc                   1000 non-null   int64
6   four_g               1000 non-null   int64
7   int_memory           1000 non-null   int64
8   m_dep                1000 non-null   float64
9   mobile_wt            1000 non-null   int64
10  n_cores               1000 non-null   int64
11  pc                    1000 non-null   int64
12  px_height             1000 non-null   int64
13  px_width              1000 non-null   int64
14  ram                   1000 non-null   int64
15  sc_h                  1000 non-null   int64
16  sc_w                  1000 non-null   int64
17  talk_time             1000 non-null   int64
18  three_g               1000 non-null   int64
19  touch_screen          1000 non-null   int64
20  wifi                  1000 non-null   int64
dtypes: float64(2), int64(19)
memory usage: 164.2 KB
```

In [4]: `data.head(5)`

Out[4]:

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_v
0	1	1043	1	1.8	1	14	0	5	0.1	19
1	2	841	1	0.5	1	4	1	61	0.8	19
2	3	1807	1	2.8	0	1	0	27	0.9	18
3	4	1546	0	0.5	1	18	1	25	0.5	9
4	5	1434	0	1.4	0	11	1	49	0.5	10

5 rows × 11 columns

In [5]: `!pip install sklearn`

```
Collecting sklearn
  Downloading sklearn-0.0.tar.gz (1.1 kB)
Requirement already satisfied: scikit-learn in c:\users\jill\anaconda3\lib\site-packages (from sklearn) (0.23.2)
Requirement already satisfied: scipy>=0.19.1 in c:\users\jill\anaconda3\lib\site-packages (from scikit-learn->sklearn) (1.5.2)
Requirement already satisfied: numpy>=1.13.3 in c:\users\jill\anaconda3\lib\site-packages (from scikit-learn->sklearn) (1.19.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\jill\anaconda3\lib\site-packages (from scikit-learn->sklearn) (2.1.0)
Requirement already satisfied: joblib>=0.11 in c:\users\jill\anaconda3\lib\site-packages (from scikit-learn->sklearn) (0.17.0)
Building wheels for collected packages: sklearn
  Building wheel for sklearn (setup.py): started
  Building wheel for sklearn (setup.py): finished with status 'done'
  Created wheel for sklearn: filename=sklearn-0.0-py2.py3-none-any.whl size=1321 sha256=85205ab8379b15bf8dda98315eb4f6d7122e4804481489c09857aca9878b48db
  Stored in directory: c:\users\jill\appdata\local\pip\cache\wheels\22\0b\40\fd3f795caaa1fb4c6cb738bc1f5610be1e57da95849bfc897
Successfully built sklearn
Installing collected packages: sklearn
Successfully installed sklearn-0.0
```

In [6]: `from sklearn import preprocessing`

In [7]: `from sklearn.model_selection import train_test_split`

In [8]: `df=pd.read_csv("test.csv")`

```
In [10]: df1=df[['battery_power','blue','clock_speed','int_memory','mobile_wt','px_height','px_width','ram']]
df1.head()
```

Out[10]:

	battery_power	blue	clock_speed	int_memory	mobile_wt	px_height	px_width	ram
0	1043	1	1.8	5	193	226	1412	3476
1	841	1	0.5	61	191	746	857	3895
2	1807	1	2.8	27	186	1270	1366	2396
3	1546	0	0.5	25	96	295	1752	3893
4	1434	0	1.4	49	108	749	810	1773

```
In [11]: x=df1.drop('blue',axis=1)
y=df1['blue']
```

```
In [12]: from sklearn.model_selection import train_test_split
x_train ,x_test,y_train,y_test =train_test_split(x,y,test_size=0.30,random_state=42)
```

```
In [13]: from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report,confusion_matrix
```

```
In [14]: model=GaussianNB()
model.fit(x_train,y_train)
```

Out[14]: GaussianNB()

```
In [15]: predictions=model.predict(x_test)
```

```
In [16]: print(confusion_matrix(y_test,predictions))
print('\n')
print(classification_report(y_test,predictions))
```

```
[[65 73]
 [68 94]]
```

	precision	recall	f1-score	support
0	0.49	0.47	0.48	138
1	0.56	0.58	0.57	162
accuracy			0.53	300
macro avg	0.53	0.53	0.53	300
weighted avg	0.53	0.53	0.53	300

In []: ▶