

Machine Learning Assignment 2

700746287

NAKKA SAI SNUSHA

Video Link : <https://drive.google.com/file/d/1u4uzQwZ2F6c0WrzNF3-qWVZSWPtJOI7g/view?usp=sharing>

Github Link: https://github.com/NSnusha/ML_Assignment2

```
In [107]: import warnings
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn import preprocessing
import matplotlib.pyplot as plt
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, recall_score, precision_score, classification_report, confusion_matrix
warnings.filterwarnings("ignore")
```

```
In [108]: #1. Read the provided CSV file 'data.csv'. https://drive.google.com/drive/folders/1h8C3mLsso-R-sIOLsvoYwPLzy2fJ4IOF?usp
df = pd.read_csv("/Users/snush/OneDrive/Desktop/dataset/data.csv")
print(df.head())
```

	Duration	Pulse	Maxpulse	Calories
0	60	110	130	409.1
1	60	117	145	479.0
2	60	103	135	340.0
3	45	109	175	282.4
4	45	117	148	406.0

```
In [109]: #2. Show the basic statistical description about the data.
print(df.describe())
```

	Duration	Pulse	Maxpulse	Calories
count	169.000000	169.000000	169.000000	164.000000
mean	63.846154	107.461538	134.047337	375.790244
std	42.299949	14.510259	16.450434	266.379919
min	15.000000	80.000000	100.000000	50.300000
25%	45.000000	100.000000	124.000000	250.925000
50%	60.000000	105.000000	131.000000	318.600000
75%	60.000000	111.000000	141.000000	387.600000
max	300.000000	159.000000	184.000000	1860.400000

```
In [110]: df.isnull().any()
```

```
In [110]: df.isnull().any()
```

```
Out[110]: Duration    False
Pulse                False
Maxpulse             False
Calories              True
dtype: bool
```

```
In [111]: #Replace the null values with the mean
df.fillna(df.mean(), inplace=True)
df.isnull().any()
```

```
Out[111]: Duration    False
Pulse                False
Maxpulse             False
Calories              False
dtype: bool
```

```
In [112]: df.agg({'Duration': ['min', 'max', 'count', 'mean'], 'Pulse': ['min', 'max', 'count', 'mean']})
```

```
Out[112]:
```

	Duration	Pulse
min	15.000000	80.000000
max	300.000000	159.000000
count	169.000000	169.000000
mean	63.846154	107.461538

📄 + 🔍 📄 ⬆️ ⬆️ ▶️ Run 🛑 ↺️ ▶️ Code ▾ 🗨

```
In [113]: df.loc[(df['Calories']>500)&(df['Calories']<1000)]
```

```
Out[113]:
```

	Duration	Pulse	Maxpulse	Calories
51	80	123	146	643.1
62	160	109	135	853.0
65	180	90	130	800.4
66	150	105	135	873.4
67	150	107	130	816.0
72	90	100	127	700.0
73	150	97	127	953.2
75	90	98	125	563.2
78	120	100	130	500.4
90	180	101	127	600.1
99	90	93	124	604.1
103	90	90	100	500.4
106	180	90	120	800.3
108	90	90	120	500.3

```
In [114]: #6. Filter the dataframe to select the rows with calories values > 500 and pulse < 100.
df.loc[(df['Calories']>500)&(df['Pulse']<100)]
```

```
Out[114]:
```

	Duration	Pulse	Maxpulse	Calories
65	180	90	130	800.4
70	150	97	129	1115.0
73	150	97	127	953.2
75	90	98	125	563.2
99	90	93	124	604.1
103	90	90	100	500.4
106	180	90	120	800.3
108	90	90	120	500.3

```
In [115]: #7. Create a new "df_modified" dataframe that contains all the columns from df except for "Maxpulse".
df_modified = df[['Duration','Pulse','Calories']]
df_modified.head()
```

```
Out[115]:
```

	Duration	Pulse	Calories
0	60	110	409.1
1	60	117	479.0
2	60	103	340.0
3	45	109	282.4
4	45	117	406.0

```
In [116]: #8. Delete the "Maxpulse" column from the main df dataframe
del df['Maxpulse']
```

```
In [117]: df.head()
```

```
Out[117]:
```

	Duration	Pulse	Calories
0	60	110	409.1
1	60	117	479.0
2	60	103	340.0
3	45	109	282.4
4	45	117	406.0

```
In [118]: df.dtypes
```

```
Out[118]: Duration      int64
Pulse      int64
Calories   float64
dtype: object
```

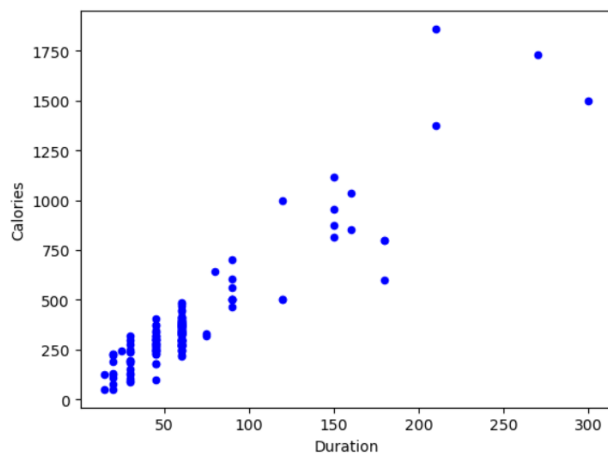
```
In [119]: df['Calories'] = df['Calories'].astype(np.int64)
df.dtypes
```

```
Out[119]: Duration      int64
Pulse      int64
Calories   int64
dtype: object
```

⏏ + 🔍 📄 ⬆️ ⬆️ ▶️ Run 🛑 ↺️ ⏩ Code ▾ 🗨

```
In [120]: df.plot.scatter(x='Duration',y='Calories',c='blue')
```

```
Out[120]: <Axes: xlabel='Duration', ylabel='Calories'>
```



```

In [16]: # In[26]:
#Question 2

import warnings
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats.stats import pearsonr
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, recall_score, precision_score, classification_report, confusion_matrix

get_ipython().run_line_magic('matplotlib', 'inline')
# Suppress warnings
warnings.filterwarnings("ignore")
glass_data = pd.read_csv("/Users/snush/OneDrive/Desktop/dataset/glass.csv")
X = glass_data.drop('Type', axis=1)
y = glass_data['Type']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

```

```

In [18]: classifier = GaussianNB()

classifier.fit(X_train, y_train)

```

```

Out[18]: GaussianNB
GaussianNB()

```

```

In [19]: y_pred = classifier.predict(X_test)
accuracy = classifier.score(X_test, y_test)
print("Accuracy:", accuracy)

print(classification_report(y_test, y_pred))

```

```

Accuracy: 0.5581395348837209

```

	precision	recall	f1-score	support
1	0.41	0.64	0.50	11
2	0.43	0.21	0.29	14
3	0.40	0.67	0.50	3
5	0.50	0.25	0.33	4
6	1.00	1.00	1.00	3
7	0.89	1.00	0.94	8
accuracy			0.56	43
macro avg	0.60	0.63	0.59	43
weighted avg	0.55	0.56	0.53	43

```

In [30]: from sklearn.svm import SVC, LinearSVC
classifier = LinearSVC()
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

```

```

In [31]: svm_classifier = SVC(kernel='linear')
svm_classifier.fit(X_train, y_train)

```

```

Out[31]: SVC
SVC(kernel='linear')

```

```

In [32]: y_pred = svm_classifier.predict(X_test)

```

```

In [33]: accuracy = svm_classifier.score(X_test, y_test)
print("Accuracy:", accuracy)

```

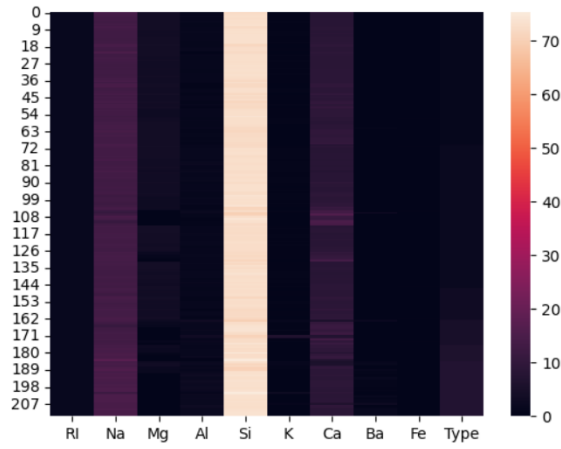
```

Accuracy: 0.7441860465116279

```

```
In [34]: sns.heatmap(data=glass) #HeatMap Visualization for above dataset
```

```
Out[34]: <Axes: >
```



```
In [35]: sns.scatterplot(data=glass) #ScatterPlot Visualization for above dataset
```

```
Out[35]: <Axes: >
```

