ML LAB ASSIGNMENT

SUPRATIM NAG -- CSE-AIML/22/057 -- GROUP-B

Q-8b:Implementation of Ensemble Techniques

------- Write a python code to show ensemble technique using Bagging mechanism.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import model_selection
from sklearn.svm import SVC
from sklearn.ensemble import BaggingClassifier
from sklearn.model_selection import train_test_split
```

In [2]:
```
data = pd.read_csv(r"C:\Users\SUPRATIM NAG\OneDrive\Documents\ML\Personal_Datasets\Dataset.csv")
data.head(1)
```

Out[2]:

	Patient ID	Age	Blood Pressure	Cholesterol Levels	Heart Rate	BMI	Diagnosis	Treatment Plan	Recovery Status	Medication Type	Follow-up Requirement
0	101	65	130	250	72	28.0	Hypertension with high cholesterol.	Medication: Lisinopril (blood pressure), Stati...	Active Recovery	Lisinopril, Statins.	Quarterly.

In [3]:
```
data.shape
```

Out[3]:
```
(100, 11)
```

In [4]:
```
meddata=data[['Age','Blood Pressure','Cholesterol Levels','Heart Rate','BMI','Diagnosis']]
meddata.head(1)
```

Out[4]:

	Age	Blood Pressure	Cholesterol Levels	Heart Rate	BMI	Diagnosis
0	65	130	250	72	28.0	Hypertension with high cholesterol.

In [5]:
```
meddata['Diagnosis'] = meddata['Diagnosis'].apply(
    lambda x: 1 if any(condition in x for condition in ['Hypertension', 'Obesity', 'Overweight']) else 0
)
```

```
C:\Users\SUPRATIM NAG\AppData\Local\Temp\ipykernel_14508\3012899904.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  meddata['Diagnosis'] = meddata['Diagnosis'].apply(
```

In [6]:
```
print(meddata['Diagnosis'].value_counts())
```

```
Diagnosis
0    63
1    37
Name: count, dtype: int64
```

In [7]:
```
x = meddata.drop('Diagnosis', axis=1)
y = meddata['Diagnosis']
```

In [9]:
```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.20)
```

In [10]:
```
# PROCESS 1
model = BaggingClassifier(SVC())
```

In [11]:
```
kfold=model_selection.KFold(n_splits=10)
results = model_selection.cross_val_score(model,X_train, y_train, cv=kfold)
```

In [12]:
```
print("Results:", results)
print("Mean Results:", results.mean())
```

```
Results: [0.5   0.75  0.75  0.75  0.875 0.25  0.625 0.625 0.625 0.75 ]
Mean Results: 0.65
```

In [13]:
```
#PROCESS 2
model = BaggingClassifier(SVC())
model.fit(X_train, y_train)
```

Out[13]:
```
BaggingClassifier
  estimator: SVC
    SVC
```

In [14]:
```
y_pred = model.predict(X_test)
```

In [15]:
```
model.score(X_test, y_test)
```

Out[15]:
```
0.55
```

In [16]:
```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,y_pred))
```

```
print(classification_report(y_test,y_pred))
```

```
[[11  0]
 [ 9  0]]
              precision    recall  f1-score   support

           0       0.55      1.00      0.71        11
           1       0.00      0.00      0.00         9

    accuracy                           0.55        20
   macro avg       0.28      0.50      0.35        20
weighted avg       0.30      0.55      0.39        20
```

```
c:\Users\SUPRATIM NAG\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning: P
recision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
c:\Users\SUPRATIM NAG\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning: P
recision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
c:\Users\SUPRATIM NAG\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning: P
recision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```