```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix

df = pd.read_csv(r"C:\Users\Jayditya\Downloads\DSBDA LAB\Lab\
Experiments\Datasets\13Iris.csv")
print(df.head())
print(df.shape)
print(df.info())
```

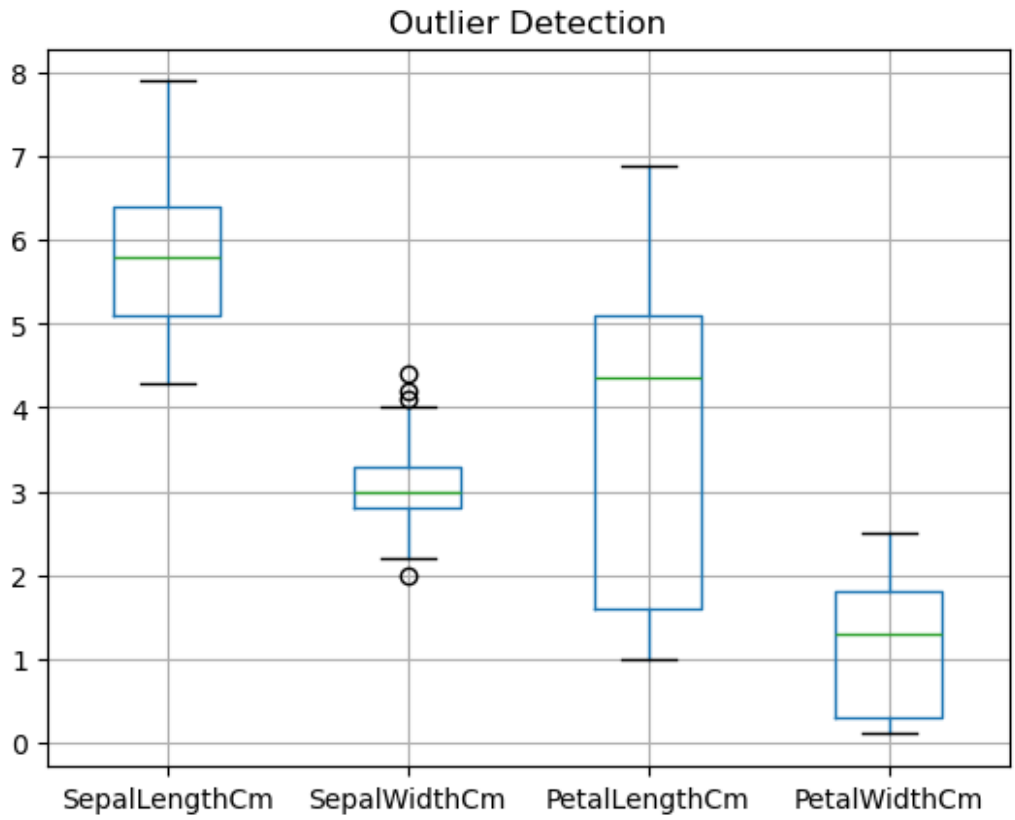
```
   Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
Species
0   1            5.1           3.5            1.4           0.2  Iris-
setosa
1   2            4.9           3.0            1.4           0.2  Iris-
setosa
2   3            4.7           3.2            1.3           0.2  Iris-
setosa
3   4            4.6           3.1            1.5           0.2  Iris-
setosa
4   5            5.0           3.6            1.4           0.2  Iris-
setosa
(150, 6)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             150 non-null    int64
 1   SepalLengthCm  150 non-null    float64
 2   SepalWidthCm   150 non-null    float64
 3   PetalLengthCm  150 non-null    float64
 4   PetalWidthCm   150 non-null    float64
 5   Species        150 non-null    object
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
None
```

```
print("Missing values: ",df.isnull().sum())
# Option 1: Drop rows with missing values
# df = df.dropna()

# Option 2: Impute missing values with mean (for numerical columns)
df['SepalLengthCm'] =
df['SepalLengthCm'].fillna(df['SepalLengthCm'].mean())
```

```
df['SepalWidthCm'] =
df['SepalWidthCm'].fillna(df['SepalWidthCm'].median())
df['PetalLengthCm'] =
df['PetalLengthCm'].fillna(df['PetalLengthCm'].mean())
df['PetalWidthCm'] =
df['PetalWidthCm'].fillna(df['PetalWidthCm'].median())

# Option 3: Impute missing values in categorical columns with mode
(for species)
df['Species'] = df['Species'].fillna(df['Species'].mode()[0])

# Verify if there are any remaining missing values
print(df.isnull().sum())
```

```
Missing values:  Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

```
df.drop('Id', axis=1, inplace=True)
df.drop('Species', axis=1).boxplot()
plt.title("Outlier Detection")
plt.show()
```




```
X = df.drop('Species', axis=1)
y = df['Species']

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y,
test_size=0.3, random_state=42)

nb = GaussianNB()
nb.fit(X_train, y_train)
y_pred = nb.predict(X_test)

cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", cm)
```

```
Confusion Matrix:
 [[19  0  0]
 [ 0 12  1]
 [ 0  0 13]]
```

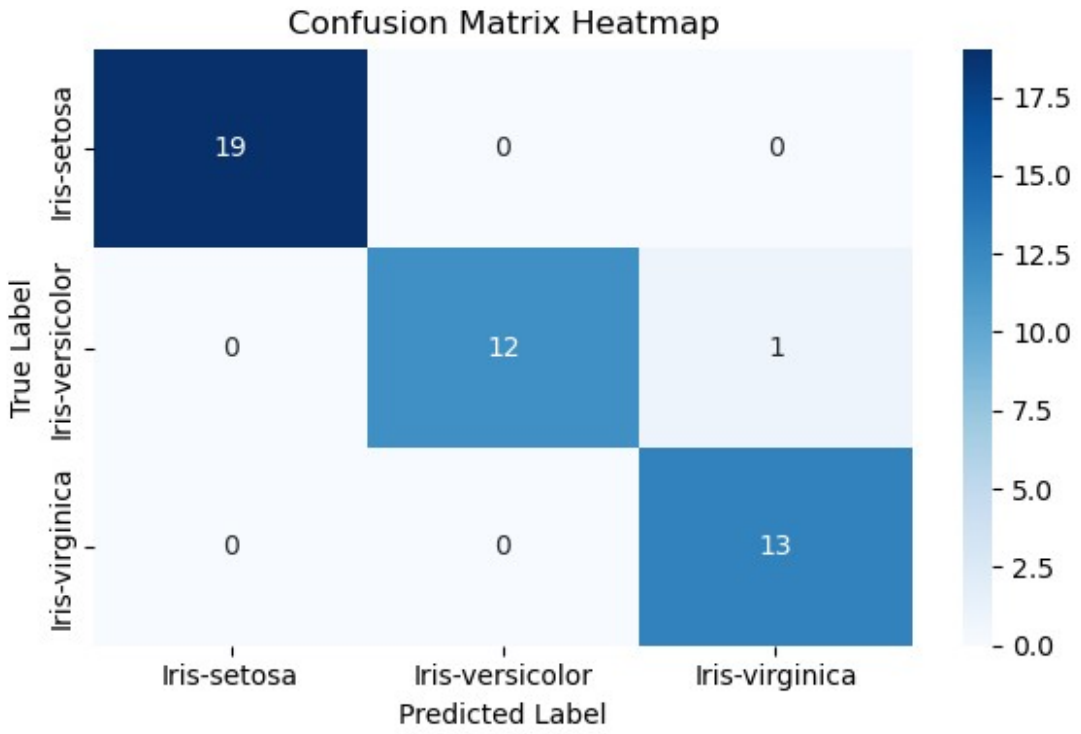
```
labels = df['Species'].unique()
cm_df = pd.DataFrame(cm, index=labels, columns=labels)
print(cm_df)
```

```
                 Iris-setosa  Iris-versicolor  Iris-virginica
Iris-setosa               19                0               0
Iris-versicolor            0               12               1
Iris-virginica             0                0              13
```

```python
# Compute TP, FP, TN, FN, Accuracy, Error Rate, Precision, Recall
total = np.sum(cm)
for i, label in enumerate(labels):
    TP = cm[i, i]
    FP = cm[:, i].sum() - TP
    FN = cm[i, :].sum() - TP
    TN = total - (TP + FP + FN)

    accuracy = (TP + TN) / total
    error_rate = 1 - accuracy
    precision = TP / (TP + FP) if (TP + FP) > 0 else 0
    recall = TP / (TP + FN) if (TP + FN) > 0 else 0

    print(f"\nClass: {label}")
    print(f"TP: {TP}, FP: {FP}, FN: {FN}, TN: {TN}")
    print(f"Accuracy: {accuracy:.2f}, Error Rate: {error_rate:.2f}")
    print(f"Precision: {precision:.2f}, Recall: {recall:.2f}")
```

```
Class: Iris-setosa
TP: 19, FP: 0, FN: 0, TN: 26
Accuracy: 1.00, Error Rate: 0.00
Precision: 1.00, Recall: 1.00

Class: Iris-versicolor
TP: 12, FP: 0, FN: 1, TN: 32
Accuracy: 0.98, Error Rate: 0.02
Precision: 1.00, Recall: 0.92

Class: Iris-virginica
TP: 13, FP: 1, FN: 0, TN: 31
Accuracy: 0.98, Error Rate: 0.02
Precision: 0.93, Recall: 1.00
```

```python
# Plot Confusion Matrix as Heatmap
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=nb.classes_, yticklabels=nb.classes_)
plt.title("Confusion Matrix Heatmap")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.tight_layout()
plt.show()
```


Confusion Matrix Heatmap
Iris-setosa
Iris-versicolor
Iris-virginica
True Label
19
0
0
0
12
1
0
0
13
Iris-setosa
Iris-versicolor
Iris-virginica
Predicted Label
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0