In [1]:
```
!pip install pandas numpy matplotlib seaborn scikit-learn
```

```
Requirement already satisfied: pandas in /home/sargam/.conda/envs/myenv/li
b/python3.11/site-packages (2.2.3)
Requirement already satisfied: numpy in /home/sargam/.conda/envs/myenv/li
b/python3.11/site-packages (2.0.1)
Requirement already satisfied: matplotlib in /home/sargam/.conda/envs/myen
v/lib/python3.11/site-packages (3.10.1)
Requirement already satisfied: seaborn in /home/sargam/.conda/envs/myenv/l
ib/python3.11/site-packages (0.13.2)
Requirement already satisfied: scikit-learn in /home/sargam/.conda/envs/my
env/lib/python3.11/site-packages (1.6.1)
Requirement already satisfied: python-dateutil>=2.8.2 in /home/sargam/.con
da/envs/myenv/lib/python3.11/site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /home/sargam/.conda/envs/my
env/lib/python3.11/site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in /home/sargam/.conda/envs/
myenv/lib/python3.11/site-packages (from pandas) (2025.2)
Requirement already satisfied: contourpy>=1.0.1 in /home/sargam/.conda/env
s/myenv/lib/python3.11/site-packages (from matplotlib) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /home/sargam/.conda/envs/my
env/lib/python3.11/site-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /home/sargam/.conda/en
vs/myenv/lib/python3.11/site-packages (from matplotlib) (4.57.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /home/sargam/.conda/en
vs/myenv/lib/python3.11/site-packages (from matplotlib) (1.4.8)
Requirement already satisfied: packaging>=20.0 in /home/sargam/.conda/env
s/myenv/lib/python3.11/site-packages (from matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in /home/sargam/.conda/envs/myen
v/lib/python3.11/site-packages (from matplotlib) (11.2.1)
Requirement already satisfied: pyparsing>=2.3.1 in /home/sargam/.conda/env
s/myenv/lib/python3.11/site-packages (from matplotlib) (3.2.3)
Requirement already satisfied: scipy>=1.6.0 in /home/sargam/.conda/envs/my
env/lib/python3.11/site-packages (from scikit-learn) (1.15.2)
Requirement already satisfied: joblib>=1.2.0 in /home/sargam/.conda/envs/m
yenv/lib/python3.11/site-packages (from scikit-learn) (1.5.0)
Requirement already satisfied: threadpoolctl>=3.1.0 in /home/sargam/.cond
a/envs/myenv/lib/python3.11/site-packages (from scikit-learn) (3.6.0)
Requirement already satisfied: six>=1.5 in /home/sargam/.conda/envs/myenv/
lib/python3.11/site-packages (from python-dateutil>=2.8.2->pandas) (1.17.
0)
```

In [3]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, precision_score, recall_score
import seaborn as sns
import matplotlib.pyplot as plt
```

In [4]:
```
df = pd.read_csv("/home/sargam/Downloads/iris (1).csv")
```

In [5]:
```
print("First few rows of the dataset:")
print(df.head())
print("\nDataset Information:")
print(df.info())
```

```
First few rows of the dataset:
   sepal_length  sepal_width  petal_length  petal_width species
0           5.1          3.5           1.4          0.2  setosa
1           4.9          3.0           1.4          0.2  setosa
2           4.7          3.2           1.3          0.2  setosa
3           4.6          3.1           1.5          0.2  setosa
4           5.0          3.6           1.4          0.2  setosa

Dataset Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
None
```

In [6]:
```
encoder = LabelEncoder()
df['species'] = encoder.fit_transform(df['species'])
```

In [7]:
```
print("\nMissing values in the dataset:")
print(df.isnull().sum())
```

```
Missing values in the dataset:
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

In [8]:
```
X = df.drop('species', axis=1)
Y = df['species']
```

In [9]:
```
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2,
```

In [10]:
```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

In [11]:
```
naive_bayes_model = GaussianNB()
naive_bayes_model.fit(X_train_scaled, y_train)
```

Out[11]:
```
▾ GaussianNB
GaussianNB()
```

In [12]:
```
y_pred = naive_bayes_model.predict(X_test_scaled)
```

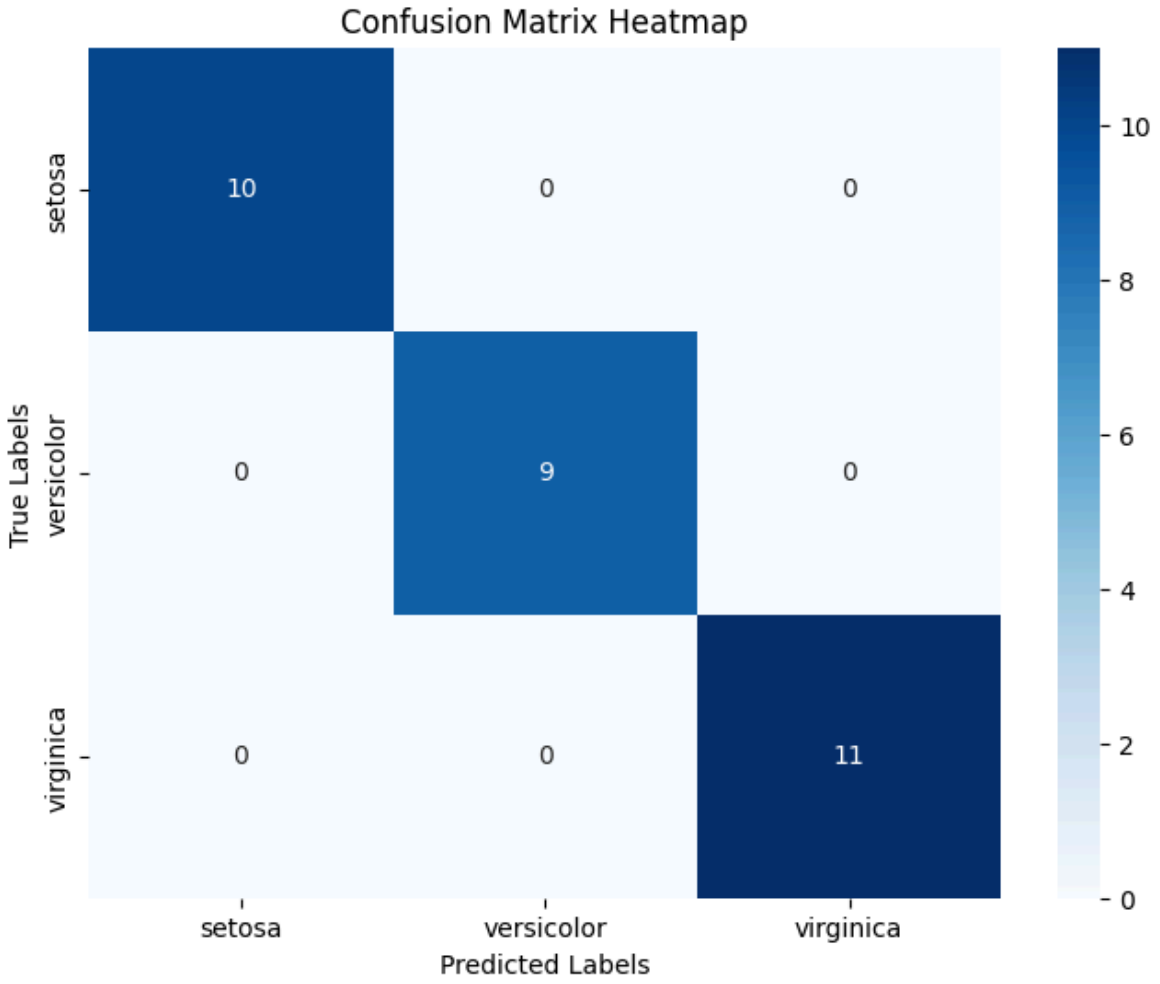
In [13]:
```
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='micro')
```

```
recall = recall_score(y_test, y_pred, average='micro')
conf_matrix = confusion_matrix(y_test, y_pred)
```

In [14]:
```
print("\nModel Evaluation:")
print(f'Accuracy: {accuracy}')
print(f'Precision: {precision}')
print(f'Recall: {recall}')
print(f'Confusion Matrix:\n{conf_matrix}')
```

```
Model Evaluation:
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
Confusion Matrix:
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
```

In [15]:
```
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=e
plt.title('Confusion Matrix Heatmap')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()
```



In [ ]:
```
```