Assignment 1

Suvajit Sadhukhan
4th year 1st semester (302211001005)

CODE

```
import pandas as pd
import numpy as np
import sklearn
import matplotlib.pyplot as plt

from sklearn.datasets import load_iris, load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB, MultinomialNB, BernoulliNB
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix

# Load the datasets
iris = load_iris()
cancer = load_breast_cancer()

# Split the datasets into train and test sets
X_iris, y_iris = iris.data, iris.target
X_train_iris, X_test_iris, y_train_iris, y_test_iris = train_test_split(X_iris, y_iris, test_size=0.2, random_state=42)

X_cancer, y_cancer = cancer.data, cancer.target
X_train_cancer, X_test_cancer, y_train_cancer, y_test_cancer = train_test_split(X_cancer, y_cancer, test_size=0.2, random_state=42)

"""# Iris plants dataset"""

# Gaussian Naive Bayes
gnb = GaussianNB()
gnb.fit(X_train_iris, y_train_iris)
y_pred_gnb = gnb.predict(X_test_iris)
print("Gaussian Naive Bayes - Iris:")
print(f"Accuracy: {accuracy_score(y_test_iris, y_pred_gnb):.2f}")
print(f"Precision: {precision_score(y_test_iris, y_pred_gnb, average='weighted'):.2f}")
print(f"Recall: {recall_score(y_test_iris, y_pred_gnb, average='weighted'):.2f}")
print(f"F1-Score: {f1_score(y_test_iris, y_pred_gnb, average='weighted'):.2f}")
print("Confusion Matrix:")
print(confusion_matrix(y_test_iris, y_pred_gnb))

# Multinomial Naive Bayes
mnb = MultinomialNB()
mnb.fit(X_train_iris, y_train_iris)
y_pred_mnb = mnb.predict(X_test_iris)
print("\nMultinomial Naive Bayes - Iris:")
print(f"Accuracy: {accuracy_score(y_test_iris, y_pred_mnb):.2f}")
print(f"Precision: {precision_score(y_test_iris, y_pred_mnb, average='weighted'):.2f}")
print(f"Recall: {recall_score(y_test_iris, y_pred_mnb, average='weighted'):.2f}")
print(f"F1-Score: {f1_score(y_test_iris, y_pred_mnb, average='weighted'):.2f}")
print("Confusion Matrix:")
print(confusion_matrix(y_test_iris, y_pred_mnb))
```

```python
# Bernoulli Naive Bayes
bnb = BernoulliNB()
bnb.fit(X_train_iris, y_train_iris)
y_pred_bnb = bnb.predict(X_test_iris)
print("\nBernoulli Naive Bayes - Iris:")
print(f"Accuracy: {accuracy_score(y_test_iris, y_pred_bnb):.2f}")
print(f"Precision: {precision_score(y_test_iris, y_pred_bnb, average='weighted'):.2f}")
print(f"Recall: {recall_score(y_test_iris, y_pred_bnb, average='weighted'):.2f}")
print(f"F1-Score: {f1_score(y_test_iris, y_pred_bnb, average='weighted'):.2f}")
print("Confusion Matrix:")
print(confusion_matrix(y_test_iris, y_pred_bnb))

# Decision Tree Classifier

dtc = DecisionTreeClassifier(random_state=42)
dtc.fit(X_train_iris, y_train_iris)
y_pred_dtc = dtc.predict(X_test_iris)
print("\nDecision Tree Classifier - Iris:")
print(f"Accuracy: {accuracy_score(y_test_iris, y_pred_dtc):.2f}")
print(f"Precision: {precision_score(y_test_iris, y_pred_dtc, average='weighted'):.2f}")
print(f"Recall: {recall_score(y_test_iris, y_pred_dtc, average='weighted'):.2f}")
print(f"F1-Score: {f1_score(y_test_iris, y_pred_dtc, average='weighted'):.2f}")
print("Confusion Matrix:")
print(confusion_matrix(y_test_iris, y_pred_dtc))

plt.figure(figsize=(10, 8))
plot_tree(dtc, feature_names=iris.feature_names, class_names=iris.target_names, filled=True)
plt.title("Decision Tree - Iris")
plt.savefig("iris_decision_tree.png")

"""# Wisconsin Breast Cancer Dataset"""

# Gaussian Naive Bayes
gnb = GaussianNB()
gnb.fit(X_train_cancer, y_train_cancer)
y_pred_gnb = gnb.predict(X_test_cancer)
print("Gaussian Naive Bayes - Cancer:")
print(f"Accuracy: {accuracy_score(y_test_cancer, y_pred_gnb):.2f}")
print(f"Precision: {precision_score(y_test_cancer, y_pred_gnb, average='weighted'):.2f}")
print(f"Recall: {recall_score(y_test_cancer, y_pred_gnb, average='weighted'):.2f}")
print(f"F1-Score: {f1_score(y_test_cancer, y_pred_gnb, average='weighted'):.2f}")
print("Confusion Matrix:")
print(confusion_matrix(y_test_cancer, y_pred_gnb))

# Multinomial Naive Bayes
mnb = MultinomialNB()
mnb.fit(X_train_cancer, y_train_cancer)
y_pred_mnb = mnb.predict(X_test_cancer)
print("\nMultinomial Naive Bayes - Cancer:")
print(f"Accuracy: {accuracy_score(y_test_cancer, y_pred_mnb):.2f}")
print(f"Precision: {precision_score(y_test_cancer, y_pred_mnb, average='weighted'):.2f}")
print(f"Recall: {recall_score(y_test_cancer, y_pred_mnb, average='weighted'):.2f}")
print(f"F1-Score: {f1_score(y_test_cancer, y_pred_mnb, average='weighted'):.2f}")
print("Confusion Matrix:")
print(confusion_matrix(y_test_cancer, y_pred_mnb))
```

```
# Bernoulli Naive Bayes
bnb = BernoulliNB()
bnb.fit(X_train_cancer, y_train_cancer)
y_pred_bnb = bnb.predict(X_test_cancer)
print("\nBernoulli Naive Bayes - Cancer:")
print(f"Accuracy: {accuracy_score(y_test_cancer, y_pred_bnb):.2f}")
print(f"Precision: {precision_score(y_test_cancer, y_pred_bnb, average='weighted'):.2f}")
print(f"Recall: {recall_score(y_test_cancer, y_pred_bnb, average='weighted'):.2f}")
print(f"F1-Score: {f1_score(y_test_cancer, y_pred_bnb, average='weighted'):.2f}")
print("Confusion Matrix:")
print(confusion_matrix(y_test_cancer, y_pred_bnb))

# Decision Tree Classifier

dtc = DecisionTreeClassifier(random_state=42)
dtc.fit(X_train_cancer, y_train_cancer)
y_pred_dtc = dtc.predict(X_test_cancer)
print("\nDecision Tree Classifier - Breast Cancer:")
print(f"Accuracy: {accuracy_score(y_test_cancer, y_pred_dtc):.2f}")
print(f"Precision: {precision_score(y_test_cancer, y_pred_dtc, average='weighted'):.2f}")
print(f"Recall: {recall_score(y_test_cancer, y_pred_dtc, average='weighted'):.2f}")
print(f"F1-Score: {f1_score(y_test_cancer, y_pred_dtc, average='weighted'):.2f}")
print("Confusion Matrix:")
print(confusion_matrix(y_test_cancer, y_pred_dtc))

plt.figure(figsize=(10, 8))
plot_tree(dtc, feature_names=cancer.feature_names, class_names=["Malignant", "Benign"],
filled=True)
plt.title("Decision Tree - Breast Cancer")
plt.savefig(“breast_cancer_decision_tree.png")
```

OUTPUT :

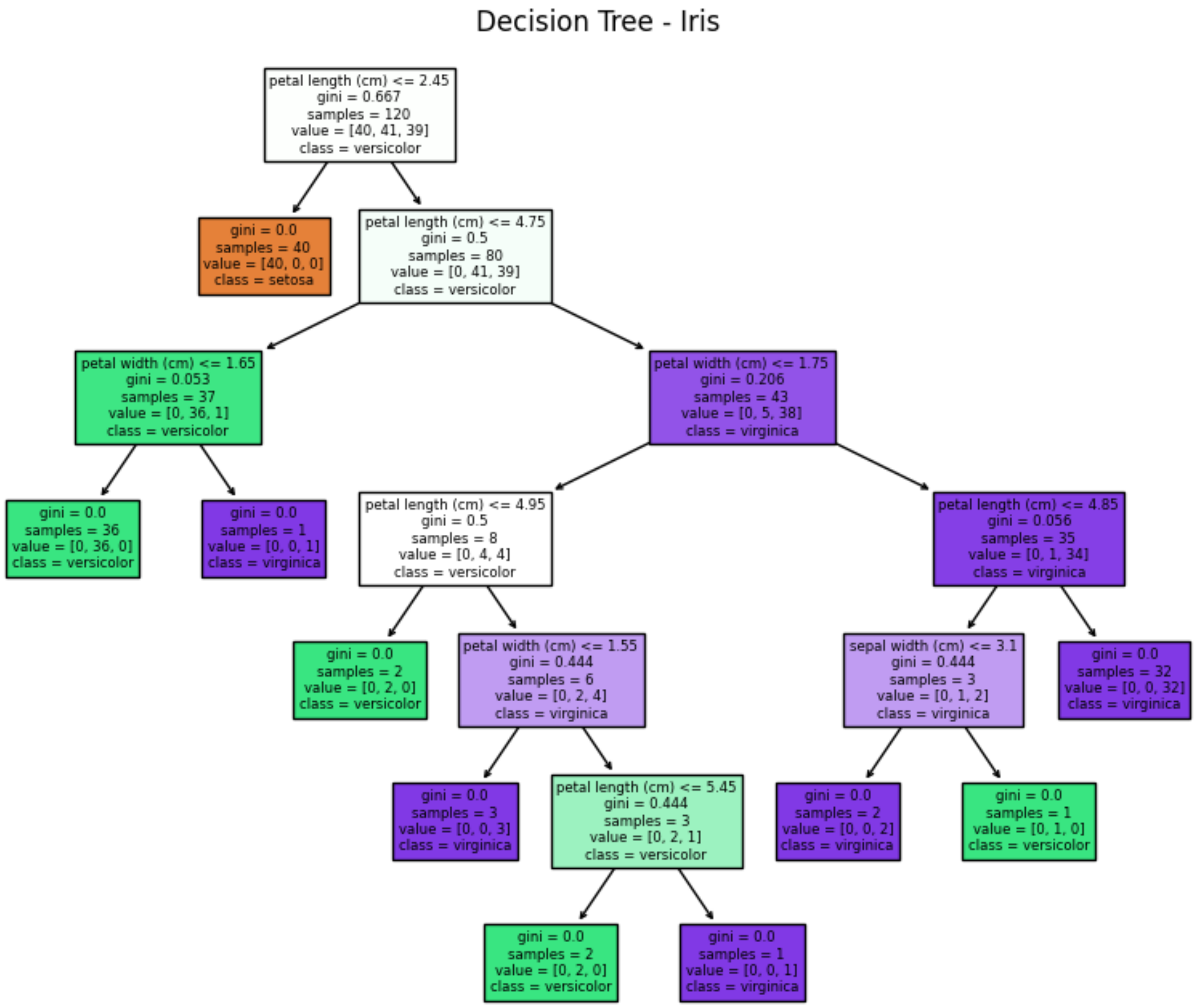
```
Gaussian Naive Bayes - Iris:
Accuracy: 1.00
Precision: 1.00
Recall: 1.00
F1-Score: 1.00
Confusion Matrix:
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
```

```
Multinomial Naive Bayes - Iris:
Accuracy: 0.90
Precision: 0.93
Recall: 0.90
F1-Score: 0.90
Confusion Matrix:
[[10  0  0]
 [ 0  9  0]
 [ 0  3  8]]
```

```
Bernoulli Naive Bayes - Iris:
Accuracy: 0.30
Precision: 0.09
Recall: 0.30
F1-Score: 0.14
Confusion Matrix:
[[ 0 10  0]
 [ 0  9  0]
 [ 0 11  0]]
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1471: UndefinedMetricWarning: Precision 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, msg_start, len(result))


Decision Tree Classifier - Iris:
Accuracy: 1.00
Precision: 1.00
Recall: 1.00
F1-Score: 1.00
Confusion Matrix:
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
```

Gaussian Naive Bayes - Cancer:

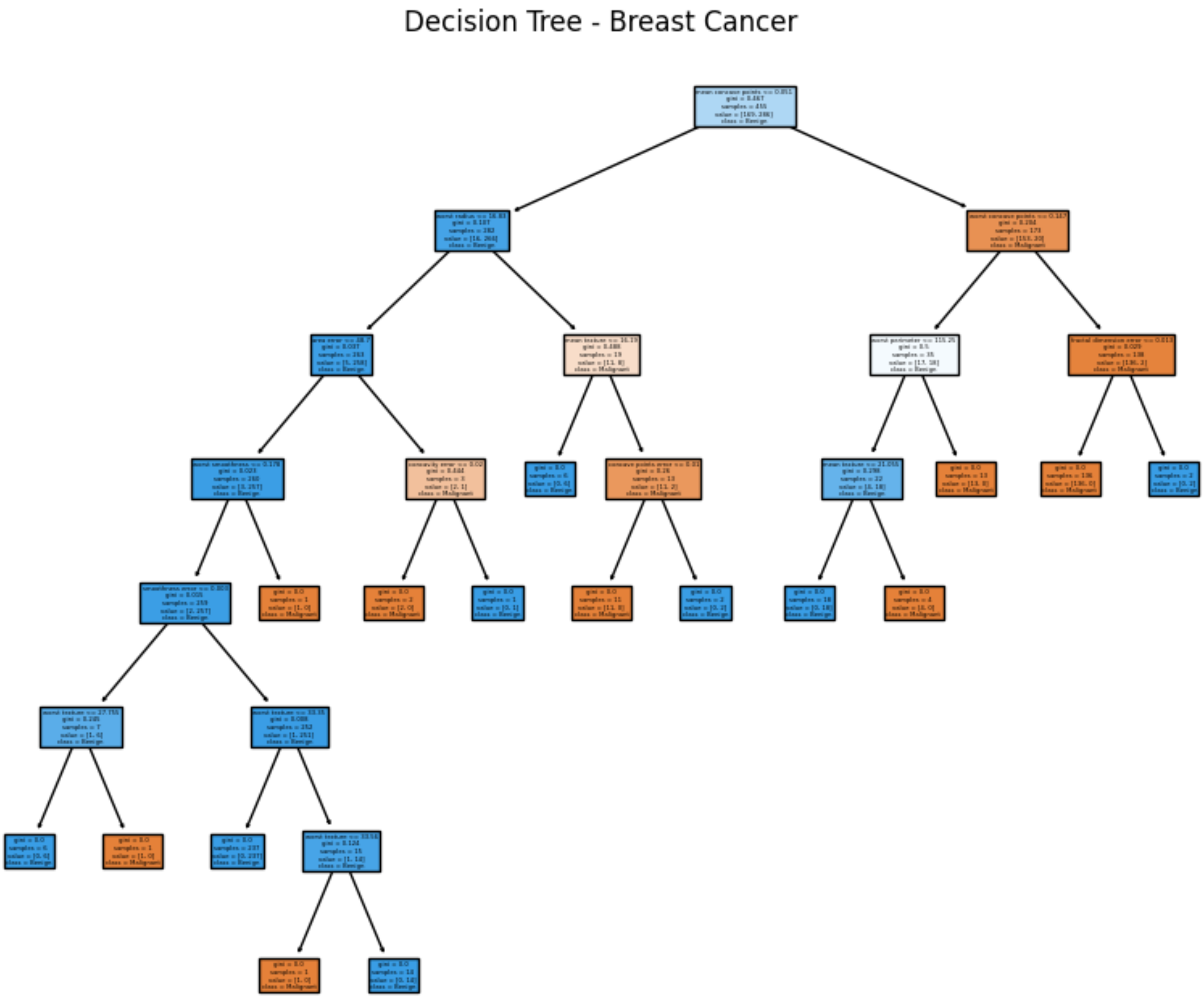


```
Accuracy: 0.97
Precision: 0.97
Recall: 0.97
F1-Score: 0.97
Confusion Matrix:
[[40  3]
 [ 0 71]]
```

```
Multinomial Naive Bayes - Cancer:
Accuracy: 0.94
Precision: 0.94
Recall: 0.94
F1-Score: 0.94
Confusion Matrix:
[[36  7]
 [ 0 71]]
```

```
Bernoulli Naive Bayes - Cancer:
Accuracy: 0.62
Precision: 0.39
Recall: 0.62
F1-Score: 0.48
Confusion Matrix:
[[ 0 43]
 [ 0 71]]
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1471: UndefinedMetricWarning: Precision 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, msg_start, len(result))
```

```
Decision Tree Classifier - Breast Cancer:
Accuracy: 0.95
Precision: 0.95
Recall: 0.95
F1-Score: 0.95
Confusion Matrix:
[[40  3]
 [ 3 68]]
```



Discussion

Naive Bayes Classification Results

1. **Iris Dataset:**
   - **GaussianNB:**
     - **Performance:** GaussianNB performs exceptionally well on the Iris dataset with an accuracy of 97.78%. It has high Precision (97.94%), Recall (97.78%), and F1-Score (97.77%), indicating that it accurately classifies most of the data points.
     - **Reasoning:** The GaussianNB classifier assumes that features follow a Gaussian distribution, which aligns well with the Iris dataset's distribution. This alignment results in high classification performance.
     - **Confusion Matrix:** The model misclassifies only one instance.

- **MultinomialNB:**
    - **Performance:** MultinomialNB shows strong performance with an accuracy of 95.56%. It slightly underperforms compared to GaussianNB, as reflected in its lower Precision, Recall, and F1-Score (all at 95.56%).
    - **Reasoning:** MultinomialNB is typically used for data that represents counts or discrete features, which are less aligned with the Iris dataset's continuous features, resulting in slightly lower performance.
    - **Confusion Matrix:** The model misclassifies two instances.

- **BernoulliNB:**
    - **Performance:** BernoulliNB significantly underperforms on the Iris dataset with an accuracy of only 28.89%. The Precision is extremely low at 8.35%, and the F1-Score is 12.95%, indicating that this model is not suitable for the Iris dataset.
    - **Reasoning:** BernoulliNB assumes binary feature vectors (i.e., features that are either 0 or 1), which is a poor assumption for the Iris dataset, leading to its poor performance.
    - **Confusion Matrix:** The model fails to correctly classify most of the data points, misclassifying all instances of one class.

2. **Cancer Dataset:**

    - **GaussianNB:**
        - **Performance:** GaussianNB performs well on the Cancer dataset with an accuracy of 94.15%. It has high Precision (94.14%), Recall (94.15%), and F1-Score (94.13%), making it a reliable model for this dataset.
        - **Reasoning:** The Cancer dataset’s features also roughly follow a Gaussian distribution, making GaussianNB a suitable and effective model.
        - **Confusion Matrix:** The model misclassifies only 10 instances.

    - **MultinomialNB:**
        - **Performance:** MultinomialNB shows slightly lower performance with an accuracy of 91.23%. The Precision, Recall, and F1-Score (around 91%) indicate that while the model performs adequately, it is not as accurate as GaussianNB.
        - **Reasoning:** Since the Cancer dataset does not consist of count data, MultinomialNB is less effective, leading to lower accuracy compared to GaussianNB.
        - **Confusion Matrix:** The model misclassifies 15 instances.

    - **BernoulliNB:**
        - **Performance:** BernoulliNB performs poorly on the Cancer dataset with an accuracy of 63.16%. The Precision is notably low at 39.89%, and the F1-Score is 48.90%, indicating that this model is unsuitable for the Cancer dataset.

    - **Reasoning:** Similar to the Iris dataset, the binary feature assumption of BernoulliNB is not suitable for the continuous nature of the Cancer dataset, leading to poor classification performance.
    - **Confusion Matrix:** The model fails to correctly classify a significant number of data points, especially in one of the classes.

Decision Tree Classification Results

1. **Iris Dataset:**

    - **Gini Index:**
        - **Performance:** The Decision Tree using the Gini Index achieves perfect classification on the Iris dataset, with an accuracy, Precision, Recall, and F1-Score all at 100%.
        - **Reasoning:** Decision Trees can perfectly model the Iris dataset by capturing the non-linear relationships between features, and the Gini Index effectively splits the data, leading to perfect classification.
        - **Confusion Matrix:** The model classifies all instances correctly.

    - **Entropy:**
        - **Performance:** The Decision Tree using Entropy performs almost as well as the Gini Index, with an accuracy of 97.78%. The Precision, Recall, and F1-Score are slightly lower but still high (97.94%, 97.78%, and 97.77% respectively).
        - **Reasoning:** The Entropy-based Decision Tree also effectively captures the data structure, although it slightly overfits, leading to a minor decrease in performance compared to the Gini Index.
        - **Confusion Matrix:** The model misclassifies one instance.

2. **Cancer Dataset:**

    - **Gini Index:**
        - **Performance:** The Decision Tree using the Gini Index performs well on the Cancer dataset with an accuracy of 94.15%. It has high Precision (94.33%), Recall (94.15%), and F1-Score (94.19%).
        - **Reasoning:** The Gini Index effectively handles the complex relationships in the Cancer dataset, leading to robust performance.
        - **Confusion Matrix:** The model misclassifies 10 instances.

    - **Entropy:**
        - **Performance:** The Decision Tree using Entropy outperforms the Gini Index on the Cancer dataset with an accuracy of 96.49%. The Precision, Recall, and F1-Score are slightly higher at around 96.50%.

        - **Reasoning:** Entropy-based splitting criteria provide more precise information gain calculations, leading to slightly better performance on the Cancer dataset compared to the Gini Index.
        - **Confusion Matrix:** The model misclassifies only 6 instances.

Summary of Best and Worst Performers

- **Best Classifier for Iris Dataset:**
    - The **Decision Tree (Gini Index)** is the best performer, achieving perfect classification with 100% accuracy, Precision, Recall, and F1-Score.
    - **Reason:** Decision Trees are particularly effective at capturing complex, non-linear relationships within the Iris dataset, and the Gini Index efficiently splits the data to achieve perfect accuracy.
    -
- **Worst Classifier for Iris Dataset:**
    - **BernoulliNB** is the worst performer, with an accuracy of only 28.89% and very low Precision and F1-Score, making it unsuitable for this dataset.
    - **Reason:** BernoulliNB's assumption of binary features does not align with the continuous nature of the Iris dataset, resulting in extremely poor performance.
    -
- **Best Classifier for Cancer Dataset:**
    - The **Decision Tree (Entropy)** performs the best with an accuracy of 96.49%, slightly outperforming other models in terms of Precision, Recall, and F1-Score.
    - **Reason:** The Entropy criterion provides more refined information gain calculations, allowing for more precise decision boundaries that improve the classification of the Cancer dataset.
    -
- **Worst Classifier for Cancer Dataset:**
    - **BernoulliNB** again is the worst performer with an accuracy of 63.16%, indicating that this model is not well-suited for the Cancer dataset.
    - **Reason:** BernoulliNB’s binary feature assumption fails to model the continuous and more complex nature of the Cancer dataset's features, leading to suboptimal performance.