Naive Bayes Classifier on Iris Dataset

Introduction

The purpose of this assignment is to implement a Naive Bayes classifier using Python on the Iris dataset. The Iris dataset contains 150 samples of flowers, with four features: sepal length, sepal width, petal length, and petal width. The target variable has three classes: *Setosa*, *Versicolor*, and *Virginica*. Gaussian Naive Bayes is applied, the dataset is split into training and testing sets, and the model performance is evaluated using accuracy, confusion matrix, and classification report. Visualizations are also included to understand the dataset and classifier results.

Code Implementation

Below is the Python code used for training and evaluating Naive Bayes on the Iris dataset:

```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, confusion_matrix,
    classification_report

iris = load_iris(as_frame=True)
X = iris.data
y = iris.target
df = iris.frame

print("Column Names:", list(X.columns))

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42, stratify=y
)

nb = GaussianNB()
nb.fit(X_train, y_train)

y_pred = nb.predict(X_test)
print("\nAccuracy:", accuracy_score(y_test, y_pred))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test,
    y_pred, target_names=iris.target_names))

sns.pairplot(df, hue="target", palette="Set2", diag_kind="kde")
plt.suptitle("Iris Dataset Pairplot by Species", y=1.02)
plt.show()

cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6,4))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
            xticklabels=iris.target_names,
            yticklabels=iris.target_names)
plt.title("Confusion Matrix Heatmap")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

feature_means = pd.DataFrame(nb.theta_, columns=X.columns, index=
    iris.target_names)
feature_vars = pd.DataFrame(nb.var_, columns=X.columns, index=
    iris.target_names)

feature_means.T.plot(kind="bar", figsize=(10,6))
plt.title("Naive Bayes - Mean Feature Values per Class")
plt.ylabel("Mean value")
plt.show()

feature_vars.T.plot(kind="bar", figsize=(10,6))
plt.title("Naive Bayes - Variance of Features per Class")
plt.ylabel("Variance")
plt.show()
```

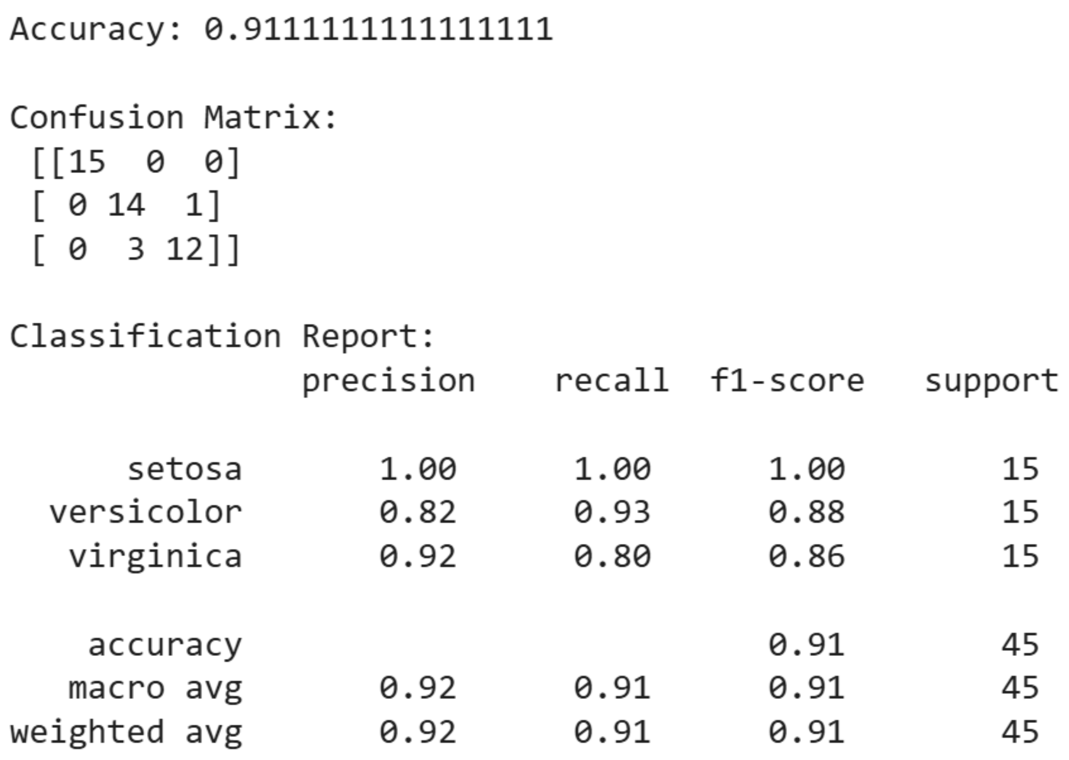
Outputs and Results

Terminal Outputs

- Accuracy score printed.
- Confusion matrix displayed.
- Classification report with precision, recall, and f1-score.

Conclusion

The Naive Bayes classifier achieved high accuracy on the Iris dataset, demonstrating the effectiveness of probabilistic classifiers on small structured datasets. Visualizations helped understand the dataset distribution and the classifier performance across different classes.



```
Accuracy: 0.9111111111111111

Confusion Matrix:
 [[15  0  0]
 [ 0 14  1]
 [ 0  3 12]]

Classification Report:
               precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        15
  versicolor       0.82      0.93      0.88        15
   virginica       0.92      0.80      0.86        15

    accuracy                           0.91        45
   macro avg       0.92      0.91      0.91        45
weighted avg       0.92      0.91      0.91        45
```

Figure 1: Terminal output showing accuracy and classification report.

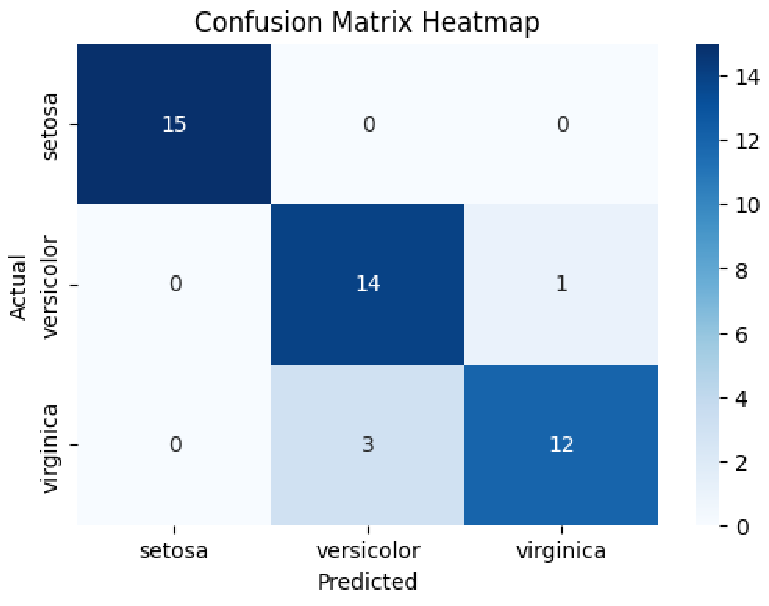


Figure 2: Confusion matrix heatmap.

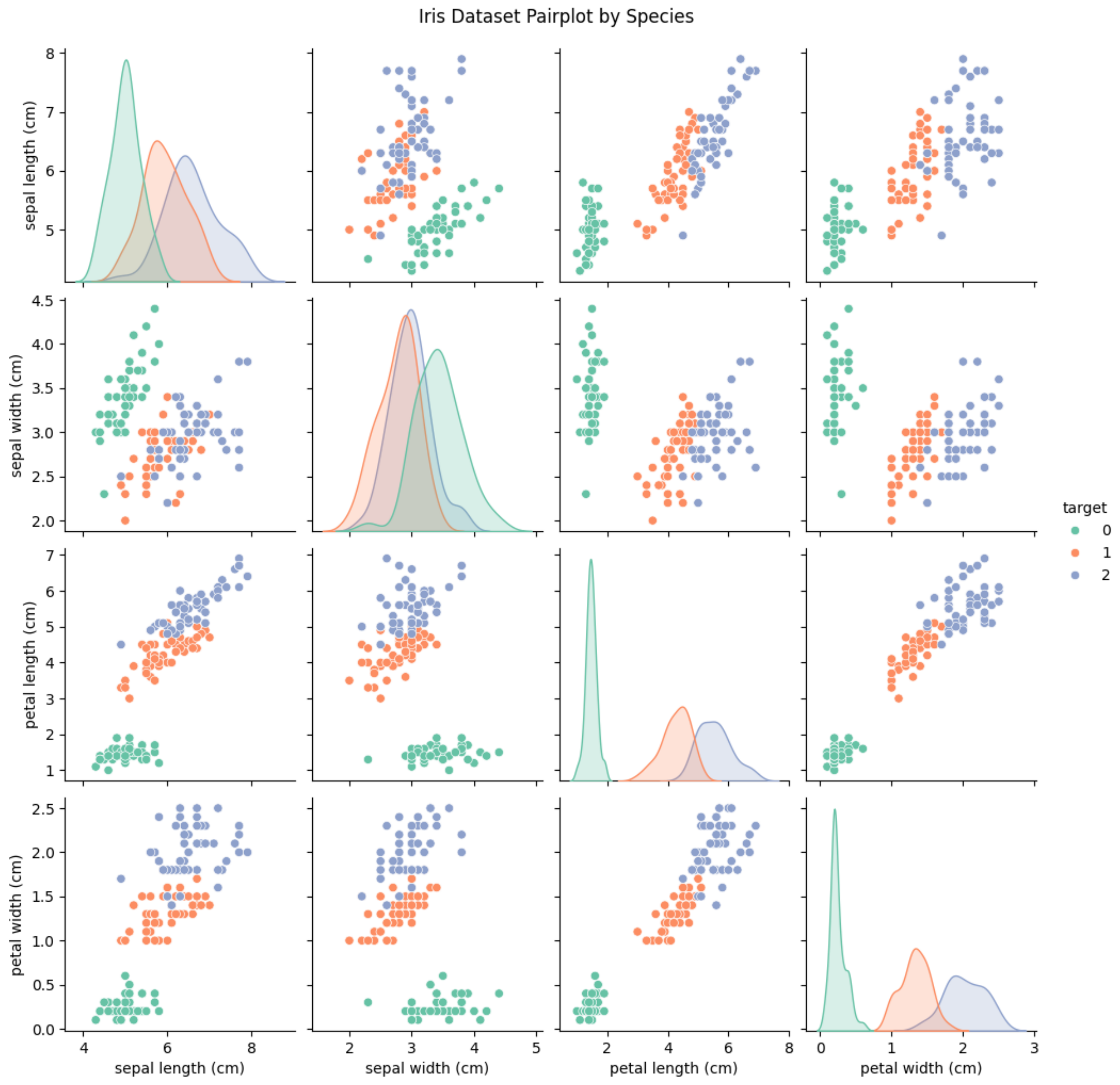


Figure 3: Pairplot of Iris dataset colored by species.

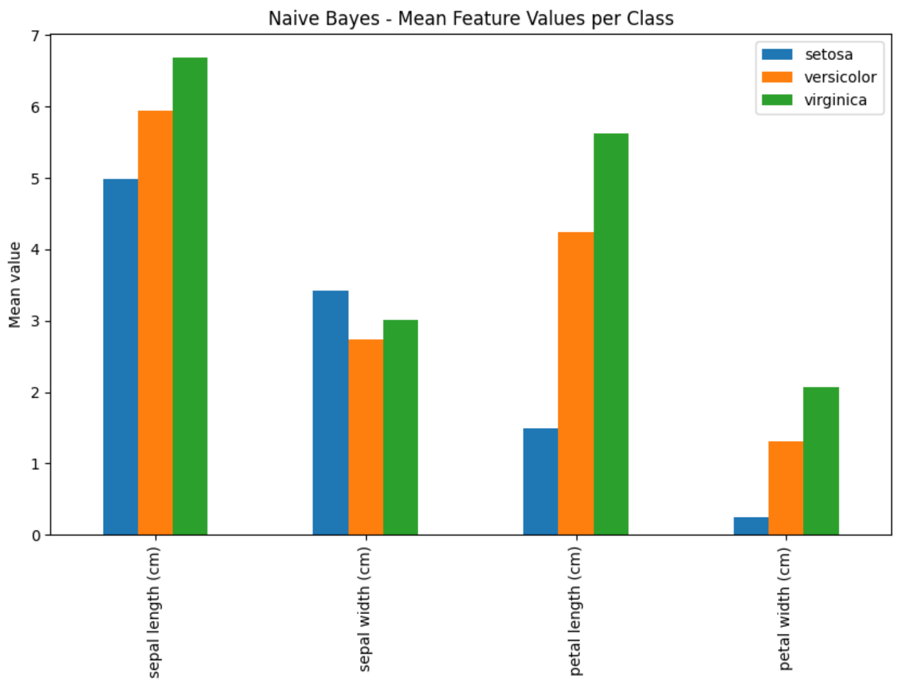


Figure 4: Mean feature values per class.

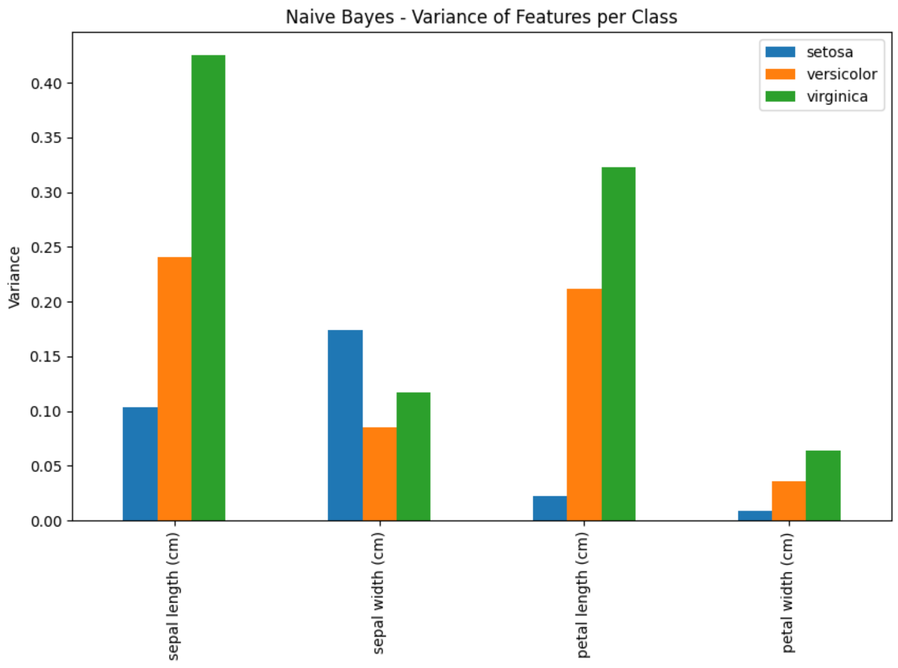


Figure 5: Feature variance per class.