Name: Kshitij Jadhav

Class: TY-3   Batch-B

RollNo.: 24

Implementation of Classification Algorithm (Decision Tree / Naïve Bayes) using Python

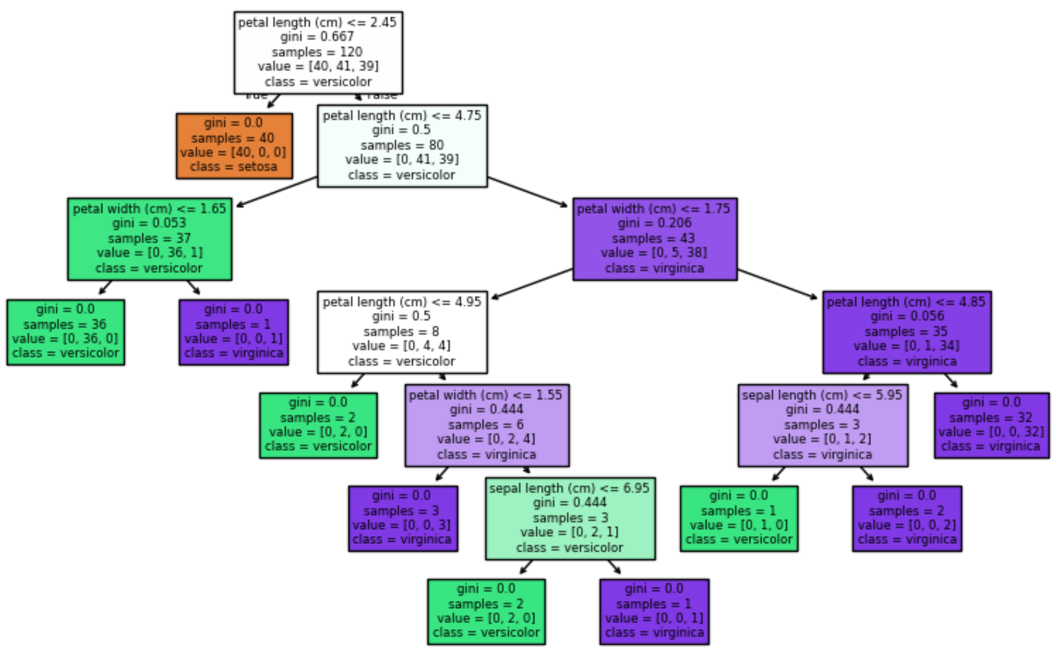
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
from sklearn.datasets import load_iris

data = load_iris()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = data.target

print("Dataset Preview:")
print(df.head())

X = df.iloc[:, :-1]
y = df.iloc[:, -1]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

dt_model = DecisionTreeClassifier()
dt_model.fit(X_train, y_train)

dt_predictions = dt_model.predict(X_test)

nb_model = GaussianNB()
nb_model.fit(X_train, y_train)
```

```
nb_predictions = nb_model.predict(X_test)

def evaluate_model(model_name, y_test, predictions):
    print(f"\n{model_name} Performance:")
    print("Accuracy:", accuracy_score(y_test, predictions))
    print("Classification Report:\n", classification_report(y_test,
predictions))
    print("Confusion Matrix:\n", confusion_matrix(y_test, predictions))

evaluate_model("Decision Tree", y_test, dt_predictions)
evaluate_model("Naïve Bayes", y_test, nb_predictions)

plt.figure(figsize=(10, 6))
plot_tree(dt_model, feature_names=data.feature_names,
class_names=data.target_names, filled=True)
plt.title("Decision Tree Visualization")
plt.show()
```

```
Dataset Preview:
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0                5.1               3.5                1.4               0.2
1                4.9               3.0                1.4               0.2
2                4.7               3.2                1.3               0.2
3                4.6               3.1                1.5               0.2
4                5.0               3.6                1.4               0.2

   target
0       0
1       0
2       0
3       0
4       0

Decision Tree Performance:
Accuracy: 1.0
Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      1.00      1.00         9
           2       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30

Confusion Matrix:
```

```
 [[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]

Naïve Bayes Performance:
Accuracy: 1.0
Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      1.00      1.00         9
           2       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30

Confusion Matrix:
 [[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
```

Decision Tree Visualization



Q.1) What is a Decision Tree classifier, and how does it work?

-> A **Decision Tree classifier** is a supervised learning algorithm used for classification and regression tasks. It works by recursively splitting the dataset into subsets based on feature values, forming a tree-like structure. Each internal node represents a decision based on a feature, branches indicate possible outcomes, and leaf nodes represent class labels. The algorithm selects the best feature split using criteria like **Gini Impurity** or **Entropy (Information Gain)**. It continues splitting until a stopping condition is met, such as maximum depth or minimum samples per leaf. Decision Trees are easy to interpret but prone to overfitting without pruning techniques.

Q.2) Explain the Naïve Bayes algorithm and its underlying assumptions.

-> The **Naïve Bayes algorithm** is a probabilistic classifier based on **Bayes' Theorem**, assuming feature independence. It calculates the probability of a class given input features and selects the highest probability class. The key assumption is that all features contribute independently to the outcome (**naïve assumption**). Common variants include **Gaussian Naïve Bayes** (normal distribution), **Multinomial Naïve Bayes** (text classification), and **Bernoulli Naïve Bayes** (binary features).

Q.3) Compare the working principles of Decision Tree and Naïve Bayes classifiers.

Feature	Decision Tree	Naïve Bayes
Type	Non-parametric, rule-based	Probabilistic, statistical
Working Principle	Splits data recursively based on feature values to create a tree structure	Uses **Bayes' Theorem** to compute class probabilities assuming feature independence
Assumptions	No strict assumptions; learns patterns from data	Assumes feature independence (naïve assumption)
Interpretability	Highly interpretable (visual tree structure)	Less interpretable (probabilistic model)
Handling of Correlated Features	Handles correlated features well	Struggles with correlated features due to independence assumption
Performance on Small Data	Works well but prone to overfitting	Performs well, especially on small datasets
Speed	Slower for deep trees	Fast computation due to probability calculations
Best Used For	Decision-making processes, structured data	Text classification, spam filtering, sentiment analysis

Q.4) What are the different types of Decision Tree splitting criteria?

1. Gini Impurity
2. Entropy (Information Gain)
3. Chi-Square
4. Redution in Variance
5. Gain Ratio