Experiment No.5

Title:Study of Confusion Matrix with appropriate example

Study of Confusion Matrix

Introduction

In classification problems, it's crucial to evaluate model performance accurately. The **confusion matrix** is one of the most insightful tools for this purpose. It provides a detailed breakdown of correct and incorrect predictions across classes, highlighting the types of errors the model is making, beyond a simple accuracy metric.

Confusion Matrix Definition

A confusion matrix is a table that is often used to describe the performance of a classification model. Each row of the matrix represents the instances in an actual class, while each column represents the instances in a predicted class. For binary classification, it is structured as:

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

- **True Positive (TP)**: Cases where the model correctly predicted the positive class.
- **True Negative (TN)**: Cases where the model correctly predicted the negative class.
- **False Positive (FP)**: Cases where the model incorrectly predicted the positive class (Type I error).
- **False Negative (FN)**: Cases where the model incorrectly predicted the negative class (Type II error).

Example

Suppose we have a binary classifier predicting whether a patient has a disease (positive class) or not (negative class). The confusion matrix for a test set of 100 patients might look like this:

	Predicted Disease	Predicted No Disease
Actual Disease	45	5
Actual No Disease	10	40

From the above:

- **True Positives (TP) = 45**: Correctly predicted patients with the disease.
- **False Negatives (FN) = 5**: Missed predictions (patients with the disease predicted as no disease).
- **False Positives (FP) = 10**: Incorrectly predicted as having the disease.

- **True Negatives (TN) = 40**: Correctly predicted no disease.

Performance Metrics Derived from the Confusion Matrix

Several performance metrics can be derived to evaluate model performance:

1. **Accuracy**: Proportion of correctly predicted instances (both positive and negative) over the total instances.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

2. **Precision**: Fraction of correctly predicted positive instances among all predicted positives.

$$\text{Precision} = \frac{TP}{TP + FP}$$

3. **Recall (Sensitivity)**: Fraction of correctly predicted positive instances among all actual positives.

$$\text{Recall} = \frac{TP}{TP + FN}$$

4. **F1-Score**: Harmonic mean of precision and recall, giving a balanced measure of performance.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

5. **Specificity**: Proportion of correctly predicted negative instances among all actual negatives.

$$\text{Specificity} = \frac{TN}{TN + FP}$$

Use Case Scenarios

The confusion matrix is particularly useful in scenarios where:

- **Class imbalance** exists, and accuracy alone would be misleading (e.g., fraud detection, medical diagnoses).
- **Error analysis** is critical. By inspecting false positives and false negatives, one can fine-tune the model to optimize for the most critical metric (e.g., improving recall in cancer detection).

Import Libraries

```
# Import necessary libraries
import numpy as np
import pandas as pd
```

```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report,
confusion_matrix,ConfusionMatrixDisplay
import matplotlib.pyplot as plt
import seaborn as sns
```

Load the Iris dataset

```
iris = sns.load_dataset('iris')
iris.head()
```

```
   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
```

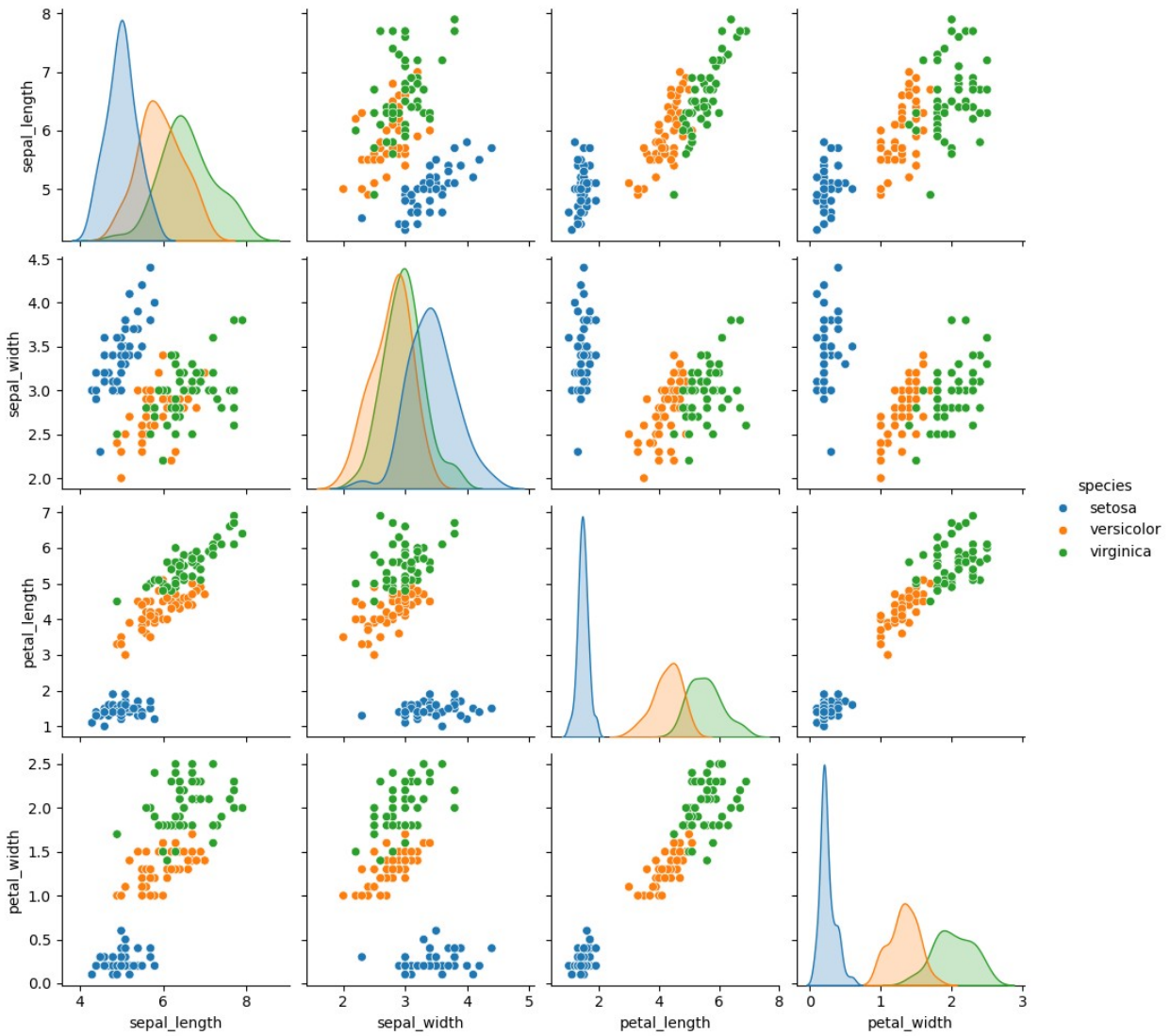
```
iris.info()
```

```
<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
```

Visualize the data

```
sns.pairplot(data=iris,hue='species')
```

```
<seaborn.axisgrid.PairGrid at 0x7dd278db8fb0>
```



Separating Feature and Target

```
X=iris[['sepal_width','petal_width']]
Y=iris['species']
```

Split data into train and test set

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.2, random_state=42)
```

Create SVM Model

```
svm_model = SVC(kernel='linear', random_state=42)  # You can use other
kernels like 'rbf', 'poly'
svm_model.fit(X_train, Y_train)

SVC(kernel='linear', random_state=42)
```

Make predictions on the test data

```
Y_pred = svm_model.predict(X_test)
Y_pred
```

```
array(['versicolor', 'setosa', 'virginica', 'versicolor',
'versicolor',
       'setosa', 'versicolor', 'virginica', 'versicolor',
'versicolor',
       'virginica', 'setosa', 'setosa', 'setosa', 'setosa',
'versicolor',
       'virginica', 'versicolor', 'versicolor', 'virginica', 'setosa',
       'virginica', 'setosa', 'virginica', 'virginica', 'virginica',
       'virginica', 'virginica', 'setosa', 'setosa'], dtype=object)
```

Evaluate the model using a confusion matrix and classification report

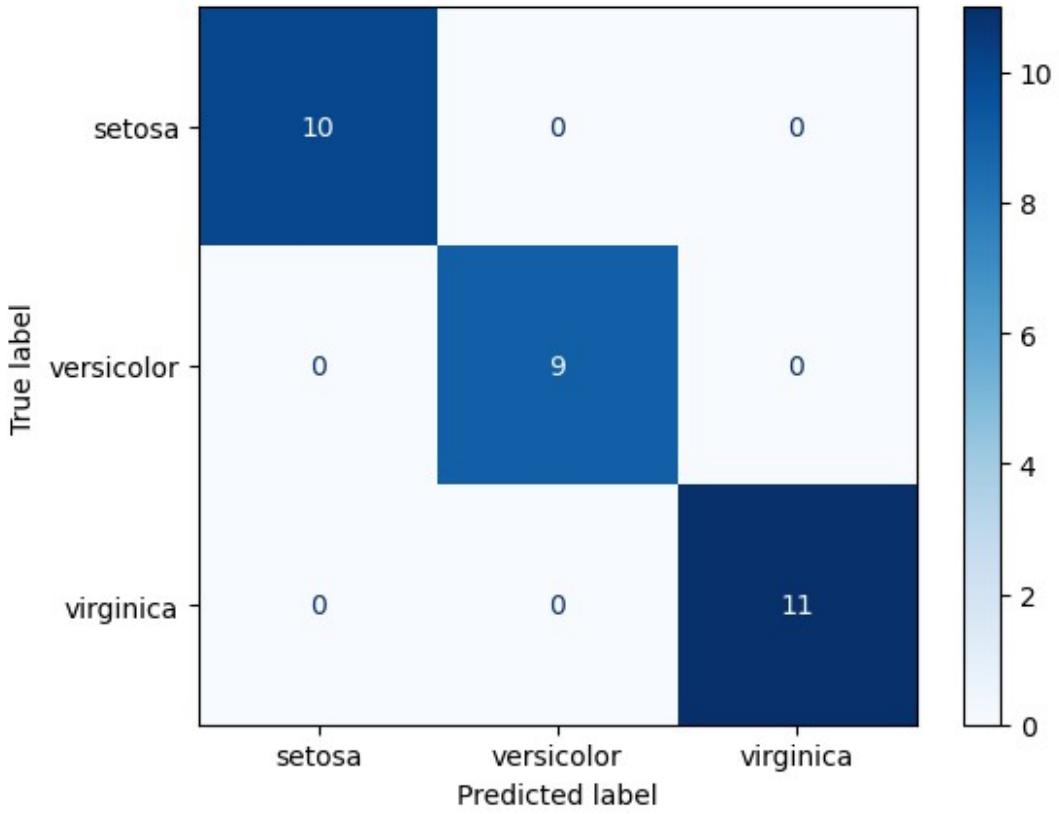
```
labels = iris['species'].unique()  # The unique class labels for the
target
cm = confusion_matrix(Y_test, Y_pred)
print("Confusion Matrix:\n", cm)
disp = ConfusionMatrixDisplay(confusion_matrix=cm,
display_labels=labels)
disp.plot(cmap='Blues')
print("\nClassification Report:\n", classification_report(Y_test,
Y_pred))
```

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

Classification Report:
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        10
  versicolor       1.00      1.00      1.00         9
   virginica       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
```



Conclusion:

The **study of the confusion matrix** provides a deeper understanding of a classification model's performance, beyond what a single metric like accuracy can offer. By examining the true positives, true negatives, false positives, and false negatives, the confusion matrix allows us to assess not only how well the model predicts each class, but also the types of errors it makes. This is particularly important when dealing with imbalanced datasets or when the cost of false positives and false negatives differs.

For instance, in the **Iris dataset** example, using a Support Vector Machine (SVM) model, we could observe the confusion matrix detailing how the model classifies each species of iris flower (setosa, versicolor, and virginica). The matrix helps us understand where the model makes errors, such as misclassifying virginica as versicolor or vice versa. Using metrics derived from the confusion matrix, like **precision**, **recall**, and **F1-score**, we can evaluate how the model balances between false positives and false negatives across each class.

Through this experiment, we learned:

- **Accuracy** alone may not be sufficient to evaluate model performance, especially in cases where class imbalance or uneven error costs exist.
- The confusion matrix provides insights into specific types of errors (false positives and false negatives), which can guide model tuning and help improve model performance for critical applications, such as medical diagnoses or fraud detection.

- By visualizing the confusion matrix, it becomes easier to interpret the performance of the classifier and take steps to mitigate the model's weaknesses, like increasing recall for one class or improving precision for another.

Overall, the confusion matrix serves as a vital tool in understanding and improving classification models by offering a granular view of predictions and errors.