Assignment No 4

Aim

To develop a machine learning model using R/Python to predict whether a customer will respond to a special offer in a cosmetics shop, and to evaluate the model using a confusion matrix with metrics like Accuracy, Precision, Recall, and F1-score.

Introduction to the Problem

In the field of machine learning, evaluating the performance of classification models is essential to ensure the quality and reliability of predictions. Classification algorithms are commonly used to categorize data into classes based on input features. However, after model training, it becomes crucial to assess **how well the model is performing**. This is where **performance evaluation metrics** such as **Accuracy**, **Precision**, **Recall**, and **F1 Score** come into play.

These metrics are derived from the **confusion matrix**, a table that describes the performance of a classification model on a set of test data for which the true values are known. The confusion matrix allows us to calculate how many predictions were correct and how many were incorrect, and in what way.

Dataset Overview (Assumed Features)

We use a customer dataset containing both demographic and behavioral features. Example attributes include:

- Customer ID
- Age
- Gender
- Annual Income (in ₹)
- Spending Score (a measure of how frequently and how much the customer shops)
- Time Spent in Shop (minutes)
- Previous Purchase History (Yes/No)
- Response to Offer (Target variable: 1 = Responded, 0 = Not Responded)

ML Workflow Steps (with Explanations)

a) Data Collection

A structured dataset is collected from the shop's CRM or marketing system. It includes customer profiles and past responses to offers.

b) Data Preprocessing

Handling Missing Values: Fill or remove incomplete rows.
Encoding: Convert categorical variables (e.g., Gender, History) into numeric.
Feature Scaling: Normalize income, age, etc., for balanced algorithm performance.

c) Data Splitting

Split dataset into:
- Training set (e.g., 70%): To build the model.
- Test set (e.g., 30%): To evaluate model performance.

d) Model Selection

Use a classification algorithm like:
- Logistic Regression – Simple and interpretable.
- Random Forest – Handles complex data with high accuracy.
- K-Nearest Neighbors (KNN) – Distance-based, good for small datasets.
- Support Vector Machine (SVM) – For well-separated classes.

e) Model Training and Prediction

Train the model on training data and test it using the test set to predict whether a customer will respond or not.

Confusion Matrix (Core Concept)

Confusion Matrix:

	Predicted: Yes	Predicted: No
Actual: Yes (Responded)	TP (True Positive)	FN (False Negative)
Actual: No (Not Responded)	FP (False Positive)	TN (True Negative)

- TP: Correctly predicted responders.
- TN: Correctly predicted non-responders.
- FP: Predicted responder, but didn't respond.
- FN: Predicted no response, but customer did respond.

Evaluation Metrics with Formulas and Interpretation

a) Accuracy

Meaning: Overall how many predictions were correct.
Formula: Accuracy = (TP + TN) / (TP + TN + FP + FN)

b) Precision

Meaning: Out of all customers predicted to respond, how many actually responded.
Formula: Precision = TP / (TP + FP)

c) Recall

Meaning: Out of all actual responders, how many we correctly predicted.
Formula: Recall = TP / (TP + FN)

d) F1-Score

Meaning: Balances Precision and Recall (useful when data is imbalanced).
Formula: F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

Real-World Applications of This Model

- Marketing Campaign Optimization: Focus only on customers likely to respond, saving costs.
- Customer Segmentation: Group customers by response probability.
- Upselling and Cross-Selling: Offer relevant products to interested customers.
- Personalized Advertising: Improve engagement through targeting.

Tools Used

Python Libraries:

- pandas for data manipulation
- scikit-learn for model building and metrics
- matplotlib/seaborn for visualization

Outputs:

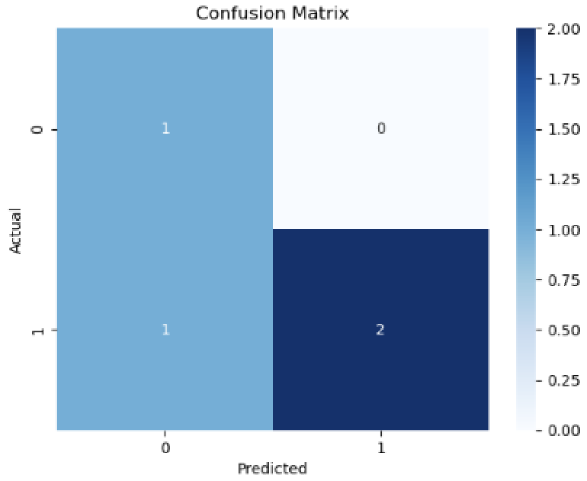
In [32]:

```
# Apply appropriate ML algorithm on a dataset collected in a cosmetics shop showing details of
# customers to predict customer response for special offer.
# Create confusion matrix based on above data and find
# a) Accuracy
# b) Precision
# c) Recall
# d) F-1 score
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score, f1_score
```

In [33]:

```
y_test=[1,2,2,2]
y_pred=[1,1,2,2]
con_matrix=confusion_matrix(y_test,y_pred)
print("Confusion Matrix:")
print(con_matrix)


sns.heatmap(con_matrix, annot=True, cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')

# Print results
print(f"Accuracy: {accuracy:.2f}")
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")
print(f"F1-Score: {f1:.2f}")
```

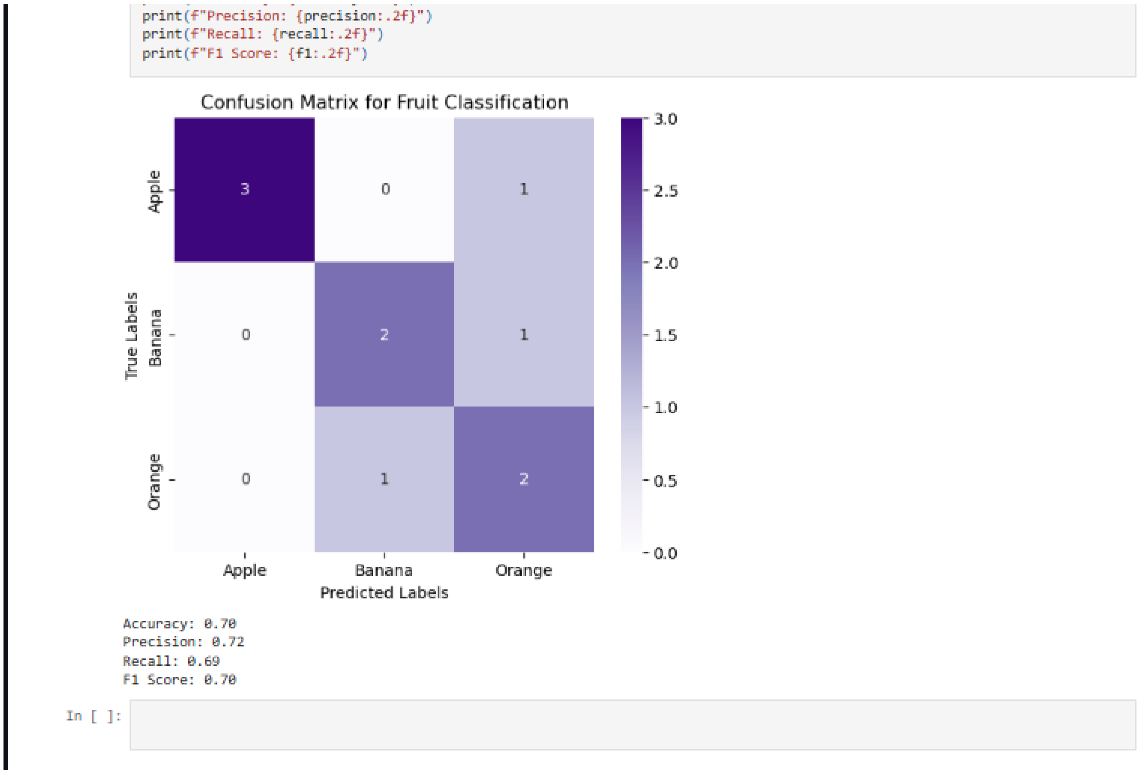
```
Confusion Matrix:
[[1 0]
 [1 2]]
```



```
Accuracy: 0.75
Precision: 0.88
Recall: 0.75
F1-Score: 0.77
```

In [34]:

```
y_true = ['Apple', 'Banana', 'Orange', 'Apple', 'Orange', 'Banana', 'Apple', 'Banana', 'Orange', 'Apple']

y_pred = ['Apple', 'Orange', 'Orange', 'Apple', 'Banana', 'Banana', 'Orange', 'Banana', 'Orange', 'Apple']

labels = ['Apple', 'Banana', 'Orange']
cm = confusion_matrix(y_true, y_pred, labels=labels)

plt.figure(figsize=(6,5))
sns.heatmap(cm, annot=True, fmt='d', cmap='Purples', xticklabels=labels, yticklabels=labels)
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix for Fruit Classification")
plt.show()

accuracy = accuracy_score(y_true, y_pred)
precision = precision_score(y_true, y_pred, average='macro')
recall = recall_score(y_true, y_pred, average='macro')
f1 = f1_score(y_true, y_pred, average='macro')

print(f"Accuracy: {accuracy:.2f}")
```



```
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")
print(f"F1 Score: {f1:.2f}")
```

```
Accuracy: 0.70
Precision: 0.72
Recall: 0.69
F1 Score: 0.70
```

```
In [ ]:
```

Conclusion

We have successfully implemented a machine learning classification model on a customer dataset from a cosmetics shop to predict responses to a special offer.