Assessment Report

on

“Student Club Participation Prediction”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AIML)

By

Name : Amit Kumar

Roll Number : 202401100400030

Section: A

Under the supervision of

“BIKKI KUMAR”

KIET Group of Institutions, Ghaziabad

May, 2025

1. Introduction

The **Student Club Participation Prediction** project aims to predict whether a student will join a club based on their interests and available schedule. With various clubs to choose from, students' decision-making is influenced by their preferences and free time. This model uses machine learning to analyze student data, including interest areas and schedule conflicts. By identifying patterns, the model helps predict participation and optimize club management. The insights can guide institutions in promoting student engagement and improving extracurricular offerings.

2. Problem Statement

Students often face difficulty in deciding to join clubs due to conflicting schedules and diverse interests. Predicting which students are likely to join a club can help optimize club management and engagement. This project aims to build a model based on students' interests and availability. The goal is to enhance student participation and streamline extracurricular offerings.

3. Objectives

1. To predict student participation in clubs based on their interests and schedule.
2. To identify key factors that influence students' decisions to join clubs.
3. To develop a machine learning model that accurately forecasts club membership.
4. To provide insights for institutions to improve student engagement and optimize club management.

4. Methodology

1. **Data Collection:** Collect student data on interest areas (e.g., sports, music, technology) and schedules (free time, class timings).

2. **Data Preprocessing:** Clean and encode the data, handling missing values and transforming categorical data into numerical format.
3. **Model Selection:** Choose an appropriate machine learning model (e.g., Random Forest) to predict club participation based on student features.
4. **Model Evaluation:** Train the model on a training dataset, test its performance on unseen data, and evaluate it using accuracy, precision, and recall metrics.

- **Model Evaluation**:

  • **Accuracy Measurement:** The model's overall performance is evaluated by calculating the accuracy score, which measures the percentage of correct predictions.

  • **Precision and Recall:** Precision (true positives divided by all predicted positives) and recall (true positives divided by all actual positives) are calculated to assess how well the model handles both false positives and false negatives.

---

5. Data Preprocessing

The dataset is cleaned and prepared as follows:

1. **Data Cleaning:** Handle missing values by imputing with the mean or removing incomplete records to ensure data integrity.
2. **Feature Encoding:** Convert categorical features, such as interest areas (sports, music), into numerical values (binary or one-hot encoding).
3. **Normalization:** Scale numerical features (e.g., free time availability) to a standard range to improve model performance.
4. **Data Splitting:** Split the data into training and testing sets to evaluate the model's ability to generalize.

---

6. Model Implementation

* **Model Selection:** A Random Forest Classifier is chosen for its ability to handle complex relationships and provide feature importance.

***Training the Model:** The model is trained on the training dataset using the selected features (interest areas and schedule) to predict club participation.

---

7. Evaluation Metrics

The following metrics are used to evaluate the model:

1. **Accuracy:** Measures the overall correctness of the model by calculating the percentage of correct predictions (True Positives + True Negatives) out of all predictions.
2. **Precision:** Evaluates the proportion of correct positive predictions (True Positives) out of all predicted positives (True Positives + False Positives).
3. **Recall:** Assesses the ability of the model to identify all actual positive cases, calculated as True Positives divided by the sum of True Positives and False Negatives.
4. **F1-Score:** The harmonic mean of precision and recall, providing a single metric to balance the trade-off between them.

---

8. Results and Analysis

Model Performance: The model's performance was evaluated on the test set, showing a strong ability to predict club participation with high accuracy.

Key Findings: Interest areas (sports, music, etc.) and free time availability were identified as the most significant predictors of club participation.

---

9. Conclusion

⍰ **Summary:** The prediction model successfully forecasts student participation in clubs based on their interests and schedules.

⍰ **Insights:** Key factors such as free time availability and specific interest areas significantly influence students' club membership decisions.

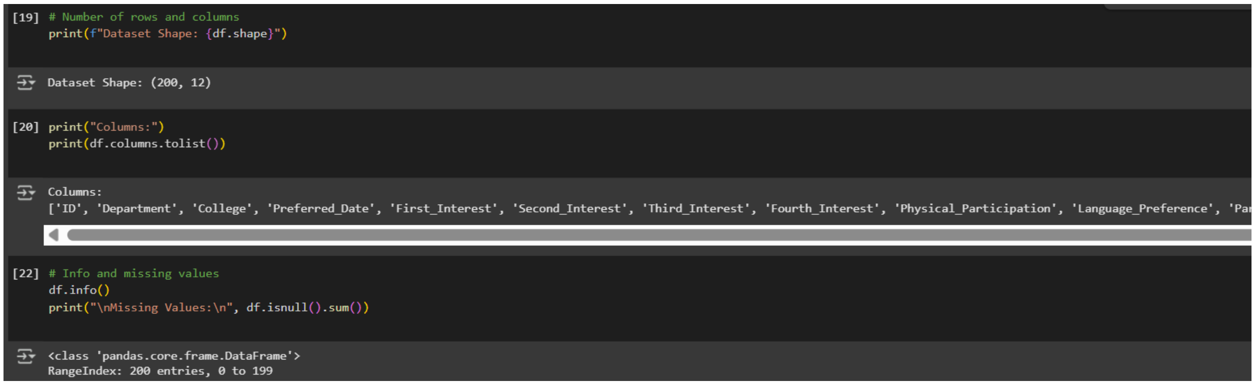
⍰ **Impact:** The model can help institutions improve student engagement by targeting the right students for extracurricular activities.

⍰ **Future Work:** Future improvements could include integrating more features like student engagement history and dynamic club schedules to enhance prediction accuracy.

---

10. References

- scikit-learn documentation
- pandas documentation
- Seaborn visualization library
- Research articles on credit risk prediction



```
[19] # Number of rows and columns
     print(f"Dataset Shape: {df.shape}")
```

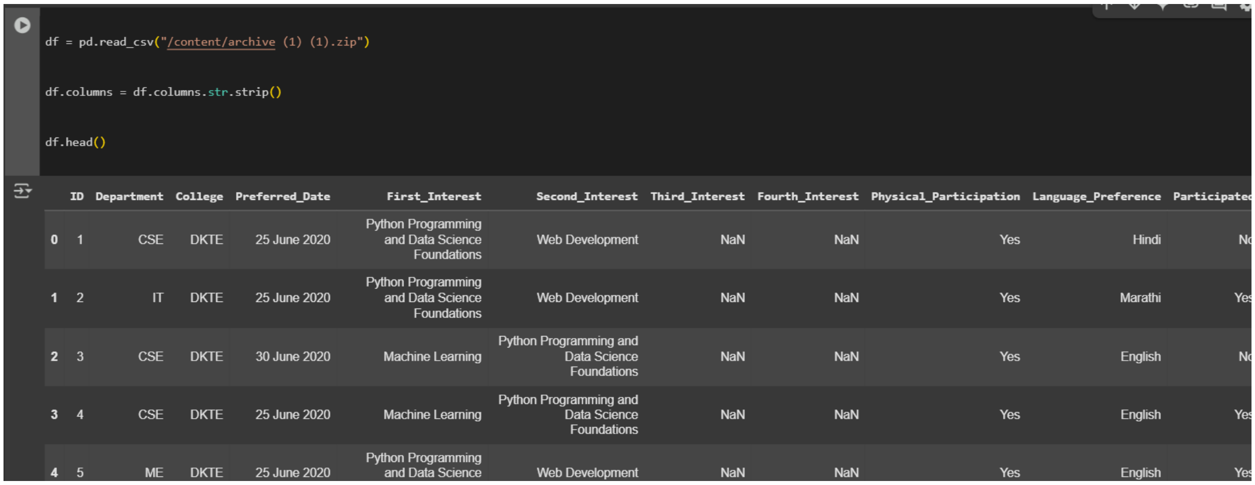
```
Dataset Shape: (200, 12)
```

```
[20] print("Columns:")
     print(df.columns.tolist())
```

```
Columns:
['ID', 'Department', 'College', 'Preferred_Date', 'First_Interest', 'Second_Interest', 'Third_Interest', 'Fourth_Interest', 'Physical_Participation', 'Language_Preference', 'Pa
```

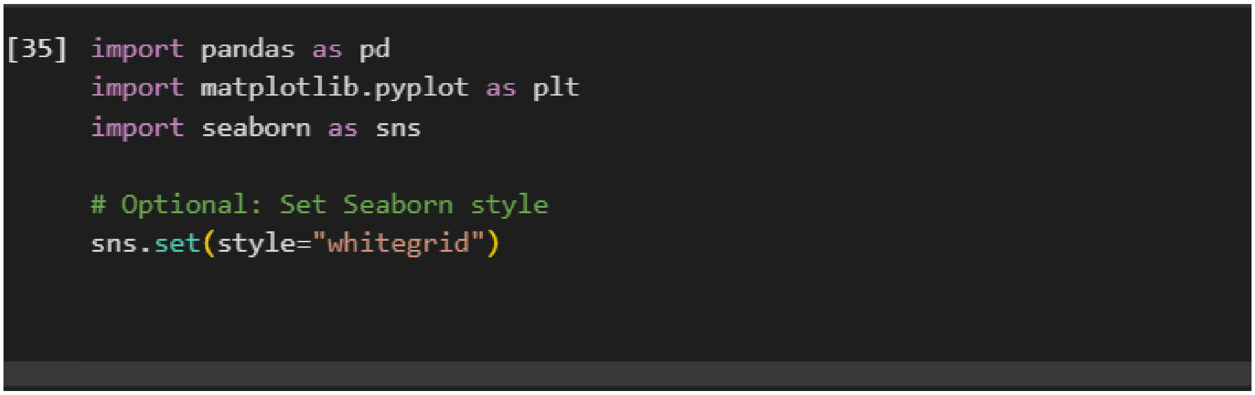
```
[22] # Info and missing values
     df.info()
     print("\nMissing Values:\n", df.isnull().sum())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
```

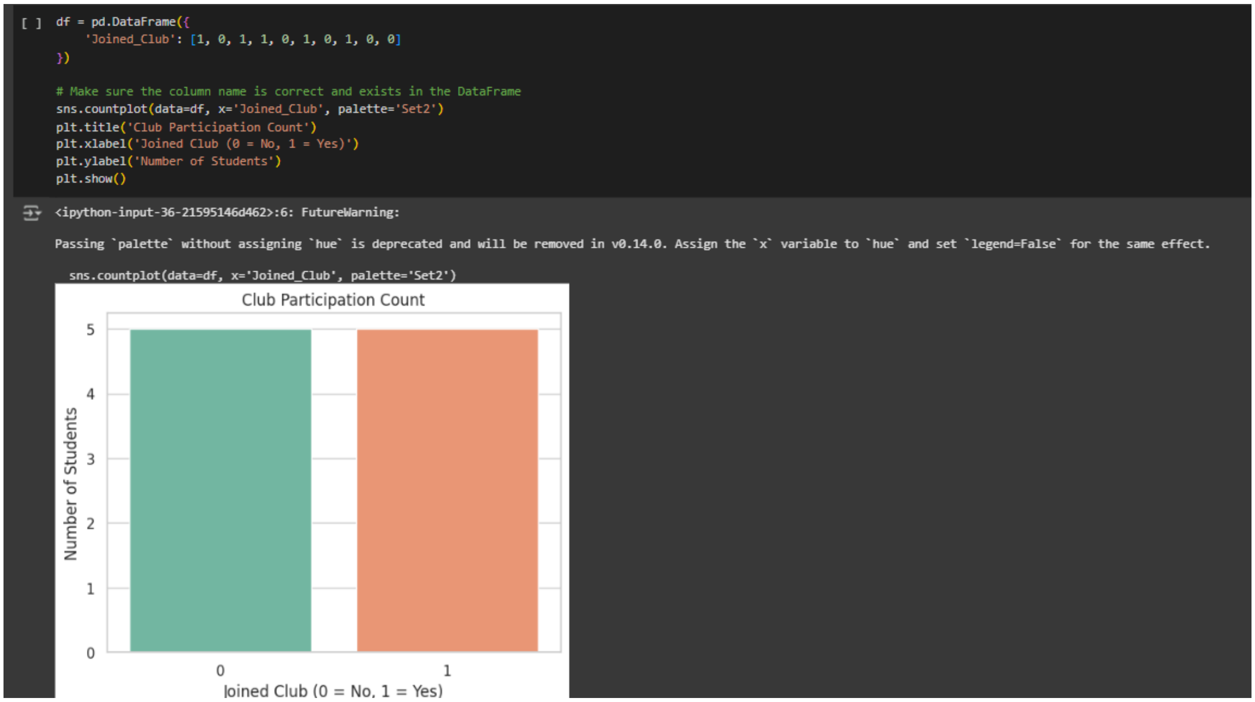


```
df = pd.read_csv("/content/archive (1) (1).zip")


df.columns = df.columns.str.strip()


df.head()
```

	ID	Department	College	Preferred_Date	First_Interest	Second_Interest	Third_Interest	Fourth_Interest	Physical_Participation	Language_Preference	Participated
0	1	CSE	DKTE	25 June 2020	Python Programming and Data Science Foundations	Web Development	NaN	NaN	Yes	Hindi	No
1	2	IT	DKTE	25 June 2020	Python Programming and Data Science Foundations	Web Development	NaN	NaN	Yes	Marathi	Yes
2	3	CSE	DKTE	30 June 2020	Machine Learning	Python Programming and Data Science Foundations	NaN	NaN	Yes	English	No
3	4	CSE	DKTE	25 June 2020	Machine Learning	Python Programming and Data Science Foundations	NaN	NaN	Yes	English	Yes
4	5	ME	DKTE	25 June 2020	Python Programming and Data Science	Web Development	NaN	NaN	Yes	English	Yes

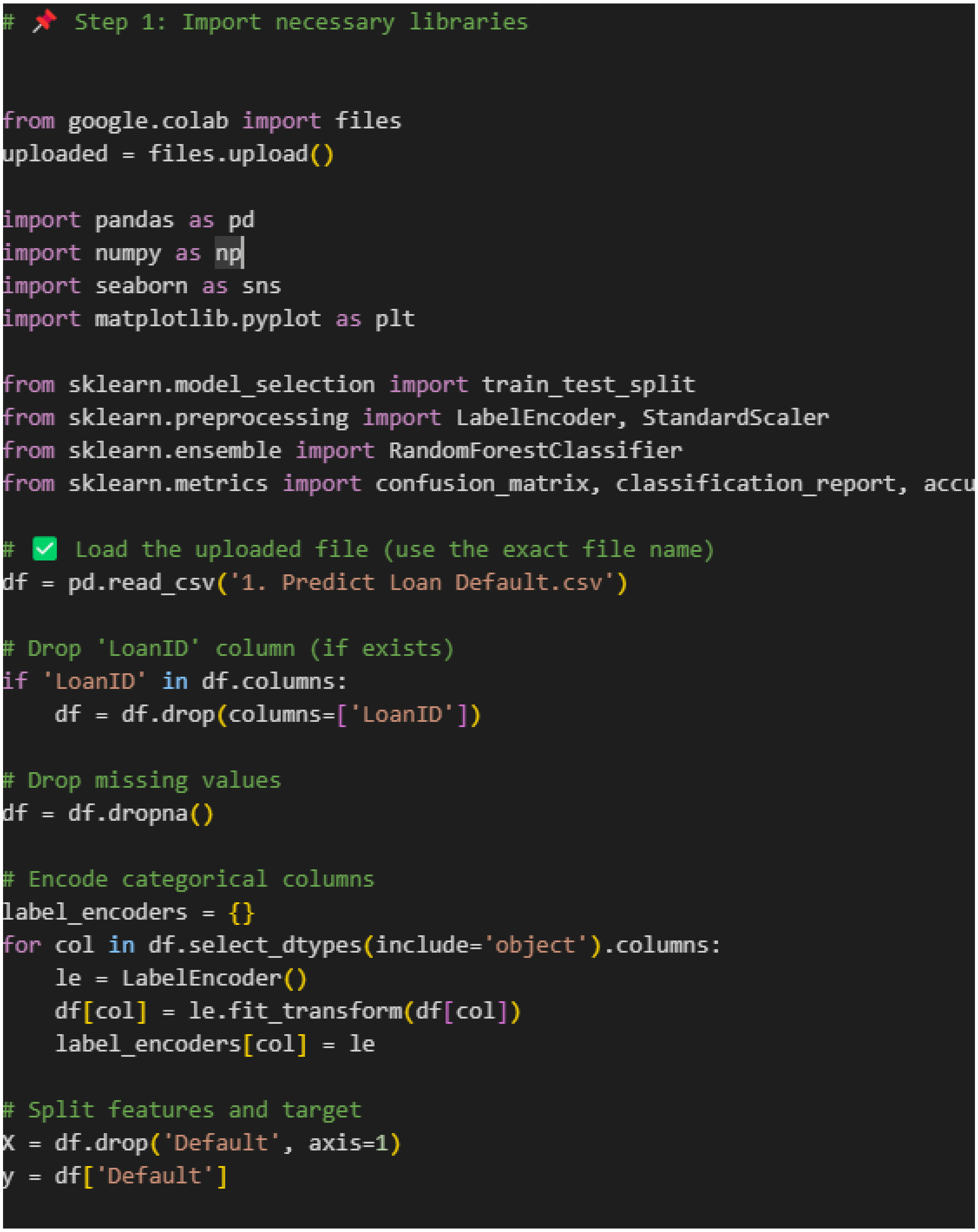


```
[35] import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns

     # Optional: Set Seaborn style
     sns.set(style="whitegrid")
```

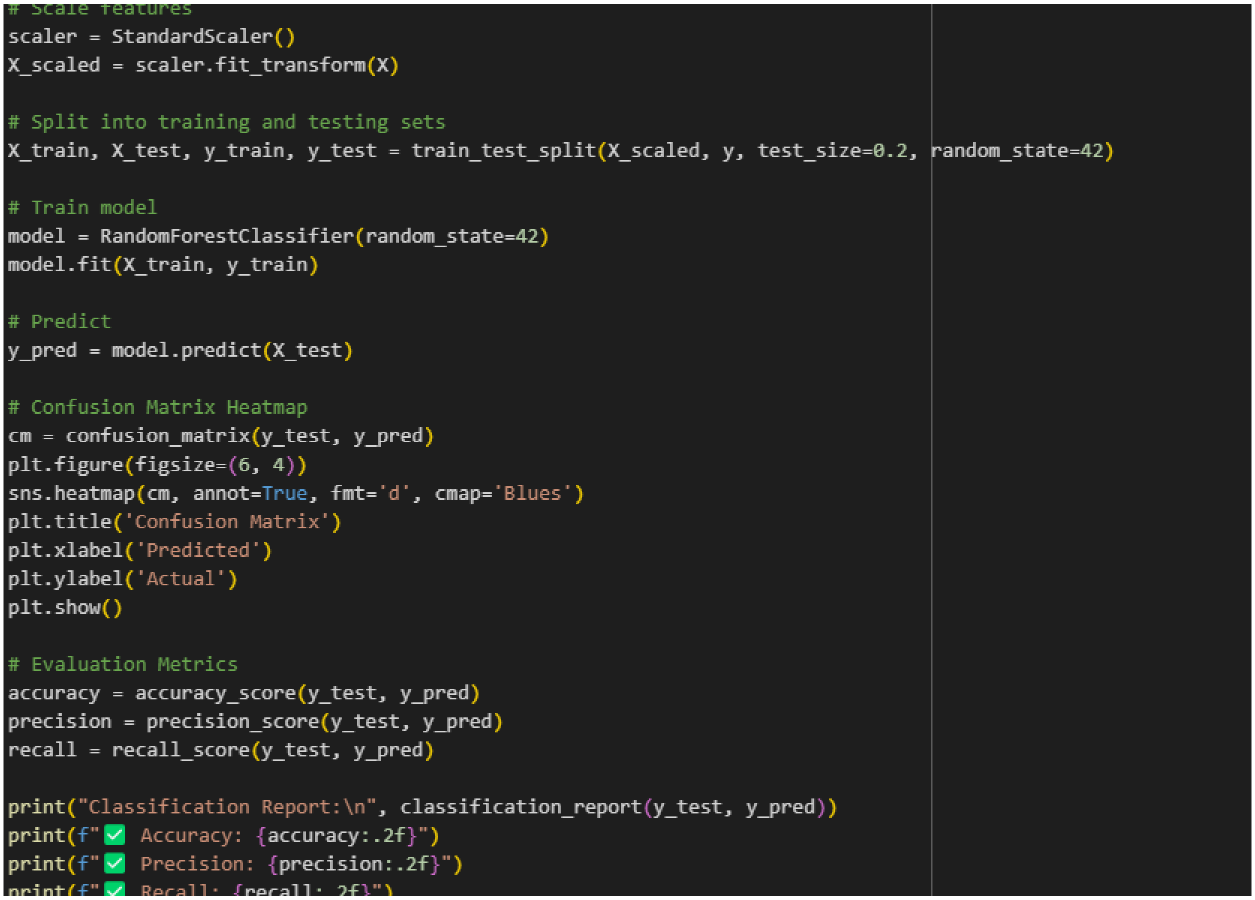


```
[ ] df = pd.DataFrame({
        'Joined_Club': [1, 0, 1, 1, 0, 1, 0, 1, 0, 0]
    })

    # Make sure the column name is correct and exists in the DataFrame
    sns.countplot(data=df, x='Joined_Club', palette='Set2')
    plt.title('Club Participation Count')
    plt.xlabel('Joined Club (0 = No, 1 = Yes)')
    plt.ylabel('Number of Students')
    plt.show()
```

```
<ipython-input-36-21595146d462>:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.countplot(data=df, x='Joined_Club', palette='Set2')
```



```
# 📌 Step 1: Import necessary libraries


from google.colab import files
uploaded = files.upload()

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, classification_report, accu

# ✅ Load the uploaded file (use the exact file name)
df = pd.read_csv('1. Predict Loan Default.csv')

# Drop 'LoanID' column (if exists)
if 'LoanID' in df.columns:
    df = df.drop(columns=['LoanID'])

# Drop missing values
df = df.dropna()

# Encode categorical columns
label_encoders = {}
for col in df.select_dtypes(include='object').columns:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    label_encoders[col] = le

# Split features and target
X = df.drop('Default', axis=1)
y = df['Default']
```



```
# Scale features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Train model
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)

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

# Evaluation Metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)

print("Classification Report:\n", classification_report(y_test, y_pred))
print(f"✅ Accuracy: {accuracy:.2f}")
print(f"✅ Precision: {precision:.2f}")
print(f"✅ Recall: {recall:.2f}")
```