Project Title: Student Club Participation Prediction using Machine Learning

Submitted By:
Name: *Deepanshu*
Roll No: *202401100300094*
Course: *Introduction to AI*

Date: *22/04/2025*

Introduction

Predicting student participation in clubs can help universities and colleges plan better engagement strategies. This project uses a machine learning approach to predict whether a student will join a club based on their interest level and free hours available per week.

The dataset contains two key features:

- **Interest Level**: A numerical score indicating student interest.
- **Free Hours Per Week**: The number of hours a student is free weekly.

The target variable is:

- **Club Participation**: Whether the student has joined a club (yes or no).

This is a binary classification problem, and we solve it using a Random Forest Classifier.

---

c. Methodology

1. **Data Preprocessing**:
    - The target variable (club_participation) was mapped from yes/no to 1/0.
    - Features were selected: interest_level and free_hours_per_week.
2. **Train-Test Split**:
    - The dataset was split into 80% training and 20% testing sets using train_test_split.
3. **Model Selection**:
    - A Random Forest Classifier was chosen due to its robustness and accuracy with tabular data.
4. **Model Evaluation**:
    - Evaluation metrics used: Accuracy, Precision, Recall, F1-score.
    - A confusion matrix was plotted for performance visualization.

- Feature importance was also visualized to understand the key drivers of prediction.

Code:

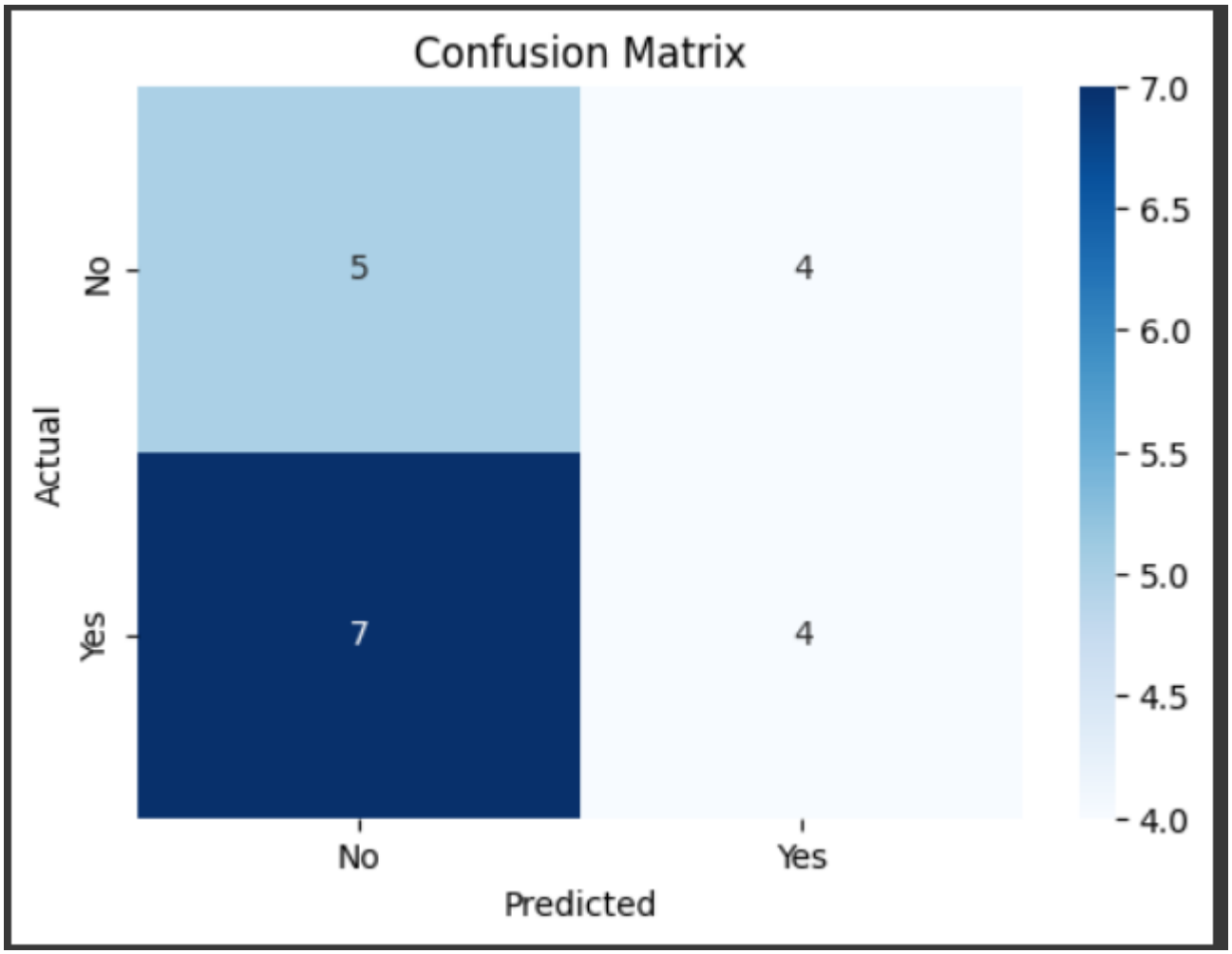
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import (
    confusion_matrix, accuracy_score, precision_score,
    recall_score, f1_score
)

# 1. Load your dataset
df = pd.read_csv("/content/club_participation.csv")

# 2. Preprocess the target variable
df["club_participation"] = df["club_participation"].map({"yes": 1, "no": 0})

# 3. Define features and target
X = df[["interest_level", "free_hours_per_week"]]
y = df["club_participation"]

# 4. Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# 5. Train model
```

```
model = RandomForestClassifier(class_weight='balanced', random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

# 6. Evaluation metrics
cm = confusion_matrix(y_test, y_pred)
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

# 7. Print metrics
print("Evaluation Metrics:")
print(f"Accuracy : {accuracy:.2f}")
print(f"Precision: {precision:.2f}")
print(f"Recall   : {recall:.2f}")
print(f"F1 Score : {f1:.2f}")

# 8. Confusion Matrix Heatmap
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['No', 'Yes'], yticklabels=['No', 'Yes'])
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()

# 9. Feature Importance
importances = model.feature_importances_
features = X.columns
indices = np.argsort(importances)[::-1]
```

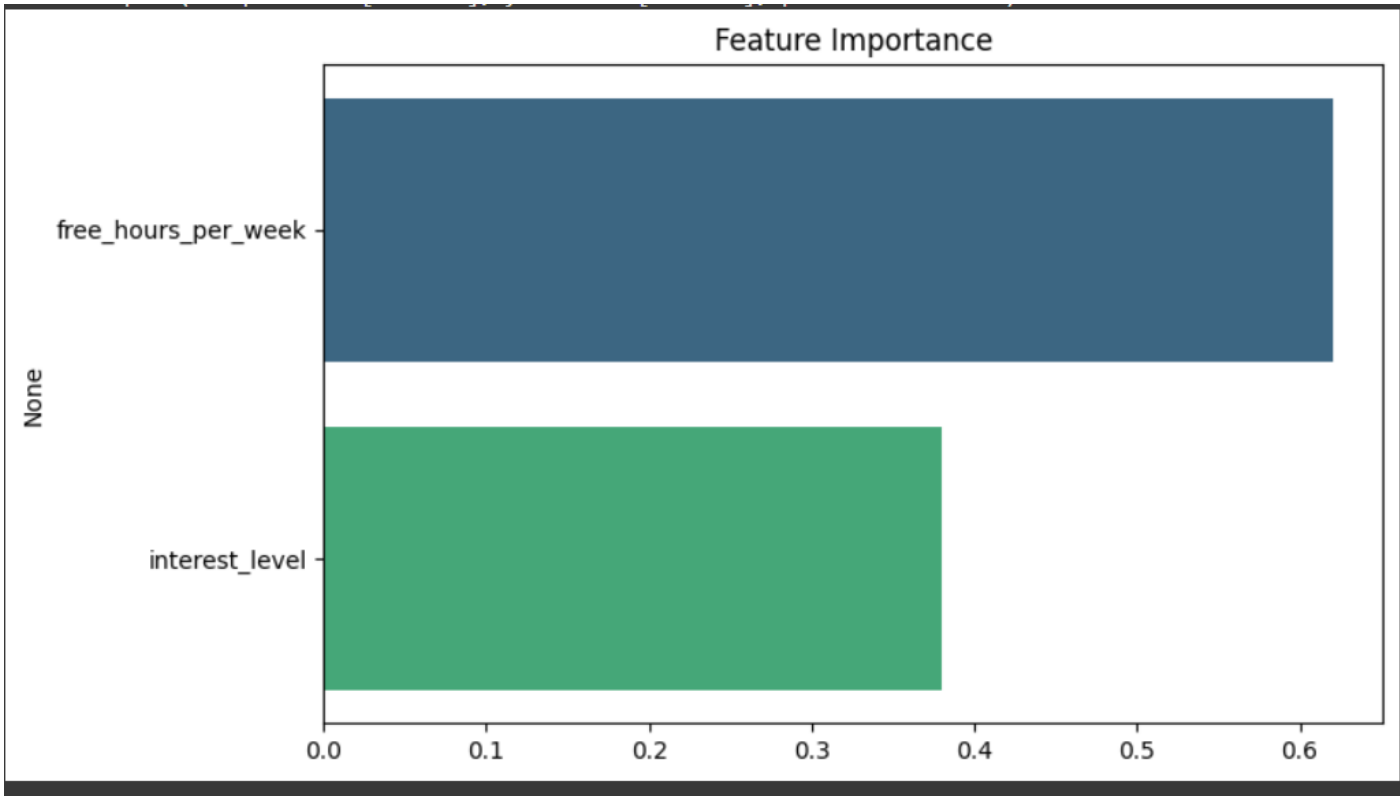
```
plt.figure(figsize=(8, 5))
sns.barplot(x=importances[indices], y=features[indices], palette='viridis')
plt.title("Feature Importance")
plt.show()
```

Output/Result:

1. **Confusion Matrix Heatmap**



2. **Feature Importance Plot**



3. **Evaluation Metrics Output**

```
Evaluation Metrics:
Accuracy : 0.45
Precision: 0.50
Recall   : 0.36
F1 Score : 0.42
```

---

f. References/Credits

- Dataset: *Provided by instructor*

S.No	Interest Level	Free Hours/Week	Club Participation
1	4	17	No
2	6	12	No
3	8	19	No
4	6	19	Yes
5	9	17	No
6	9	3	Yes
7	2	0	No
8	1	17	Yes
9	2	0	No
10	5	12	Yes
11	4	19	Yes
12	4	19	Yes
13	9	9	No
14	8	10	No
15	6	8	No
16	1	8	Yes
17	2	4	No
18	6	0	Yes
19	2	16	No
20	5	1	Yes

- Python Libraries: pandas, numpy, seaborn, matplotlib, sklearn

- Tool: Google Colab