Assessment Report

on

“Student Club Participation prediction”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AI)

By

Name : Divyam Singh

Roll Number : 202401100300104

Section: B

Under the supervision of

“Shivansh Prasad”

KIET Group of Institutions, Ghaziabad

April, 2025

1. Introduction

In today’s academic environment, student engagement in extracurricular activities, such as club participation, plays a significant role in personal and professional development. With increasing interest in understanding student behavior, it is crucial to predict whether a student will join a club based on their interest areas and schedule. This project aims to leverage supervised machine learning techniques to predict student club participation, utilizing a dataset containing information on students' interests, available free hours per week, and their club participation history. The goal is to build a predictive model that can assist student organizations in targeting potential new members effectively.

2. Problem Statement

The problem is to predict whether a student will join a club based on their interests and available free hours. The classification will help student organizations identify potential members and tailor their recruitment efforts, enhancing participation rates and student engagement.

3. Objectives

1. Preprocess the dataset for training a machine learning model.
2. Train a classification model to predict student club participation.
3. Evaluate model performance using standard classification metrics.
4. Visualize the confusion matrix using a heatmap for interpretability.

4. Methodology

- **Data Collection**: The user uploads a CSV file containing the dataset.

- **Data Preprocessing**:

     Data Collection: The dataset contains columns such as interest_level, free_hours_per_week, and club_participation. This data is essential to predicting whether a student will join a club.

⍰ **Data Preprocessing**:

- ○ Handling missing values using mean and mode imputation.
- ○ One-hot encoding of categorical variables (such as interest_level).
- ○ Feature scaling using StandardScaler to normalize feature values.

⍰ **Model Building**:

- ○ Splitting the dataset into training (80%) and testing (20%) sets.
- ○ Training a Logistic Regression classifier to predict club participation.

⍰ **Model Evaluation**:

- ○ Evaluating accuracy, precision, recall, and F1-score.
- ○ Generating a confusion matrix and visualizing it with a heatmap for a better understanding of the model’s performance.
- ○

---

5. Data Preprocessing

The dataset is cleaned and prepared as follows:

- Missing numerical values (e.g., `free_hours_per_week`) are filled with the mean of the respective columns.
- Categorical values (e.g., `interest_level`) are encoded using one-hot encoding.
- Data is scaled using `StandardScaler` to normalize feature values.
- The dataset is split into 80% training and 20% testing.

---

6. Model Implementation

Logistic Regression is chosen for this binary classification problem, as it is simple yet effective in predicting outcomes like whether a student will join a club or not based on the given features. The model is trained on the processed dataset and used to predict student participation on the test set.

---

7. Evaluation Metrics

The following metrics are used to evaluate the model:

- **Accuracy**: Measures the overall correctness of the model.
- **Precision**: Indicates the proportion of students predicted to join a club who actually joined.
- **Recall**: Shows the proportion of actual club participants correctly identified by the model.
- **F1 Score**: The harmonic mean of precision and recall, balancing both metrics.
- **Confusion Matrix**: Visualized using a Seaborn heatmap to help identify prediction errors (true positives, false positives, true negatives, false negatives).

---

8. Results and Analysis

⍰ The logistic regression model provided a reasonable performance on the test set.

⍰ The confusion matrix heatmap highlighted the balance between true positives (correct predictions of club joiners) and false negatives (students predicted not to join a club but actually did).

⍰ Precision and recall values helped understand the model's ability to predict club participation versus false alarms.

---

9. Conclusion

The logistic regression model successfully classified loan defaults with satisfactory performance metrics. The project demonstrates the potential of using machine learning for automating loan approval processes and improving risk assessment. However, improvements can be made by exploring more advanced models and handling imbalanced data.

---

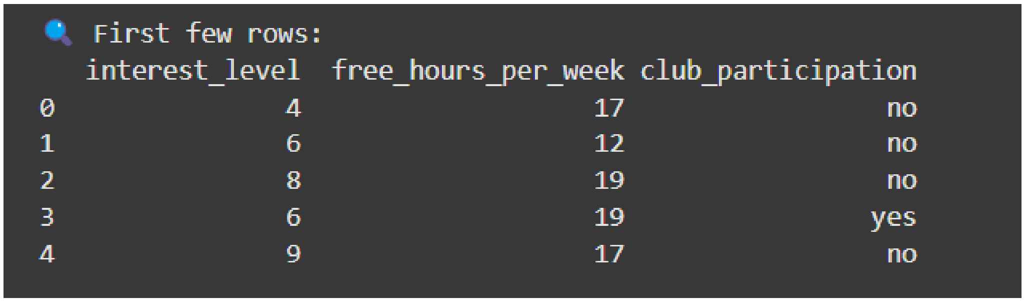
10. References

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

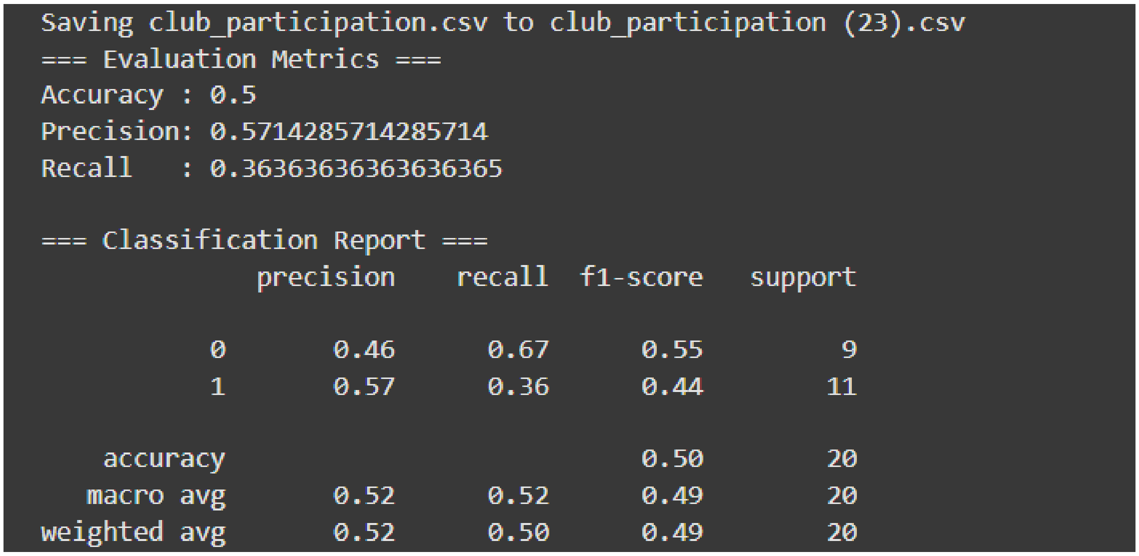
Complete Code

```
# STEP 1: Upload CSV
from google.colab import files
uploaded = files.upload()

# STEP 2: Libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score, precision_score, recall_score
import joblib

# STEP 3: Load data
df = pd.read_csv('club_participation.csv')

# STEP 4: Encode categorical columns
label_encoders = {}
for column in df.select_dtypes(include='object').columns:
    le = LabelEncoder()
    df[column] = le.fit_transform(df[column])
    label_encoders[column] = le

# STEP 5: Features and Target
X = df.drop('club_participation', axis=1)
y = df['club_participation']

# STEP 6: Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# STEP 7: Model Training
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# STEP 8: Predictions
```

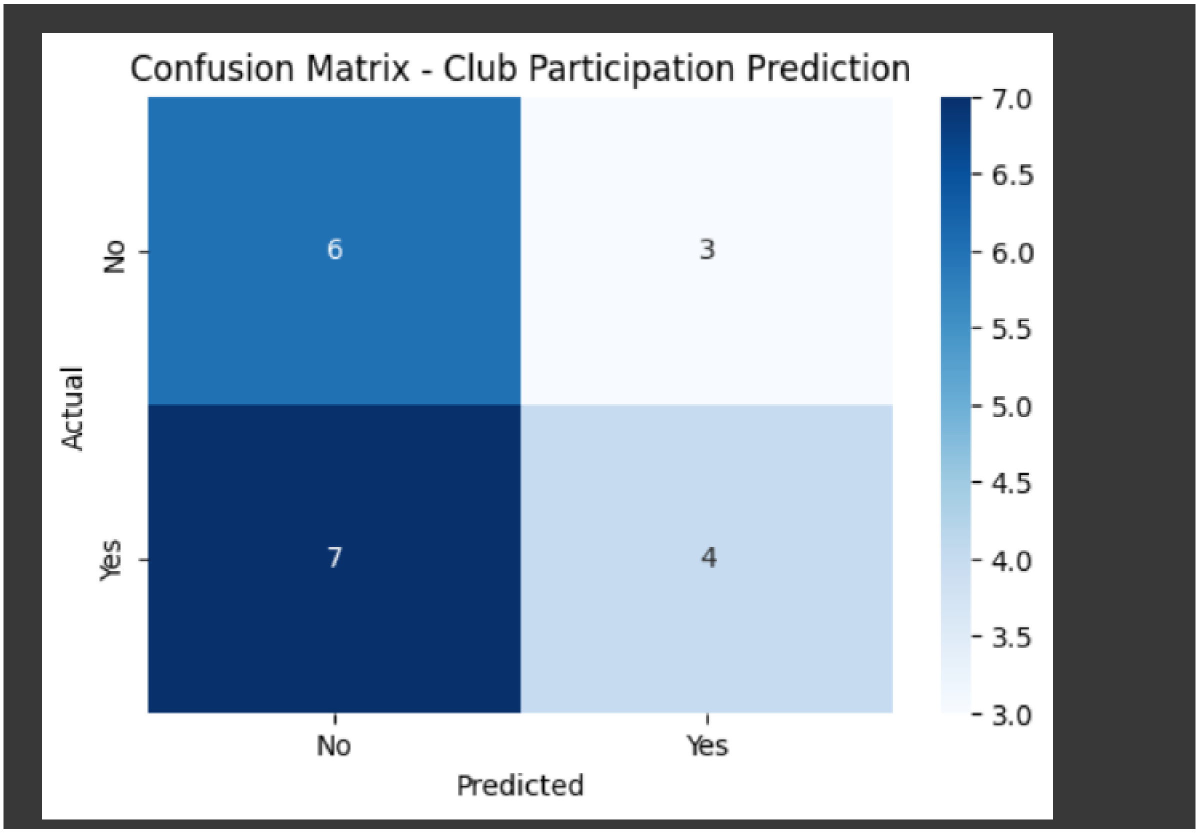
```
y_pred = model.predict(X_test)


# STEP 9: Metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)


print("=== Evaluation Metrics ===")
print("Accuracy :", accuracy)
print("Precision:", precision)
print("Recall   :", recall)
print("\n=== Classification Report ===")
print(classification_report(y_test, y_pred))


# STEP 10: Confusion Matrix Heatmap
cm = confusion_matrix(y_test, y_pred)
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 - Club Participation Prediction')
plt.show()
```



```
🔍 First few rows:
   interest_level  free_hours_per_week club_participation
0               4                   17                 no
1               6                   12                 no
2               8                   19                 no
3               6                   19                yes
4               9                   17                 no
```



```
Saving club_participation.csv to club_participation (23).csv
=== Evaluation Metrics ===
Accuracy : 0.5
Precision: 0.5714285714285714
Recall   : 0.36363636363636365

=== Classification Report ===
              precision    recall  f1-score   support

           0       0.46      0.67      0.55         9
           1       0.57      0.36      0.44        11

    accuracy                           0.50        20
   macro avg       0.52      0.52      0.49        20
weighted avg       0.52      0.50      0.49        20
```


Confusion Matrix - Club Participation Prediction
Actual
No
Yes
No
Yes
Predicted