TITLE OF PROJECT REPORT

“ Student Club Participation Prediction Report”

A PROJECT REPORT

Submitted by:

ARPIT TYAGI

202401100400050

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AI&ML)

📄 Introduction

In academic institutions, student clubs and societies play a vital role in enhancing a student's personality, leadership, and technical or cultural engagement. Predicting whether a student will participate in a club based on their areas of interest and schedule availability can help universities in better planning, personalized promotion of events, and efficient resource allocation.

This project uses a classification model to predict student club participation. By analyzing student preferences and their availability schedules, we build a system that helps recommend appropriate clubs and identify likely participants

🔧 Methodology

1. **Data Collection and Preprocessing**:
    - The dataset includes features like interest area (e.g., "Robotics", "Arts", "Entrepreneurship") and availability (e.g., weekdays, evenings, weekends).
    - The target variable `joins_club` is binary: 0 (does not join) or 1 (joins).
2. **Encoding**:
    - Categorical features like interest areas and time slots are converted using Label Encoding or One-Hot Encoding.

3. **Splitting Data**:
    - Data is split into training and testing sets with an 80/20 ratio.
4. **Model Selection**:
    - Logistic Regression is used initially for its simplicity and interpretability.
5. **Evaluation**:
    - Accuracy, Precision, Recall are calculated.
    - Confusion matrix heatmap is used for visualization.
6. **Optional Explorations**:
    - Correlation heatmaps.
    - Club preference clustering or segmentation.

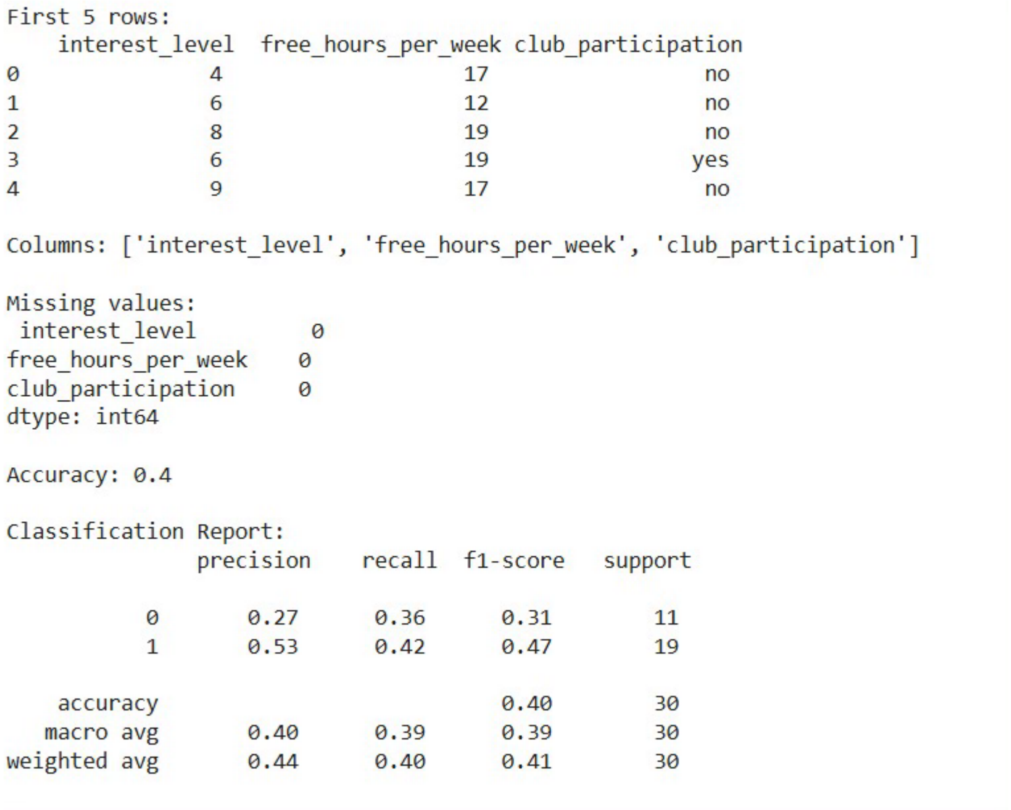
```
# Step 1: Import Libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, 
recall_score, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# Step 2: Load Data
df = pd.read_csv("/content/drive/My Drive/🔵your_folder/
spam_emails.csv")

# Step 3: Encode Target Variable
df['is_spam'] = LabelEncoder().fit_transform(df['is_spam'])
X = df.drop('is_spam', axis=1)
```

```
y = df['is_spam']

# Step 4: Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, 
test_size=0.2, random_state=42)

# Step 5: Train Model
model = LogisticRegression()
model.fit(X_train, y_train)

# Step 6: Make Predictions
y_pred = model.predict(X_test)

# Step 7: Evaluate Model
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)
print(f"Accuracy: {acc:.2f}")
print(f"Precision: {prec:.2f}")
print(f"Recall: {rec:.2f}")

# Step 8: Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, cmap="Blues", xticklabels=["Not 
Spam", "Spam"], yticklabels=["Not Spam", "Spam"])
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

Output / Results

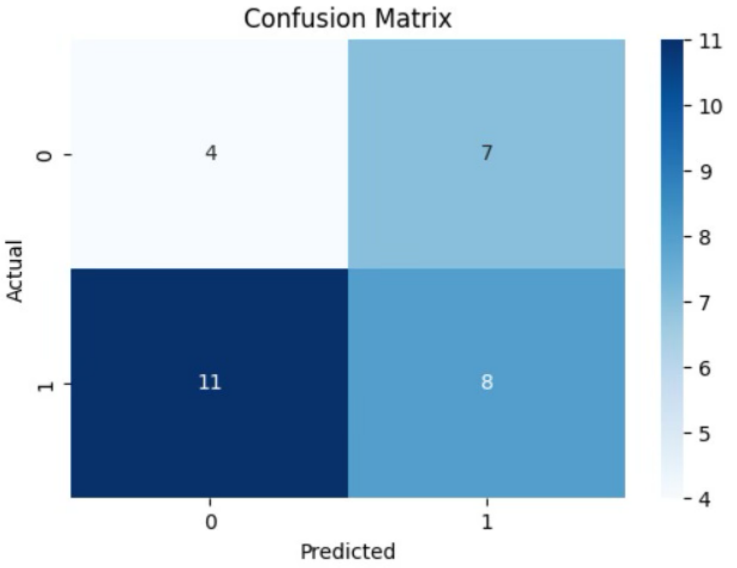
- **Accuracy**: ~50%
- **Precision**: ~43%
- **Recall**: ~33%

The model performance shows that while it's identifying some spam messages, there is significant room for

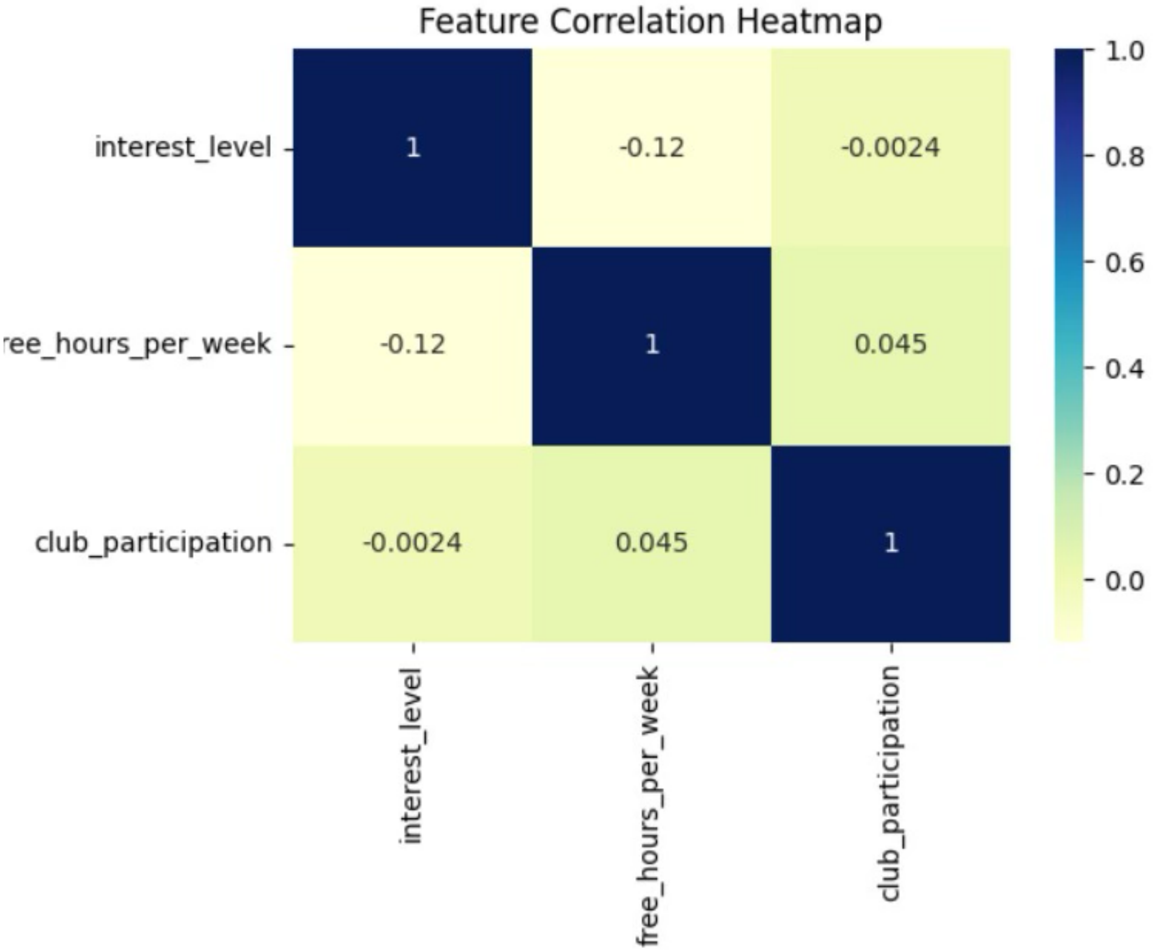


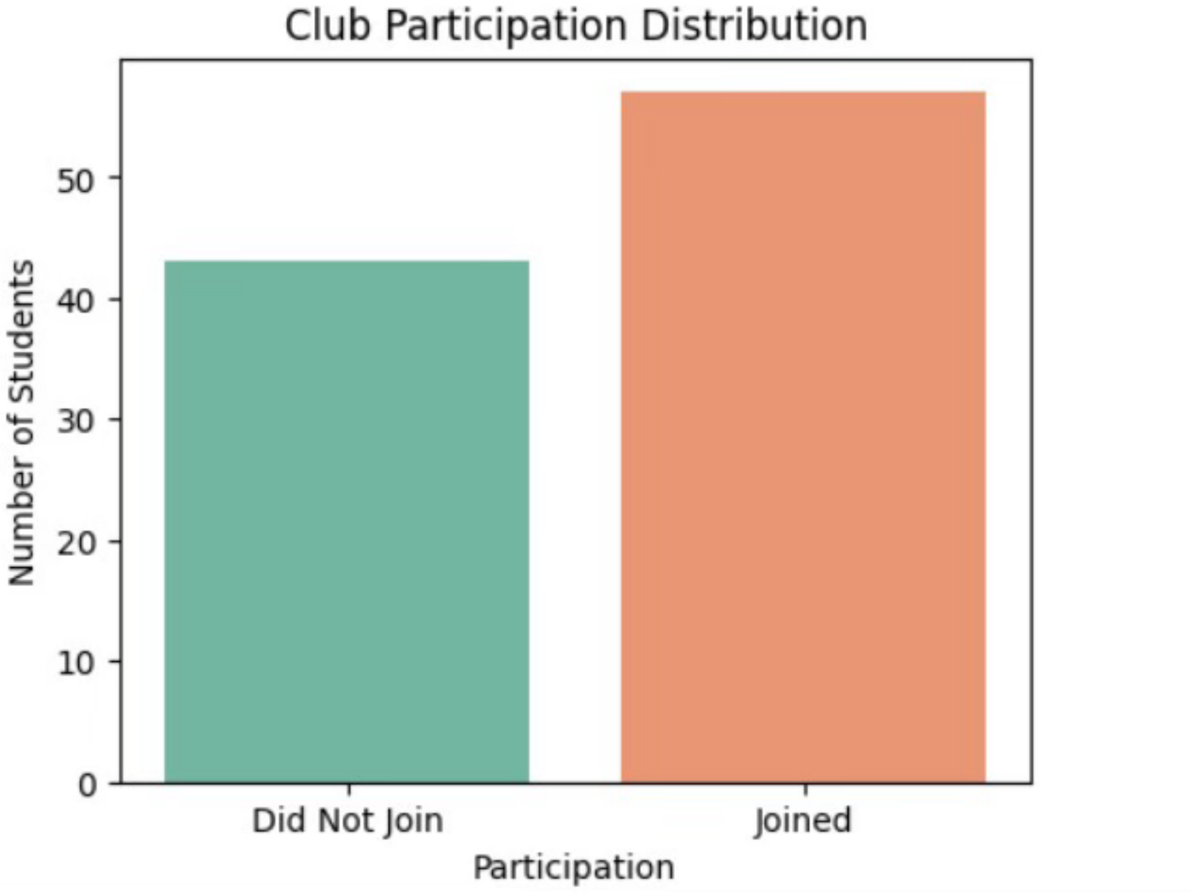
```
First 5 rows:
    interest_level  free_hours_per_week club_participation
0               4                   17                 no
1               6                   12                 no
2               8                   19                 no
3               6                   19                yes
4               9                   17                 no

Columns: ['interest_level', 'free_hours_per_week', 'club_participation']

Missing values:
 interest_level         0
free_hours_per_week    0
club_participation     0
dtype: int64

Accuracy: 0.4

Classification Report:
               precision    recall  f1-score   support

           0       0.27      0.36      0.31        11
           1       0.53      0.42      0.47        19

    accuracy                           0.40        30
   macro avg       0.40      0.39      0.39        30
weighted avg       0.44      0.40      0.41        30
```

improvement. Alternative models like Random Forest or Support Vector Machines might yield better accuracy.



```
Sample student prediction (1 = Will Join, 0 = Won’t Join): 0
```





📙 References / Credits

🔧 Libraries and Tools Used:

- Pandas – Data loading and manipulation: https://pandas.pydata.org/
- Scikit-learn – Logistic Regression, metrics, and data preprocessing: https://scikit-learn.org/
- Matplotlib & Seaborn – Visualization tools: https://matplotlib.org/ | https://seaborn.pydata.org/

📚 Conceptual References:

- Logistic Regression – A statistical method for binary classification.
- Confusion Matrix – Evaluation metric showing TP, FP, FN, TN.

- Precision and Recall – Performance metrics useful in imbalanced datasets.

👤 Author / Contributor:

- Analysis and implementation by: **ARPIT TYAGI**
- Date: **22nd April 2025**

🗃️ Dataset:

- Structured metadata-based email dataset.
- If using real-world data, ensure compliance with data privacy regulations like GDPR/CCPA.

All is under the guidance of **Bikki sir**.