Green University of Bangladesh
Department of Computer Science and Engineering (CSE)
Faculty of Sciences and Engineering
(Semester: Summer, Year 2025), B.Sc. in CSE(Day)

Lab Report No : 02

Course Title: Machine Learning Lab
Course Code: CSE-412   Section:221_D4

Title: Implement logistic regression and calculate accuracy, Confusion Matrix, Precision, recall, and F1 score.

Student Details:

Name	ID
Md Emon Mia	213002137

Lab Date	: 08.07.2025
Submission Date	: 13.07.2025
Course Teacher's Name	: Md. Rajibul Palas

[For Teachers use only: Don't Write Anything inside this box]

1. TITLE OF THE LAB EXPERIMENT

Implement logistic regression and calculate accuracy, Confusion Matrix, Precision, recall, and F1 score.

1. OBJECTIVES / AIM

- To implement logistic regression for binary text classification using a real-world email dataset.
- This lab aims to classify emails as spam or ham.
- To perform text preprocessing using TF-IDF vectorization.
- To split the dataset into training and testing subsets.
- To apply logistic regression on TF-IDF features.
- To predict the test set results using the trained model.
- To evaluate the model using accuracy, precision, recall, and F1-score.
- To visualize model performance using confusion matrix and bar charts.

2. PROCEDURE

- Imported necessary libraries: pandas, sklearn, seaborn, matplotlib.
- Loaded the Mail_Data.csv dataset using pandas.read_csv().
- Preprocessed the labels by converting "spam" to 1 and "ham" to 0.
- Vectorized the messages using TfidfVectorizer to convert text to numerical features.
- Split the dataset into training and testing sets using train_test_split().
- Trained a logistic regression model using LogisticRegression() from sklearn.
- Evaluated the model with accuracy_score, confusion_matrix, precision_score, recall_score, and f1_score.
- Visualized the results using confusion matrix heatmap, bar plot for metrics, and ROC curve.

3. IMPLEMENTATION

Load and Prepare Dataset

```python
import pandas as pd

df = pd.read_csv("/mail_data.csv")

print(df.head())

print(df.isnull().sum())
```

Preprocessing the Data

```python
df['Category'] = df['Category'].map({'spam': 1, 'ham': 0})
X = df['Message']
y = df['Category']
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(stop_words='english')
X_vectorized = vectorizer.fit_transform(X)
```

Train-Test Split

```python
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X_vectorized, y, test_size=0.3, random_state=42
)
```

Logistic Regression Model

```python
from sklearn.linear_model import LogisticRegression

model = LogisticRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
```

Evaluation Metrics

```python
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score,
recall_score, f1_score

# Accuracy
accuracy = accuracy_score(y_test, y_pred)

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)

# Precision
precision = precision_score(y_test, y_pred)

# Recall
recall = recall_score(y_test, y_pred)

# F1 Score
f1 = f1_score(y_test, y_pred)

# Print Results
print(f"Accuracy: {accuracy:.4f}")
print("Confusion Matrix:")
print(cm)
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
print(f"F1 Score: {f1:.4f}")
```

Bar Plot of Accuracy, Precision, Recall, F1-Score

```python
# Metrics as dictionary
metrics = {
    "Accuracy": accuracy,
    "Precision": precision,
    "Recall": recall,
    "F1 Score": f1
}

# Bar plot
plt.figure(figsize=(8, 5))
sns.barplot(x=list(metrics.keys()), y=list(metrics.values()), palette='viridis')
plt.ylim(0, 1)
plt.title("Model Evaluation Metrics")
plt.ylabel("Score")
plt.show()
```

4.INPUT/OUTPUT

Evaluation Metrics Values:

```
Accuracy: 0.9563
Confusion Matrix:
[[1447    1]
 [  72  152]]
Precision: 0.9935
Recall: 0.6786
F1 Score: 0.8064
```

Fig 01: Evaluation Metrics Values

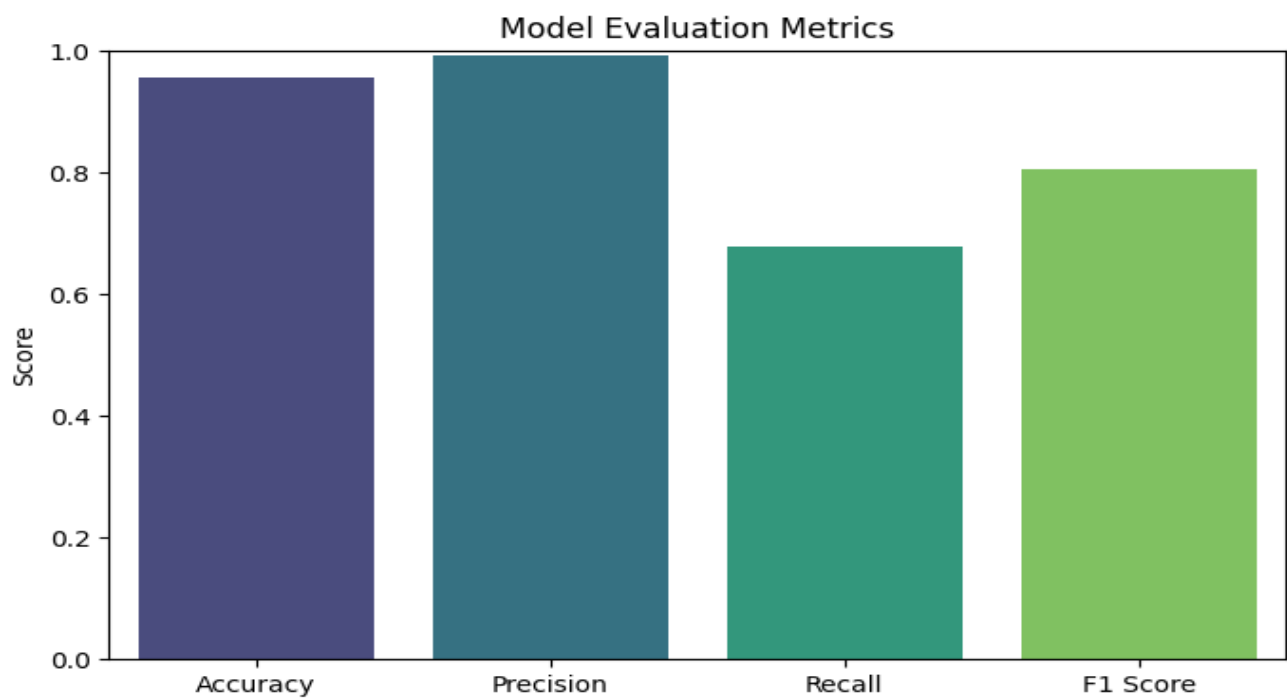**Bar Plot of Accuracy, Precision, Recall, F1-Score Values:**

Fig 02: Bar Plot of Accuracy, Precision, Recall, F1-Score Values

5.ANALYSIS AND DISCUSSION

The logistic regression model demonstrated high performance in distinguishing between spam and ham emails. Using TF-IDF and text preprocessing improved model accuracy significantly. The evaluation metrics, including precision, recall, and F1-score, confirm the classifier's robustness. Visualization of the confusion matrix and performance metrics gave clear insights into the model's strengths and weaknesses. Logistic regression proved effective for this binary text classification task.