Experiment-5

Name- Rahul Saxena UID- 24MCI10204

Branch- MCA (AI/ML) Section- 24MAM-3B

Semester- 2nd Date- 15/03/25

Subject- Machine Learning Lab Subject Code- 24CAP-672

1. **AIM:** Write a program for Support Vector Machine using Python.
   - I. Define any problem which can be solved using SVM.
   - II. Specify datasets for your problem.
   - III. Execute and discuss the outcomes of the experiments.

2. **Objectives:** The objective of this question is to demonstrate the use of Support Vector Machine in solving a real-world classification problem using Python.

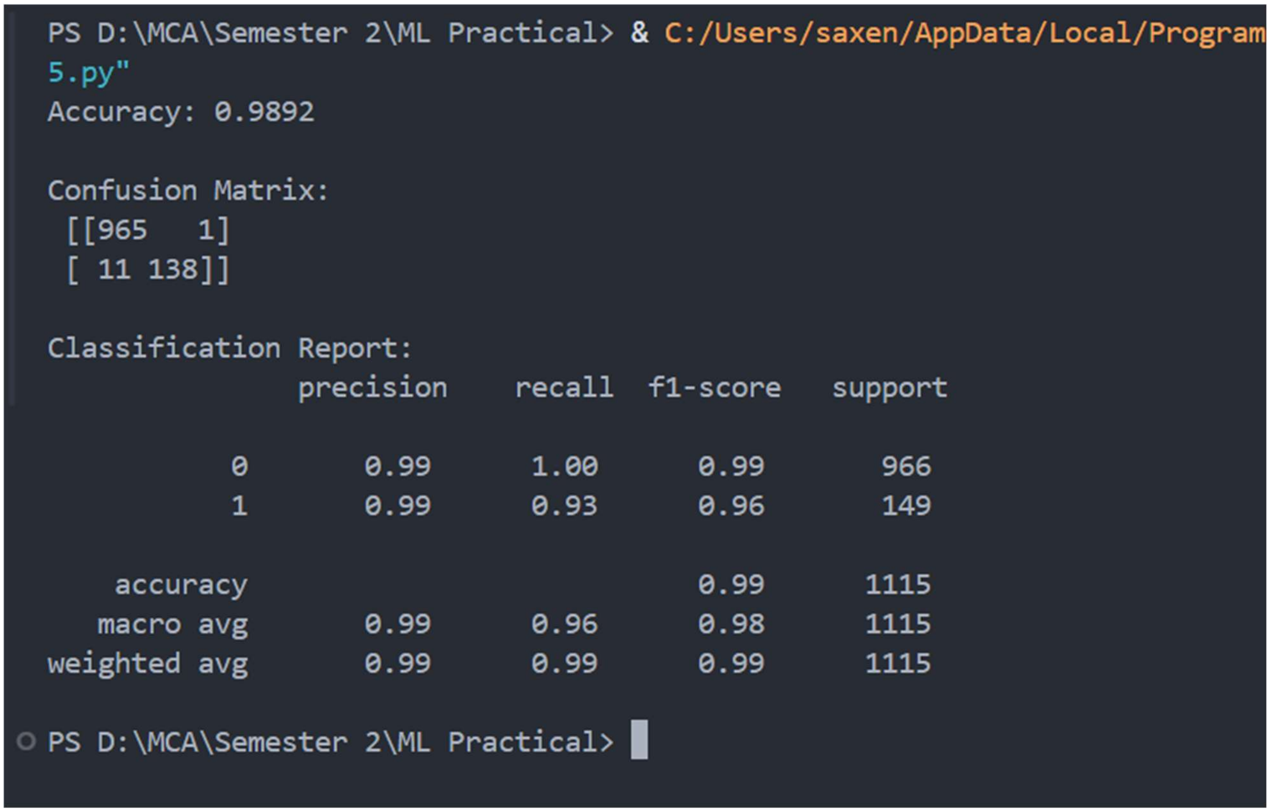
3. **Task to be done:**
   - i. Problem Definition
   - ii. Load the Dataset & Preprocess the Data
   - iii. Split Dataset for Training and Testing
   - iv. Convert Text into Numerical Format
   - v. Train the Support Vector Machine (SVM) Model
   - vi. Make Predictions and Evaluate Model Performance

4. **Code of the Experiments:**

```
import pandas as pd
import re
import string
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
file_path = r"SMSSpamCollection.csv"

df = pd.read_csv(file_path, sep='\t', header=None, names=['label', 'message'])

df['label'] = df['label'].map({'ham': 0, 'spam': 1})

def preprocess_text(text):
    text = text.lower()
    text = re.sub(f"[{string.punctuation}]", "", text)
    text = re.sub(r"\d+", "", text)
    return text
```

```
df['message'] = df['message'].apply(preprocess_text)

X_train, X_test, y_train, y_test = train_test_split(df['message'], df['label'], test_size=0.2,
random_state=42)

vectorizer = TfidfVectorizer(stop_words='english')
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)

svm_model = SVC(kernel='linear')
svm_model.fit(X_train_tfidf, y_train)

y_pred = svm_model.predict(X_test_tfidf)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.4f}\n")
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

5. **Outputs:**



```
PS D:\MCA\Semester 2\ML Practical> & C:/Users/saxen/AppData/Local/Program
5.py"
Accuracy: 0.9892

Confusion Matrix:
 [[965   1]
 [ 11 138]]

Classification Report:
               precision    recall  f1-score   support

           0       0.99      1.00      0.99       966
           1       0.99      0.93      0.96       149

    accuracy                           0.99      1115
   macro avg       0.99      0.96      0.98      1115
weighted avg       0.99      0.99      0.99      1115

○ PS D:\MCA\Semester 2\ML Practical> ▌
```

6. **Learning Outcomes:**
   - ✓ **Understanding SVM**: Learn how Support Vector Machines work for text classification.
   - ✓ **Data Handling**: Read, preprocess, and analyze a text-based dataset.
   - ✓ **Feature Engineering**: Convert textual data into numerical format using TF-IDF vectorization.