

**CSBB311: MACHINE LEARNING LAB**  
**ASSIGNMENT 3 :- Classification Using SVM**

**Submitted By:**

**Name: Kartik Mittal**

**Roll No: 221210056**

**Branch: CSE**

**Semester: 5th Sem**

**Group : 2**

**Submitted To: Dr. Preeti Mehta**

Department of Computer Science and Engineering



**NATIONAL INSTITUTE OF TECHNOLOGY DELHI**

2024

## Code :-

```
1  import pandas as pd
2  from sklearn.model_selection import train_test_split
3  from sklearn.svm import SVC
4  from sklearn.metrics import accuracy_score, confusion_matrix
5  from sklearn.preprocessing import StandardScaler, LabelEncoder
6  import matplotlib.pyplot as plt
7  import seaborn as sns
8
9  # Load the Titanic dataset
10 data = pd.read_csv('tested.csv')
11
12 # Preprocessing
13 data['Age'].fillna(data['Age'].mean(), inplace=True)
14 data['Embarked'].fillna(data['Embarked'].mean(), inplace=True)
15
16 # Encode categorical features
17 label_encoder = LabelEncoder()
18 data['Sex'] = label_encoder.fit_transform(data['Sex'])
19 data['Embarked'] = label_encoder.fit_transform(data['Embarked'])
20
21 # Drop irrelevant columns (Passengerid and 'zero' columns)
22 data = data.drop(columns=['Passengerid'] + [col for col in data.columns if 'zero' in col])
23
24 # Features (X) and Target (y)
25 X = data.drop(columns=['Survived']) # target column
26 y = data['Survived']
27
28 # Split data into training (60%), validation (20%), and testing (20%) sets
29 X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.4, random_state=42)
30 X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5, random_state=42)
31
32 # Standardize features
33 scaler = StandardScaler()
34 X_train = scaler.fit_transform(X_train)
35 X_val = scaler.transform(X_val)
36 X_test = scaler.transform(X_test)
37
38 # Train SVM model
39 svm_model = SVC(kernel='linear')
40 svm_model.fit(X_train, y_train)
41
42 # Validate the model
43 y_val_pred = svm_model.predict(X_val)
44 val_accuracy = accuracy_score(y_val, y_val_pred)
45 print(f'Validation Accuracy: {val_accuracy:.2f}')
46
47 # Test the model
48 y_test_pred = svm_model.predict(X_test)
49 test_accuracy = accuracy_score(y_test, y_test_pred)
50 print(f'Test Accuracy: {test_accuracy:.2f}')
51
52 # Plot confusion matrix for the test data
53 conf_matrix = confusion_matrix(y_test, y_test_pred)
54 plt.figure(figsize=(6,4))
55 sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['Did not survive', 'Survived'], yticklabel
56 plt.title('Confusion Matrix - Test Data')
```

```

56 plt.title('Confusion Matrix - Test Data')
57 plt.ylabel('True Label')
58 plt.xlabel('Predicted Label')
59 plt.show()
60
61 # Plot accuracy comparison
62 plt.figure(figsize=(6,4))
63 accuracy_scores = [val_accuracy, test_accuracy]
64 labels = ['Validation', 'Test']
65 sns.barplot(x=labels, y=accuracy_scores, palette='viridis')
66 plt.title('Accuracy Comparison: Validation vs Test')
67 plt.ylabel('Accuracy')
68 plt.show()

```

Output :-

```

hine Learning\lab 3 svm\svm.py"
Validation Accuracy: 75%
Test Accuracy: 70%

```

Plots :-



