A PYTHON PROGRAM TO IMPLEMENT SVM CLASSIFIER MODEL

Expt no. 6

Name: Jayasuriya.j

Roll no: 241801102

PROGRAM:

```
import numpy as np
import pandas as pd
from sklearn import svm
import matplotlib.pyplot as plt
import seaborn as sns; sns.set(font_scale=1.2)
from sklearn.metrics import confusion_matrix,
classification_report
from sklearn.model_selection import
train_test_split

recipes =
pd.read_csv('C:\\Users\\Luqman\\Downloads\\arc
hive (6)\\recipes_muffins_cupcakes.csv')
print(recipes.shape)
print(recipes.head())

sugar_butter = recipes[['Sugar','Flour']].values type_label
= np.where(recipes['Type']=='Muffin',
0, 1)

x_train, x_test, y_train, y_test =
```

```
train_test_split(sugar_butter,
type_label, test_size=0.2,
random_state=42)
model = svm.SVC(kernel='linear')
model.fit(sugar_butter,
type_label)
model1 = svm.SVC(kernel='linear')
model1.fit(x_train, y_train)

pred = model1.predict(x_test)
print("Predictions:", pred)
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, pred))
print("\nClassification Report:")
print(classification_report(y_test,
pred))
plt.figure(figsize=(8, 6)) sns.lmplot(x='Sugar',
y='Flour', data=recipes, hue='Type',
palette='Set1',
        fit_reg=False, scatter_kws={"s": 70})
plt.show()

plt.figure(figsize=(8, 6))

w = model.coef_[0]
hyperplane = -w[0] / w[1]
xx = np.linspace(5, 30) yy = hyperplane *
xx - (model.intercept_[0] / w[1])
```

```
sns.scatterplot(x='Sugar', y='Flour', hue='Type',
data=recipes, palette='Set1', s=70)
plt.plot(xx, yy, linewidth=2, color='black',
label='Decision Boundary')
plt.xlabel('Sugar')
plt.ylabel('Flour')

plt.legend()
plt.xlim(5, 30)
plt.ylim(35, 55)
plt.show()

plt.figure(figsize=(8, 6))

b = model.support_vectors_[0]
yy_down = hyperplane * xx + (b[1] - hyperplane
* b[0])
b = model.support_vectors_[-1]
yy_up = hyperplane * xx + (b[1] - hyperplane *
b[0])

sns.scatterplot(x='Sugar', y='Flour', hue='Type',
data=recipes, palette='Set1', s=70)
plt.plot(xx, yy, linewidth=2, color='black',
label='Decision Boundary')
plt.plot(xx, yy_down, 'k--', linewidth=1,
label='Margin')
plt.plot(xx, yy_up, 'k--', linewidth=1)
plt.scatter(model.support_vectors_[:, 0],
model.support_vectors_[:, 1],
        s=80, facecolors='none',
```

edgecolors='black', label='Support Vectors')

plt.xlabel('Sugar')

plt.ylabel('Flour')

plt.legend()

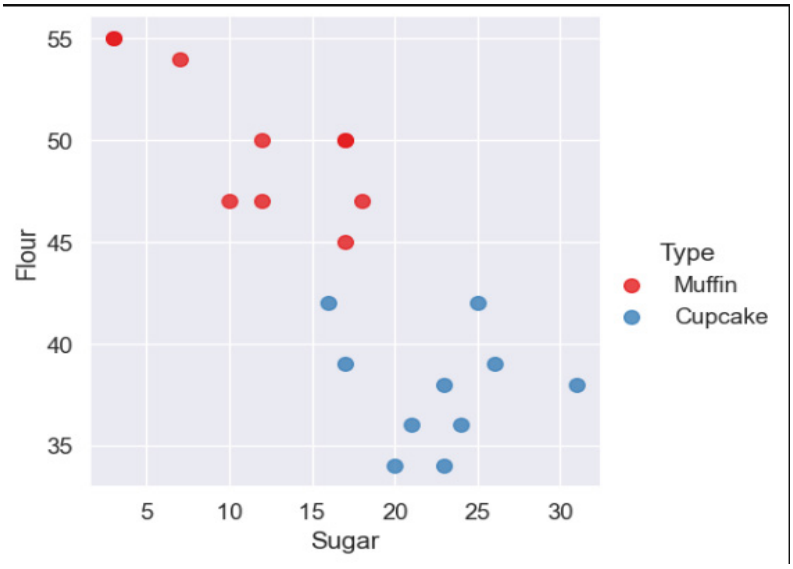
plt.xlim(5, 30)

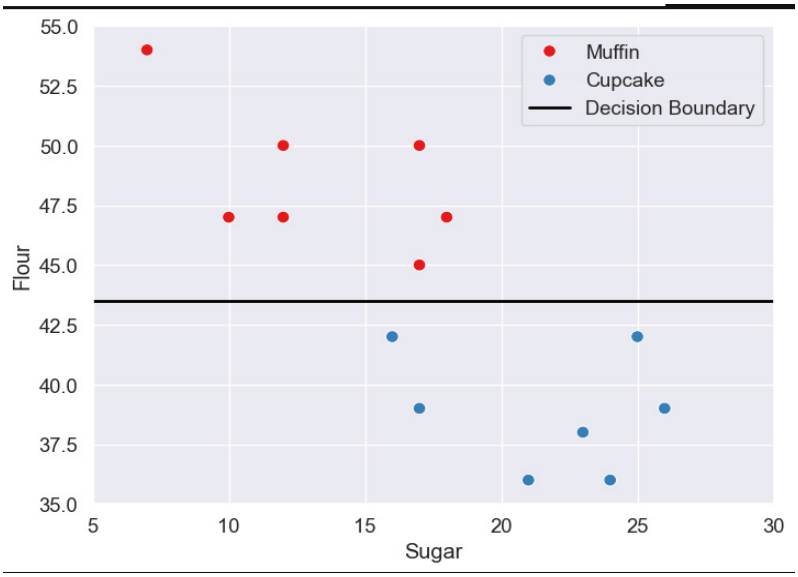
plt.ylim(35, 55)

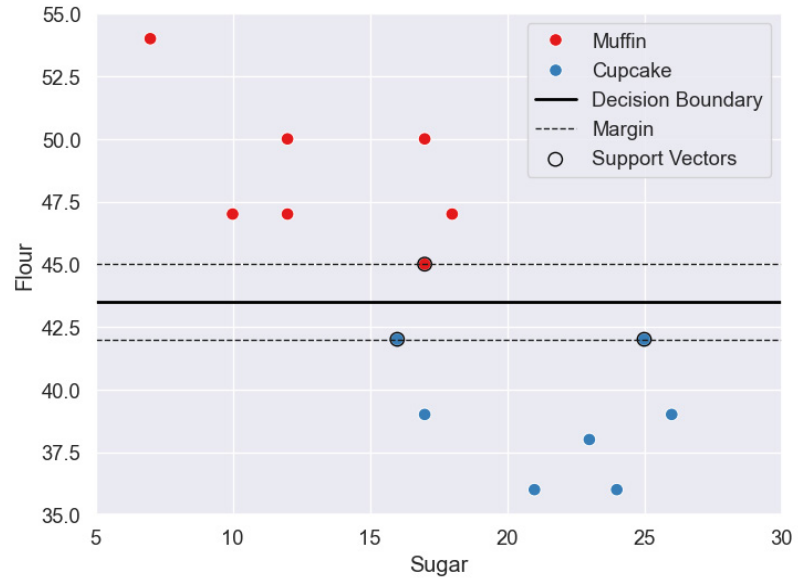
plt.show()

OUTPUT:

```
(20, 9)
     Type  Flour  Milk  Sugar  Butter  Egg  Baking Powder  Vanilla  Salt
0  Muffin     55    28      3       7    5              2        0     0
1  Muffin     47    24     12       6    9              1        0     0
2  Muffin     47    23     18       6    4              1        0     0
3  Muffin     45    11     17      17    8              1        0     0
4  Muffin     50    25     12       6    5              2        1     0
Predictions: [0 1 0 0]

Confusion Matrix:
[[2 0]
 [1 1]]

Classification Report:
              precision    recall  f1-score   support

           0       0.67      1.00      0.80         2
           1       1.00      0.50      0.67         2

    accuracy                           0.75         4
   macro avg       0.83      0.75      0.73         4
weighted avg       0.83      0.75      0.73         4
```


Flour
Sugar
Type
Muffin
Cupcake


Flour
Sugar
Muffin
Cupcake
Decision Boundary


Flour
Sugar
Muffin
Cupcake
Decision Boundary
Margin
Support Vectors