Lab Assignment 6

Fundamentals of Machine Learning

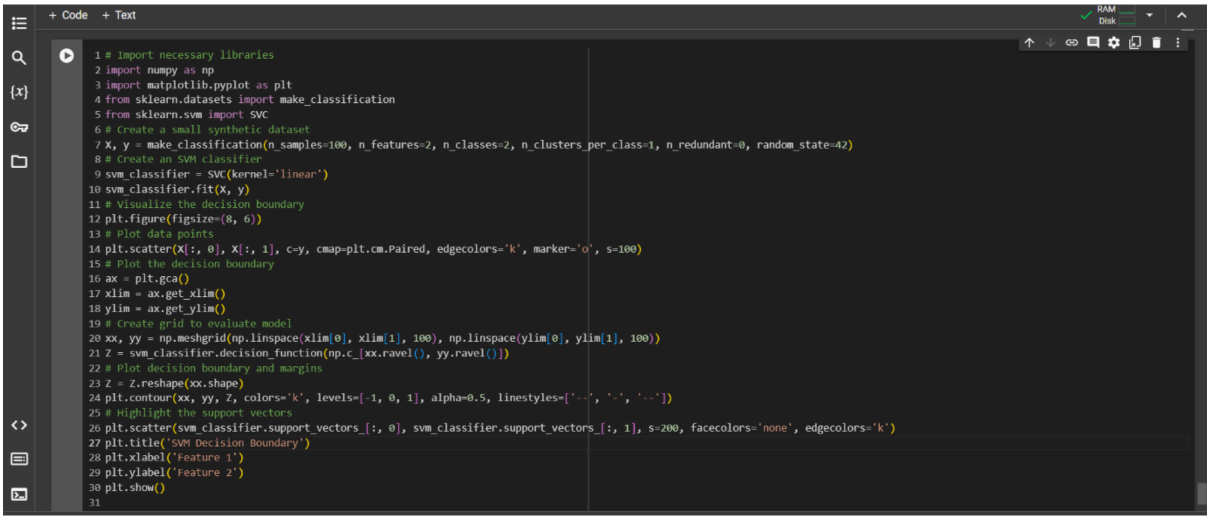
Name: M Jeyapathy

USN: 22BTRAD016

1. Implementing SVM algorithm

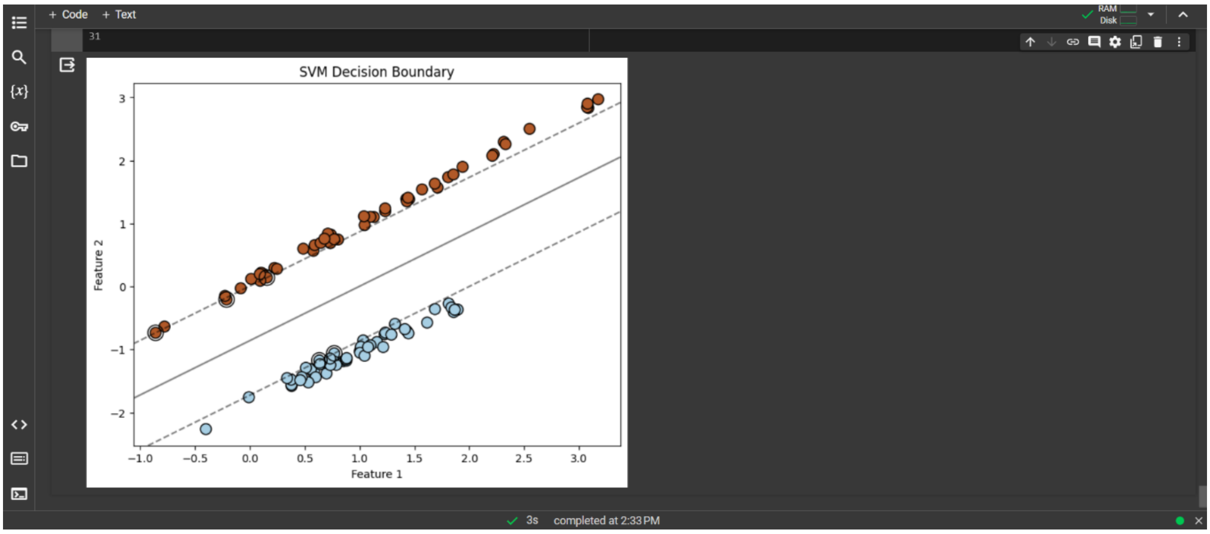
```
# Import necessary libraries
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_classification
from sklearn.svm import SVC

# Create a small synthetic dataset
X, y = make_classification(n_samples=100, n_features=2, n_classes=2,
n_clusters_per_class=1, n_redundant=0, random_state=42)

# Create an SVM classifier
svm_classifier = SVC(kernel='linear')
svm_classifier.fit(X, y)

# Visualize the decision boundary
plt.figure(figsize=(8, 6))

# Plot data points
plt.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.Paired, edgecolors='k', marker='o', s=100)

# Plot the decision boundary
ax = plt.gca()
xlim = ax.get_xlim()
```

```
ylim = ax.get_ylim()


# Create grid to evaluate model
xx, yy = np.meshgrid(np.linspace(xlim[0], xlim[1], 100), np.linspace(ylim[0], ylim[1], 100))
Z = svm_classifier.decision_function(np.c_[xx.ravel(), yy.ravel()])


# Plot decision boundary and margins
Z = Z.reshape(xx.shape)
plt.contour(xx, yy, Z, colors='k', levels=[-1, 0, 1], alpha=0.5, linestyles=['--', '-', '--'])


# Highlight the support vectors
plt.scatter(svm_classifier.support_vectors_[:, 0], svm_classifier.support_vectors_[:, 1], s=200, facecolors='none', edgecolors='k')


plt.title('SVM Decision Boundary')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.show()
```



```
# Import necessary libraries
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_classification
from sklearn.svm import SVC
# Create a small synthetic dataset
X, y = make_classification(n_samples=100, n_features=2, n_classes=2, n_clusters_per_class=1, n_redundant=0, random_state=42)
# Create an SVM classifier
svm_classifier = SVC(kernel='linear')
svm_classifier.fit(X, y)
# Visualize the decision boundary
plt.figure(figsize=(8, 6))
# Plot data points
plt.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.Paired, edgecolors='k', marker='o', s=100)
# Plot the decision boundary
ax = plt.gca()
xlim = ax.get_xlim()
ylim = ax.get_ylim()
# Create grid to evaluate model
xx, yy = np.meshgrid(np.linspace(xlim[0], xlim[1], 100), np.linspace(ylim[0], ylim[1], 100))
Z = svm_classifier.decision_function(np.c_[xx.ravel(), yy.ravel()])
# Plot decision boundary and margins
Z = Z.reshape(xx.shape)
plt.contour(xx, yy, Z, colors='k', levels=[-1, 0, 1], alpha=0.5, linestyles=['--', '-', '--'])
# Highlight the support vectors
plt.scatter(svm_classifier.support_vectors_[:, 0], svm_classifier.support_vectors_[:, 1], s=200, facecolors='none', edgecolors='k')
plt.title('SVM Decision Boundary')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.show()

```

Output:



3s completed at 2:33 PM

Github Link:

https://github.com/Jeyapathy/Machine-Learning