In [2]:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix, accuracy_score

# Load the digits dataset
digits = datasets.load_digits()
x, y = digits.data, digits.target
r,c = x.shape

#Dimensions of the dataset
print(f"Shape of dataset': {r} {c}")

# Split the dataset into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=42)

# Create and train an SVC classifier
classifier = SVC(kernel='linear')
classifier.fit(x_train, y_train)

# Make predictions on the test set
y_pred = classifier.predict(x_test)

# Evaluate the classifier
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))

#Accuracy
accuracy = accuracy_score(y_test, y_pred)
print("\nAccuracy: ", accuracy)

# Print the number of support vectors for each class
print("\nSupport Vectors:")
print(classifier.support_vectors_)

# Print the total number of support vectors
print(f"Total number of support vectors: {len(classifier.support_vectors_)}")
```

```
Shape of dataset': 1797 64
Confusion Matrix:
[[53  0  0  0  0  0  0  0  0  0]
 [ 0 49  0  0  0  0  0  0  1  0]
 [ 0  0 47  0  0  0  0  0  0  0]
 [ 0  0  1 52  0  1  0  0  0  0]
 [ 0  1  0  0 59  0  0  0  0  0]
 [ 0  0  0  0  0 64  0  1  0  1]
 [ 0  0  0  0  0  0 53  0  0  0]
 [ 0  0  0  0  0  0  0 54  0  1]
 [ 0  0  0  0  0  1  0  0 42  0]
 [ 0  0  0  0  1  0  0  1  1 56]]

Accuracy:  0.9796296296296296

Support Vectors:
[[ 0.  0.  3. ...  5.  0.  0.]
 [ 0.  0.  0. ...  6.  0.  0.]
 [ 0.  0.  2. ...  1.  0.  0.]
 ...
 [ 0.  0.  2. ... 14. 10.  1.]
 [ 0.  0.  0. ...  7.  0.  0.]
 [ 0.  0.  2. ... 10.  0.  0.]]
Total number of support vectors: 374
```

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js