A7: Use sci-kit-learn to implement a support vector machine classifier for binary and multiclass classification tasks.

In [1]:
```python
# Import necessary libraries
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
```

--

PART 1: Binary Classification (Breast Cancer Dataset)

--

In [2]:
```python
# Load the dataset
binary_data = datasets.load_breast_cancer()
X_binary = binary_data.data
y_binary = binary_data.target
```

In [3]:
```python
# Split dataset into training and testing sets
X_train_bin, X_test_bin, y_train_bin, y_test_bin = train_test_split(X_binary, y_binary, test_size=0.2, random_state=42)
```

In [4]:
```python
# Create and train the SVM classifier
svm_binary = SVC(kernel='linear', C=1.0, random_state=42)
svm_binary.fit(X_train_bin, y_train_bin)
```

Out[4]:

▾ SVC ⓘ ?

▸ Parameters

	C	1.0
	kernel	'linear'
	degree	3
	gamma	'scale'
	coef0	0.0
	shrinking	True
	probability	False
	tol	0.001
	cache_size	200
	class_weight	None
	verbose	False
	max_iter	-1
	decision_function_shape	'ovr'
	break_ties	False
	random_state	42

In [5]:

```
# Predict on test data
y_pred_bin = svm_binary.predict(X_test_bin)
```

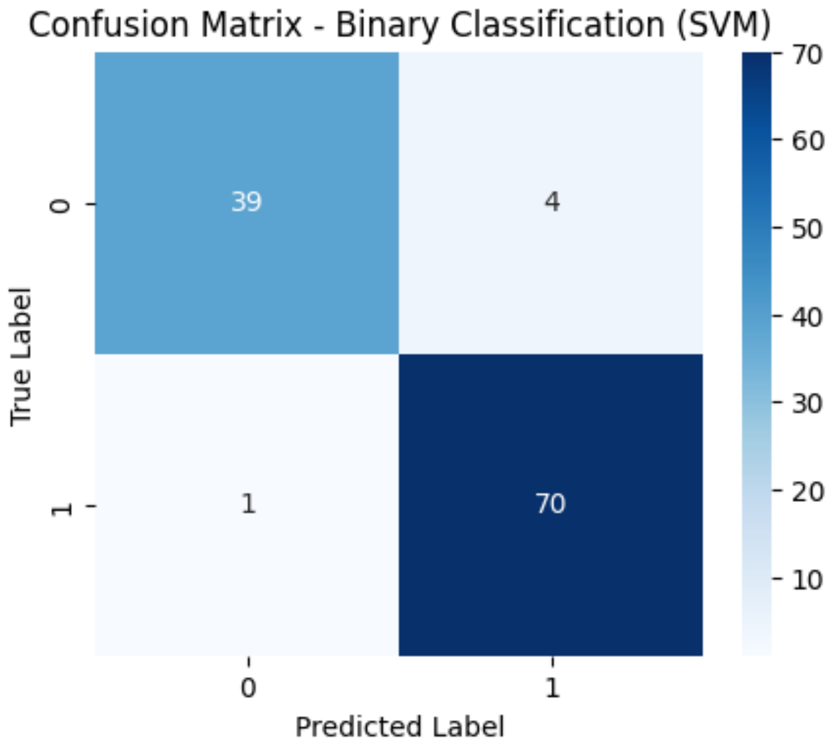
In [6]:

```
# Evaluate performance
print("🔹 Binary Classification (Breast Cancer Dataset)")
print("Accuracy:", accuracy_score(y_test_bin, y_pred_bin))
print("\nClassification Report:\n", classification_report(y_test_bin, y_pred_bin))
```

```
🔷 Binary Classification (Breast Cancer Dataset)
Accuracy: 0.956140350877193

Classification Report:
               precision    recall  f1-score   support

           0       0.97      0.91      0.94        43
           1       0.95      0.99      0.97        71

    accuracy                           0.96       114
   macro avg       0.96      0.95      0.95       114
weighted avg       0.96      0.96      0.96       114
```

In [7]:

```python
# Confusion Matrix Visualization
plt.figure(figsize=(5,4))
sns.heatmap(confusion_matrix(y_test_bin, y_pred_bin), annot=True, fmt='d', cmap='Blues')
plt.title("Confusion Matrix - Binary Classification (SVM)")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.show()
```



In [8]:

```
# -----------------------------------------
# PART 2: Multiclass Classification (Iris Dataset)
# -----------------------------------------
```

In [9]:

```
# Load the dataset
iris = datasets.load_iris()
X_multi = iris.data
y_multi = iris.target
```

In [10]:

```
# Split dataset
X_train_multi, X_test_multi, y_train_multi, y_test_multi = train_test_split(X_multi, y_multi, test_size=0.2, random_stat
```

In [11]:

```
# Create and train the SVM classifier for multiclass
# Scikit-learn automatically uses One-vs-Rest strategy for multiclass SVM
svm_multi = SVC(kernel='rbf', C=1.0, gamma='scale', random_state=42)
svm_multi.fit(X_train_multi, y_train_multi)
```

Out[11]:

▾ SVC ⓘ ?

▶ Parameters

	C	1.0
	kernel	'rbf'
	degree	3
	gamma	'scale'
	coef0	0.0
	shrinking	True
	probability	False
	tol	0.001
	cache_size	200
	class_weight	None
	verbose	False
	max_iter	-1
	decision_function_shape	'ovr'
	break_ties	False
	random_state	42

In [12]:

```
# Predict
y_pred_multi = svm_multi.predict(X_test_multi)
```

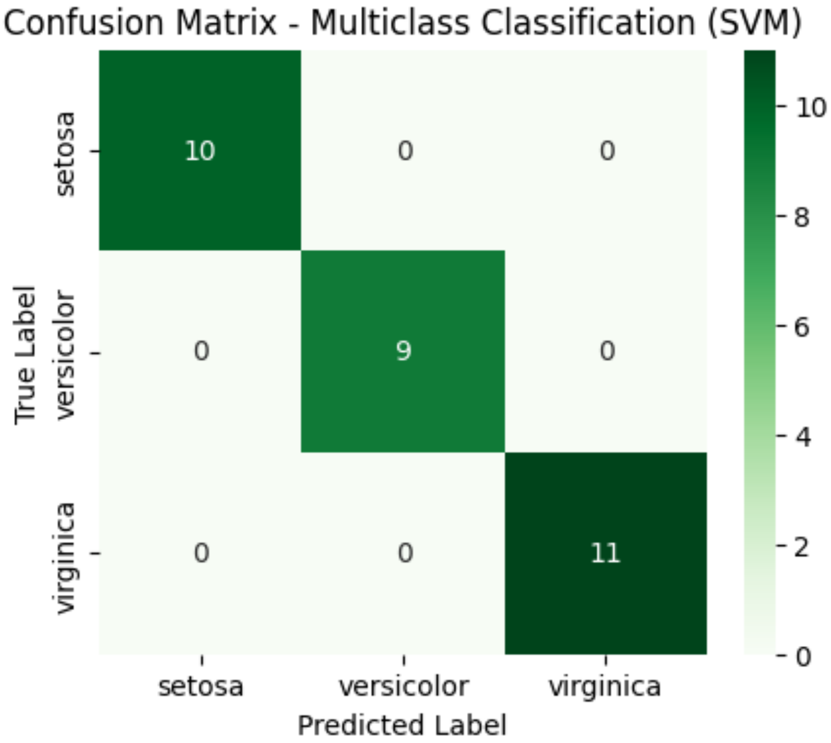
In [13]:

```
# Evaluate
print("\n🔷 Multiclass Classification (Iris Dataset)")
print("Accuracy:", accuracy_score(y_test_multi, y_pred_multi))
print("\nClassification Report:\n", classification_report(y_test_multi, y_pred_multi, target_names=iris.target_names))
```

```
🔷 Multiclass Classification (Iris Dataset)
Accuracy: 1.0

Classification Report:
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        10
  versicolor       1.00      1.00      1.00         9
   virginica       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```

In [14]:

```
# Confusion Matrix Visualization
plt.figure(figsize=(5,4))
sns.heatmap(confusion_matrix(y_test_multi, y_pred_multi), annot=True, fmt='d', cmap='Greens',
            xticklabels=iris.target_names, yticklabels=iris.target_names)
plt.title("Confusion Matrix - Multiclass Classification (SVM)")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.show()
```


Confusion Matrix - Multiclass Classification (SVM)
True Label
setosa
versicolor
virginica
10
0
0
0
9
0
0
0
11
setosa
versicolor
virginica
Predicted Label
10
8
6
4
2
0