svm-amirtha-ganesh-r

May 10, 2025

0.1 Name: Amirtha Ganesh R

0.2 SVM Lab assignment

0.3 reg No: 24MSD7035

[1]:
```
from sklearn.datasets import load_iris
import pandas as pd

# Load the dataset
iris = load_iris()

# Convert to DataFrame
df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['target'] = iris.target  # Classes: 0 = setosa, 1 = versicolor, 2 = virginica

df.head()
```

[1]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

[2]:
```
# for sum i'am using iris inbuilt dataset
```

1 Normal svm

[5]:
```
X = df.drop(columns = ['target'] )
y = df['target']
```

[6]:
```
from sklearn.datasets import load_iris
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt



# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)
```

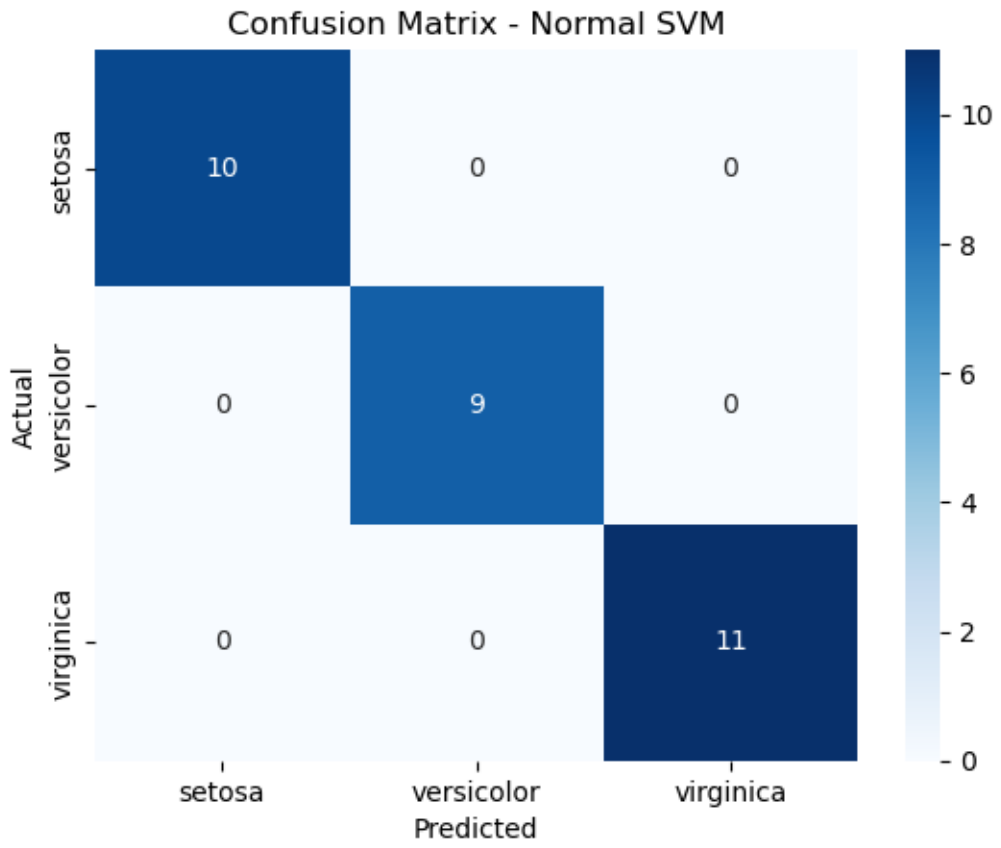
[7]:
```
# Train a basic SVM classifier
svm_model = SVC(kernel='linear')  # You can change kernel to 'rbf', 'poly', etc.
svm_model.fit(X_train, y_train)
```

[7]:
```
SVC(kernel='linear')
```

[8]:
```
# Make predictions
y_pred = svm_model.predict(X_test)

# Print classification report
print("  Normal SVM Classification Report:")
print(classification_report(y_test, y_pred, target_names=iris.target_names))

# Confusion matrix
cm = confusion_matrix(y_test, y_pred)
```

```
 Normal SVM 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
```

[9]:
```
# Visualize it
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=iris.target_names,
            yticklabels=iris.target_names)
plt.title("Confusion Matrix - Normal SVM")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```



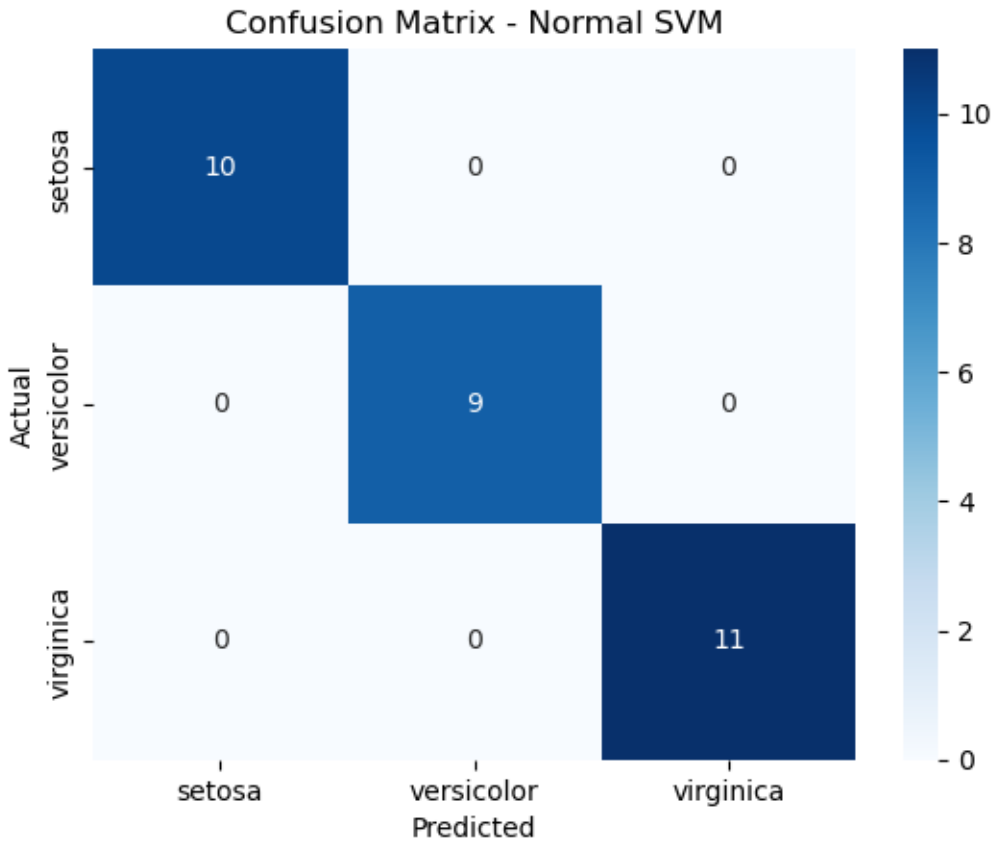
2 svm one vs all

[12]:
```
from sklearn.datasets import load_iris
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report

# Load data
iris = load_iris()
X, y = iris.data, iris.target
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# SVM with One-vs-Rest (OvR)
svm_ovr = SVC(decision_function_shape='ovr')  # default is ovr
svm_ovr.fit(X_train, y_train)
y_pred_ovr = svm_ovr.predict(X_test)

print(" SVM (One-vs-Rest) Classification Report:")
print(classification_report(y_test, y_pred_ovr, target_names=iris.target_names))
```

```
 SVM (One-vs-Rest) 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
```

[13]:
```
cm = confusion_matrix(y_test, y_pred_ovr)

# Visualize it
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=iris.target_names,
            yticklabels=iris.target_names)
plt.title("Confusion Matrix - Normal SVM")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```



3 SVM one vs one

[14]:
```
# SVM with One-vs-One (OvO)
svm_ovo = SVC(decision_function_shape='ovo')
svm_ovo.fit(X_train, y_train)
y_pred_ovo = svm_ovo.predict(X_test)

print(" SVM (One-vs-One) Classification Report:")
print(classification_report(y_test, y_pred_ovo, target_names=iris.target_names))
```

```
 SVM (One-vs-One) 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
```

[15]:
```
cm = confusion_matrix(y_test, y_pred_ovo)

# Visualize it
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=iris.target_names,
            yticklabels=iris.target_names)
plt.title("Confusion Matrix - Normal SVM")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```