

TP1 - Exercice 2.1 : SVM RBF sur Iris

Ce programme utilise un classifieur SVM avec un noyau RBF sur les deux premières caractéristiques de la base **Iris**. Nous analysons les performances avec une matrice de confusion et visualisons la frontière de décision.

Importation des bibliothèques

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.svm import SVC
from sklearn import datasets
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
```

Chargement de la base de données iris

```
iris = datasets.load_iris()
X = iris.data[:, :2] # On prend seulement les deux premières
caractéristiques pour la visualisation
y = iris.target
```

Normalisation des données

```
scaler = StandardScaler()
X = scaler.fit_transform(X)
```

Séparation en données d'entraînement et de test

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=0)
```

Entraînement du svm avec noyau rbf

```
svm = SVC(kernel='rbf', C=10, gamma=1)
svm.fit(X_train, y_train)

SVC(C=10, gamma=1)
```

Prédictions

```
y_pred = svm.predict(X_test)
```

Calcul de la matrice de confusion

```
cm = confusion_matrix(y_test, y_pred)

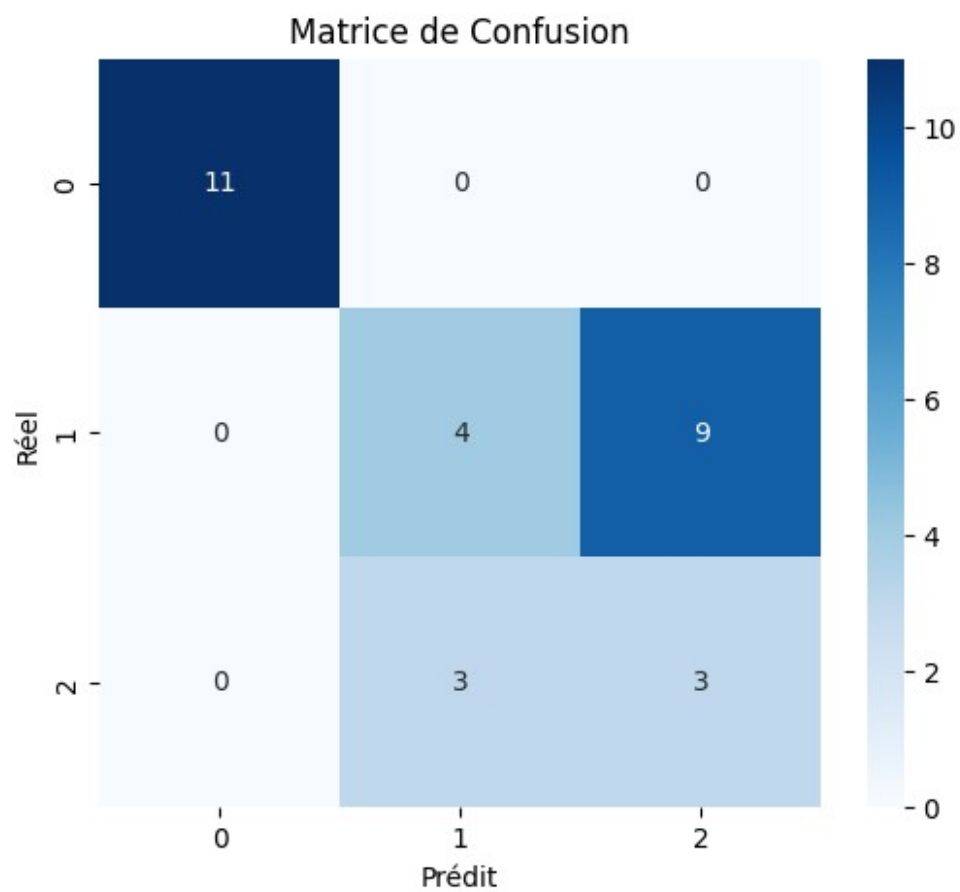
def afficher_matrice_confusion(cm, labels):
    """Affiche la matrice de confusion."""
    plt.figure(figsize=(6, 5))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=labels, yticklabels=labels)
    plt.xlabel('Prédit')
    plt.ylabel('Réel')
    plt.title('Matrice de Confusion')
    plt.show()

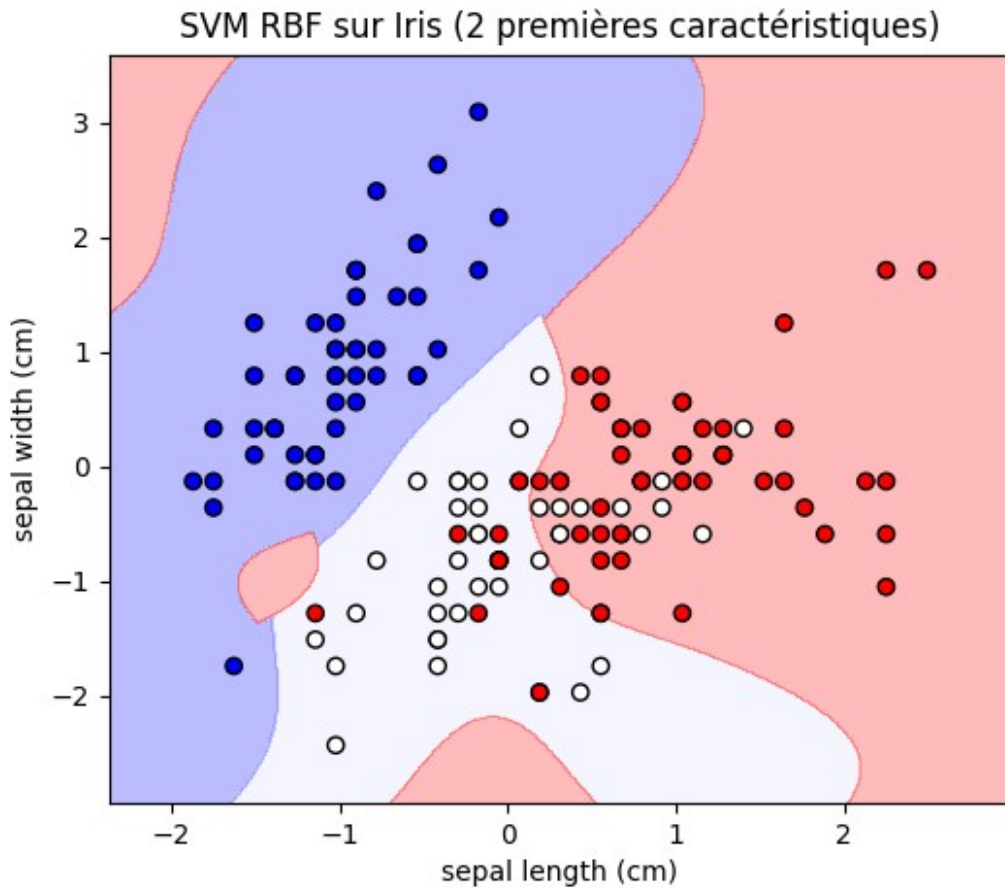
def afficher_frontiere_decision(X, y, model, feature_names):
    """Affiche la frontière de décision."""
    x_min, x_max = X[:, 0].min() - 0.5, X[:, 0].max() + 0.5
    y_min, y_max = X[:, 1].min() - 0.5, X[:, 1].max() + 0.5
    xx, yy = np.meshgrid(np.linspace(x_min, x_max, 500),
np.linspace(y_min, y_max, 500))
    Z = model.predict(np.c_[xx.ravel(), yy.ravel()]).reshape(xx.shape)

    plt.figure(figsize=(6, 5))
    plt.contourf(xx, yy, Z, alpha=0.3, cmap='bwr')
    plt.scatter(X[:, 0], X[:, 1], c=y, cmap='bwr', edgecolors='k')
    plt.xlabel(feature_names[0])
    plt.ylabel(feature_names[1])
    plt.title("SVM RBF sur Iris (2 premières caractéristiques)")
    plt.show()
```

Affichage séparé des résultats

```
afficher_matrice_confusion(cm, np.unique(y))
afficher_frontiere_decision(X, y, svm, iris.feature_names)
```





Analyse :

Le classifieur SVM avec noyau RBF est particulièrement adapté aux données non-linéaires. Ici, les deux premières caractéristiques de la base Iris sont utilisées, ce qui peut limiter la performance globale.

La matrice de confusion permet d'évaluer les erreurs de classification, et la frontière de décision aide à comprendre comment le modèle sépare les classes.