

svm_rbf_iris

March 21, 2025

Je n'ai eu le temps que de traiter l'arbre de décision.

0.1 Importation des bibliothèques

```
[1]: 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
```

0.2 Chargement de la base de données IRIS

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

0.3 Normalisation des données

```
[3]: scaler = StandardScaler()
X = scaler.fit_transform(X)
```

0.4 Séparation en train/test

```
[4]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
    ↪ random_state=0)
```

0.5 Entraînement du SVM avec noyau RBF

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

```
[5]: SVC(C=10, gamma=1)
```

0.6 Prédiction

```
[6]: y_pred = svm.predict(X_test)
```

0.7 Calcul de la matrice de confusion

```
[7]: cm = confusion_matrix(y_test, y_pred)

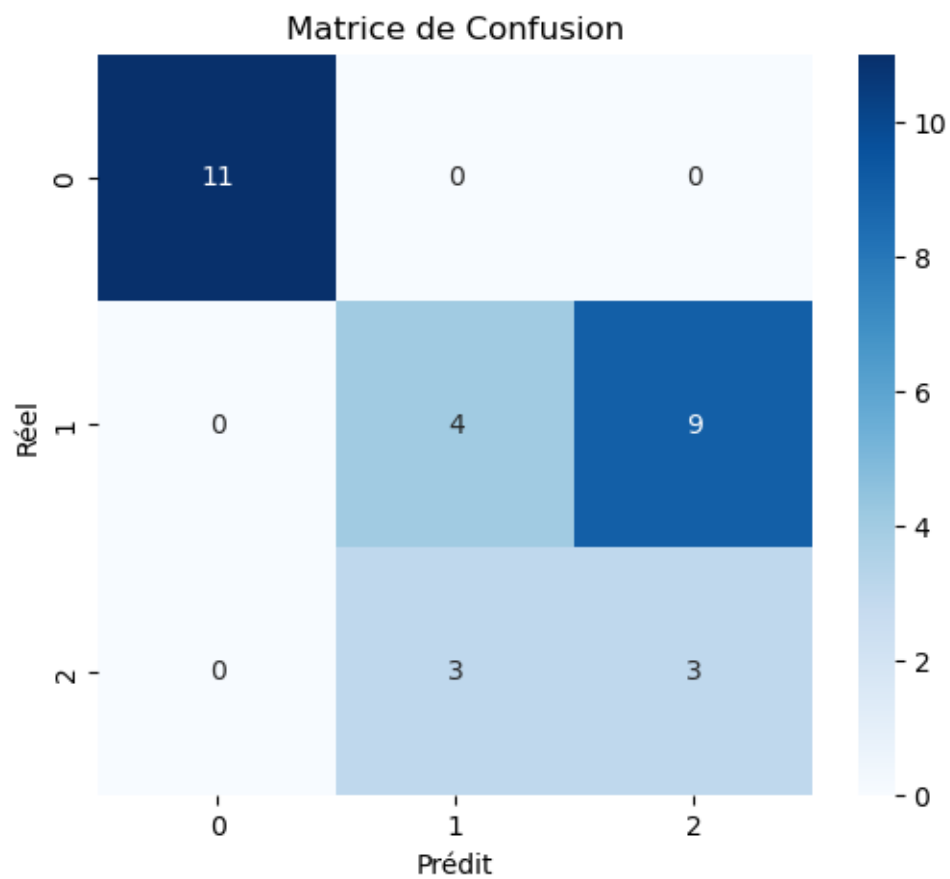
def afficher_matrice_confusion(cm, labels):
    """Affiche la matrice de confusion."""
    plt.figure(figsize=(6, 5))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=labels,
    yticklabels=labels)
    plt.xlabel('Prédit')
    plt.ylabel('Réel')
    plt.title('Matrice de Confusion')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

0.8 Affichage séparé des résultats

```
[8]: afficher_matrice_confusion(cm, np.unique(y))
    afficher_frontiere_decision(X, y, svm, iris.feature_names)
```



SVM RBF sur Iris (2 premières caractéristiques)

