Exercices



Exercice 6: Classification des iris avec SVM

- 1. Charger la base de données IRIS du fichier « IRIS.csv ».
- 2. Afficher l'en tête de la base de données.
- 3. Tracer des relations par paires dans l'ensemble de données en utilisant la fonction **pairplot()** de la bibliothèque seaborn. Interpréter les résultats.
- 4. Sélectionner les quatre premières colonnes pour les données et la dernière colonne pour les labels. Puis séparer les données et les labels en 70% en données d'apprentissage et 30% en données de test.

Les SVM à noyaux sont implémentées dans Scikit-learn dans les classes **svm.SVC** pour la classification et **svm.SVR** pour la régression. Dans ces deux classes, on peut spécifier un noyau grâce au paramètre « kernel ». Ce noyau peut être un des grands classiques (linéaire, polynomial, RBF).

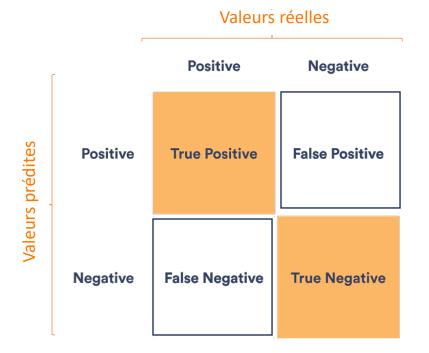
- 5. Importer et instancier le modèle SVC pour créer un SVM avec un noyau linéaire, puis entraîner le modèle.
- 6. Calculer le vecteur des prédictions en utilisant la fonction predict() sur les données du test.
- 7. Calculer la justesse (accuracy) en utilisant la fonction score().
- 8. Calculer la matrice de confusion en utilisant sklearn.metrics.





Exercice 6: Classification des iris avec SVM

Sachant que la matrice de confusion est sous cette forme :



La diagonale représente les prédictions correctes.

Exercices



Exercice 6: Classification des iris avec SVM

- 9. Calculer le rapport de classification.
- 10. Importer puis instancier le modèle SVC pour créer un SVM avec un noyau gaussien (RBF kernel et gamma=0.001)
- 11. Entraîner ce modèle puis prédire les résultats.
- 12. Calculer la matrice de confusion.
- 13. Calculer le rapport de classification.

Il faut mettre en œuvre de bonnes pratiques de sélection du modèle (par exemple une validation croisée) pour choisir la meilleure valeur de C et gamma.

14. Utiliser une validation croisée sur l'ensemble d'entraînement pour sélectionner les valeurs optimales de C et de gamma parmi une grille de valeurs.





Exercice 6: Classification des iris avec SVM

15. Exécuter ce bout de code pour calculer le noyau rbf (gaussien) sur les données d'apprentissage :

```
from sklearn import metrics
import matplotlib.pyplot as plt
kmatrix = metrics.pairwise.rbf_kernel(x_train, gamma=0.1)
plt.pcolor(kmatrix, cmap=plt.cm.PuRd)
plt.colorbar()
plt.gca().invert_yaxis()
plt.gca().xaxis.tick_top()
plt.show()
```

- 16. Utiliser le package joblib pour enregistrer le modèle de la classification dans le fichier « iris_model.h5 » afin de pouvoir le réutiliser sans effectuer l'entraînement à chaque fois.
- 17. Charger le modèle SVM qui a été sauvegardé dans la phase d'entraînement.
- 18. Prédire la classe des nouvelles données.