Rapport PCA & SVM

Introduction

Contexte & Objectifs

Durant ce premier TP, nous nous sommes familiarisé avec l'analyse en composantes principales (Principal Component Analysis, ou PCA) et les machines à vecteurs de support (Support Vector Machines, SVM).

L'objectif principal est de classer des données botaniques associées à des espèces de fleurs, en utilisant un apprentissage supervisé.
Pour ce faire, nous entrainerons un modèle avec des données se trouvant dans le fichier `flowerTrain.data`. Se trouve dans ce fichier un set de 48 échantillons pour chacune des trois espèces de fleurs : *interior*, *versicolor*, et *convoluta*.
Quatre caractéristiques ont été mesurées pour chaque échantillon : la *longueur* et la *largeur* des **sépales**, ainsi que la *longueur* et la *largeur* des **pétales**.
En se basant sur la combinaison de ces quatre caractéristiques, nous allons développer un modèle qui pourra déterminer si un nouvel échantillon est une fleur de type interior ou non (c'est-à-dire, appartenant à l'une des deux autres espèces).
Une fois le modèle obtenu, nous devrons déterminer l'espèce des fleurs du fichier `new_flowerData.data`.

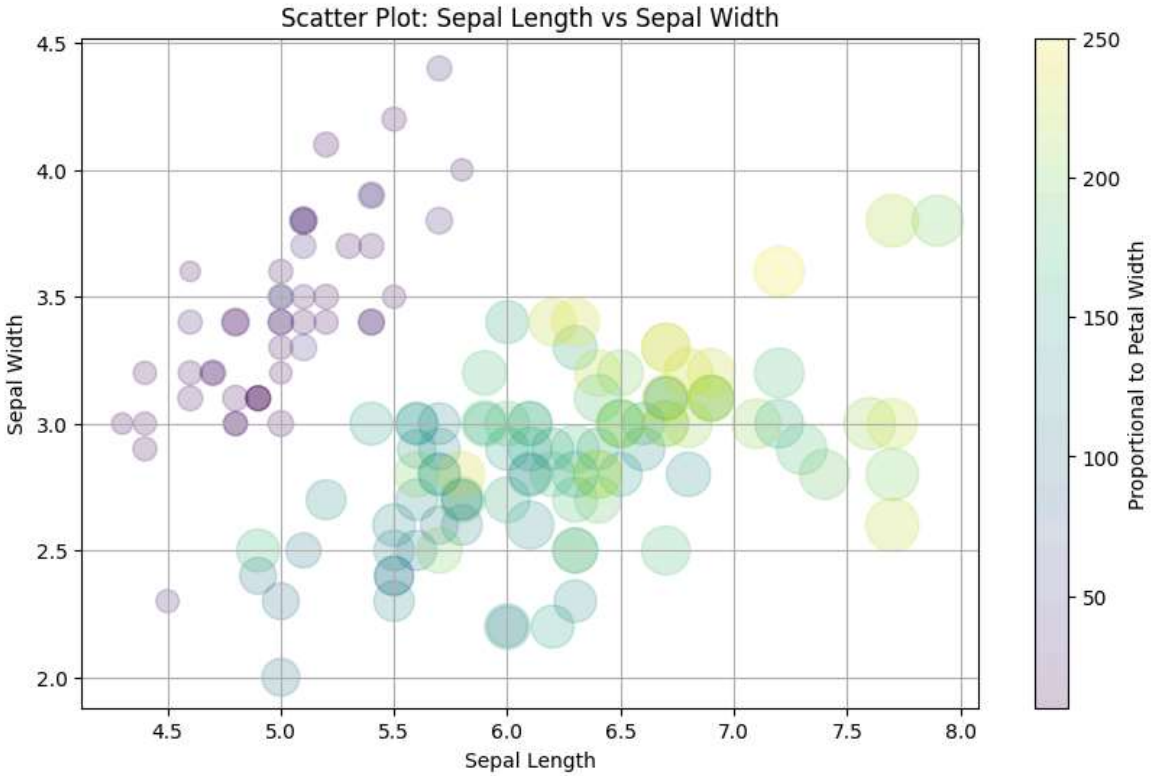
Dans ce TP, nous suivrons les étapes suivantes :

1. Comme entrainement, nous allons tout d'abord écrire puis tester une fonction PCA sur des ensembles de données en 2 dimensions (avec 4 points) afin de les réduire à 1 dimension.
2. Utiliser la fonction PCA sur le jeu de données d'entrainement (4 dimensions) pour le réduire à 2 dimensions.
3. Entrainer une machine à vecteurs de support (SVM) avec les données d'entrainement dont on a réduit les dimensions pour classifier les échantillons comme étant « interior » ou « non interior ».
4. Tester le modèle sur de nouveaux échantillons.

Enfin, ce rapport se conclura sur la présentation de nos résultats.

Présentation des données

Les données traitées dans ce TP sont présentes dans le fichier `flowerTrain.data`.



Ce plot permet d'afficher les 4 dimensions des données de chaque fleur sur un seul plot : en abscisse et ordonnée, on retrouve les dimensions du **sépal** tandis que la taille et la couleur des points indiquent les dimensions du **pétals**.

I. L’analyse en composantes principales (AKA PCA)

Définitions & Utilité

Le PCA, pour Analyse en Composantes Principales, est une méthode statistique de réduction de dimension qui permet de réduire les dimensions d'un jeu de données à plusieurs dimensions, tout en préservant au maximum l’information présente dans les données.
Cette méthode est utilisée lorsque les données présentent un grand nombre de variables, rendant l’analyse et l’apprentissage difficile et long.

Dans le cadre de notre TP, nous avons les longueurs et largeurs des pétales et des sépales, soit 4 dimensions (données). Nous allons passer la dimension de ces données à 2.

Etapes

L'Analyse suit les étapes suivantes :

1. Standardisation des données

La standardisation des données se déroulent en deux étapes : le calcul de la moyenne des données, puis le centrage des données.

- Moyenne des données:
  Pour une matrice de données X (n échantillons et d caractéristiques), cette étape calcule la moyenne de chaque colonne $\bar{X}_j$:

$$\bar{X}_j = \frac{1}{n}\sum_{i=1}^{n} X_{ij}$$

  où $\bar{X}_j$ est la moyenne de la j-ième caractéristique.
- Centrage des données
  On centre les données en soustrayant la moyenne de chaque caractéristique :

$$X'_{ij} = X_{ij} - \bar{X}_j$$

  Cela garantit que chaque caractéristique est centrée autour de 0.

Cette première étape sert à s'assurer que les variables ayant de grandes valeurs absolues n'éclipsent pas les autres, ainsi qu'à préparer les données pour une analyse basée sur la variance.

2. Calcul de la matrice de covariance

La matrice de covariance C est définie comme :

$$C = \frac{1}{n-1} X'^{T} X'$$

- X' est la matrice centrée.
- C est une matrice symétrique de dimension $d \times d$, où chaque élément C_{ij} représente la covariance entre les caractéristiques i et j.

Cette seconde étape permet d'identifier les relations linéaires entre les caractéristiques.

3. Calcul des valeurs et des vecteurs propres

On effectue une décomposition propre de la matrice de covariance C :

$$Cv_i = \lambda_i v_i$$

- v_i : vecteur propre correspondant à la i-ième composante principale.
- λ_i : valeur propre indiquant la quantité de variance expliquée par v_i.

Le calcul des valeurs propres permet de comprendre quelles caractéristiques dominent dans la structure des données. Le calcul des vecteurs propres permet quand à lui d'identifier les directions principales maximisant la variance des données.

4. Sélection des p premières composantes principales

On sélectionne les p premiers vecteurs propres correspondant aux plus grandes valeurs propres. Cela permet de réduire la dimensionnalité à p dimension(s).

Cette étape permet de choisir un sous-espace plus simple tout en maintenant la structure des données.

5. Projection des données

Les données centrées X' sont projetées sur les p composantes principales :

$$Z = X' \cdot W$$

- W : matrice contenant les p premiers vecteurs propres ($d \times p$).
- Z : données projetées ($n \times p$).

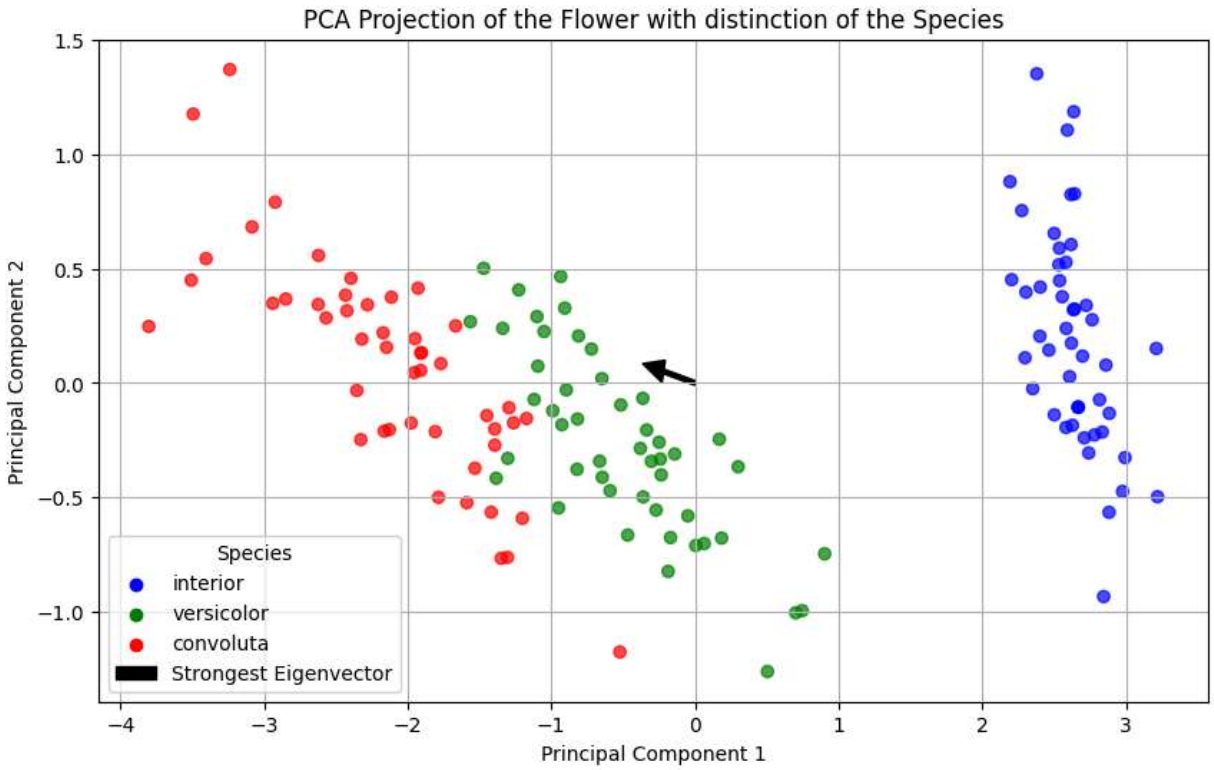
Enfin, cette étape finale permet d'obtenir la représentation des données avec la réduction de dimension.

Implémentation

Voir la fonction ***PCA*** du fichier TP1.ipynb

Résultat

Une fois le PCA implémenté et sa prise en main acquise, nous l'avons utilisé pour réduire les dimensions des fleurs de 4 à 2.
Sur les données d'entrainement, nous obtenons ce plot :



Comparé au plot plus tôt en 4 dimensions, ce plot des données à dimensions réduites permet de clairement identifier les différentes classes de données.

Ainsi, les fleurs de l'espèce *Interior* se trouvent à droite du plot tandis que les fleurs des deux autres espèces (*Versicolor* et *Convoluta*) se trouvent à la gauche. Ces deux dernières ont des valeurs beaucoup trop similaire pour que l'on espère pouvoir en dégager deux classes distinctes, c'est pourquoi nous les considèreront comme faisant partie d'une même classe : *non-Interior*.

II. Support Vector Machine (SVM)

Définitions & Utilité

Le **Support Vector Machine (SVM)** est une méthode d'apprentissage supervisé utilisée pour résoudre des problèmes de classification et de régression. L'objectif principal d'un SVM est de séparer des données en classes distinctes à l'aide d'un hyperplan optimal qui sera déterminé.

Dans le cadre de notre TP, le SVM sera utilisé pour classer les échantillons projetés par le PCA.
Cela nous permettra de différencier les espèces de fleurs à l'aide des données, et nous permettra de déterminer par la suite à quel classe appartient un échantillon sans étiquette.

Etapes de calcul

Une fois les dimensions des données d'entrainement réduites par le PCA (pour rappel, nous passons de 4 dimensions à 2), nous allons entrainer un SVM linéaire sur ces dernières.
Le modèle a été optimisé pour maximiser la séparation entre les classes tout en utilisant les vecteurs supports comme base de la frontière de décision.

1. Définition de l'hyperplan

Un hyperplan est une frontière qui sépare les classes dans un espace de caractéristiques. En 2 dimensions, un hyperplan correspond à une ligne définie par l'équation suivante :

$$w^T x - b = 0$$

où :

- w est le vecteur des coefficients (ou poids) qui définit la direction de l'hyperplan,
- b est le biais, ajustant la position de l'hyperplan,
- x est le vecteur de données.

2. Maximisation de la marge

La marge est la distance entre l'hyperplan et les points de données les plus proches, appelés **vecteurs supports**. Un SVM cherche à maximiser cette marge pour une séparation optimale.

Pour chaque point x_i, appartenant à une classe $y_i \in \{-1, +1\}$, la contrainte de classification est donnée par :

$$y_i(w^T x_i - b) \geq 1$$

Cela garantit que les points des deux classes sont correctement séparés.

3. Fonction objectif

Pour maximiser la marge, le SVM minimise la norme $||w||^2$ tout en respectant les contraintes de classification. Cela revient à résoudre le problème d'optimisation suivant :

$$\min_{w,b} \frac{1}{2}||w||^2$$

4. Données non linéaires et noyaux

Dans le cas où les données ne sont pas linéairement séparables, le SVM utilise une transformation $\phi(x)$ pour projeter les données dans un espace de dimension supérieure, où elles deviennent séparables. La séparation est alors définie dans cet espace projeté. Cette transformation est effectuée implicitement grâce à une fonction noyau $K(x_i, x_j)$, comme :

- **Noyau linéaire** :

$$K(x_i, x_j) = x_i^T x_j$$

- **Noyau RBF (gaussien)** :

$$K(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2)$$

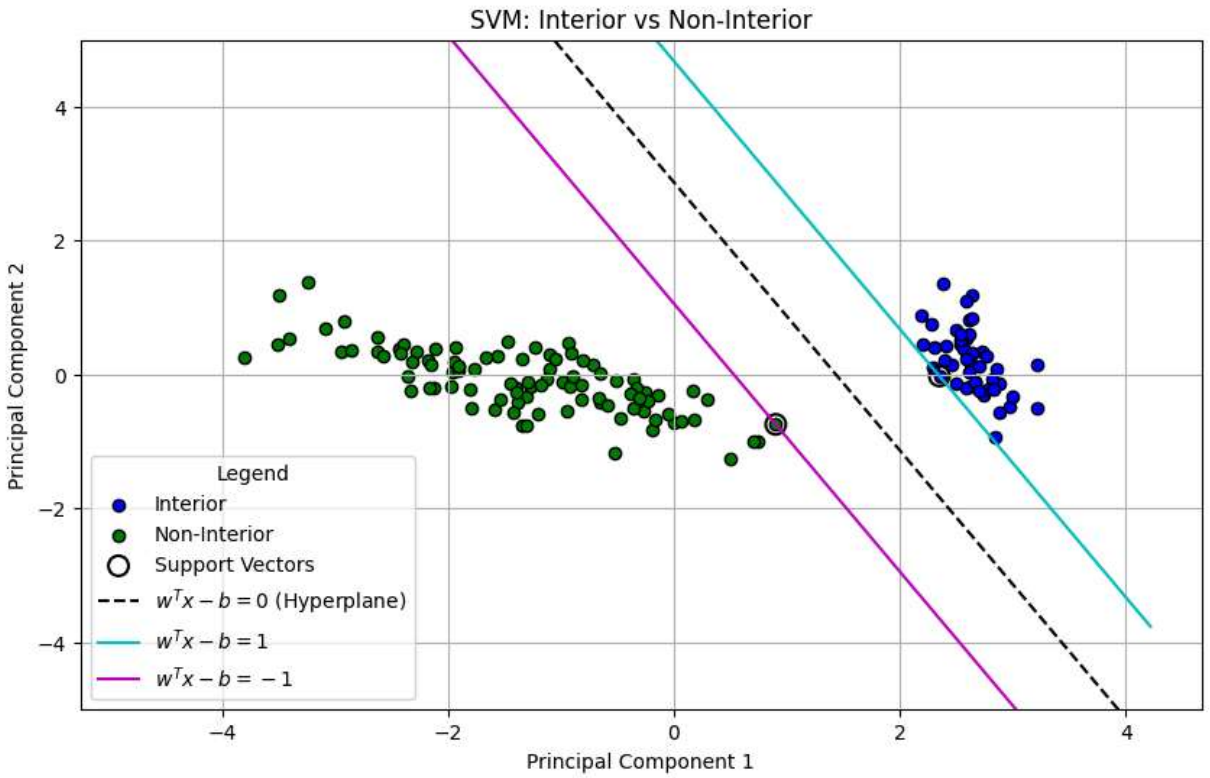
Implémentation

Nous avons utilisé le bibliothèque *sklearn* pour programmer la SVM.

Nous avons utilisé comme argument de la SVM `kernel='linear', C=1E10` .

- `kernel='linear'` : on souhaite que le SVM utilise un hyperplan linéaire pour séparer les classes.
- `C=1E10` : le paramètre C de la fonction SVM de **seaborn** contrôle le compromis entre la maximisation de la marge maximale entre les classes et la minimisation des erreurs de classification. On demande un C égal à 10 milliards, ce qui veut dire que le modèle va fortement pénaliser les erreurs de classification.

Résultat



Sur le graphique, deux classes distinctes se dégagent : la classe des fleurs *Interior* et celle des *Non-Interior*. Tous les points situés à gauche de l'hyperplan sont classés comme appartenant à la classe *Non-Interior*, tandis que ceux situés à droite sont attribués à la classe *Interior*. La classe *Non-Interior* comporte les fleurs des espèces *Versicolor* et *Convoluta*. Les points de ces deux espèces créant un cluster groupé, il est trop difficile de trouver un hyperplan qui pourrait les séparer en réduisant les erreurs de classification.

Le compromis entre les erreurs de classification et le nombre de classes disponibles est un aspect central ici.
Dans notre cas, il aurait été trop complexe de séparer correctement les fleurs des labels Versicolor et Convoluta. Nous avons donc privilégié une approche qui réduit les erreurs de classification en regroupant ces fleurs dans des classes plus larges.

III. Classées des données sans étiquettes

Introduction & Etapes

Maintenant que nous avons un modèle prédictif entrainé, nous pouvons l'utiliser pour categoriser des données sans étiquettes.
Il nous a été fourni un fichier `new_flowerData.data` , un fichier qui contient les *longueur* et *largeur* de **pétals** et **sépals** de fleur. Leurs espèces ne nous ont cependant pas été donné.
A l'aide des fonctions et du modèle utilisé dans les deux première parties du TP, nous allons classer ces données.

Pour classer ces données, nous allons réaliser les étapes suivantes dans cet ordre :

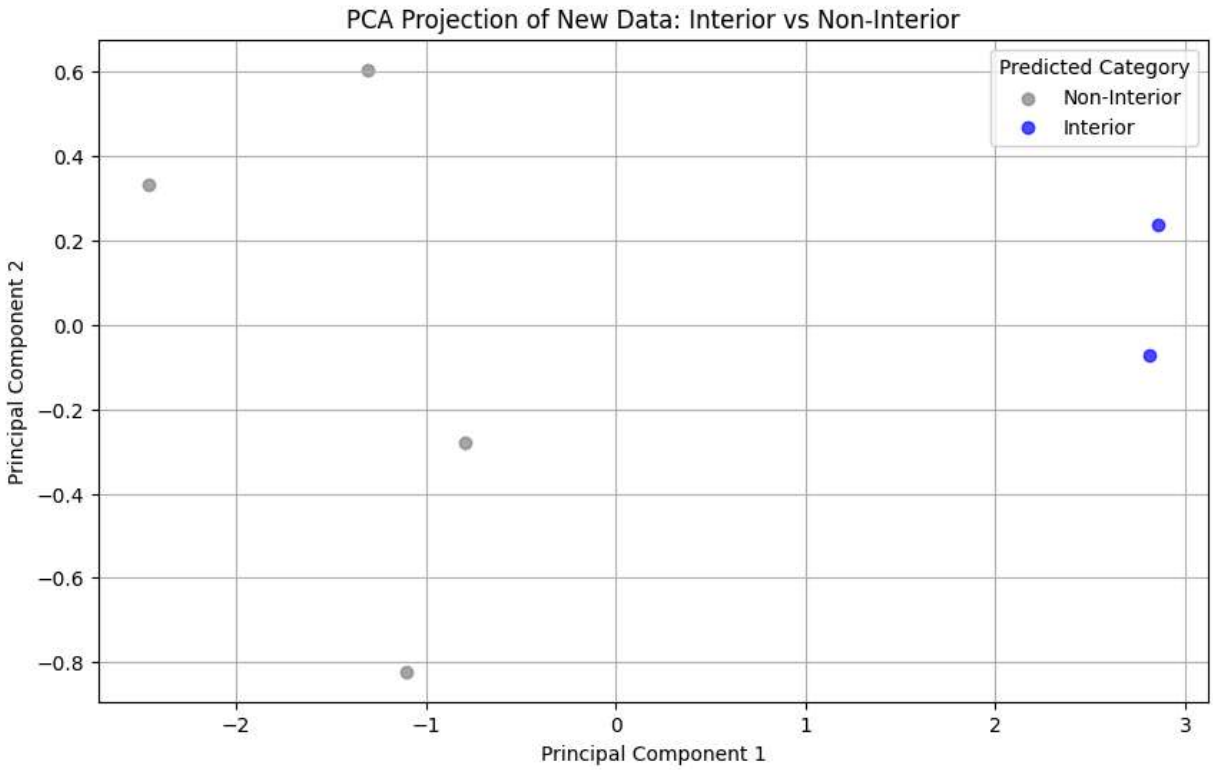
1. **Projection avec le PCA** :
   Les nouvelles données sont projetées dans un espace réduit à 2 dimensions, de la même manière que les données d'entraînement.
2. **Prédiction avec le SVM** :
   On utilise le modèle SVM entrainé avec les données d'entrainement projetées pour prédire l'espèce de chaque fleur.

Résultat

Les prédictions du modèle SVM pour les nouvelles données sont présentées ci-dessous :

Échantillon	Prédiction
1	interior
2	non-interior
3	non-interior
4	non-interior
5	non-interior
6	interior

Sous forme de plot, nous obtenons ceci :



On peut voir une ressemblance avec le plot précédent du modèle SVM avec les données d'entrainement : les fleurs d'intérieurs sont à droite du plot tandis que les fleurs d'extérieurs sont à gauche.

Conclusion

En conclusion de ce TP, notre modèle linéaire fonctionne.
Notre modèle est capable de prédire si une fleur appartient à l'espèce *Interior* ou *non-Interior*. Il n'est cependant pas possible de prédire si une feuille *non-Interior* est de l'espère *Versicolor* ou *Convoluta* sans risquer des erreurs de classification : la marge entre ces deux classes est trop fine.