STATISTIQUE
SCIENCE DES DONNÉES
UNIVERSITÉ DE MONTPELLIER

Master 2 Statistiques et Sciences des Données

Travaux Pratique
Support Vector Machines

LAHJIOUJ Aïcha

Année 2024 - 2025

Les Support Vector Machines (SVM) sont des algorithmes d'apprentissage supervisé utilisés principalement pour la classification. Leur principe fondamental repose sur la recherche d'un hyperplan qui sépare les différentes classes de données, tout en maximisant la marge entre elles. Cette approche permet de garantir une meilleure généralisation sur de nouvelles observations.

Dans un premier temps, nous allons nous intéresser au jeu de données Iris mise à disposition sur Python. Il contient 150 échantillons de fleurs d'iris, répartis en trois espèces : Iris setosa, Iris versicolor et Iris virginica. Chaque échantillon est décrit par quatre caractéristiques :

- Longueur du sépale
- Largeur du sépale
- Longueur du pétale
- Largeur du pétale

Nous allons classifier la classe 1 contre la classe 2 du data set iris en utilisant les deux premières variables et un noyau linéaire, en laissant la moitié des données de côté.

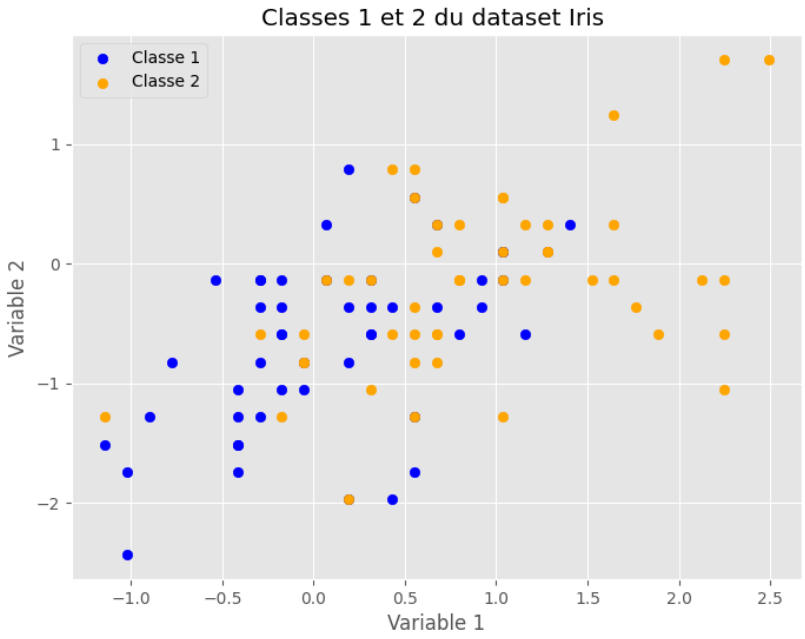


Figure 1: Répartition des différentes espèces de fleurs selon les deux premières variables

Notre objectif est d'évaluer la capacité de généralisation du modèle.

1 Question 1

Synthèse du Modèle SVM avec validation croisée

1. **Mélange des données :** Les données sont mélangées aléatoirement pour éviter les biais liés à un ordre spécifique.
2. **Division des ensembles :** Les données sont séparées en deux ensembles :
   - Entraînement (50%)
   - Test (50%)
3. **Paramètres :** Le paramètre de régularisation C varie de 10^{-3} à 10^{3} sur une échelle logarithmique.
4. **Optimisation :** Les paramètres du modèle SVM sont optimisés via validation croisée, en divisant l'ensemble d'entraînement en cinq sous-ensembles pour cinq itérations d'entraînement et validation.
5. **Entraînement et évaluation :** Le modèle est entraîné sur les données d'entraînement avec les paramètres optimaux, puis évalué :
   - Score moyen d'entraînement (10 itérations) : 0.75 (indiquant une bonne performance sur les données d'entraînement).
   - Score moyen de test (10 itérations) : 0.67 (inférieur au score d'entraînement, suggérant une capacité de généralisation limitée).
6. **Analyse des performances :** La différence entre les scores d'entraînement et de test peut indiquer un léger sur-apprentissage, signifiant que le modèle s'adapte trop aux spécificités des données d'entraînement.

Nous allons maintenant observer la frontière de décision :

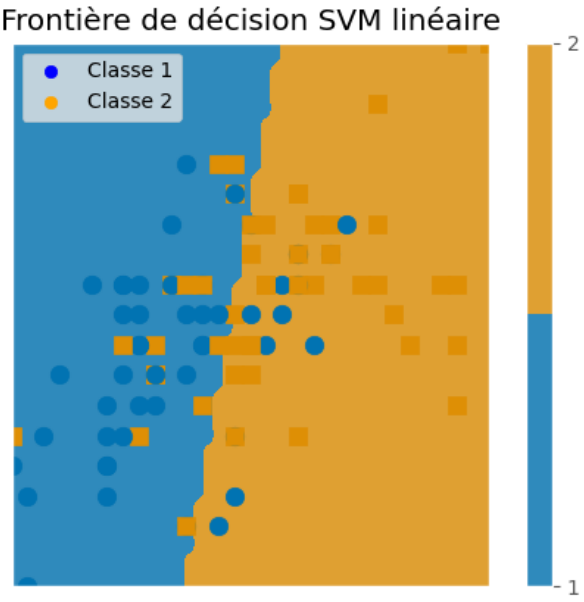


Figure 2: Frontière de décision SVM linéaire

La distribution des points montre que certains points de Classe 1 apparaissent dans la région de Classe 2 et vice-versa, ce qui indique que le modèle peut avoir des difficultés à correctement séparer les deux classes dans ces zones.

Cette tendance est confirmée par la matrice de confusion ci-dessous, avec un taux d'erreur de 44% .

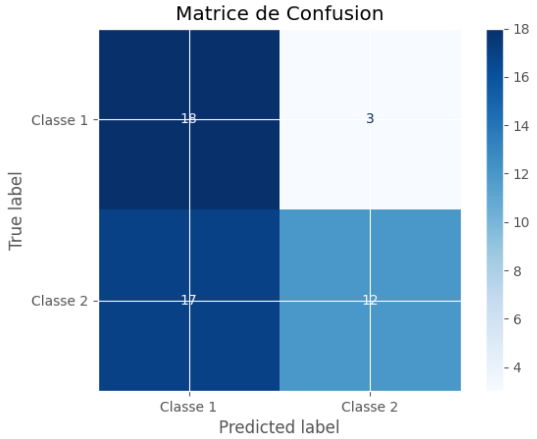


Figure 3: Matrice de Confusion

Les hyper-paramètres sont sélectionnés par validation croisée à l'aide la fonction `GridSearchCV`. Nous allons maintenant nous intéresser au cas sans la validation croisée, à l'aide de seulement la fonction SVC et un noyau linéaire. Nous obtenons un score d'entrainement moyen (10 itérations) de 0.75, et un score de test moyen (10 itérations) de 0.58.

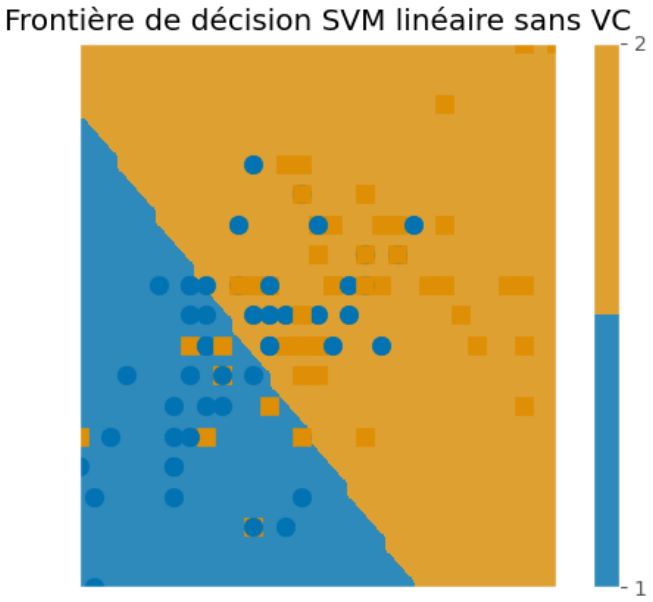


Figure 4: Frontière de décisions avec SVM linéaire sans Validation Croisée

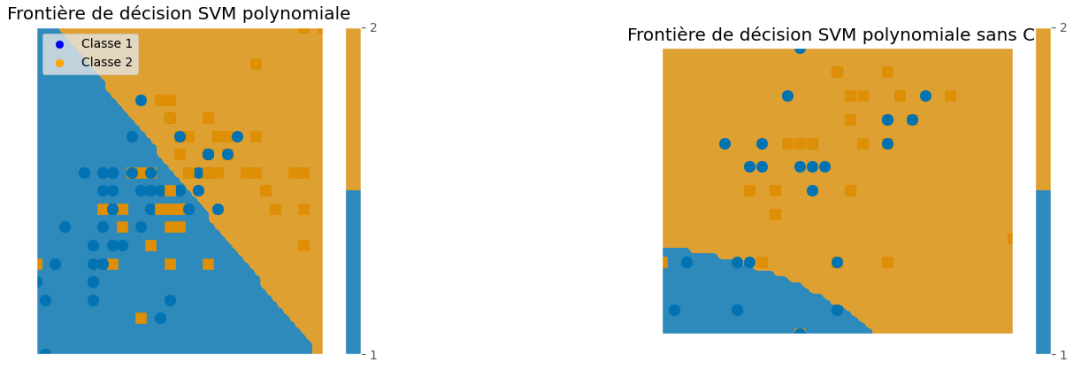
Nous observons que le score d'entraînement reste constant, tandis que le score de test est plus élevé lors de la validation croisée. Cela s'explique par une optimisation des paramètres obtenue grâce à cette méthode.

Néanmoins, l'erreur reste d'environ 0,4, ce qui indique un taux d'erreur relativement élevé. De plus, l'écart entre le score sur les données d'entrainement et de test est assez important, de l'ordre 0.15, ce qui un signe de sur-apprentissage. Nous pouvons donc en conclure que le modèle linéaire n'est pas l'idéal pour nos données, et qu'il faudrait penser à une autre méthode afin d'améliorer les résultats.

2 Question 2

Nous allons maintenant nous intéresser au noyau polynomial.

Nous allons tout d'abord optimiser les paramètres à l'aide de la validation croisée, puis sans validation croisée. Nous obtenons les frontières de décisions suivantes :



(a) Frontière de décision SVM polynomiale

(b) Sans Validation Croisée

Figure 5: Comparaison des frontières de décision

Nous allons maintenant comparer les scores d'entrainement et de test, avec et sans validation croisée.

	Score d'entraînement	**Score de test**
Avec Validation Croisée	0.70	0.67
Sans Validation Croisée	0.62	0.60

Table 1: Scores avec et sans validation croisée

L'écart entre les scores d'entraînement et de test a été réduit, mais demeure en faveur du score d'entraînement, ce qui est tout à fait logique. Le modèle performe mieux sur les données sur lesquelles il a été entraîné. En revanche, une différence significative apparaît entre le score d'entraînement avec et sans validation croisée.

Lorsque nous appliquons la validation croisée avec un noyau polynomial, il est intéressant de noter que, dans la plupart des cas, le degré du polynôme sélectionné est de 1. Cela signifie que, malgré notre choix initial d'un noyau polynomial pour modéliser des relations plus complexes, le modèle opte souvent pour une solution linéaire.

Cela dit, certaines frontières générées avec le noyau polynomial ont effectivement l'apparence d'un polynôme.

Avec la validation croisée, nous observons une légère amélioration du score, ce qui suggère l'existence de relations non linéaires, bien que celles-ci minoritaire.

Nous constatons que la méthode du noyau polynomiale n'apporte pas d'avantage significatif par rapport au noyau linéaire, ce qui laisse présagé qu'une autre méthode comme celle du noyau gaussien, peut-être mieux adaptées aux données.

3 Question 3 (Bonus)

L'un des paramètres les plus importants de l'algorithme SVM est le coefficient C . Afin d'étudier son influence, nous allons utiliser une application interactive permettent de comprendre l'impact des différents paramètres des SVM. Pour cela, nous allons lancer le script `svm_gui.py` mis à notre disposition.

Le paramètre de régulation C détermine le niveau de tolérance aux erreurs. Il influence ainsi la flexibilité ou la rigidité avec laquelle le SVM construit l'hyperplan de séparation entre les classes.

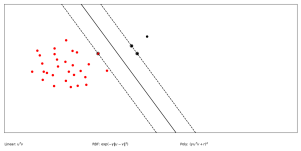


Figure 6: C=1

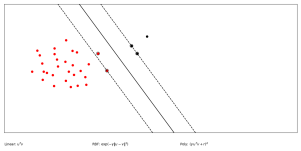


Figure 7: C=0.01

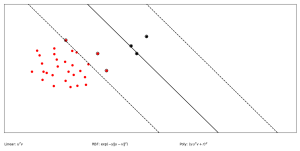


Figure 8: C=0.001

Ce paramètre a également un impact sur la taille de la fenêtre de séparation des classes, qui est de 2 γ où γ représente la marge maximale.

Nous pouvons donc observer, pour chaque valeur de C, les différents vecteurs supports (cf. figures 6, 7 et 8).

Grand C

- Plus C est grand, plus la marge est petite, et donc moins le modèle est tolérant aux erreurs.
- Le modèle accorde une grande importance à la minimisation des erreurs d'entraînement, au détriment de la généralisation.
- Risque de sur-apprentissage.

Petit C

- Lorsque C est plus petit, la marge est plus importante, rendant le modèle plus souple face aux erreurs.
- Un petit C rend le modèle plus général, car il ne cherche pas à séparer parfaitement les classes.
- Risque de sous-apprentissage

Nous allons maintenant générer un jeu de données très déséquilibré avec deux classes (90% vs 10%), et observé l'influence de C sur les résultats.

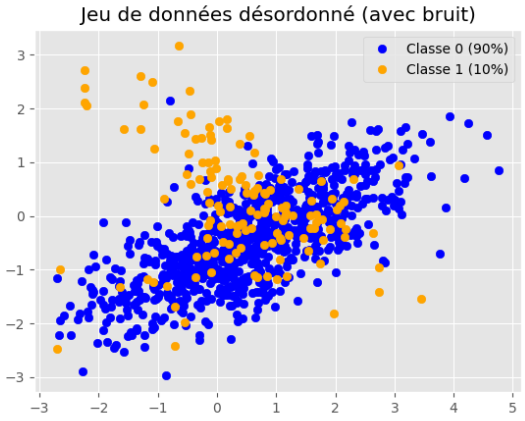


Figure 9: Distribution des deux classes

Nous allons pour cela entrainer un SVM avec un noyau linéaire, et observer l'influence du paramètre de régulation C sur le score.

C	Score	Matrice de Confusion	Taux de prédiction 1 à tort
0.001	0.8633	$\begin{pmatrix} 259 & 0 \\ 41 & 0 \end{pmatrix}$	13.6667%
0.01	0.87	$\begin{pmatrix} 259 & 0 \\ 39 & 2 \end{pmatrix}$	13%
1	0.9033	$\begin{pmatrix} 258 & 1 \\ 28 & 13 \end{pmatrix}$	9.333%
100	0.9033	$\begin{pmatrix} 258 & 1 \\ 28 & 13 \end{pmatrix}$	9.333%

Table 2: Récapitulatif des résultats

Résultats des expériences

Les résultats obtenus pour différentes valeurs de C sont présentés ci-dessous.

Nous constatons de plus que le score augmente au fur à mesure que l'on augmente le paramètre C, avant de se stabiliser.

Nous observons rapidement, grâce aux matrices de confusion, que le modèle a tendance à prédire à tort la classe majoritaire de manière fréquente. En réalité, presque toutes les erreurs du modèle proviennent de ce phénomène.

Notre objectif va être d'y remédier, pour cela, nous allons utiliser le paramètre "$class_weight$" afin de pouvoir pondérer les erreurs en fonction de la rareté de la classe. Les erreurs sur la classe la moins présente seront donc plus lourdement pénalisées, ce qui permettra au modèle de mieux prendre en compte cette classe au moment de l'entrainement.

Nous obtenons les résultats suivants :

Score	Matrice de Confusion
0.92	$\begin{pmatrix} 258 & 1 \\ 23 & 18 \end{pmatrix}$

Le taux d'erreur provoqué par la prédiction de la classe 1 à tort à diminuer, passant à 7.667%. Nous remarquons que le score a augmenté, atteignant 0.92, ce qui permet de dire que grâce à la pondération des classes, le modèle a mieux appris.

4 Question 4

Nous allons maintenant nous intéresser à la classification d'image. Pour cela, nous allons utiliser les données "Labeled Faces in the Wild" (LFW) qui contient des images de visages de célébrités. Nous allons conserver seulement les personnes ayant au moins 70 photos.

Ici, deux personnes sont sélectionnées, par exemple "Tony Blair" et "Colin Powell".

Les images des deux personnes sélectionnées sont regroupées. y est le vecteur de labels : 0 pour la première personne (Tony Blair) et 1 pour la seconde (Colin Powell).On se placera donc dans le cadre de la classification binaire.

Figure 10: Image de Tony Blair et Colin Powell tiré aléatoirement

Nous générons cette fois encore un échantillon de test et d'entrainement (50% chacun), et nous allons entrainer un classifieur SVM avec un noyau linéaire.

Nous allons maintenant visualiser les prédictions faites par le modèle sur les données de test.

Figure 11: Prédiction du modèle

Nous remarquons que dans notre cas, sur 12 prédictions, il y a une seule erreur de prédictions, ce qui représente un très bon résultat.

Nous allons maintenant nous intéresser au paramètre de régularisation C pour le classificateur SVM linéaire. Nous entrainons le modèle pour des valeurs C allant de 10^{-5} à 10^{5} et sélectionne la valeur C qui permet d'obtenir le meilleur score.

C optimal	Best Score	Temps d'exécution
0.0001	0.9105263157894737	3.678 s

Table 3: Résultats des paramètres optimaux

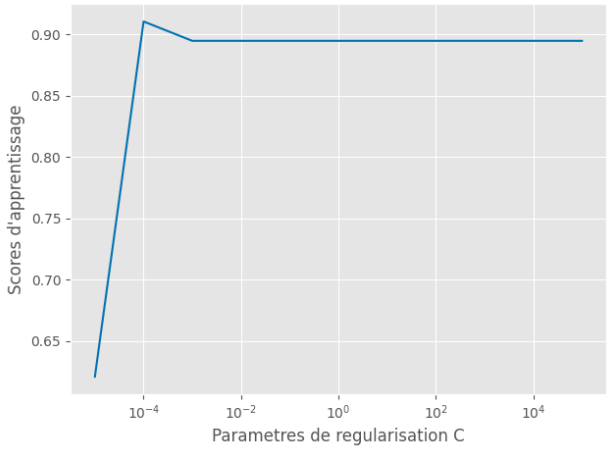


Figure 12: Évolution du score en fonction de la valeur de C

En observant le graphique, nous constatons que le score augmente de manière exponentielle entre 10^{-5} à 10^{-4} et atteint son maximum à 10^{-4}, avant de converger vers un score de 0.89.

Dans l'ensemble, la capacité de généralisation du modèle semble être satisfaisante, comme en témoigne le score relativement élevé.

Nous allons maintenant étudier une visualisation des coefficients (les poids) du modèle SVM après l'entraînement, sous la forme d'une carte de chaleur. Ces poids indiquent quelles sont les parties de l'image influant le plus sur la classification.

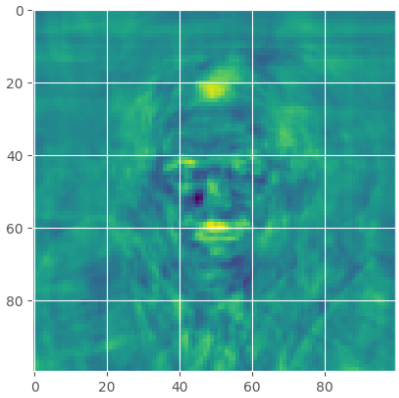


Figure 13: Carte de chaleur des coefficients

Les zones claires représentent les pixels ayant un coefficient élevé, ce qui signifie que ces zones sont particulièrement importantes pour la décision du modèle. Ces zones correspondent aux traits du visage communs entre Tony Blair et Colin Powell.

À l'inverse, les régions sombres sont des pixels ayant des coefficients proches de zéro, ce qui illustre le fait

qu'ils ne jouent pas un rôle important dans la classification.

5 Question 5

Nous allons étudier l'effet d'un ajout de variables de nuisances sur la performance de prédiction.

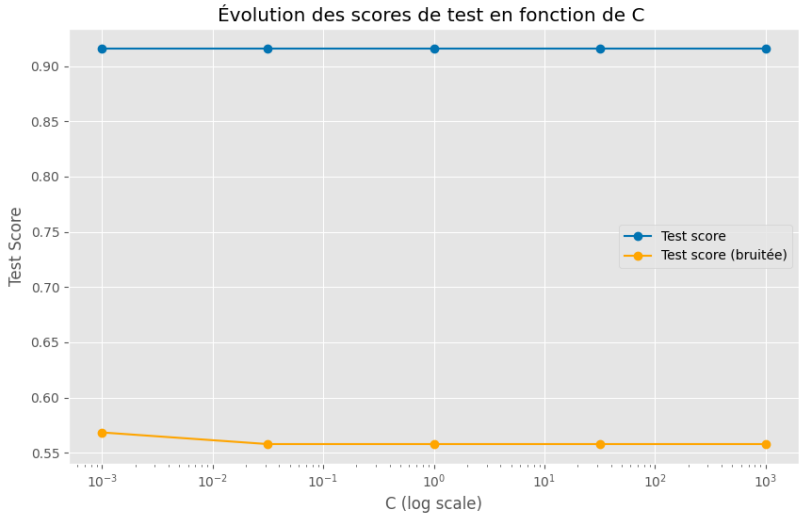


Figure 14: Comparaison des scores obtenue avec les données bruitées et non bruitées

Nous observons clairement à partir du graphique ci-dessus que la performance prédictive diminue de manière significative, avec une réduction d'environ 0.35, passant d'environ 0.91 à 0.57. Cette baisse marquée souligne les limites du modèle dans la gestion des données bruitées.

6 Question 6

Afin d'améliorer la performance du modèle sur les données bruitées, nous allons procéder à une réduction des dimensions avec une ACP.

Nombre de composantes	Score d'entraînement	Score de test
3	0.6368421052631579	0.6052631578947368
10	0.6578947368421053	0.5842105263157895
20	0.7210526315789474	0.5736842105263158
80	0.8789473684210526	0.5105263157894737
90	0.9736842105263158	0.5052631578947369
120	0.8473684210526315	0.4789473684210526
150	0.8894736842105263	0.5263157894736842

Table 4: Scores d'entraînement et de test en fonction du nombre de composantes

Nous constatons que plus le nombre de composantes est élevé, plus le score sur les données d'entrainement est élevé, alors que le score sur les données de test est en moyenne de 0.5, ce qui semble nous illustrer un phénomène de sur-apprentissage.

Dans le cadre d'un nombre de composantes assez faible, l'écart entre le score des données d'entrainement et de test est toujours présent en faveur des données d'entrainement, mais cette différence est beaucoup moins flagrante.

Nous pouvons donc en déduire que la réduction des dimensions semble permettre de réduire l'impact du sur-apprentissage.

Le temps d'exécution du code étant très long, pour un nombre de composantes inférieur à 80, je n'ai malheureusement pas pu approfondir l'analyse.

7 Conclusion

Ce TP sur les Support Vector Machines (SVM) a permis d'explorer différentes approches de classification supervisée à travers l'utilisation de plusieurs types de noyaux, ainsi que des techniques d'optimisation des hyperparamètres comme la validation croisée.

L'étude a mis en évidence l'efficacité des noyaux linéaires et polynomiaux, ainsi que leurs limites. Nous avons aussi étudié l'influence du paramètre C joue un rôle clé dans la capacité du modèle à équilibrer la précision et la généralisation.

L'ajout de variables de nuisance a montré une diminution marquée des performances prédictives, ce qui a ensuite été partiellement compensé par l'utilisation de la réduction de dimension via l'Analyse en Composantes Principales (ACP). Cependant, la réduction des dimensions n'a pas totalement éliminé le sur-apprentissage, soulignant la nécessité d'envisager d'autres méthodes plus adaptés aux données bruitées.