01/12/2023

Rapport de projet

Machine Learning

DOUBABI Mustapha
NEVEU Pierre

Dauphine | PSL
UNIVERSITÉ PARIS

Table des matières

I) Modèles

1) Régression logistique

En théorie, ce modèle est robuste face aux valeurs aberrantes mais sa précision peut faiblir en cas de grandes dimensions.

La régression logistique, consiste en la maximisation de la fonction de coût suivante pour estimer le paramètre B chapeau:

$$\arg\max_{\beta \in \mathbf{R}^{p+1}} \sum_{i=1}^{n} y_i \log \sigma(\beta^\top x_i) + (1 - y_i) \log \left(1 - \sigma(\beta^\top x_i)\right)$$

De ce fait, le gradient est:

$$\nabla_\beta \log \mathcal{L} = \sum_{i=1}^{n} \left(y_i - \underbrace{\frac{1}{1 + e^{-\beta^\top x_i}}}_{\sigma(\beta^\top x_i)} \right) x_i$$

Afin de maximiser cette fonction, nous allons adapter la méthode de la descente de gradient. Cette méthode consiste en un algorithme itératif qui permet de trouver un minimum global d'une fonction convexe. Nous allons donc chercher le maximum global d'une fonction concave.

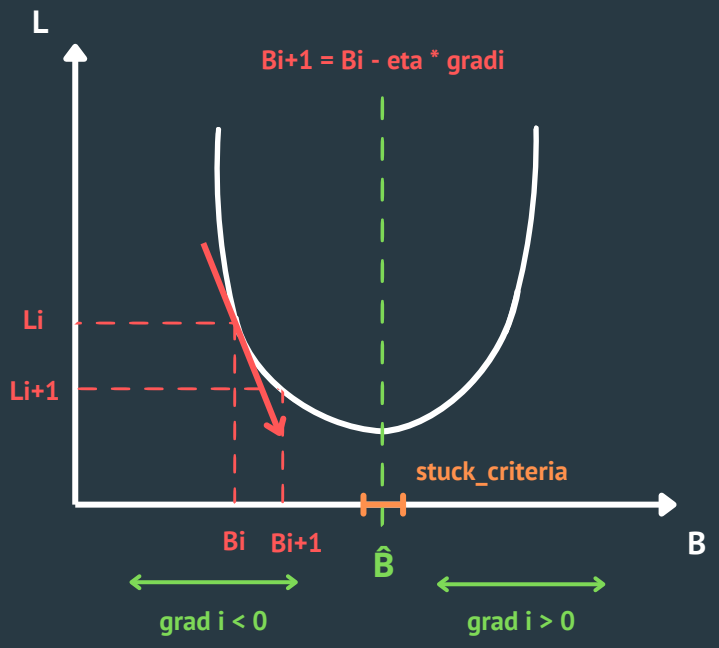
Soient eta le learning rate et stuck_criteria qui assure les deux hyperparamètres de notre algorithme.
Voici comment adapter la descente de gradient pour maximiser une fonction:

Descente de gradient minimisation

L_0 = +infinite
B_0 = random()
while not convergent
$B_{i+1} = B_i$ - eta * $grad_i$
$L_{i+1} = L(B_{i+1})$
if $L_{i+1} > L_i$:
end
if $B_{i+1} - B_i <$ stuck_criteria:
end

Critères de convergence:

- **1) Bi+1 - Bi < stuck_criteria**
- **2) Li+1 > Li**

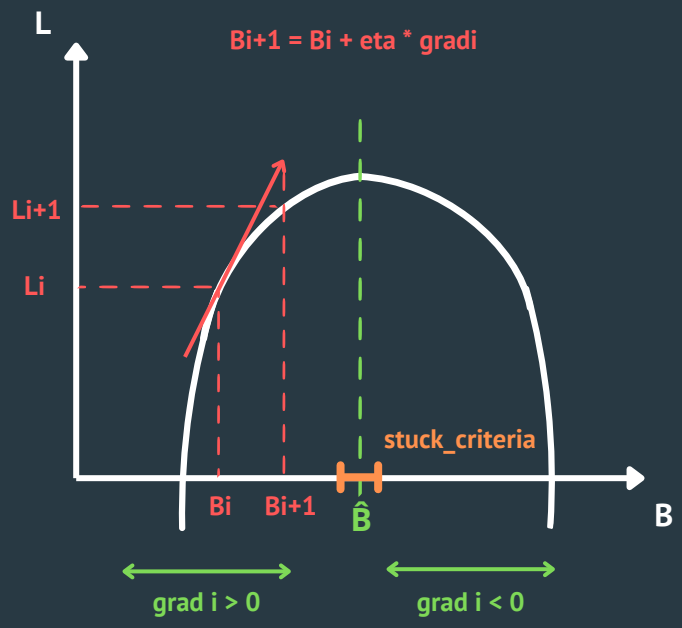


Montée de gradient maximisation

L_0 = -infinite
B_0 = random()
while not convergent
$B_{i+1} = B_i$ + eta * $grad_i$
$L_{i+1} = L(B_{i+1})$
if $L_{i+1} < L_i$:
end
if $B_{i+1} - B_i <$ stuck_criteria:
end

Critère de convergence:

- **1) Bi+1 - B < stuck_criteria**
- **2) Li+1 < Li**



NB: Python arrondi les valeurs inférieurs à 1e-15 à 0. Après avoir tester l'algorithme, nous nous sommes rendu compte qu'à certaines itérations, la valeur de sigmoid(X, b) était inférieur à 1e-15 et donc nous avions np.log(0) ce qui est mathématiquement impossible. Nous nous proposons donc de rajouter une constante err_correct = 1e-15 afin de contourner ce problème.

2) Analyse Discriminante Linéaire (LDA)

Le modèle de la LDA fait l'hypothèse que les données ont une distribution gaussienne (iid) et que les matrices de covariance des différentes classes sont égales. Elle suppose également que les données sont linéairement séparables, ce qui signifie qu'une frontière de décision linéaire peut classer avec précision les différentes classes.

L'équation de l'hyperplan séprateur repose sur la probabilité de chaque classe, ainsi que sur leur centre. Son implémentation est assez simple et se repose sur l'équation suivante:

$$x^\top \underbrace{\Sigma^{-1}(\mu_0 - \mu_1)}_{w} - \underbrace{\frac{1}{2}(\mu_0 - \mu_1)^\top \Sigma^{-1}(\mu_0 + \mu_1) + \log(\frac{\pi_0}{\pi_1})}_{b} = 0$$

Avec:

- π_k : la probabilité de la classe k
- mu_k : le centre de la variable k
- Sigma: la matrice de covariance

3) Machine à Vecteurs de Supports (SVM)

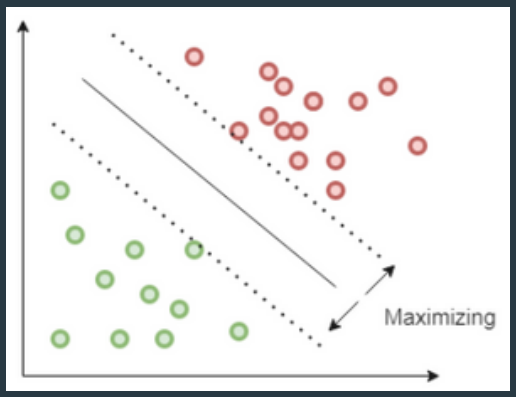
Les Machines à Vecteurs de Support (SVM) sont des méthodes d'apprentissage supervisé utilisées pour la classification et la régression, qui généralisent les classificateurs linéaires. Elles se sont avérées efficaces pour traiter des données de **grandes dimensions** avec **un nombre restreint d'hyperparamètres.**

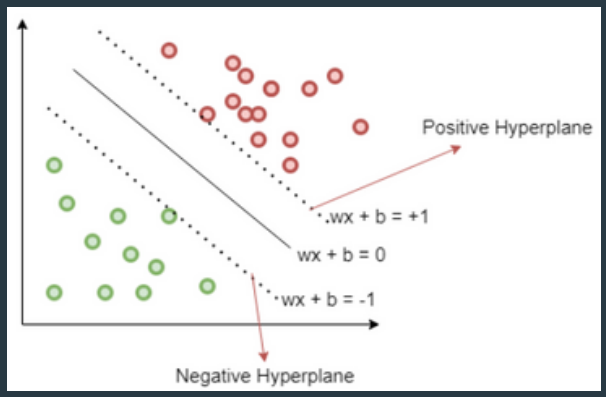
Les SVM ont trouvé des applications dans divers secteurs tels que la **bioinformatique, la recherche d'information, la vision par ordinateur et la finance**. Leur efficacité est **comparable, voire supérieure**, à celle d'autres techniques de pointe telles que les **réseaux de neurones ou les modèles de mélanges gaussiens.**

Comment elles fonctionnent ?

Les SVM sont considérées comme des classificateurs à large marge, utilisant **des vecteurs supports**, qui sont les points de données **les plus proches de l'hyperplan pour maximiser cette marge (= la distance entre la frontière de séparation et les échantillons les plus proches).**
L'algorithme peut être adapté à des données non linéaires grâce à l'astuce du noyau **(kernel trick)**, qui permet de traiter les données dans un espace de dimensions plus élevées sans calculs lourds.

Illustration avec un problème





Mathématiques pour la recherche de la solution optimale

On donne le label 1 aux points au dessus de l'hyperplan et - 1 aux autres.
On rappelle que l'hyperplan à trouver est paramétré par les poids w et b (w0) et son équation :

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

Avec les équations des hyperplans positifs et négatifs (voir graphique ci-dessus), on peut en déduire la largeur entre eux deux:

$$w^T x_1 + b - w^T x_2 + b = 1 - -1$$

D'où:

$$w^T (x_1 - x_2) = 2$$

En utilisant le vecteur unitaire de w, c'est à dire en divisant par la norme de w:

$$(x_1 - x_2) = \frac{2}{\|w\|}$$

On transforme ce problème de maximisation en minimisation:

$$min \frac{1}{2}||w||^2 \quad \text{sous contraintes} \quad y_n \begin{Bmatrix} +1 & w^T x + b \geq 1 \\ -1 & w^T x + b \leq -1 \end{Bmatrix}$$

i.e.

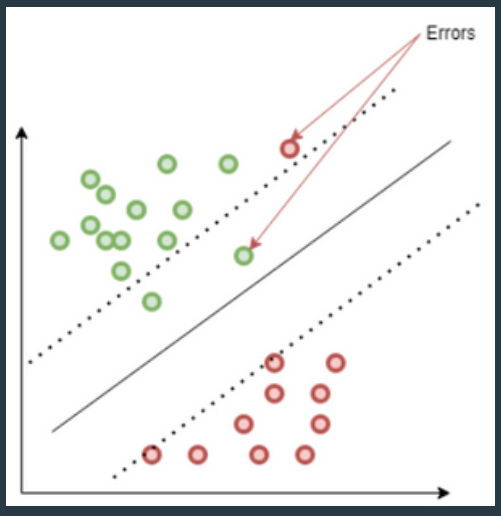
$$min \frac{1}{2}||w||^2 \quad \text{sous contraintes} \quad y_n(w^T x + b) \geq 1$$

Nous utilisons de plus la fonction de coût de Hinge:

$$f(x) = max\{0, 1 - t\}$$

Cela permet d'inclure les erreurs dans notre modèle et éviter le surentraînement, voici donc notre modèle:

$$\underbrace{\min \frac{1}{2}\|w\|^2}_{\text{regularizer}} + C_i \underbrace{\sum_{i=1}^{n} \max\{0, 1 - y_n(w^T x + b)\}}_{\text{error term}}$$



Remarque

Comme la mise en place de l'hyperplan séparateur ne dépend que des points les plus "proches", les valeurs aberrantes n'auront pas d'effet sur les poids paramètres.

Implémentation: Mini Batch Gradient Descent

Batch Gradient Descent	Stochastic Gradient Descent	Mini Batch Gradient Descent
Parcours toutes les données, calcule de chaque erreur avant d'ajuster les paramètres	Choisit aléatoirement une donnée, calcule l'erreur avant d'ajuster les paramètres	Choisit aléatoirement un lot de données (dont la taille est fixé en hyper paramètre), calcule les erreurs avant d'ajuster les paramètres

C'est donc pour des raisons de flexibilité et de complexité que nous choisissons la descente de gradient par mini lots, voici le pseudo code pour plus de détails:

Soit X, y les matrices de données et labels, pour un taux d'erreur C fixé:

```
w, b = 0
For i in range(epochs):
    L_i = Hinge(w, b)
    batch = random
    for j in range(0, n, batch_size):
        gradw, gradb = 0
        for k in range(j, j + batch_size):
            condition_i = y_i * (w^T X + b)
            if condition_i < 1:
                update gradw
                update gradb
        w = w - eta * w + eta * grad_w
        b = b + eta * grad_b
```

Hyper paramètres:

- **eta**: c'est le learning rate, qui suit le même principe que vu précédemment
- **epochs**: C'est le nombre de fois que l'on réitérer le calcul des erreurs pour ajuster les paramètres
- **batch_size**: la taille des lots pris au hasard pour calculer les erreurs

II) Dataset du cancer

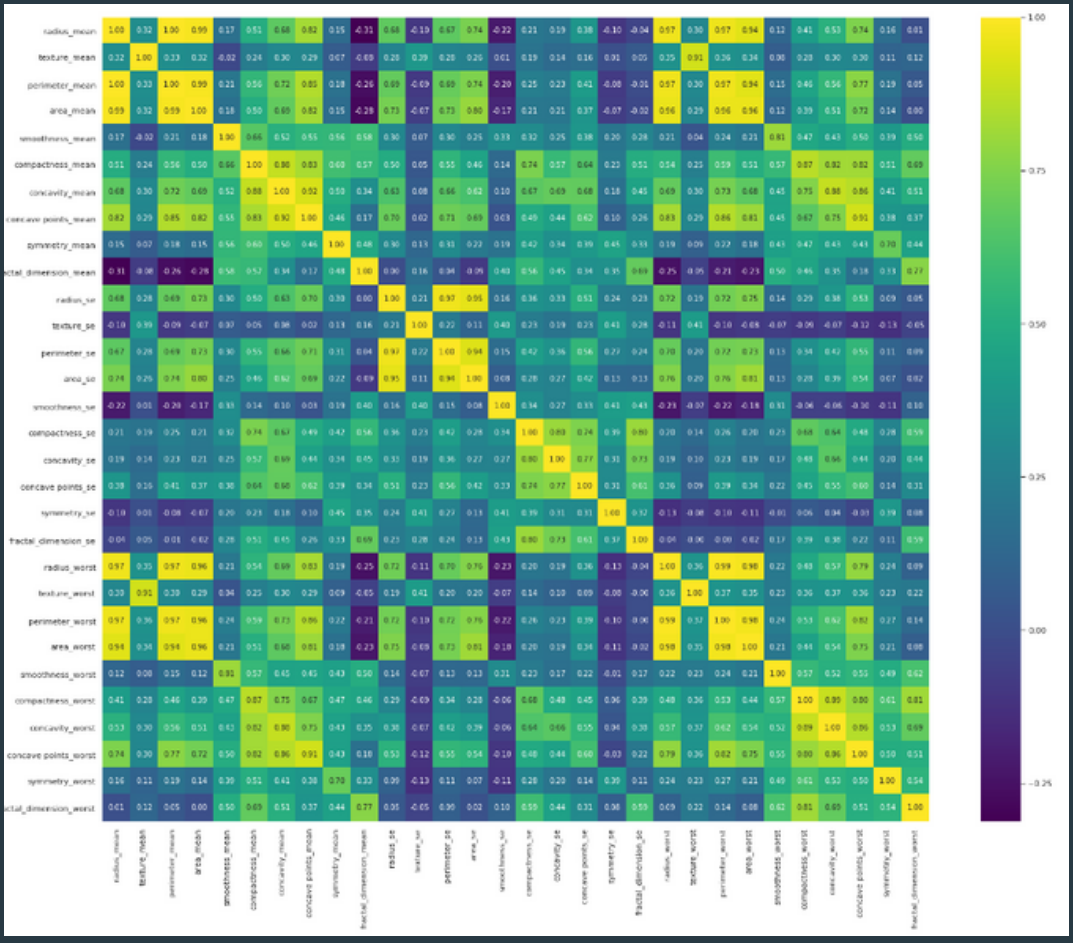
Nous avons appliqué les modèles sur le dataset du cancer du sein, largement utilisé et disponible dans scikit-learn.

Cet ensemble de données contient 30 features indépendantes calculées à partir d'images numérisées. L'objectif est de classer les tumeurs comme **malignes** ou **bénignes** sur la base de ces caractéristiques.

Notons que l'ensemble de ces données sont de type float64. Dans le cadre d'utilisation du SVM, nous procédons à un pré-traitement des labels en transformant les valeurs négatives en -1 et les valeurs strictement positives en 1.

Matrice de corrélation:

Les features qui composent le data set ont majoritairement des covariances positives.



Valeurs aberrantes:

(Voir notebook)

III) Dataset du projet de marketing

Contexte

En novembre 2023, dans le cadre du Master 1 MIAGE de l'université Paris Dauphine, nous avons travaillé sur un projet de marketing dont le but était de proposer une innovation sur le marché de notre choix. Notre proposition d'innovation était la sortie d'un nouveau software à l'occasion de la sortie de la nouvelle Apple Watch visant à améliorer le sommeil et le bien être des utilisateurs accompagne de la sortie d'un nouveau bracelet conçu pour la nuit. La fonctionnalité la plus importante est un réveil intelligent qui réveillerait l'utilisateur au moment le plus optimisé en fonction de l'heure de réveil maximale qu'il a programmé et l'analyse de ses cycles de sommeil. Le projet WHOOP (www.whoop.com) est un projet existant comparable.

Image générée par DALL.E

Source des données

Dans le cadre de l'étude des attentes des consommateurs, nous avons diffusé un questionnaire permettant de récolter des informations démographiques et comportementales des consommateurs ainsi que leur motivation et rapport au sommeil / bien-être ainsi que leurs différentes propension à payer pour les nouveaux produits.

Traitement des données

Nous avons donc un jeu de données qui comporte 150 échantillons caractérisés par 18 variables aussi bien qualitatives que quantitatives. Ces variables sont la résultante des réponses aux questions posées aux consommateurs. Google Forms permet d'extraire un fichier excel dont les colonnes sont les questions posées et les valeurs sont les réponses des consommateurs.

Hypothèses

Nous allons traiter les données en faisant les hypothèses de simplification suivante:

- Nous ne gardons pas les variables dont la réponse est un choix multiple
- Nous considérons qu'une réponse "Non" peut se résumer à "Non" quelle que soit la nuance de la réponse. Nous faisons le même raisonnement pour "Oui". La question "Utilisez vous actuellement un bracelet ou une application de suivi de sommeil ?", pour laquelle nous considérons la réponse "Non", mais j'ai déjà utilisé" comme un "Oui", est la seule exception
- L'échelle des réponses qui évaluent un niveau (valeurs de 1 à 5) est linéaire.
- Une intervalle peut se résumer par son centre.

One-hot encoding

Considérant ces hypothèses, nous avons utilisé la méthode de one-hot encoding pour traduire les variables qualitatives. Nous avons alors deux ensembles de variables à encoder :

- Les variables de valeurs binaire “Oui” ou “Non” qui seront codés respectivement par 1 et 0.
- Les variables non binaires : les n-valeurs possibles sont respectivement représentées par n colonnes dont les valeurs possibles sont binaires.

Choix du label

Nous décidons de désigner la variable “new_sleeping_bracelet_desire”: comme label : 1 si la personne souhaite acheter le nouveau bracelet, 0 sinon. En effet, cette variable est binaire de valeurs 1 (Oui) et 0 (Non) et de quantités relativement équilibrées. Nous décidons donc de nous débarrasser des échantillons ayant répondu “Peut-être” à cette question. Cela nous permet de simplifier le jeu de données et de ne pas émettre d’hypothèses non fondées quant à la signification de la valeur ‘Peut-être”.

Mapping post et pré-traitement des données

Questions	Nom de colonne formatée	Valeurs des données pré-traitement	Type des données pré-traitement	Valeurs des données post-traitement	Type des données post-traitement
Horodateur	Horodateur		Date		
Etes vous un utilisateur Apple?	is_apple_user	“Oui” “Non”	String	**1** **0**	**Int**
Possédez-vous une montre connectée?	is_connected_watch_user	“Oui” “Oui, pas une Apple Watch” “Non”	String	**1** **1** **0**	**Int**
Pourquoi n’utilisez-vous pas de montre connectée ?	why_no_watch_user	Réponses multiples	String		
Comment évaluez-vous la qualité de votre réveil au quotidien ?	awakening_quality_level	1, 2, 3, 4 ou 5 1: Très mauvais 5: Très bien	Int		

A quel point souhaitez-vous améliorer la qualité de votre réveil ?	awakening_improvment_desire__level	1, 2, 3, 4 ou 5 1: Pas du tout 5: Énormément	Int		
Utilisez vous actuellement un bracelet ou une application de suivi de sommeil ?	is_sleep_tracking_user	“Oui” “Non” “Non, j’ai déjà utilisé mais j’ai arrêté”	String	**1 1 0**	**Int**
A quel point êtes-vous intéressé par l’achat de cette nouvelle Apple Watch ?	new_watch_desire_level	1, 2, 3, 4 ou 5 1: Pas du tout intéressé 5: Extrêmement intéressé	Int		
Quelles fonctionnalités vous intéressent le plus ?	wanted_fonctionnalities	Réponses multiples	String		
A quel point dormir avec votre Apple Watch vous gênerait-il ?	bracelet_sleeping_discomfort_level	1, 2, 3, 4 ou 5 1: Pas du tout gêné 5: Très dérangeant	Int		

Etes-vous intéressé par des bracelets spéciaux pour le confort pendant le sommeil avec l'Apple Watch ?	new_sleeping_bracelet_desire	"Oui" "Non" "Peut-être"	String	**1** **0** **Supprimée**	**Int**
Seriez-vous prêt à payer 450 euros pour la nouvelle Apple Watch ?	new_watch_purchase_willing	1) "Oui" 2) "Non" 3) "Non, j'ai déjà une Apple Watch et les fonctionnalités ne suffisent pas"	String	**1** **0** **0**	**Int**
Seriez-vous prêt à payer 99 euros pour le nouveau bracelet ?	new_bracelet_purchase_willing	1) "Oui" 2) "Non"	String	**1** **0**	**Int**
Quel est votre âge ?	age	Réponses multiples: intervalles d'âge	String	**Centre de l'intervalle**	**Int**
Quel est votre genre ?	gender	1) "Homme" 2) "Femme" 3) "Autre"	String	**Supprimé**	**Supprimé**

Quelle est votre situation professionnelle ?	professional_ situation	1) “Étudiant” 2) “Travailleur actif” 3) “Retraité” 4) “Sans profession”	String	**Supprimé**	**Supprimé**
	man			**1 0**	**Int**
	woman			**1 0**	**Int**
	other			**1 0**	**Int**
	student			**1 0**	**Int**
	worker			**1 0**	**Int**
	unemployed			**1 0**	**Int**
	retired			**1 0**	**Int**

IV) Comparaison des modèles en pratique

Les observations suivantes sont le fruit d'implémentations plus ou moins simplifiées. De ce fait, les commentaires porteront majoritairement sur la pratique plutôt que sur la théorie. Il se peut qu'en complexifiant les données (dimensions très élevées ou autres), les modèles ne reflètent pas la théorie.

*RL = régression logistique
*LDA = Analyse discriminante linéaire
*SVM = Machine à vecteurs de support

Données initiales

A priori, on attribuerait visuellement la classe 0 au point (0,0). Cependant, on observe un surentraînement (apprentissage par coeur) de la RL qui met à défaut la généralisation du modèle. De ce fait, la RL prédit, contrairement à la LDA et au SVM, la classe 1. La RL a alors un meilleur taux d'erreur sur les données actuelles mais se trompe probablement quant à la prédiction d'un nouveau point. Des librairies, telles que scikit-learn, permettent d'appliquer un biais à la RL pour empêcher un surentraînement.

Données avec valeurs aberrantes

Ces observations confirment la théorie (i.e le cours). En effet, on observe avec les validations croisées un meilleur taux d'erreur de la RL par rapport à la LDA traduisant une sensibilité moindre face à des valeurs aberrantes. Le SVM confirme également la théorie puisque sa frontière de décision n'a pas ou peu bougé et son taux d'erreur sur les données test n'a pas changé.

Données avec dimensions augmentées et variables corrélées

- **Augmentation des variables corrélées: dim = 4, n_redundant = 3**

L'augmentation du nombre de variables corrélées (non indépendantes) met en évidence une augmentation du taux d'erreur de la LDA. En effet celle-ci fait l'hypothèse de variables indépendantes et identiquement distribuées. La régression logistique quant à elle y est plus robuste.

- **Augmentation de la dimension sans corrélation : dim = 20, n_redundant = 0**

La performance de la fonction d'entraînement de la RL, telle qu'implémentée, est affectée par une dimension augmentée. De plus, cette configuration permet d'observer avec la validation et les taux d'erreurs que la RL est moins robuste que la LDA et le SVM face à des données en plus grande dimension.

Dataset du cancer

Le modèle SVM, tel qu'implémenté, de par sa flexibilité et sa sensibilité au choix des hyper paramètres, est plus difficile. La RL, est de même moins efficace sur ce dataset qui est plus complexe. Pour résumer, les validations croisées montrent que les modèles SVM et surtout RL tels qu'implémentés ne répondent pas bien au modèle.
Cependant, on observe que les modèles de RL de scikit-learn, confirment la théorie de par leur gestion plus efficace des hyperparamètres. Effectivement, la régression logistique de scikit-learn obtient de meilleurs résultats que la LDA en raison de la corrélation positive des features entre elles, confirmant encore une fois les théories du cours de par le fait qu'une corrélation peut aller en défaveur de la LDA.
Également, notons que la présence de beaucoup de valeurs aberrantes (voir notebook) dans le dataset joue en faveur de la régression logistique qui est moins sensible par rapport à la LDA.

Dataset marketing

Les taux d'erreurs et valdiations croisées sont assez satisfaisantes compte tenu du fait que les données sont essentiellement qualitatives ce qui valide l'encoding et les choix de pré-traitement qui ont été fait. Sur les features non binaire, on détecte quelques valeurs aberrantes grâce à l'IQR (Inter Quartile Range) ce qui pourrait expliquer un résultat moins satisfaisant pour la LDA en plus d'une possible corrélation des variables. Quant au SVM, c'est en grande partie à cause de la difficulté de trouver les hyper paramètres satisfaisants que les résultats sont moins bons.
Les validations croisées remettent toutefois en questions les modèles. Cepandant, en ayant essayer les modèles de scikit learn des trois modèles sur les données marketing, on trouve des résultats satisfaisants.

Pour conclure sur ce jeu de donnée, il faut prendre du recul et se rappeler que des hypothèses de simplifications ont été faites et peuvent être plus ou moins fondées. De ce fait, ces modèles sont sûrement peu généralisables pour ces données. Cependant, il est intéressant de voir que l'on puisse obtenir des estimations convenables sur des données concrètes avec des modèles quelque peu simplifiés.

V) Annexe

Recherches SVM

https://www.pycodemates.com/2022/10/implementing-SVM-from-scratch-in-python.html
https://www.pycodemates.com/2022/09/primal-formulation-of-svm-simplified.html
https://www.youtube.com/watch?v=IU5fuoYBTAM
https://www.youtube.com/watch?v=dSqKGNT0qaw

Recherche dataset cancer

https://www.kaggle.com/code/yifanxia0/breastcancer-final