



5

# Modèles SKLEARN

# 5. Modèles - sklearn



Modèle	Package	Description	Gère Overfit?	Gère outliers?	Gère gd nbre var?	adaptive learns regularization	Large dataset	Non linear	Usage?	Usage+?	Avantages	Désavantage
<b>LinearRegression</b>	linear	Algorithme des moindres carrés ordinaires							Hautement interprétable, aucun biais introduit	Les données sont composées de peu de valeurs aberrantes La variance entre les étiquettes de sortie est faible Toutes les caractéristiques d'entrée sont non seulement indépendantes mais aussi non corrélées.	Facilité d'interprétation des résultats Faible niveau de complexité	Risque de multicolinéarité si les caractéristiques d'entrée sont corrélées. De petites erreurs ou valeurs aberrantes dans les valeurs cibles peuvent avoir un impact considérable sur le modèle
<b>Ridge</b>	linear RidgeCV	Les moindres carrés ordinaires avec un Oui terme de régularisation L2 ajouté. Le poids du terme de régularisation est contrôlé par un argument alpha supplémentaire.	Oui						Prévention de l'overfitting	Les données sont composées de quelques valeurs aberrantes Il peut y avoir une certaine corrélation entre les caractéristiques d'entrée. Éviter le surajustement On veut utiliser toutes les caractéristiques pour faire des prédictions.	Facile à interpréter les résultats Moins affecté par les caractéristiques corrélées Prévient le surajustement et diminue la variance des poids appris	En réduisant la variance, il incorpore un certain degré de biais dans le modèle. Il peut être difficile d'ajuster alpha pour atteindre un équilibre souhaitable entre les MCO et le terme de régularisation
<b>Lasso</b>	linear LassoCV lasso_path LassoLarsCV LassoLarsIC MultiTaskLasso	Algorithme des moindres carrés ordinaires avec ajout d'un terme de régularisation L1. Le poids du terme de régularisation est contrôlé par un argument alpha supplémentaire. Le poids du terme MCO est inversement proportionnel au nombre d'échantillons.	Oui	Oui					Seules quelques caractéristiques sont importantes, sélection des caractéristiques	Les données sont composées de quelques valeurs aberrantes Il peut y avoir une certaine corrélation entre les caractéristiques d'entrée. Vous souhaitez donner plus de poids à quelques caractéristiques importantes et ne pas ignorer les caractéristiques moins importantes. Sélection des caractéristiques	Réduit le nombre de caractéristiques utilisées Peut gérer de grandes quantités de caractéristiques. Peut souvent être utilisé comme un test préliminaire pour effectuer une sélection de caractéristiques	En réduisant la variance, incorpore un certain degré de biais dans le modèle. Il peut être difficile de régler alpha pour atteindre un équilibre souhaitable entre les MCO et le terme de régularisation

# 5. Modèles - sklearn



Modèle	Package	Description	Overfit?	Gère gd outliers?	Gère gd nbre var?	adaptive learns regularization	Large dataset	Non linear	Usage?	Usage+?	Avantages	Désavantage
ElasticNet	linear ElasticNetCV MultiTaskElasticNet	Moindres carrés ordinaires avec un terme de régularisation L1 et L2. Les poids des termes de régularisation L1 et L2 sont contrôlés par un paramètre l1_ratio.	Oui	Oui					Mélange de Ridge et Lasso	Les données sont composées de quelques valeurs aberrantes Il peut y avoir une certaine corrélation entre les caractéristiques d'entrée. Éviter le surajustement Sélection des caractéristiques	Incorporer les capacités de sélection de caractéristiques de Lasso avec les capacités de régularisation de Ridge.	En réduisant la variance, incorpore un certain degré de biais dans le modèle. Il peut être difficile de régler l'alpha pour atteindre un équilibre souhaitable entre les MCO et les termes de régularisation. Coût de calcul plus élevé que Ridge ou Lasso
Lars (Least Angle Regression)	linear lars_path lars_path_gram	Une régression pas à pas qui modifie les poids des caractéristiques les plus corrélées à l'étiquette de sortie à chaque étape. Au fur et à mesure des étapes, d'autres caractéristiques deviennent tout aussi corrélées. Lorsque cela se produit, l'algorithme se met à jour pour se déplacer dans la direction déterminée par les angles commutatifs des poids de ces caractéristiques.			Oui				Plus de caractéristiques que d'échantillons de données	Plus de caractéristiques d'entrée que d'échantillons de données Les données comportent peu de valeurs aberrantes Peu de variance entre les étiquettes de sortie	Les poids des caractéristiques également corrélées sont mis à jour à des taux similaires. Efficacité numérique et informatique pour de nombreuses caractéristiques. Résultats et étapes de formation interprétables	Le bruit ou de petites erreurs ou valeurs aberrantes dans les valeurs cibles peuvent avoir un impact considérable sur le modèle
OrthogonalMatchingPursuit	linear orthogonal_mp	Une régression par étapes vers l'avant qui permet à l'utilisateur de spécifier un nombre maximal de coefficients non nuls ou une valeur d'erreur cible à atteindre.			Oui				Sélection des caractéristiques	Vous cherchez à sélectionner les N caractéristiques les plus importantes pour représenter un modèle composé de plusieurs caractéristiques	Peut déterminer les caractéristiques les plus étroitement liées aux étiquettes. Peut spécifier le nombre exact de coefficients non nuls à retourner	Applications limitées Non utilisé très souvent (et donc pas beaucoup de ressources en ligne)

# 5. Modèles - sklearn



Modèle	Package	Description	Overfit?	Gère outliers?	Gère gd nbre var?	adaptive learns regularization	Large dataset	Non linear	Usage?	Usage+?	Avantages	Désavantage
<b>BayesianRidge</b>	linear	Similaire à Ridge, mais le paramètre de régularisation est réglé pour s'adapter aux données pendant le processus de formation.	Oui			Oui		Ridge mais ne veut pas fixer de constante de régularisation	Vous recherchez des résultats similaires à ceux de Ridge, mais vous êtes prêt à sacrifier l'interprétabilité pour gagner du temps en évitant de tester différents poids de régularisation.	Aucun besoin de régler la valeur alpha s'adapte bien aux données disponibles	Des résultats moins interprétables	
<b>ARDRegression</b>	linear	BayesianRidge avec des valeurs de poids plus espacées. Presque comme une version de BayesianLasso.	Oui		Oui	Oui		Lasso mais pas de constante de régularisation.	Vous recherchez des résultats similaires à ceux de Lasso, mais vous êtes prêt à sacrifier l'interprétabilité pour gagner du temps en évitant de tester différents poids de termes de régularisation.	Il n'est pas nécessaire de régler la valeur alpha. S'adapte bien aux données disponibles Réduit le poids des caractéristiques non importantes	Résultats moins interprétables Coûteux en termes de calcul (ne peut pas traiter de très grands ensembles de données)	
<b>SGDRegressor</b>	linear	Ajusté en minimisant une perte empirique régularisée avec une descente de gradient stochastique (SGD). Le type de fonction de perte utilisé peut être changé en utilisant l'argument "loss".			Oui		Oui	Le nombre de caractéristiques et d'échantillons est très élevé rapidement des résultats.	Avoir un très grand ensemble de données et vouloir obtenir beaucoup de flexibilité pour changer entre différentes fonctions de perte	Fonctionne bien sur de grandes quantités de données et de fonctions Fonction personnalisable qui offre différentes fonctions de perte	Résultats moins interprétables -Beaucoup de paramètres à régler	
<b>PassiveAggressiveRegressor</b>	linear	Ajusté en utilisant une variante de la perte charnière. Efficace pour l'apprentissage à grande échelle avec plusieurs points de données et caractéristiques.			Oui		Oui	Le nombre de caractéristiques et d'échantillons est très élevé souhaitez obtenir rapidement des résultats.	Vous disposez d'un très grand ensemble de données et vous souhaitez obtenir rapidement des résultats.	Fonctionne bien sur de grandes quantités de données et de fonctionnalités	Résultats moins interprétables Ne sont pas très utilisés (et donc peu de ressources en ligne)	
<b>RANSACRegressor</b>	linear	Un modèle linéaire conçu pour traiter les valeurs aberrantes dans les données et/ou les données corrompues. Ajuste de manière itérative un modèle linéaire à un sous-ensemble de données. Si le modèle s'adapte mieux aux données en ligne que le modèle précédent, il est enregistré comme le meilleur modèle. Cette opération est effectuée pendant un nombre spécifié d'itérations ou jusqu'à ce qu'un critère d'arrêt soit rempli.		Oui				Valeurs aberrantes dans les étiquettes de l'ensemble de données	Vous avez des valeurs aberrantes dans les étiquettes de votre ensemble de données.	Il s'adapte mieux aux valeurs aberrantes de grande taille dans la direction Y que les autres algorithmes de calcul des valeurs aberrantes. Plus rapide que TheilSen et s'adapte beaucoup mieux à un grand nombre d'échantillons	S'effondre avec un grand nombre de caractéristiques d'entrée. -Ignore toutes les données qu'il considère comme aberrantes	

# 5. Modèles - sklearn



Modèle	Package	Description	Overfit?	Gère outliers?	Gère gd nbre var?	adaptive learning regularization	Large dataset	Non linear	Usage?	Usage+?	Avantages	Désavantage
TheilSenRegressor	linear	Un modèle linéaire conçu pour traiter les valeurs aberrantes dans les données et/ou les données corrompues. Calcule les solutions des moindres carrés sur un certain nombre de sous-échantillons de données, puis détermine la médiane L1 de ces calculs pour choisir le meilleur modèle.		Oui					Valeurs aberrantes dans les caractéristiques de l'ensemble de données	Il y a des valeurs aberrantes dans les caractéristiques de votre ensemble de données.	mieux gérer les valeurs aberrantes de taille moyenne dans la direction X	S'effondre avec un grand nombre de caractéristiques d'entrée. Ignore toutes les données qu'il considère comme aberrantes
HuberRegressor	linear	Un modèle linéaire conçu pour traiter les valeurs aberrantes des données et/ou les données corrompues. Il n'ignore pas les valeurs aberrantes, mais leur accorde plutôt un poids plus faible.		Oui					Données aberrantes et recherche de l'algorithme le plus rapide	Vous voulez des analyses rapides des données en ignorant les valeurs aberrantes.	Plus rapide que RANSAC et TheilSen (tant que le nombre d'échantillons n'est pas trop grand) N'ignore pas complètement les points de données qu'il considère comme aberrants	S'effondre avec un grand nombre de caractéristiques d'entrée
DecisionTreeRegressor	tree	Développe un arbre de décision en divisant sur les caractéristiques d'entrée. Très similaire à un arbre de décision typique, sauf que les noeuds de sortie correspondent à des sorties linéaires.						Oui	Données non linéaires regroupées dans des buckets	Les données ne sont pas linéaires et sont plutôt composées de ""buckets"". Nombre d'échantillons > nombre de caractéristiques Il y a des caractéristiques dépendantes dans les données d'entrée. DTR gère bien ces corrélations.	Peut exporter la structure de l'arbre pour voir sur quelles caractéristiques l'arbre se divise. Gère bien les données éparses et corrélées. Possibilité d'ajuster le modèle pour résoudre les problèmes d'overfitting	Risque d'ajustement excessif, surtout lorsque le nombre de caractéristiques est proche ou supérieur au nombre d'échantillons.
GaussianProcessRegressor	gaussian_process	Crée un modèle en prenant une distribution sur un certain nombre de fonctions ajustées aux données. Met à jour le poids de ces différentes fonctions en utilisant la règle de Baye.						Oui	Données non linéaires, pas sûr de la structure des données	Lorsque les données sont non linéaires ou que vous n'êtes pas sûr de la structure des données et que vous voulez un modèle qui s'adapte bien aux données d'entrée. Ne sont pas concernés par l'overfitting"	S'adapte très bien aux données de structures diverses	Très enclin à l'ajustement excessif et plus difficile à éviter que DecisionTreeRegressor. -Très coûteux en termes de calcul

# 5. Modèles - sklearn



Modèle	Package	Description	Overfit?	Gère outliers?	Gère gd nbre var?	adapte ve learns regular ization	Large dataset	Non linear	Usage?	Usage+?	Avantages	Désavantage
MLPRegressor	neural_network	Fonctionne comme un réseau neuronal avec plusieurs neurones à chaque couche et des fonctions d'activation non linéaires entre chaque couche. Un certain nombre de paramètres peuvent être réglés pour obtenir des résultats optimaux.		Oui			Oui	Données non linéaires, beaucoup de caractéristiques importantes	Les données se composent de plusieurs caractéristiques et de modèles linéaires de lutte Toutes ou la plupart des caractéristiques d'entrée sont importantes pour faire des prédictions".	Peut apprendre des modèles en temps réel Les réseaux neuronaux peuvent obtenir des résultats impressionnants Peuvent traiter un grand nombre de caractéristiques importantes	Nécessité de mettre les données à l'échelle -Difficile d'interpréter les résultats Beaucoup de paramètres à régler	
KNeighborsRegressor	neighbors	Crée un modèle basé sur les k plus proches voisins à un point donné. Où k est un argument d'entrée.					Oui	Données non linéaires, l'interprétabilité est importante, caractéristiques non importantes	Lorsque vous n'êtes pas sûr de la structure de vos données et que vous souhaitez un modèle qui s'ajuste bien. Non concerné par l'overfitting L'interprétabilité est importante	S'adapte très bien aux données de structures variées Plus facile à interpréter que les autres modèles non linéaires	Très impacté par les valeurs aberrantes et les données corrompues Il faut beaucoup plus d'échantillons que de caractéristiques pour obtenir des résultats de qualité. Difficulté à traiter un grand nombre de caractéristiques	
RadiusNeighborsRegressor	neighbors	Crée un modèle basé sur tous les plus proches voisins dans un rayon r donné pour un point donné. Où r est un argument d'entrée.					Oui	Données non linéaires, l'interprétabilité est importante	Lorsque vous n'êtes pas sûr de la structure de vos données et que vous souhaitez un modèle qui s'ajuste bien Non concerné par l'overfitting L'interprétabilité est importante	S'adapte très bien aux données de structures variées Plus facile à interpréter que les autres modèles non linéaires	Très impacté par les valeurs aberrantes et les données corrompues Il faut beaucoup plus d'échantillons que de caractéristiques pour obtenir des résultats de qualité. Difficulté à traiter un grand nombre de caractéristiques	

# 5. Modèles - sklearn



Modèle	Package	Description	Overfit?	Gère outliers?	Gère gd nbre var?	adaptive learns regularization	Large dataset	Non linear	Usage?	Usage+?	Avantages	Désavantage
SVR	svm	Une implémentation de l'algorithme de la machine à vecteurs de support pour effectuer une régression. Le type de noyau peut être changé entre 'linéaire', 'poly', 'rbf', 'sigmoïde', ou une version personnalisée pour changer la façon dont le modèle apprend.		Oui				Oui	Données non linéaires, grand nombre de caractéristiques sans importance	Lorsque vous avez un jeu de données avec un grand nombre d'éléments Vous voulez que le modèle apprenne à partir d'un sous-ensemble de données qu'il juge le plus important. Vous n'êtes pas concerné par l'interprétabilité.	Efficace avec de nombreuses fonctionnalités Fournit de la flexibilité pour tester différents modèles avec des options d'algorithme de noyau	Très difficile d'interpréter les résultats Calculs coûteux pour les grands ensembles de données. Problèmes liés aux échelles variables des données d'entrée (peuvent être évités en mettant les données à l'échelle)
NuSVR	svm	Similaire à SVR avec les mêmes options de noyau. La différence réside dans la manière dont le modèle est implémenté.		Oui				Oui	Données non linéaires, grand nombre de caractéristiques non importantes.	Lorsque vous avez un jeu de données avec un grand nombre d'éléments Vous voulez que le modèle apprenne à partir d'un sous-ensemble de données qu'il juge le plus important. Vous n'êtes pas concerné par l'interprétabilité.	Efficace avec de nombreuses fonctionnalités Fournit de la flexibilité pour tester différents modèles avec des options d'algorithme de noyau	Très difficile d'interpréter les résultats Calculs coûteux pour les grands ensembles de données. Problèmes liés aux échelles variables des données d'entrée (peuvent être évités en mettant les données à l'échelle). Moins utilisé en pratique que le SVR (donc moins d'aide de la part de la communauté)
LinearSVR	svm	Une implémentation de plus bas niveau du SVR qui ne considère que les solutions linéaires de la machine à vecteurs de support. Mais ce faisant, elle est beaucoup plus efficace en termes de calcul pour les grands ensembles de données.		Oui					SVR, mais avec un très grand nombre d'échantillons et/ou de caractéristiques.	Vous souhaitez implémenter le SVR sur un très grand jeu de données (>100k).	Efficace avec beaucoup de caractéristiques S'adapte mieux aux grands ensembles de données que SVR et NuSVR	Difficile d'interpréter les résultats Problèmes liés aux échelles variables des données d'entrée (peuvent être évités par la mise à l'échelle des données)

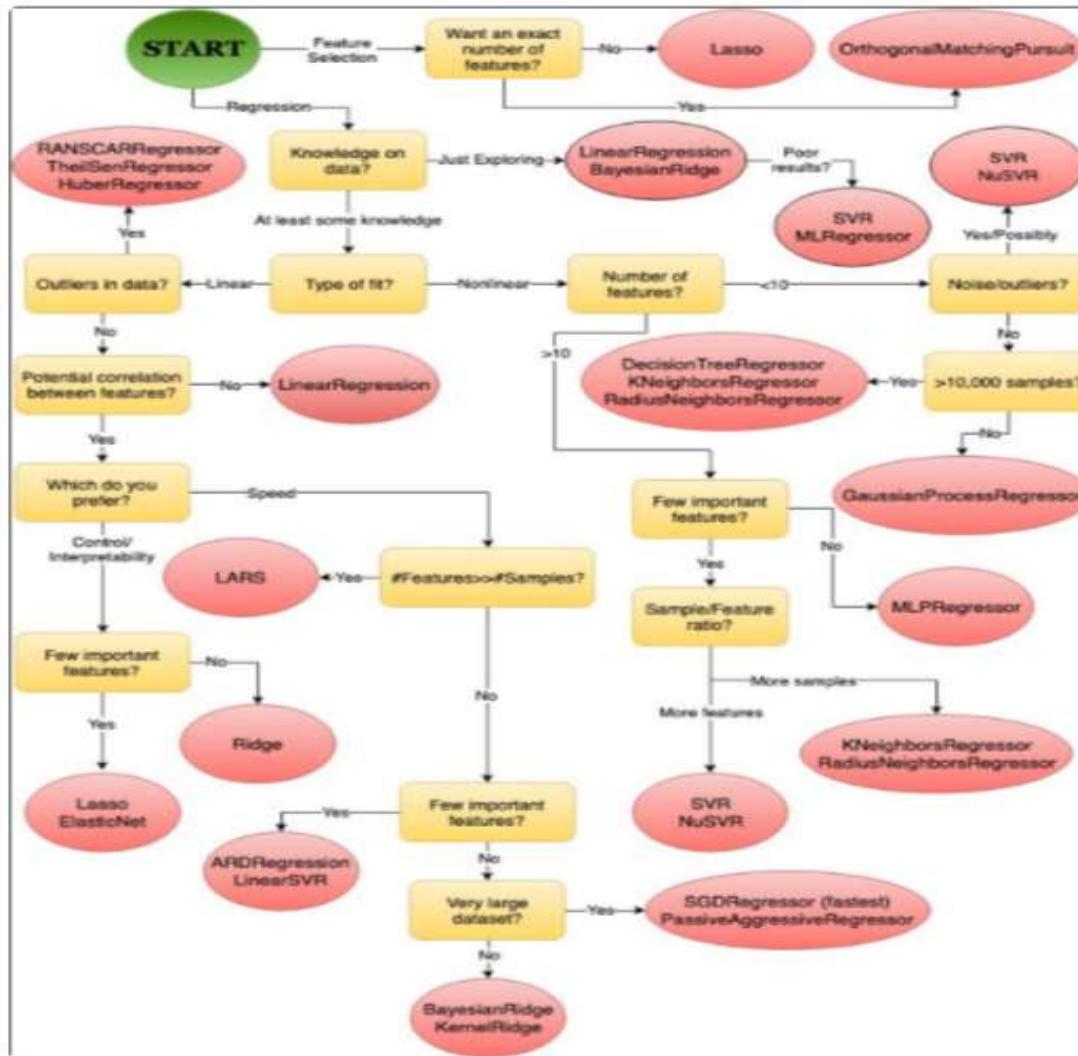
# 5. Modèles - sklearn



Modèle	Package	Description	Overfit?	Gère outliers?	Gère gd nbre var?	adaptive learns regularization	Large dataset	Non linear Usage?	Usage+?	Avantages	Désavantage
KernalRidge	kernel_ridge	Une implémentation kernelisée de Ridge. Les performances sont très similaires à celles du SVR, mais les résultats sont moins épars.			Oui				Une autre déviation du SVR Vous voulez tester un algorithme en plus du SVR qui optimise une fonction de perte différente ?	Efficace avec de nombreuses fonctionnalités	
IsotonicRegression	isotonic	Ajuste une ligne par morceaux aux données qui n'est pas décroissante. Utilise la fonction d'erreur quadratique moyenne pour ajuster la ligne.					Oui	Le jeu de données présente des sauts importants dans les valeurs données qu'il serait difficile d'étiquettes.	Il existe de grands écarts entre les valeurs d'un ensemble de données qui seraient difficiles à ajuster à une ligne continue.	Souvent utilisé pour approximer les résultats prédictifs par d'autres modèles moins interprétables. Peut gérer une grande variance dans les points de données. Résultats interprétables	Utilisé peu dans la pratique et peu utile pour l'application est limité à l'augmentation (ne peut pas traiter des données qui ne sont pas en augmentation)

Source : <https://towardsdatascience.com/choosing-a-scikit-learn-linear-regression-algorithm-dd96b48105f5>

## 5. Modèles - sklearn



# 5. Modèles – Régression linéaire

La **régression linéaire** est un modèle linéaire qui suppose une relation linéaire entre les variables d'entrée (variables indépendantes "x") et la variable de sortie (variable dépendante "y"), de sorte que "y" peut être calculé à partir d'une combinaison linéaire des variables d'entrée (x).

Pour une variable à entrée unique, la méthode est appelée régression linéaire simple, tandis que pour des variables à entrées multiples, elle est appelée régression linéaire multiple.

Les coefficients sont trouvés **en maximisant la vraisemblance** (la probabilité d'observer nos n observations étant donné  $\beta$ , si erreurs normalement distribuée et centrées en 0 et les variables iid (indépendantes et identiquement distribuées) et non corrélées)  $\Leftrightarrow$  **minimiser la somme des carrés des erreurs**

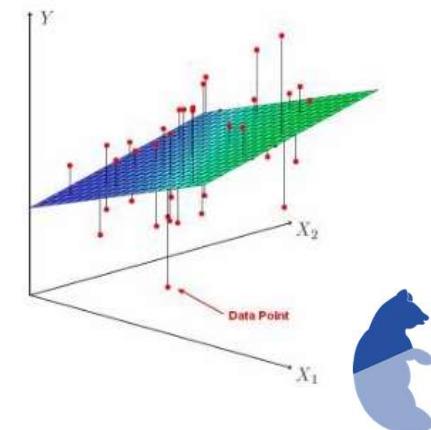
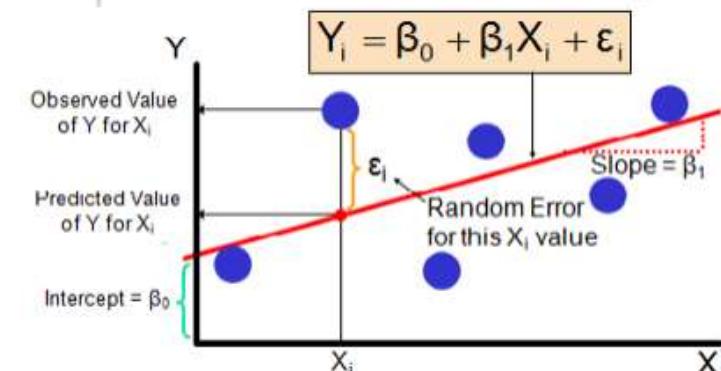
$$\min_{\beta \in \mathbb{R}^2} \left( \sum_{i=1}^n (y^i - f_{\beta}(x^i))^2 \right)$$

**Avantage :**

- Interprétable. Pas de paramètres à régler.
- Si plus de features que de données de tests. Gros jeux de données ou grandes dimensions (features).

**Inconvénient :**

- Si les variables sont CORRELÉES : coefficient très variant et solution non unique, dur à interpréter  $\rightarrow$  et problème de sur apprentissage  $\rightarrow$  régulariser.
- Sujets sur-apprentissage (pas régularisation).
- Ne peut pas contrôler la complexité du modèle (pas de param).



# 5. Modèles – Régularisation



**Régularisation** : Pour limiter le sur-apprentissage, on peut utiliser une technique, la **régularisation**, qui consiste à contrôler simultanément **l'erreur du modèle sur le jeu d'entraînement et la complexité du modèle**. → **minimiser une fonction objective** qui est la somme d'un terme d'erreur et d'un terme mesurant la complexité du modèle.

$$\arg \min_{\beta \in \mathbb{R}^{p+1}} (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta) + \lambda \text{Regularisateur}(\beta)$$

Le **régularisateur**, qui mesure la complexité du modèle, est une fonction des poids  $\beta$  du modèle.

$\lambda$  = **hyperparamètre**, le **coefficent de régularisation**, qui contrôle l'importance relative du terme d'erreur et du terme de régularisation.

- Plus  $\lambda$  est grand, plus le terme de régularisation est important.
- Plus il est petit, est plus l'erreur est importante
- s'il est suffisamment faible (et en particulier s'il est égal à zéro), on retrouvera la solution de la régression linéaire non-régularisée.

Quelle valeur donner à cet hyperparamètre ? En général, c'est une question que l'on réglera en utilisant une **validation croisée**.

## 5. Modèles – Ridge

**Régularisation de Tykhonov** : La régularisation de Tykhonov est un cas particulier de régularisation, dans lequel on utilise pour régulariser la régression linéaire **le carré de la norme euclidienne l2** du vecteur de poids  $\beta$ .

Il faut optimiser :  $\arg \min_{\beta \in \mathbb{R}^{p+1}} \|y - X\beta\|_2^2 + \lambda \|\beta\|_2^2$  ou minimiser la somme des carrés des erreurs

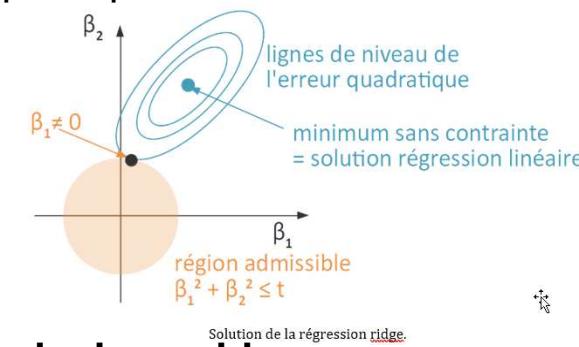
$$||\beta||_2^2 = \sum_{j=0}^p \beta_j^2$$

Le modèle linéaire que l'on apprend en résolvant cette équation est appelé **régression ridge**.

Elle réduit les poids des coefficients de pondération (les plus petits possibles) associés à chacune des variables.

Comment la solution de la régression ridge évolue-t-elle en fonction du coefficient de régularisation  $\lambda$  ?

- Si  $\lambda$  est très grand, alors la régularisation prend le dessus, le terme d'erreur n'importe plus et la solution est  $\beta=0$ , on pénalise la complexité du modèle et plus on a des  $\beta$  petits..
  - À l'inverse, si  $\lambda$  est très faible, le terme de régularisation n'est plus utilisé,
  - et on retrouve la solution de la régression linéaire non régularisée.



**Paramètre :** alpha à régler

## Avantage :

- La régression ridge admet *toujours* une solution *analytique unique*.
  - La régression ridge permet **d'éviter le surapprentissage en restreignant l'amplitude des poids.**
  - La régression ridge a un effet de **sélection groupée** : les variables corrélées ont le même coefficient.

### Inconvénient :

- Standardiser les variables



# 5. Modèles – Lasso



**Lasso** (*Least Absolute Shrinkage and Selection Operator*): modèle **parcimonieux de sélection de variables** et de **réduction de dimension supervisée** : les variables qui ne sont pas nécessaires à la prédiction de l'étiquette sont éliminées → *annuler* certains coefficients.

Les variables qui auront un coefficient égal à zéro ne feront plus partie du **modèle**, qui en sera simplifié d'autant.

Régularisation = norme  **$\ell_1$**  :  $\|\beta\|_1 = \sum_{j=1}^p |\beta_j|$

Il faut optimiser :  $\arg \min_{\beta \in \mathbb{R}^{p+1}} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1$ . mais pas de solution explicite donc algorithme de gradient pour le résoudre.

**Paramètre** : alpha à régler, contrôle l'impact sur la valeur des coefficients.

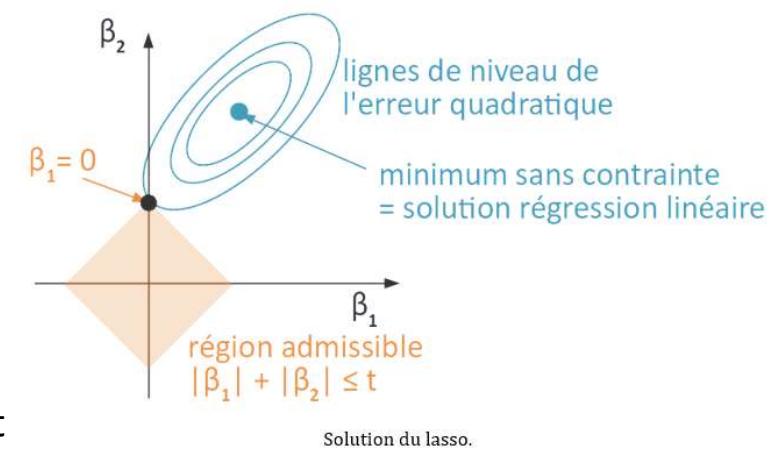
Basse : moins complexe et bon résultat, très basse : surapprentissage.

**Avantage** :

- Parcimonieux : **sélection de variables** et de **réduction de dimension supervisée** : les variables qui ne sont pas nécessaires à la prédiction de l'étiquette sont éliminées.

**Inconvénient** :

- Instable : Si plusieurs variables corrélées contribuent à la prédiction de l'étiquette, le lasso va avoir tendance à choisir *une seule* d'entre elles (affectant un poids de 0 aux autres), plutôt que de répartir les poids équitablement comme la régression ridge, mais aléatoire → instable



# 5. Modèles – Elastic Net



**Elastic net**, qui combine les deux termes de régularisation en un (norme  $\ell_2$  nous permet d'éviter le sur-apprentissage avec une solution unique, et l'utilisation de la norme  $\ell_1$  d'avoir un modèle parcimonieux mais instable).

$$\arg \min_{\beta \in \mathbb{R}^{p+1}} \|y - X\beta\|_2^2 + \lambda ((1-\alpha)\|\beta\|_1 + \alpha\|\beta\|_2^2)$$

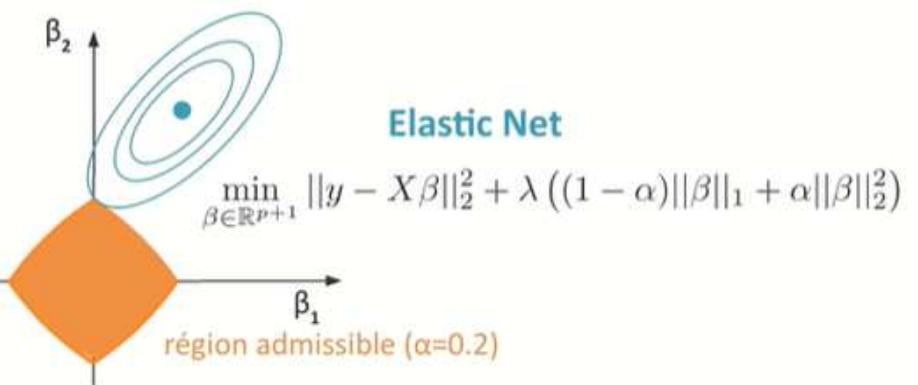
L'**Elastic net** combine les normes  $\ell_1$  et  $\ell_2$  pour obtenir une solution *moins parcimonieuse* que le lasso, mais plus stable et dans laquelle toutes les variables corrélées pertinentes pour la prédiction de l'étiquette sont sélectionnées et reçoivent un poids identique.

$0 \leq \alpha \leq 1$

- Si alpha = 0  $\rightarrow$  LASSO
- Si alpha = 1  $\rightarrow$  RIDGE

## Avantage :

- Stabilité : plus stable et dans laquelle toutes les variables corrélées pertinentes pour la prédiction de l'étiquette sont sélectionnées et reçoivent un poids identique.



## Inconvénient :

- 2 hyperparamètres, alpha et lambda ce qui demande plus de Ressources computationnelles.