

Chapitre : REGRESSION LINEAIRE appliquée au ML

Niveau: Ing2_GL

Dr.Manel SEKMA

Plan

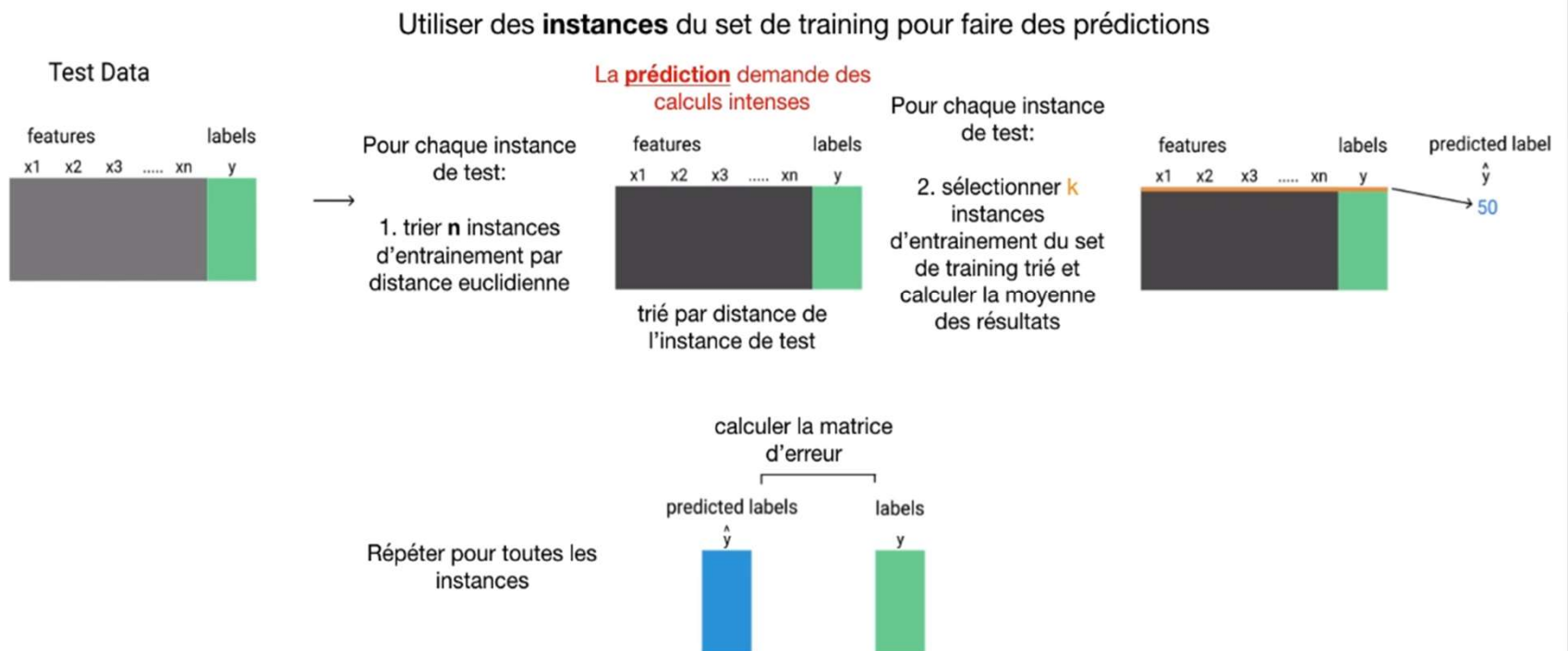
- ▶ Apprentissage basé sur une instance Vs. Apprentissage basé sur un modèle
- ▶ Régression linéaire vs Régression logistique
- ▶ Modèle de régression linéaire simple
- ▶ Moindres carrés ordinaires
- ▶ Gradient Descente
- ▶ Modèle de régression linéaire multiple
- ▶ Régression régularisée

Apprentissage basé sur une instance Vs. Apprentissage basé sur un modèle

Training Process



Testing Process



Apprentissage basé sur une instance Vs. Apprentissage basé sur un modèle

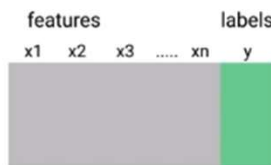
Apprentissage basée sur un modèle

$$y = a_1 x_1 + a_2 x_2 + a_3 x_3 + \dots + a_n x_n$$

Adapter le modèle aux données

Training Process

Training Data



Select Features

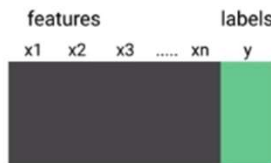
Trouver le meilleur modèle qui **approxime** le set de training

L'entraînement est l'étape la plus demandeuse en terme de calcul

$$\hat{y} = 2.5x_1 + 11x_2 + 3.5x_3 + \dots + a_n x_n$$

Testing Process

Test Data



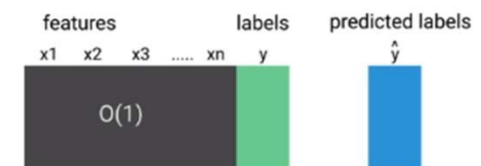
Prédire un libellé pour une instance en passant les valeurs des caractéristiques dans le modèle entraîné.

$$\hat{y} = 2.5x_1 + 11x_2 + 3.5x_3 + \dots + a_n x_n$$

La **prédiction** demande peu de ressources

Utiliser le **modèle** pour faire des prédictions

Calculer la métrique d'erreur



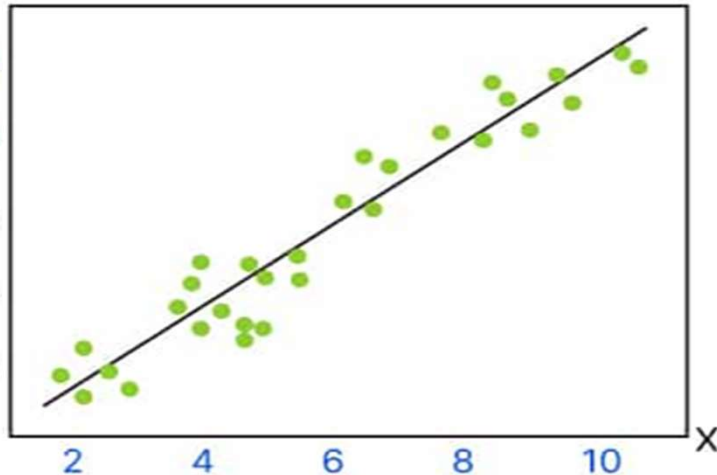
Recap

	Apprentissage basé sur une instance	Apprentissage basé sur un modèle
Principe	Mémorise les exemples d'entraînement et compare directement les nouvelles instances	Apprend un modèle général à partir des données
Moment de la prédiction	Au moment du test	Après la phase d'apprentissage
Exemples d'algorithmes	k-Nearest Neighbors (k-NN)	Régression linéaire, Arbre de décision, Réseau de neurones
Avantage	Simple, sans phase d'entraînement complexe	Rapide à la prédiction, bonne généralisation
Inconvénient	Lent à la prédiction, sensible au bruit	Phase d'apprentissage parfois coûteuse
Type d'approche	Basée sur la similarité	Basée sur la modélisation

Régression linéaire vs Régression logistique

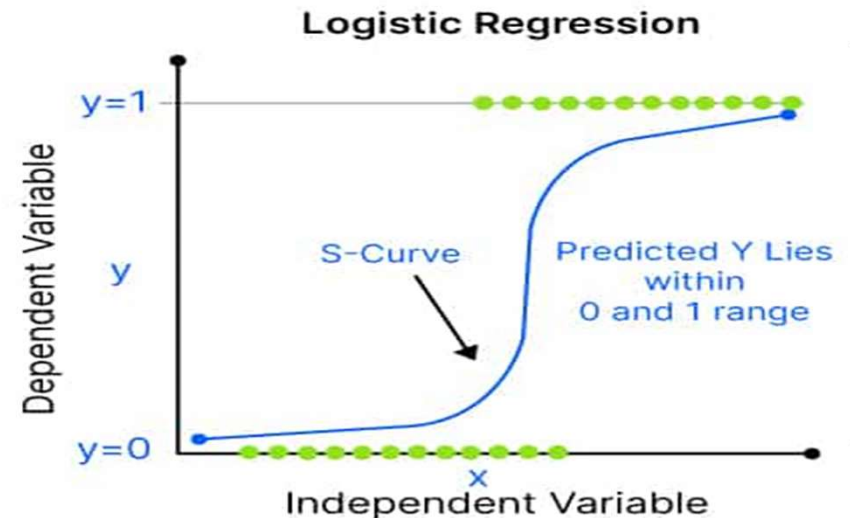
Régression Linéaire

- Type de variable dépendante : Continue
- Modèle mathématique : Equation linéaire
- Objectif : Prédiction de valeurs numériques.
- Sortie du modèle : Valeurs continues
- Exemple: la prévision des stocks



Régression Logistique

- Type de variable dépendante : Binaire ou nominale
- Modèle mathématique : Fonction logistique (sigmoïde)
- Objectif : Prédiction de la probabilité d'appartenance à une classe
- Sortie du modèle : Probabilité entre 0 et 1
- Exemple: la classification d'images.

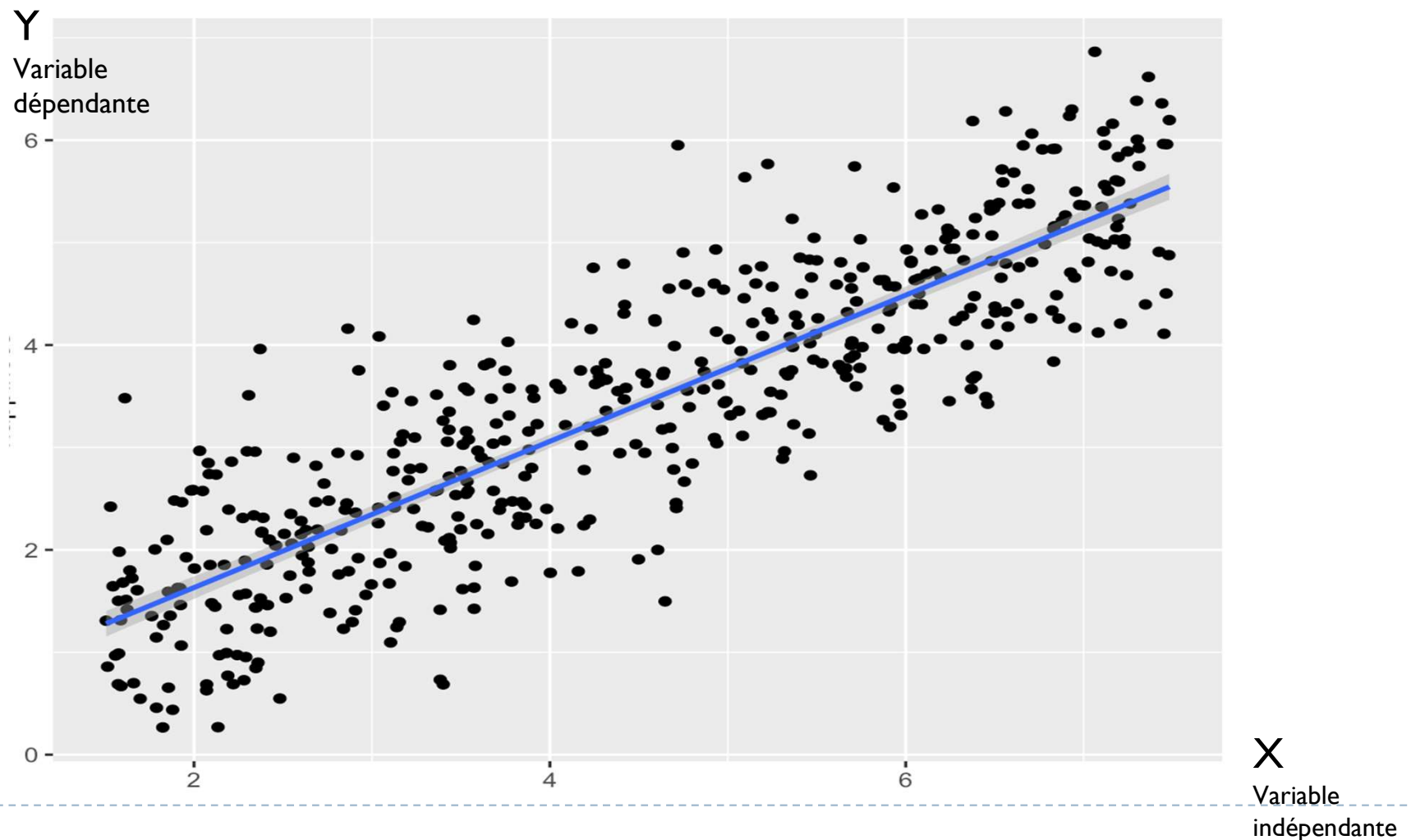


Introduction à la Régression Linéaire

- ▶ La régression linéaire est une technique d'analyse statistique qui permet de modéliser la relation entre une variable dépendante (Y) et une ou plusieurs variables indépendantes (X).
- ▶ Apprentissage basé sur un modèle.
- ▶ Elle est basée sur l'hypothèse que la relation entre ces variables est linéaire, c'est-à-dire qu'elle peut être approximée par une ligne droite.

Quand Utiliser la Régression Linéaire ?

- La régression linéaire est utilisée pour **prédire** ou **expliquer** la valeur d'une variable dépendante en fonction de variables indépendantes.



Objectif de la Régression Linéaire

- ▶ L'objectif principal de la régression linéaire est de trouver la meilleure ligne (modèle) qui représente la relation entre les variables, minimisant ainsi l'erreur de prédiction.
- ▶ Cette ligne est appelée "ligne de régression" ou "droite de régression".

Types de Régression Linéaire

Il existe plusieurs types de régression linéaire,

- ▶ La régression linéaire simple (une seule variable indépendante)
- ▶ La régression linéaire multiple (plusieurs variables indépendantes).
- ▶ D'autres variantes incluent la régression linéaire robuste, la régression linéaire polynomiale, etc

Régression Linéaire Simple

- ▶ La régression linéaire simple est une technique d'analyse statistique qui modélise la relation entre une variable dépendante (Y) et **une seule variable indépendante (X)**.
- ▶ L'équation de la régression linéaire simple :

$$Y = \alpha + \beta X + \varepsilon$$

- ▶ **α** et **β** sont les coefficients de régression, **ε** est l'**erreur résiduelle**.

Forme de la Régression Linéaire Simple

- ▶ L'objectif est de trouver les meilleurs coefficients (α et β) pour ajuster la ligne qui minimise l'erreur résiduelle.

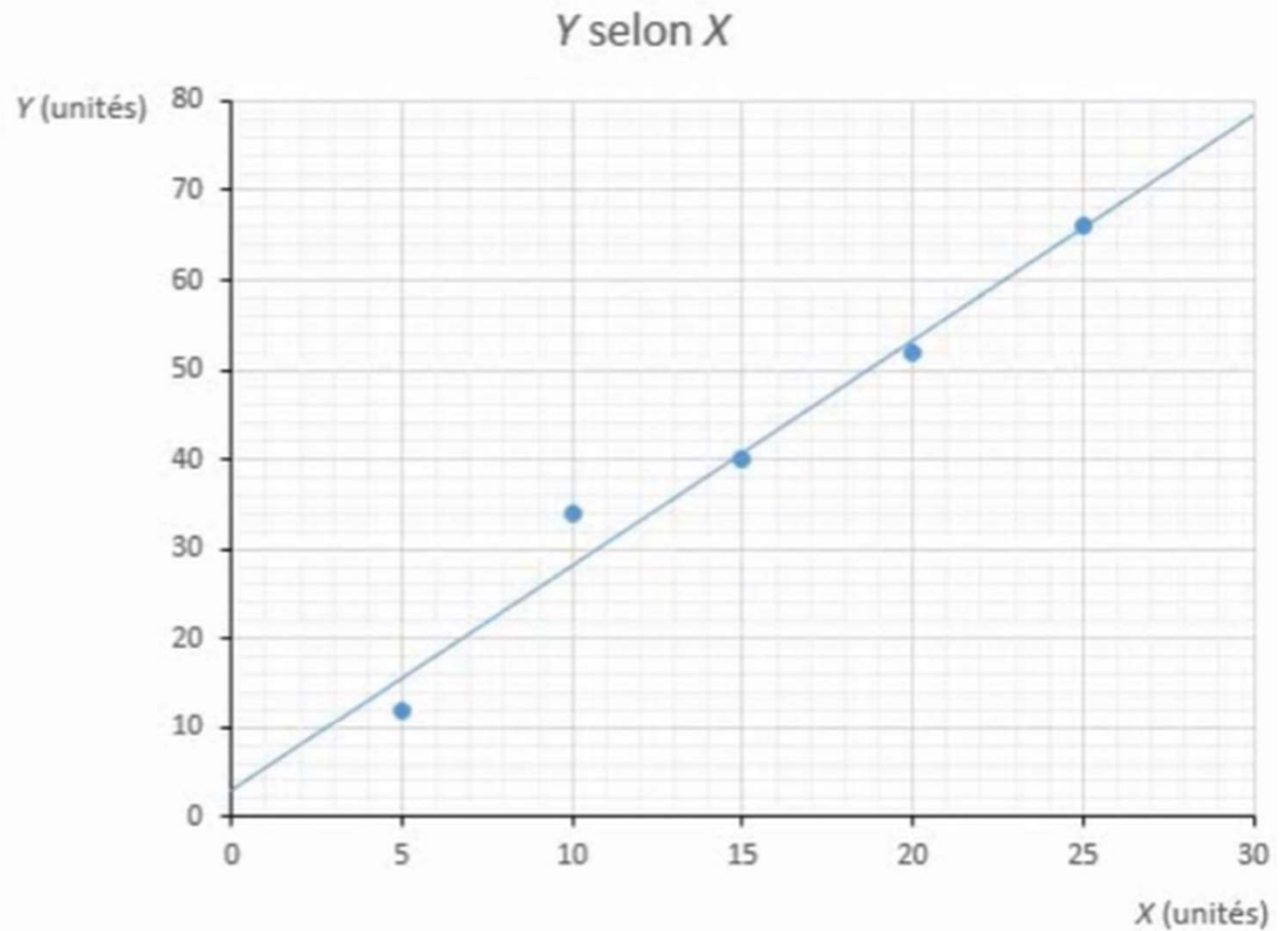
$$Y = \alpha + \beta X + \varepsilon$$

- ▶ L'erreur résiduelle mesure la différence entre les valeurs prédites et les valeurs réelles.

Exemple

Y selon X	
X	Y
(unités)	(unités)
± 1	± 2
5	12
10	34
15	40
20	52
25	66

X = une variable
Y = autre variable



$$y = 2,52x + 3$$

Trouver les Coefficients: méthode des moindres carrés

- ▶ Dans les modèles de régression linéaire les coefficients sont trouvés en utilisant la **méthode des moindres carrés**.
- ▶ La méthode des moindres carrés est une technique d'optimisation utilisée pour ajuster un modèle aux données **en minimisant la somme des carrés des écarts** entre les valeurs prédites et les valeurs réelles.

L'équation de la régression linéaire simple :

$$Y = \alpha + \beta X + \varepsilon$$

Rappelons

- ▶ $\text{Var}(X)$ représente la variance de la variable indépendante $X \rightarrow$ mesure la variation de la variable indépendante X en elle-même
- ▶ La covariance mesure comment les deux variables X et Y varient ensemble,

▶

Trouver les Coefficients: méthode des moindres carrés

Interprétation des Coefficients:

- ▶ α est l'ordonnée à l'origine et représente la valeur de Y lorsque X est égal à zéro.
- ▶ β est la pente de la ligne de régression et indique comment Y change lorsque X augmente d'une unité.

Les formules des valeurs estimées sont :

1. La pente $= \beta = \frac{\text{cov}(x, y)}{S_x^2} = \frac{\sum_{i=1}^n (x_i - \bar{X})(y_i - \bar{Y})}{\sum_{i=1}^n (x_i - \bar{X})^2}$

2. L'ordonnée à l'origine $\alpha = \bar{Y} - \beta \bar{X}$

Méthode des Moindres Carrés

- ▶ Le modèle peut servir à prédire Y si on connaît le point x :

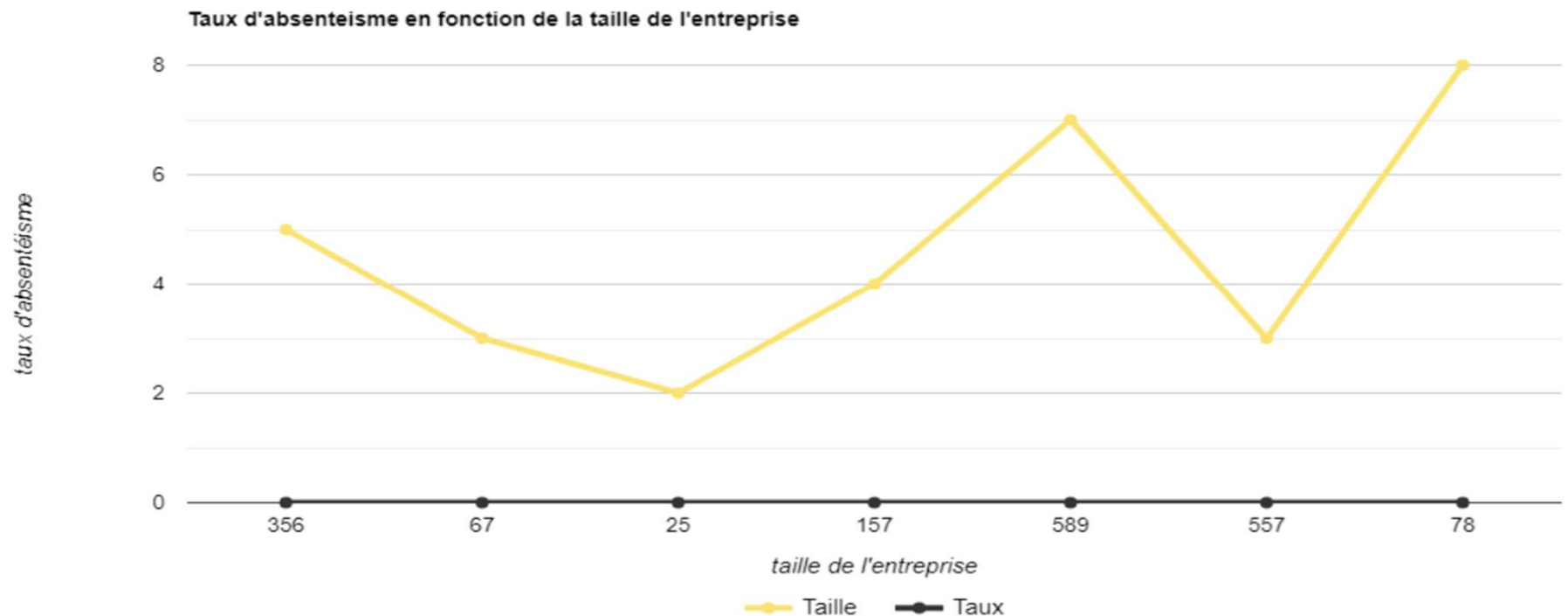
$$Y = \alpha + \beta X$$

- ▶ Exemple: Si $\alpha = 3$ et $\beta = 100$ alors pour $x = 255$ la valeur de Y donné par le modèle est de $Y = 3 + 100 \times 255 = 25503$
- ▶ Dans ce modèle on dit qu'un changement d'une unité dans X est associé à un changement de 100 unités dans Y , ce qui reflète la pente de la régression.

Exemple. Méthode des Moindres Carrés

Considérons la relation entre le nombre d'employés d'une usine et le taux d'absentéisme.

- ▶ On veut prédire le **taux d'absentéisme** en fonction de la **taille de l'entreprise**, mesurée en termes du nombre d'employés.
- ▶ L'objectif est de modéliser ce taux en fonction de la taille de l'entreprise, afin de déterminer:
 1. s'il y a une relation entre les deux,
 2. quelle est l'influence de la taille de l'entreprise sur le taux d'absentéisme.



Exemple. Méthode des Moindres Carrés

Soit des valeurs suivantes de 7 entreprises :

X_i Nombre d'employés	356	67	25	157	589	557	78
Y_i Taux d'absentéisme %	5	3	2	4	7	3	8

$$1. \text{ La pente } = \beta = \frac{\text{cov}(x, y)}{S_x^2} = \frac{\sum_{i=1}^n (x_i - \bar{X})(y_i - \bar{Y})}{\sum_{i=1}^n (x_i - \bar{X})^2}$$

$$2. \text{ L'ordonnée à l'origine } \alpha = \bar{Y} - \beta \bar{X}$$

I- Calculer les moyennes de X et Y

- La moyenne de X noté $\bar{x} = (356+67+25+157+589+557+78)/7 = 261,29$
- La moyenne $\bar{y} = (5+3+2+4+7+3+8)/7 = 4,5714$

Exemple. Méthode des Moindres Carrés

1. La pente = $\beta = \frac{\text{cov}(x, y)}{S_x^2} = \frac{\sum_{i=1}^n (x_i - \bar{X})(y_i - \bar{Y})}{\sum_{i=1}^n (x_i - \bar{X})^2}$

2. L'ordonnée à l'origine $\alpha = \bar{Y} - \beta \bar{X}$

X_i Nombre d'employés	356	67	25	157	589	557	78
Y_i Taux d'absentéisme %	5	3	2	4	7	3	8

► 2- Covariance de X et Y:

$$\text{Cov}(x, y) = \sum_i (x_i - \bar{x})(y_i - \bar{y}) = 715.8571$$

► 3- $S^2_x = \sum (x_i - \bar{x})^2 = 341861.4$

Exemple. Méthode des Moindres Carrés

► $Y = \alpha + \beta X$

$$\beta = \frac{\text{Cov}(x,y)}{S^2_x} = 715,8571 / 341861,4 = 2,0940 \times 10^{-3} \approx 0,002$$

$$\alpha = \bar{Y} - \beta \bar{X} = 4,5714 - (2,0940 \times 10^{-3}) \times 261,29 = 4,0243$$

L'équation de la régression sera alors: $y = 4,0243 + 0,002 x$

► **Interprétation:**

- Une entreprise ayant 200 employés devrait avoir un taux d'absentéisme de $4,024 + 0,002 \times (200) = 4,424$
- De plus une augmentation de 100 du nombre des employés augmente de $0,002 \times 100 = 0,2$ le taux en %

Exercice

- Dans le but d'expliquer la consommation sur carte de crédit, des données sur le revenu et sur la dépense sont obtenues :

Dépenses	Revenu
8900	21000
9400	25000
14500	30000
25400	45000
26600	50000

Le modèle estimé doit permettre d'obtenir les dépenses sur carte de crédit en fonction des revenus.

La variable dépendante est : $y = \text{Dépenses}$

La variable indépendante est $x = \text{Revenu}$

Optimiser les paramètres du modèle:

Gradient Descent

- ▶ Le Gradient Descent (GD) est un algorithme d'optimisation utilisé pour ajuster les paramètres d'un modèle afin de minimiser une fonction de coût.
- ▶ Convergence: le GD assure que l'optimisation converge vers un minimum local de la fonction de coût,
- ▶ Pour rappel, nous souhaitons que la *somme des carrés des erreurs* soit la plus petite possible.

Optimiser les paramètres du modèle: Gradient Descent

- ▶ On va non pas prendre la somme mais la moyenne. Cette moyenne est appelée *fonction coût*.
- ▶ Elle s'écrit de la manière suivante :

$$J(\alpha, \beta) = \frac{1}{m} \sum_{i=1}^m \text{erreur}^i = \frac{1}{m} \sum_{i=1}^m ((\alpha \beta^{(i)} + b) - y^{(i)})^2$$

m = représente le nombre

d'observations.

(rem. la littérature divise la fonction coût par $2m$ et non par m , je choisis ici de diviser par m pour ne pas compliquer l'explication).

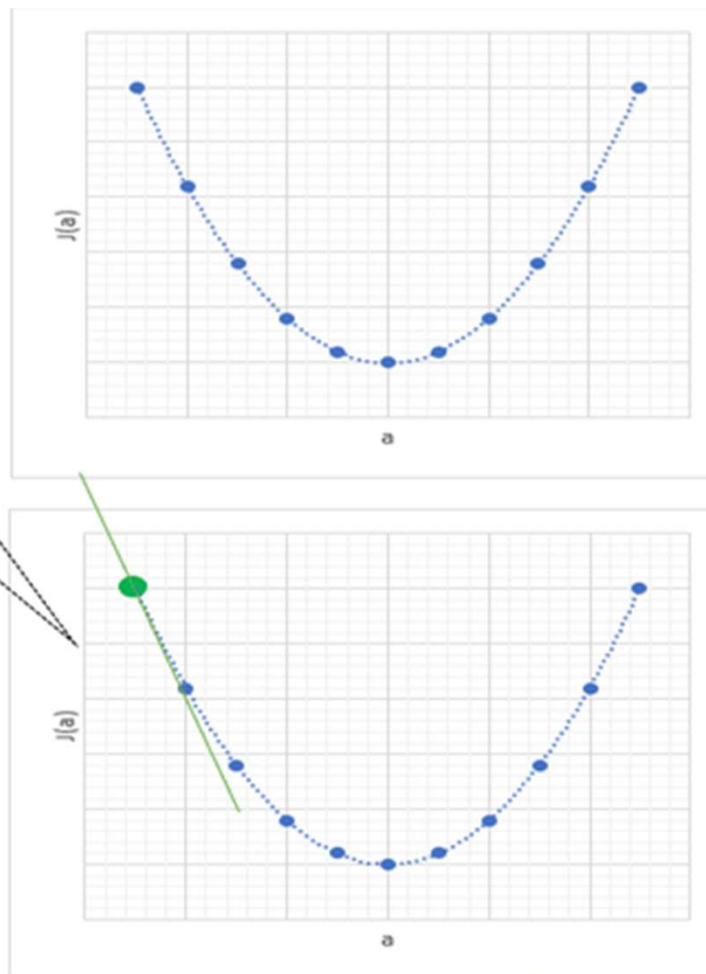
Etape de calcul de GD

Cas I: une seul variable a

$$J(a) = \frac{1}{m} \sum_{i=1}^m \text{erreur}^i = \frac{1}{m} \sum_{i=1}^m (a * x^{(i)} - y^{(i)})^2$$

La machine calcule la dérivée afin de savoir si elle est à gauche ou à droite du point minimum.

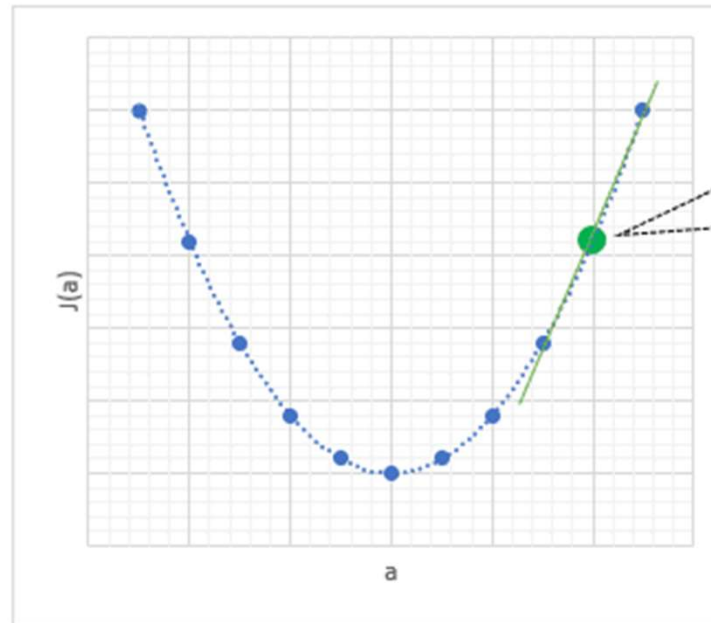
La dérivée est significativement **négative**, il faut **augmenter** significativement la valeur de a.



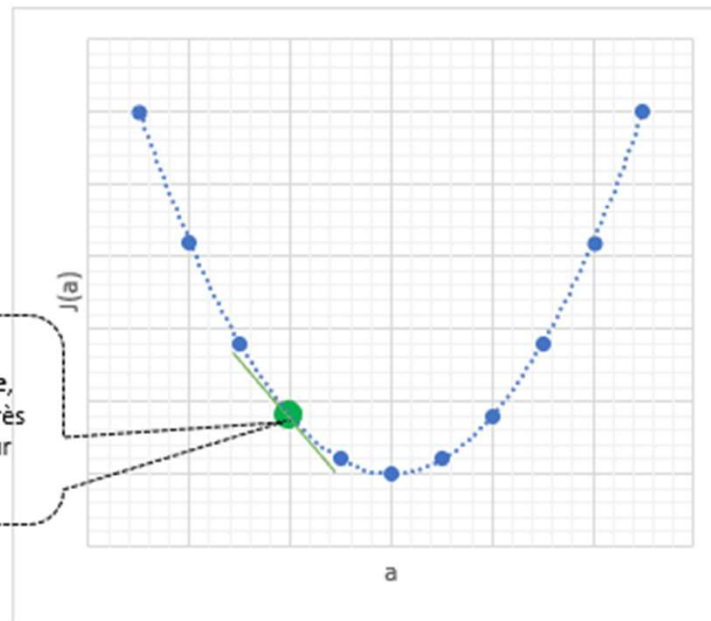
a	J(a)
0	1290,85
0,01	1165,41
0,02	1046,44
0,03	933,92
0,04	827,87
0,05	728,28
0,06	635,15
0,07	548,49
0,08	468,29
0,09	394,55
0,1	327,27
0,11	266,46
0,12	212,11
0,13	164,22
0,14	122,79
0,15	87,83

Etape de calcul de GD

Cas I: une seul variable a



La dérivée est relativement **positive**, il faut **diminuer** relativement la valeur de a .



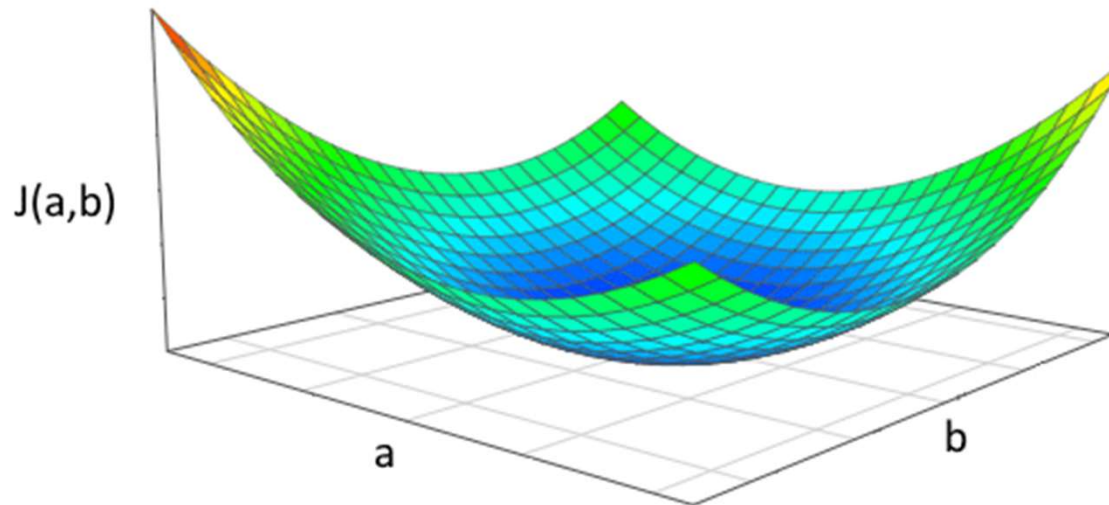
La dérivée est faiblement **négative**, il faut **augmenter** très légèrement la valeur de a .

Etape de calcul de GD

► Cas2: Deux variables a et b

L'objectif reste identique, mais l'ajout d'une nouvelle variable, en l'occurrence b , transforme la fonction coût en un graphique en 3D dimensions.

- la dérivée de $J(a,b)$ par rapport à a ($\frac{\partial J(a,b)}{\partial a}$)
- la dérivée de $J(a,b)$ par rapport à b ($\frac{\partial J(a,b)}{\partial b}$)



Descente de gradient « optimisée »

- Plutôt que proposer aléatoirement à chacune des étapes une nouvelle valeur pour **a**, on va soustraire à **a** un certain pourcentage de la dérivée calculée à l'étape précédente, pourcentage c'est « **learning rate = taux d'apprentissage** ».
- La nouvelle valeur de **a** est calculée de la manière suivante :

$$\text{« Nouveau a »} = \text{« Ancien a »} - \text{« learning_rate »} \times \text{« Dérivée »}$$

	a	b	Dérivée de J(a,b) par rapport à a	Dérivée de J(a,b) par rapport à b	J(a,b)	Conclusion	Learning rate	Nouveau a	Nouveau b
Etape 1	10,00	12,00	1198,00	263,00	6842,25	On continue	0,001	8,80	11,74
Etape 2	8,80	11,74	1065,23	235,07	5421,08	On continue	0,001	7,74	11,50
Etape 3	7,74	11,50	947,14	210,22	4296,93	On continue	0,001	6,79	11,29
Etape 4	6,79	11,29	842,12	188,12	3407,72	On continue	0,001	5,95	11,10
Etape 5	5,95	11,10	748,72	168,47	2704,35	On continue	0,001	5,20	10,94
Etape 6	5,20	10,94	665,65	150,98	2147,96	On continue	0,001	4,53	10,78
Etape 7	4,53	10,78	591,77	135,43	1707,84	On continue	0,001	3,94	10,65
Etape 8	3,94	10,65	526,06	121,60	1359,69	On continue	0,001	3,42	10,53
Etape 9	3,42	10,53	467,62	109,30	1084,29	On continue	0,001	2,95	10,42
Etape...
Etape 2000	0,67	2,48	-0,14	0,68	0,71	On continue	0,001	0,67	2,48

REGRESSION LINEAIRE appliquée au ML

Notions traitées dans cette partie:

- ▶ Régression linéaire vs Régression logistique
- ▶ Modèle de régression linéaire simple
- ▶ Moindres carrés ordinaires
- ▶ Gradient Descente
- ▶ **Modèle de régression linéaire multiple**

► Modèle de régression linéaire multiple

Modèle de régression linéaire multiple

- ▶ Le modèle de régression multiple repose sur les hypothèses suivantes:
 - ▶ Il existe une **relation linéaire** entre les variables dépendantes et les **variables indépendantes**
 - ▶ Les variables **indépendantes** ne sont pas trop fortement corrélées entre elles
 - ▶ y_i les observations sont sélectionnées **indépendamment** et au hasard dans la population
 - ▶ Les résidus doivent être normalement **distribués** avec une moyenne de 0 et une **variance σ**

Modèle de régression linéaire multiple

- La régression linéaire multiple s'exprime comme :

$$\mathbf{a} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

- \mathbf{a} est le vecteur de coefficients,
- \mathbf{X} est la matrice des caractéristiques,
- \mathbf{y} est le vecteur de la variable cible,
- \mathbf{X}^T est la transposée de \mathbf{X} ,
- $(\mathbf{X}^T \mathbf{X})^{-1}$ est l'inverse de la matrice produit de la transposée de \mathbf{X} par \mathbf{X} .

Modèle de régression linéaire multiple

$$y = a_0 + a_1x_1 + \dots + a_px_p + \varepsilon$$

← Résume l'information de y que l'on n'arrive pas à capter avec les (x_1, \dots, x_p) variables prédictives.

Disposant d'un échantillon de taille n , nous cherchons les paramètres estimés $\hat{a} = (\hat{a}_0, \hat{a}_1, \dots, \hat{a}_p)$ qui minimise l'erreur quadratique moyenne (MSE : *mean squared error*).

$$\min_{a_0, a_1, \dots, a_p} \frac{1}{n} \sum_{i=1}^n \left(y_i - \left(a_0 + \sum_{j=1}^p a_j x_{ij} \right) \right)^2$$

Estimateur des moindres carrés ordinaires (écriture matricielle)

$$\hat{a}_{MCO} = (X'X)^{-1}X'Y$$

Régression Linéaire en Python avec Scikit-learn

1- Importation des Bibliothèques

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
```

2 - Division des Données

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
                                                    random_state=42)
```

3- Création du Modèle

```
model = LinearRegression()
model.fit(X_train, y_train)
```

Régression Linéaire en Python avec Scikit-learn

4- Prédiction

```
y_pred = model.predict(X_test)
```

5- Évaluation du Modèle

```
mse = mean_squared_error(y_test, y_pred)
```

Régression régularisée

Ridge – Lasso – Elasticnet

Multicollinéarité vs Corrélation

Corrélation

- ▶ Mesure la relation entre une variable indépendante et la variable dépendante ou entre deux variables indépendantes.
- ▶ Utilise le coefficient de corrélation pour quantifier la force et la direction de la relation.
- ▶ La corrélation peut être utile pour sélectionner des variables pertinentes pour un modèle.

Multicollinéarité vs Corrélation

Multicollinéarité

- ▶ Se produit lorsque deux ou plus **de variables indépendantes** dans un modèle de régression sont **fortement corrélées entre elles**.
- ▶ Impact sur la capacité à **estimer l'effet individuel de chaque variable** indépendante sur la variable dépendante.
- ▶ Peut rendre les coefficients de régression **instables** et **difficiles à interpréter**.
- ▶ **Perturbe la sélection** des variables significatives.

Multicollinéarité vs Corrélation

Multicollinéarité – Exemple I: prédire le prix de maison

- Superficie de la maison (en m^2)
- Nombre de chambres
- Nombre de salles de bains
- Taille du terrain (en m^2)
- Proximité des centres commerciaux
-



le nombre de chambres et le nombre de salles de bains peuvent également être corrélés

➔ **plusieurs variables mesurent la même chose**

Multicollinéarité vs Corrélation

Multicollinéarité – Exemple 2: prédire le prix d'une voiture

- Marque et modèle :
- Année de fabrication
- Kilométrage
- Type de carburant
-



- La distance parcourue par une voiture (kilométrage) en multicollinéarité avec l'âge de la voiture(exprimé en années),

➔ **plusieurs variables mesurent la même chose**

Gérer la multicollinéarité

1. **Éliminer une variable** : Si deux variables sont fortement corrélées, éliminer une du modèle.
2. **Combiner des variables** : Créer une nouvelle variable qui combine les effets des variables corrélées (par exemple, une variable composite pour les caractéristiques de la maison).
3. **Utiliser des techniques de régularisation** : Des méthodes comme la régression **Ridge** ou **Lasso** (utiliser aussi pour gérer le problème du Surajustement) peuvent aider à atténuer les effets de la multicollinéarité en pénalisant les coefficients.

Conséquences de la Multicolinéarité :

- ▶ Coefficients instables.
- ▶ Difficulté d'interprétation des résultats.
- ▶ Augmentation de la variance des estimations.

Importance de la Régularisation

▶ Objectif de la Régularisation :

- ▶ Réduire la complexité du modèle.
- ▶ Améliorer la généralisation sur les données nouvelles.

▶ Techniques Principales :

- ▶ Ridge (L2)
- ▶ Lasso (L1)
- ▶ Elastic Net

▶ Utilisation

- ▶ Applicable à la régression linéaire et logistique, mais aussi à d'autres méthodes dès lors que l'on a des combinaisons linéaires de variables avec des coefficients à estimer (ex. réseaux de neurones, perceptron, SVM linéaire, etc.).

Régression Ridge (L2)

Principe :

- ▶ Ajout d'une pénalité au carré des coefficients.

Formule :

Minimiser $\sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^p \beta_j^2$

λ ($\lambda \geq 0$) est un paramètre (coefficient de pénalité) qui permet de contrôler :
l'impact de la pénalité : à fixer

$\sum_{j=1}^p \beta_j^2$: Fonction de pénalité

➡ L'estimateur Ridge s'écrit alors : $\hat{\beta}_{Ridge} = (X'X + \lambda I_p)^{-1} X'y$ I_p est la matrice identité

- On peut avoir une estimation même si $(X'X)$ n'est pas inversible
- On voit bien que $\lambda = 0$, alors on a l'estimateur des MCO
moindres carrés ordinaires (MCO)

Régression Ridge (L2)

Avantages :

- ▶ Réduit les coefficients mais ne les annule jamais.
- ▶ Gère bien la multicolinéarité en redistribuant les poids des variables corrélées.

Régression Lasso (L1)

Principe :

Ajout d'une pénalité à la somme des valeurs absolues des coefficients.

Formule :

$$\text{Minimiser } \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

Avantages :

- ▶ Peut annuler complètement certains coefficients, ce qui permet la sélection de variables.
- ▶ Efficace pour réduire le surajustement.

Elastic Net

Principe :

- ▶ Combine les pénalités L1 et L2.

Formule :

$$\text{Minimiser } \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda_1 \sum_{j=1}^p |\beta_j| + \lambda_2 \sum_{j=1}^p \beta_j^2$$

Avantages :

- ▶ Gère bien la multicolinéarité tout en permettant la sélection de variables.
- ▶ Efficace lorsque le nombre de variables est supérieur au nombre d'observations.

Exemples Réels d'Utilisation des Techniques de Régularisation(1)

► **Régression Ridge (L2)**

► **Exemple d'Application : Prédiction de Prix Immobiliers**

- **Contexte** : Les prix des maisons sont souvent influencés par de nombreuses caractéristiques (superficie, nombre de chambres, localisation, etc.), qui peuvent être corrélées.
- **Utilisation** : **Ridge** est utilisé pour prédire les prix, car il permet de conserver toutes les variables tout en réduisant les effets des variables corrélées, garantissant ainsi une estimation plus stable des prix.

Exemples Réels d'Utilisation des Techniques de Régularisation(2)

► **Régression Lasso (L1)**

Exemple d'Application : Sélection de Caractéristiques en Biologie

- **Contexte** : Dans des études génétiques, des milliers de gènes peuvent être mesurés pour prédire la présence d'une maladie.
- **Utilisation** : Lasso est utilisé pour sélectionner un sous-ensemble de gènes significatifs tout en éliminant les gènes non pertinents, facilitant ainsi l'interprétation des résultats et la réduction de la complexité du modèle.

Exemples Réels d'Utilisation des Techniques de Régularisation(3)

► **Elastic Net**

Exemple d'Application : Classification de Textes

- **Contexte** : Dans le traitement du langage naturel, des milliers de mots peuvent être présents dans un corpus de texte, créant une matrice de caractéristiques très large.
- **Utilisation** : Elastic Net est utilisé pour gérer la multicolinéarité et effectuer une sélection de caractéristiques simultanément, améliorant ainsi les performances des modèles de classification tout en maintenant l'interprétabilité.

Régularisation avec knn

Importation des bibliothèques nécessaires

```
from sklearn.linear_model import Ridge
```

```
from sklearn.linear_model import Lasso
```

```
from sklearn.linear_model import ElasticNet
```

```
knn = KNeighborsRegressor(n_neighbors=5, algorithm='brute')
```

alpha est le paramètre de régularisation

```
knn= Ridge(alpha=1)
```

```
# knn = Lasso(alpha=0.1)
```

```
# knn=ElasticNet(alpha=0.1)
```

```
knn.fit(X_train, y_train)
```

Prédictions sur l'ensemble de test

```
y_pred = knn.predict(X_test)
```

Calcul de l'erreur quadratique moyenne

```
mse = mean_squared_error(y_test, y_pred)
```

```
print(f"Mean Squared Error (MSE) : {mse}")
```

Affichage des coefficients du modèle

```
print(f"Coefficient : {knn.coef_}")
```

```
print(f"Intercept : {knn.intercept_}")
```

Le Compromis Biais-Variance en Machine Learning

Concepts et Implications pour les Modèles de Prédiction

Compromis biais-Variance

Concepts et Implications pour les Modèles de Prédiction

Définition de Biais et Variance

Biais : Erreur due à des simplifications excessives dans le modèle.

Variance : Erreur due à la complexité excessive du modèle.

Biais ?

- ▶ Imaginez que vous essayez d'estimer la relation entre **la taille et le poids** des personnes en utilisant une **ligne droite** (régression linéaire).
- ▶ Si la **relation réelle est non linéaire** (comme une courbe), votre modèle linéaire **simplifiera trop cette relation**, entraînant des prédictions incorrectes.
- ▶ Cela se traduit par un **biais élevé**, car le modèle ne capture pas correctement la forme réelle de la relation entre les variables.

Biais ?

Caractéristiques d'un Modèle à Biais Élevé :

Sous-apprentissage (Underfitting) : Le modèle n'arrive pas à bien apprendre les motifs dans les données d'entraînement.

Mauvaises performances sur les données d'entraînement et les données de test.

Erreur de biais élevée : L'erreur ne diminue pas même si on augmente la quantité de données.

□ Le biais élevé se produit souvent lorsque l'on choisit des modèles trop simples, comme les modèles linéaires pour des relations non linéaires ou les arbres de décision peu profonds.

Comment Réduire le Biais :

- ▶ Pour réduire le biais, il est nécessaire d'utiliser des modèles plus complexes ou plus flexibles, capables de mieux s'adapter aux données, comme les réseaux de neurones ou les arbres de décision plus profonds.

La variance ?

- ▶ La variance fait référence à l'erreur causée par un modèle trop complexe qui s'adapte trop étroitement aux données d'entraînement,
 - ▶ capturant non seulement les tendances réelles mais aussi le **bruit** ou les fluctuations aléatoires des données.
- ▶ Cela conduit à un problème appelé **surapprentissage** (overfitting).

La variance ?

- ▶ Imaginez que vous essayez de prédire la relation entre la **taille et le poids** des personnes, mais cette fois, vous utilisez un **polynôme de très haut degré** pour ajuster vos données.
- ▶ Bien que ce polynôme puisse parfaitement correspondre aux points de données d'entraînement (même les petites variations ou erreurs), **il sera probablement mauvais pour faire des prédictions sur de nouvelles données**,
 - ▶ car il est trop spécifique aux exemples sur lesquels il a été formé.

La variance ?

Caractéristiques d'un Modèle à Variance Élevée :

- ▶ **Surapprentissage (Overfitting)** : Le modèle apprend non seulement les motifs sous-jacents mais aussi le bruit spécifique des données d'entraînement.
- ▶ **Bonne performance sur les données d'entraînement** mais mauvaise performance sur les données de test.
- ▶ **Sensibilité élevée** aux petites variations dans les données d'entraînement, ce qui rend le modèle instable pour de nouvelles données

Comment Réduire la Variance :

- ▶ Pour réduire la variance,
 - ▶ simplifier le modèle
 - ▶ d'utiliser des techniques de régularisation (comme la régularisation L1/L2),
 - ▶ d'augmenter la taille de l'ensemble de données,
 - ▶ d'utiliser des méthodes d'ensemble comme le bagging et le boosting.

Erreur Totale dans le Modèle

Décomposition de l'Erreur Totale :

**Erreur totale = Erreur due au biais + Erreur due à la variance + Bruit
(erreur irréductible)**

