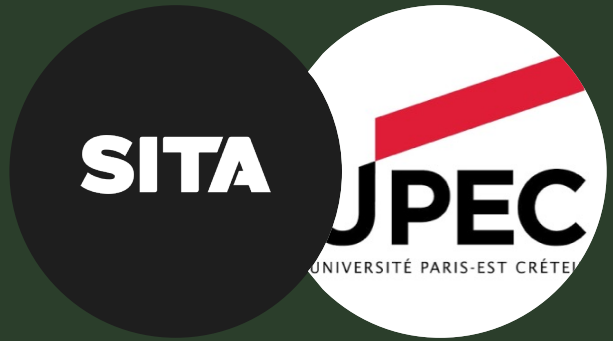



## EPISEN – ING3. SI

# Machine Learning

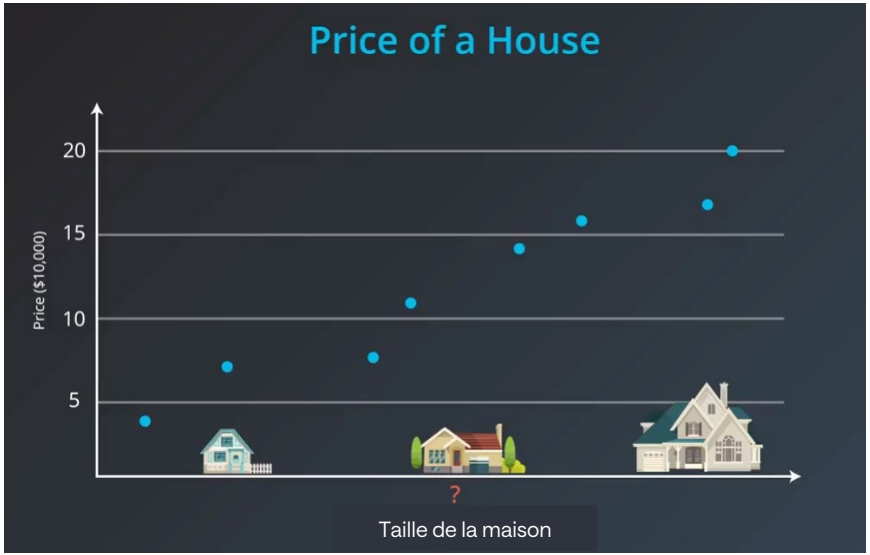
**Abdallah EL HIDALI**
Tech Lead Sita For Aircraft
abdallah.el-hidali@sita.aero

**EPISEN**
2024/2025

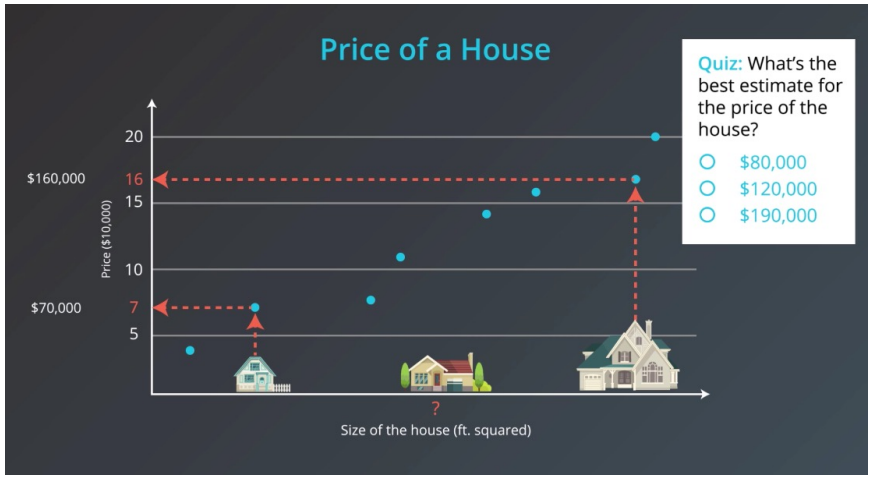
SITA

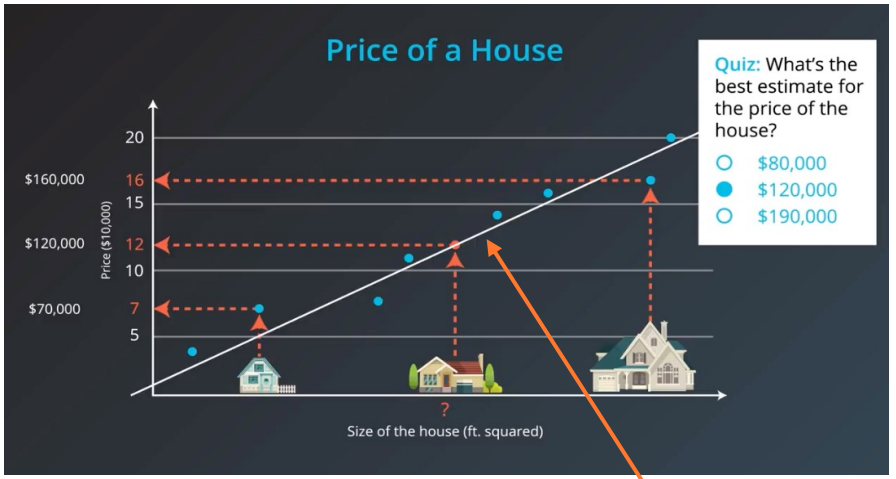
# II. La Régression Linéaire

# La Régression Linéaire: introduction

# Régression Linéaire: introduction



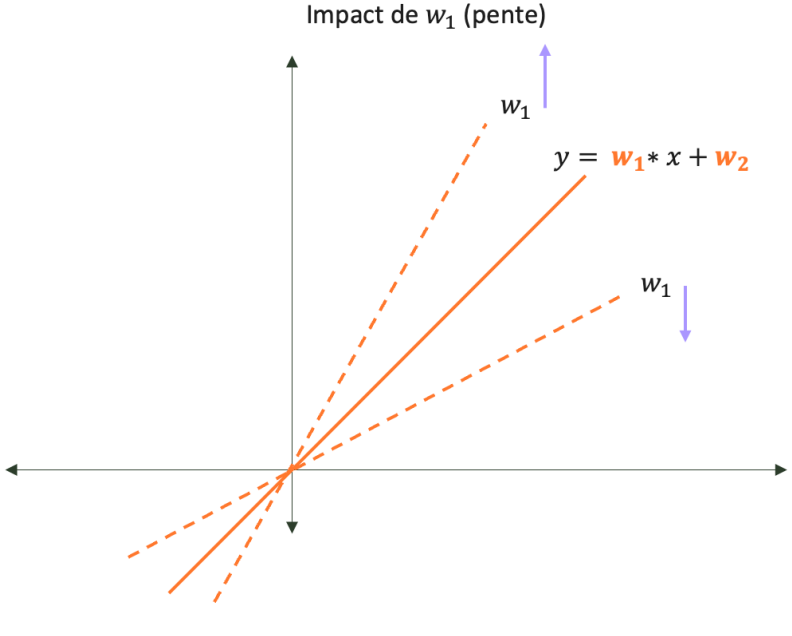



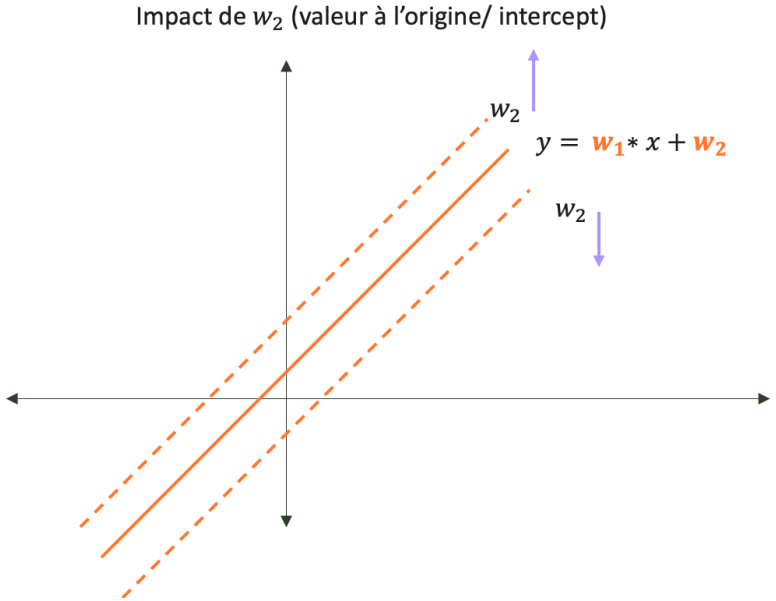

Quelle est la meilleure estimation du prix de la maison ?

- $80,000
- $120,000
- $190,000

Comment trouver cette droite de régression?

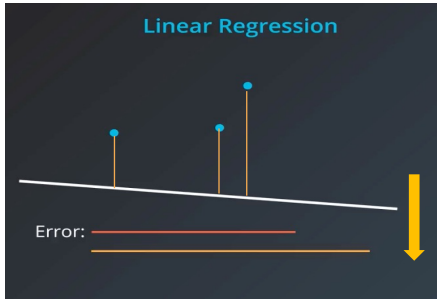
# Régression Linéaire: Ajuster une droite

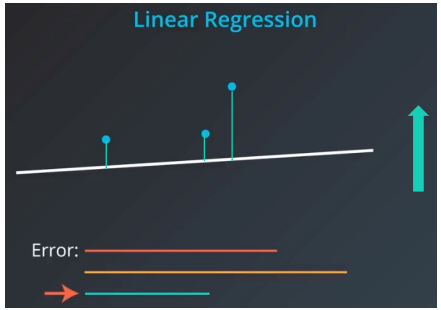


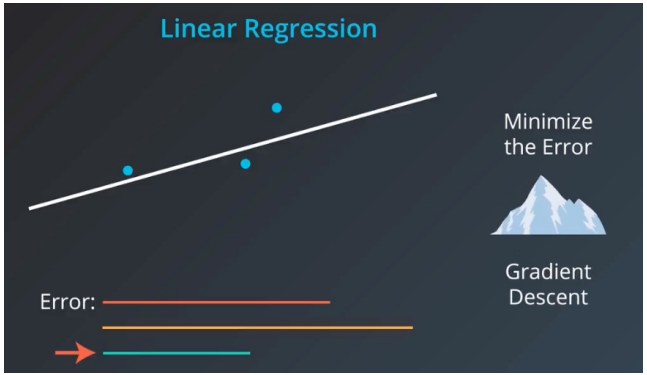
# Régression Linéaire: Ajuster une droite avec les données



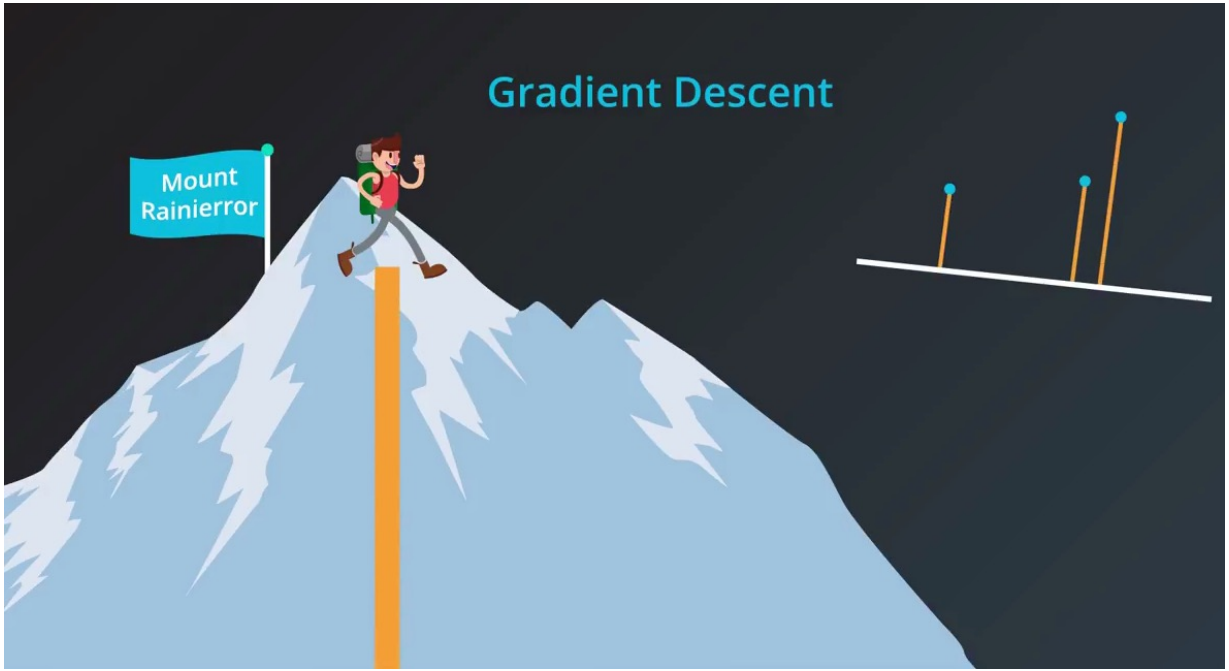


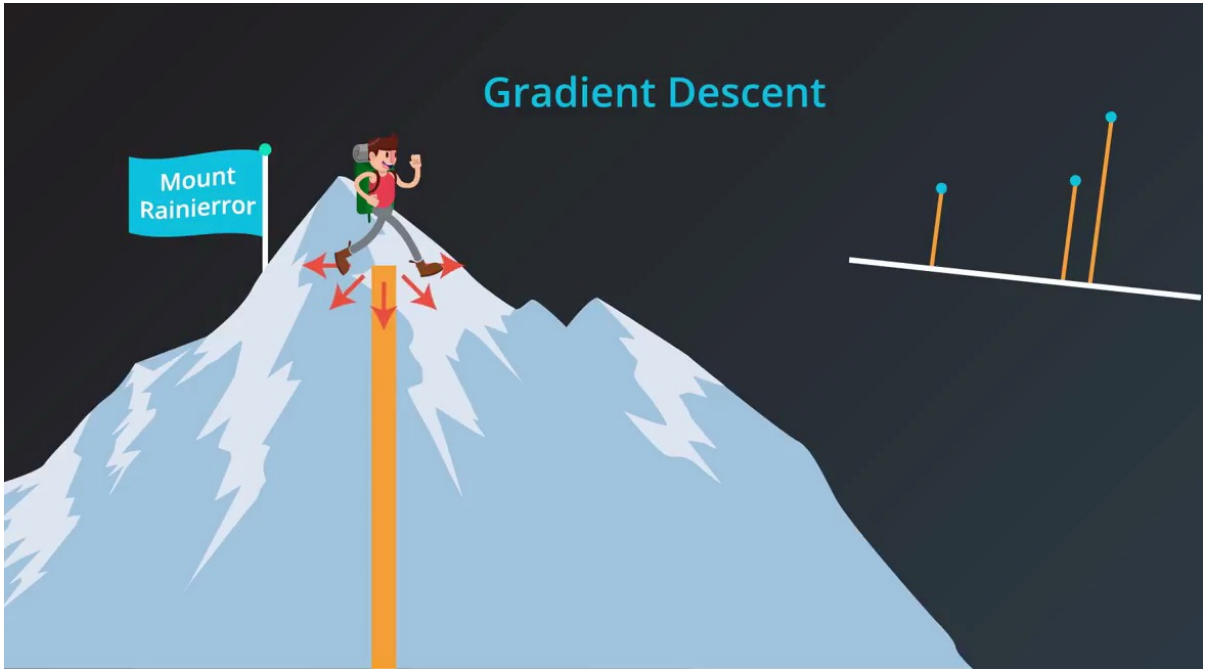

- On répète ce processus plusieurs fois jusqu'à ce qu'on minimise l'erreur.
- **La descente de gradient** est l'algorithme employé pour réaliser cette optimisation.



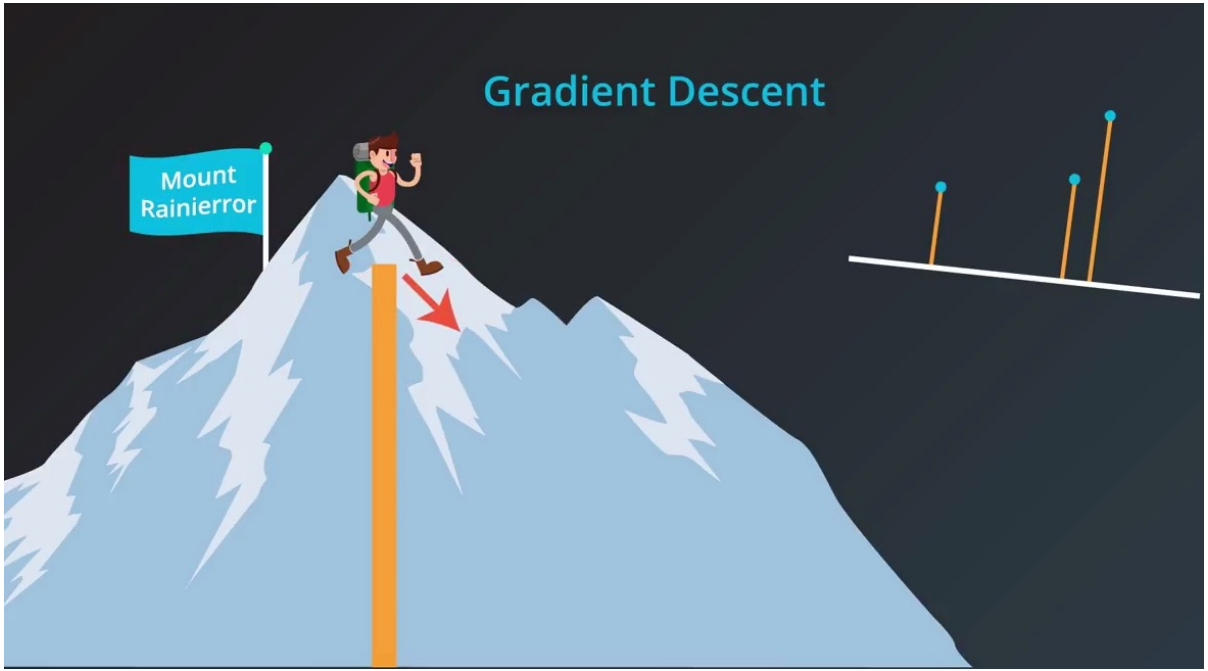
# La Régression Linéaire: La descente de gradient



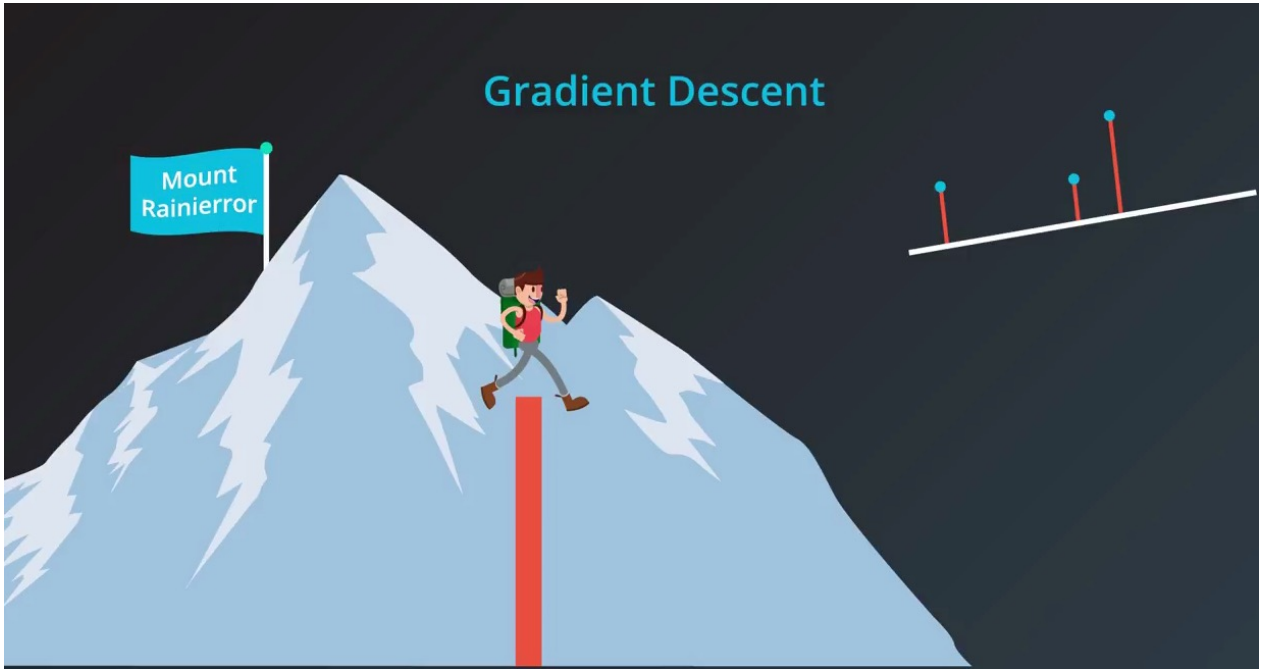
# La Régression Linéaire: La descente de gradient



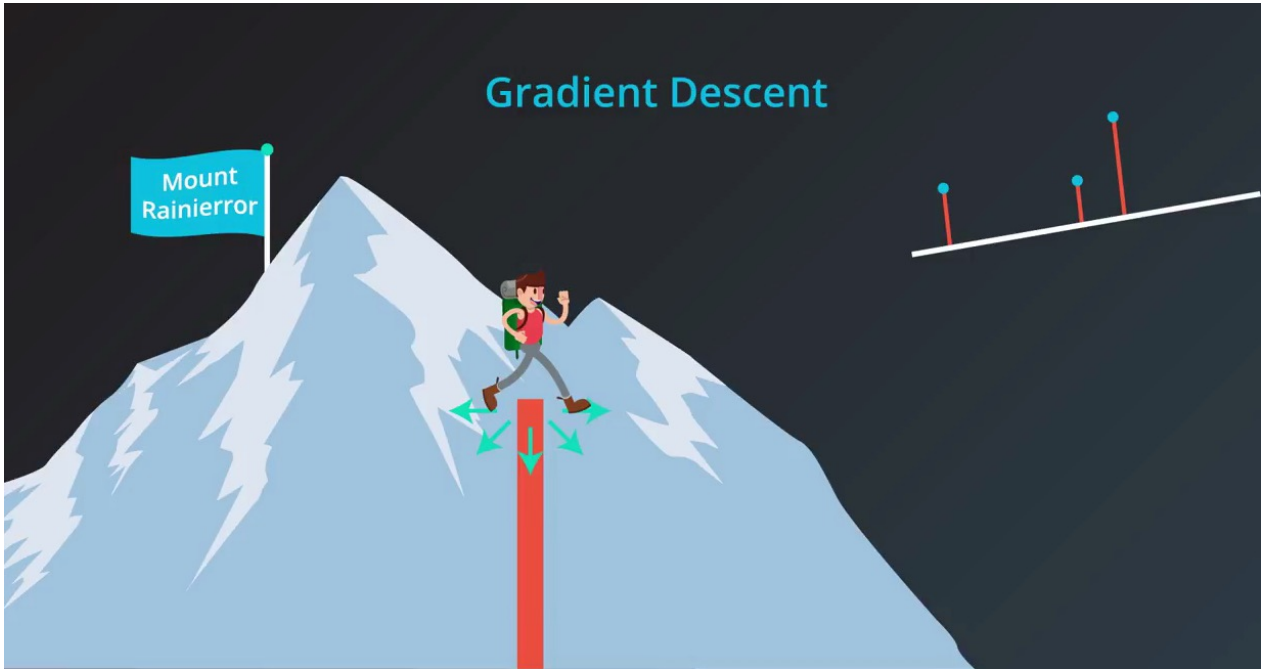
# La Régression Linéaire: La descente de gradient



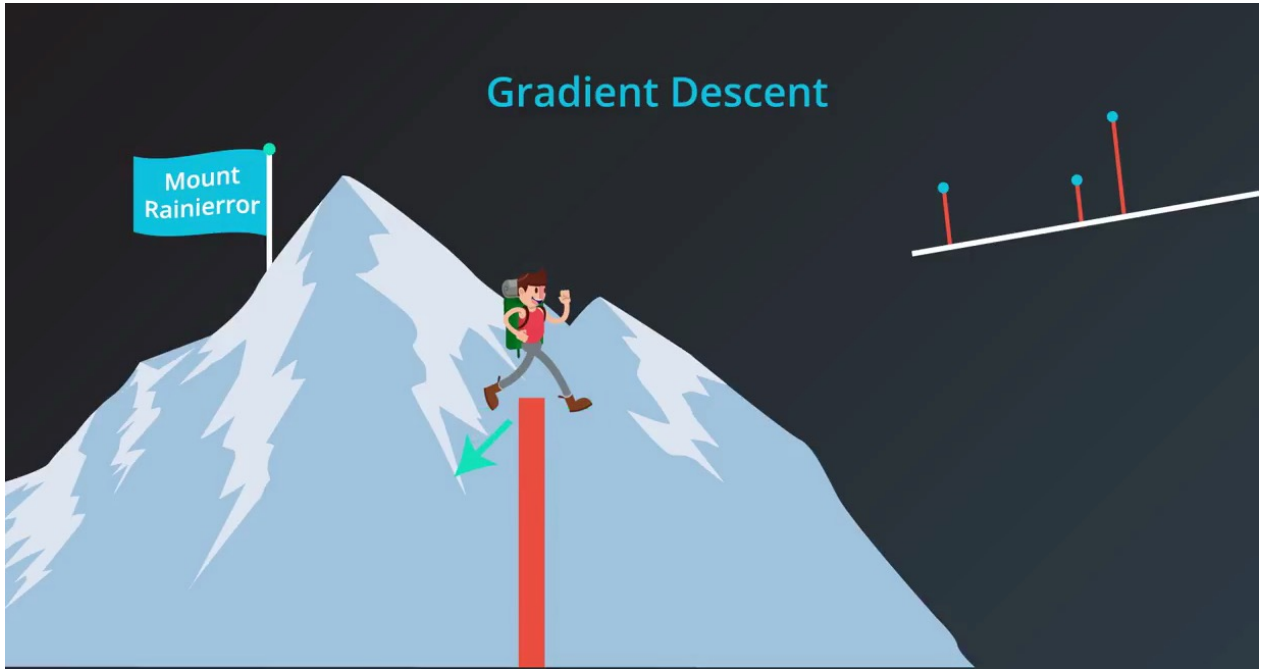
# La Régression Linéaire: La descente de gradient



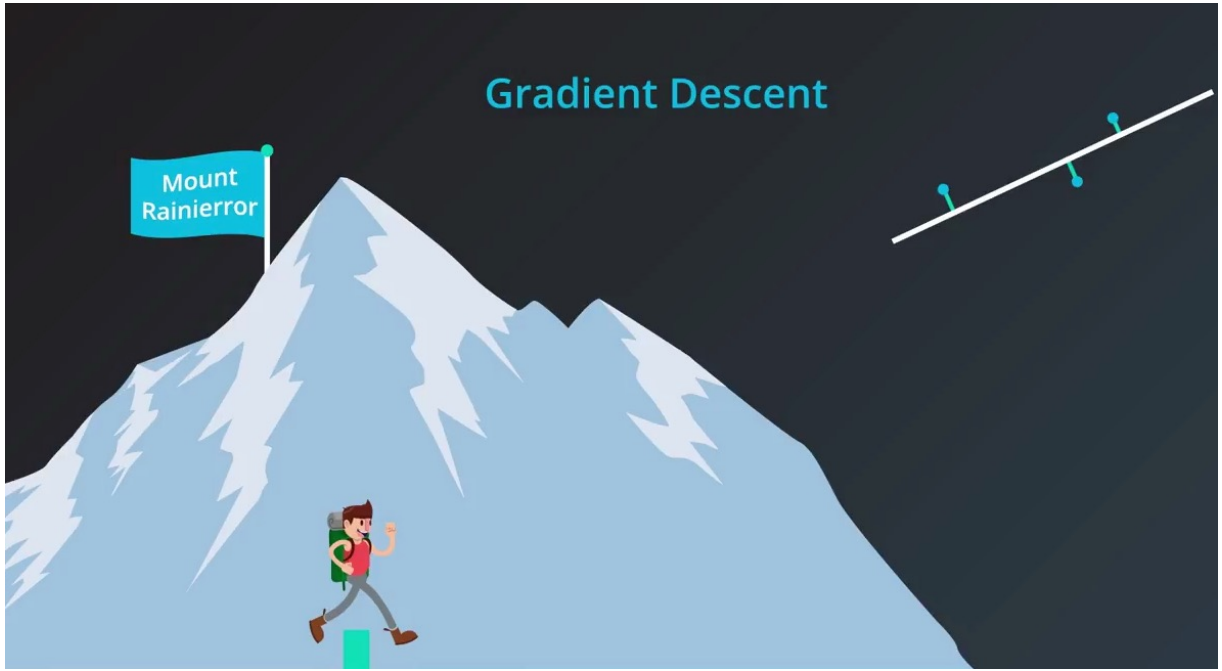
# La Régression Linéaire: La descente de gradient



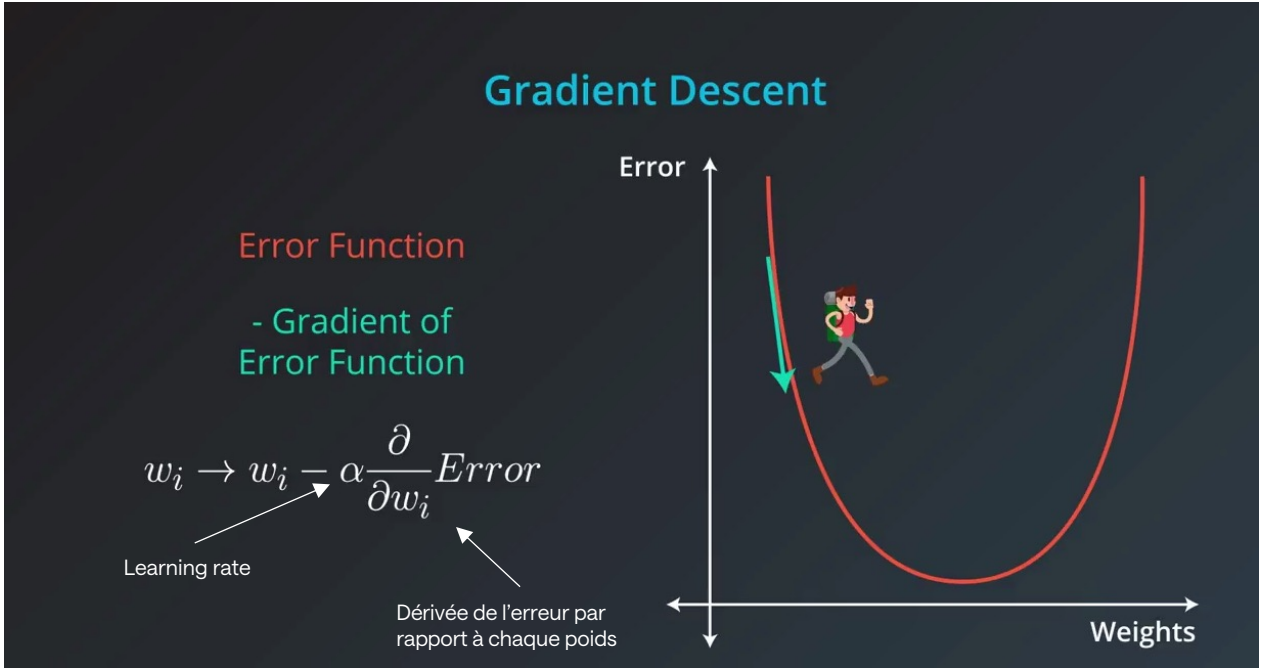
# La Régression Linéaire: La descente de gradient



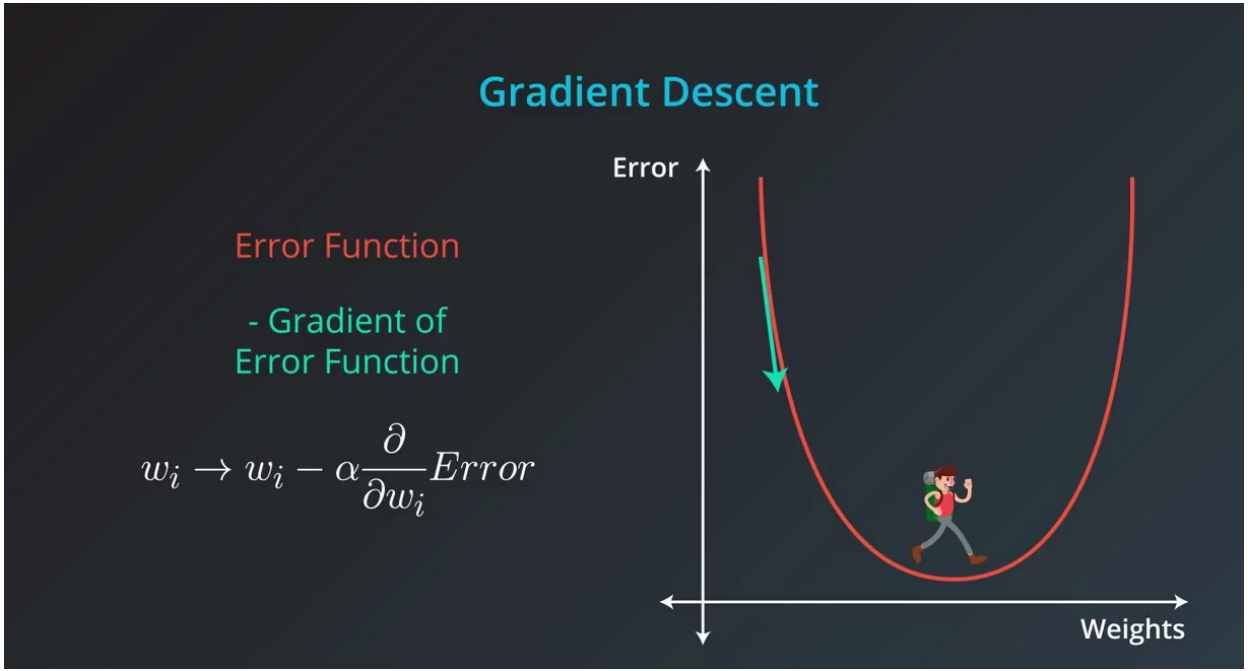
# La Régression Linéaire: La descente de gradient



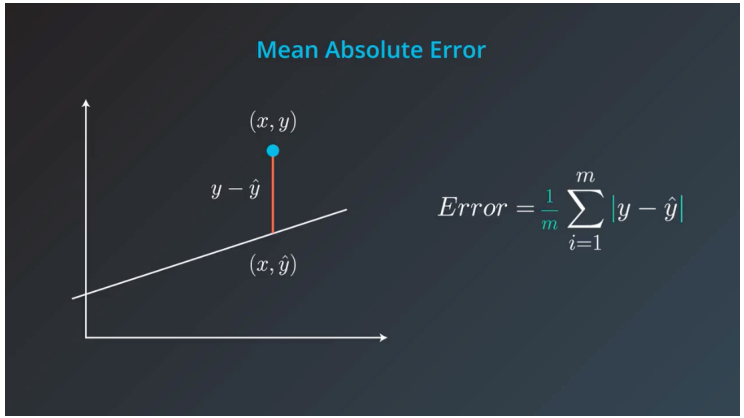
# La Régression Linéaire: La descente de gradient

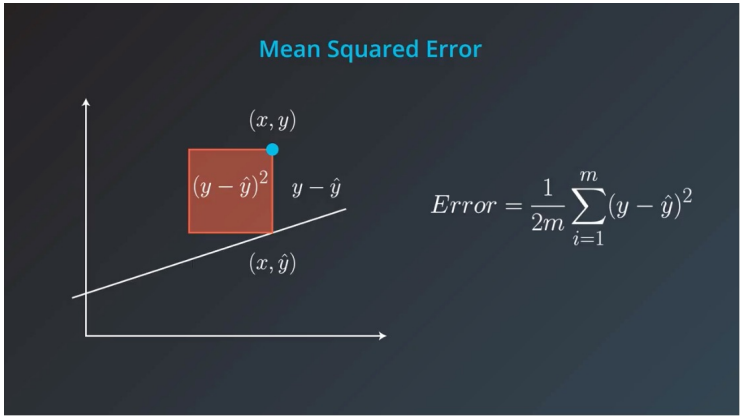


# La Régression Linéaire: La descente de gradient



# La Régression Linéaire: La fonction coût à minimiser



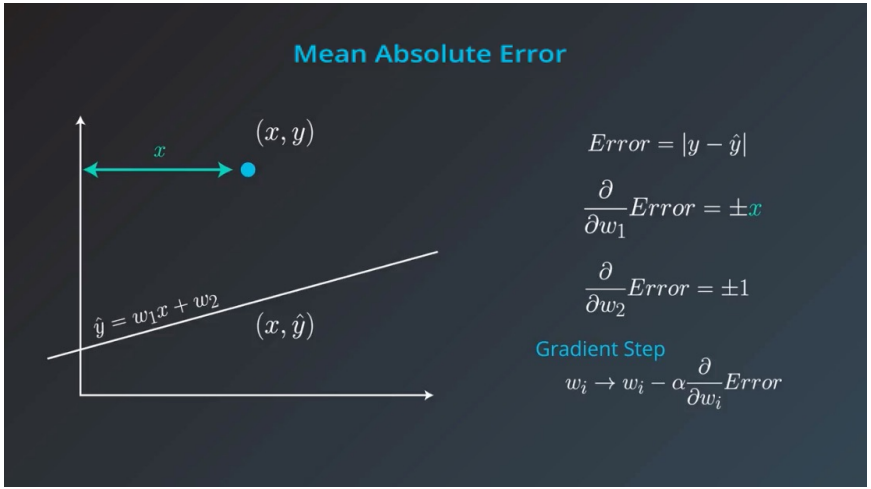

## Caractéristiques clés :

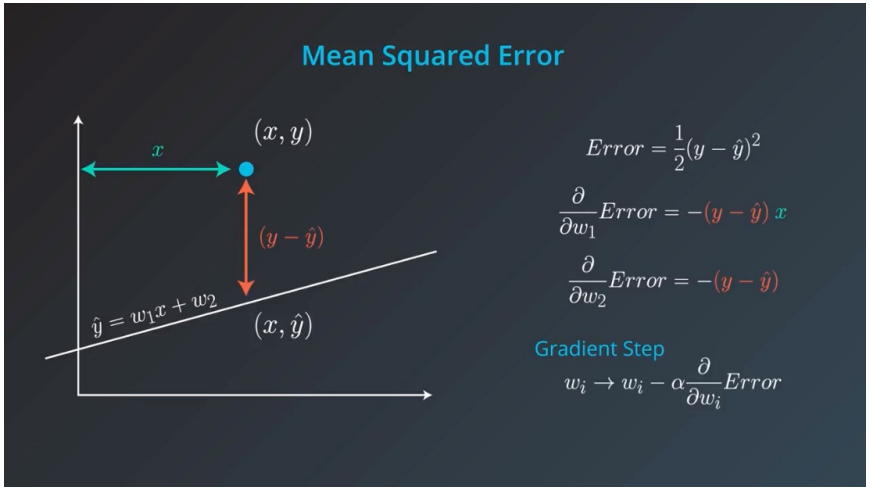
- Exprimée dans la même unité que la variable cible
- Moins sensible aux valeurs aberrantes que l'erreur quadratique moyenne

## Caractéristiques clés :

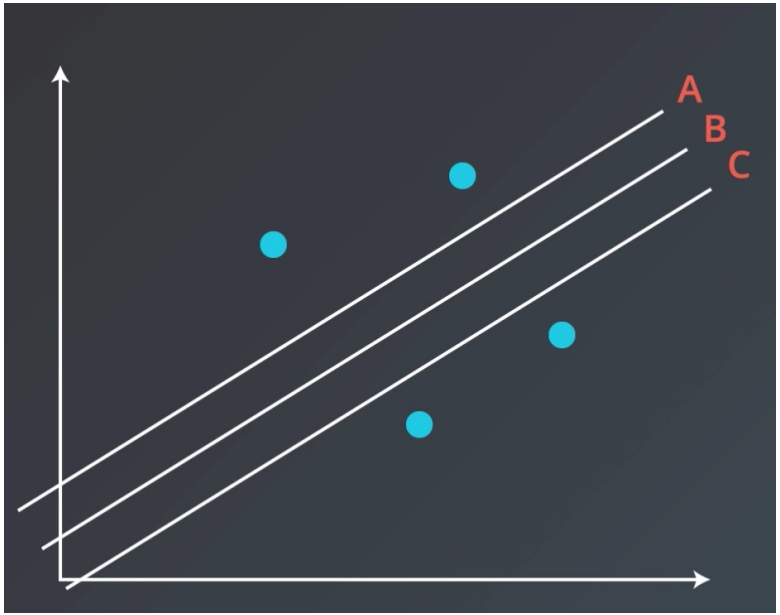
- Pénalise davantage les grandes erreurs
- Couramment utilisée en régression linéaire

# La Régression Linéaire: Minimiser la fonction coût



# La Régression Linéaire: MAE vs MSE




1 - Quelle droite donne la MAE la plus faible?

- A
- B
- C
- Les 3 donnent la même erreur

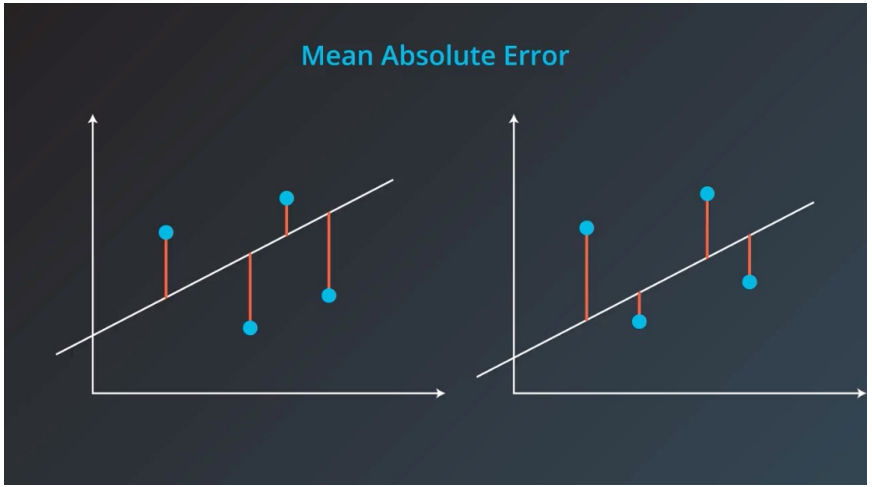
2 - Quelle droite donne la MSE la plus faible?

- A
- B
- C
- Les 3 donnent la même erreur

# La Régression Linéaire: MAE vs MSE

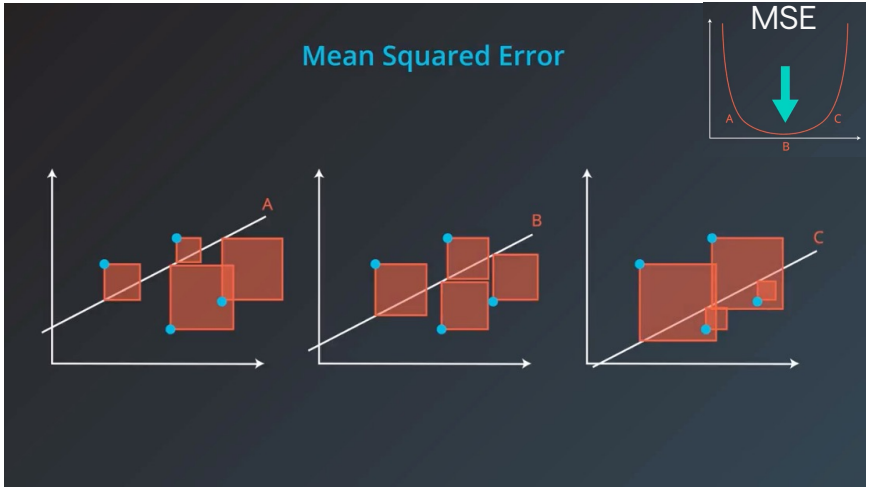
## 1 - Quelle droite donne la MAE la plus faible?

- A
- B
- C
- Les 3 donnent la même erreur



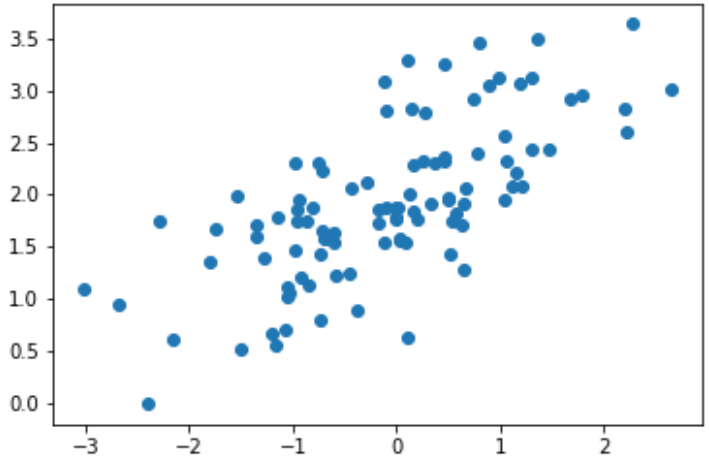

## 2 - Quelle droite donne la MSE la plus faible?

- A
- B (la MSE est une fonction quadratique, elle admet un minimum)
- C
- Les 3 donnent la même erreur



# La Descente de Gradient  - Exercice

Dans cet exercice, on vous donnera l'ensemble de données échantillon suivant (comme dans data.csv), et votre objectif sera d'écrire une fonction qui exécute la descente de gradient pour trouver la droite de régression la mieux ajustée.




https://github.com/elhidali/EPISEN-2024

# La régression linéaire - Exercice

Dans cet exercice, vous utiliserez la régression linéaire pour prédire l'espérance de vie à partir de l'indice de masse corporelle (IMC [BMI en anglais])

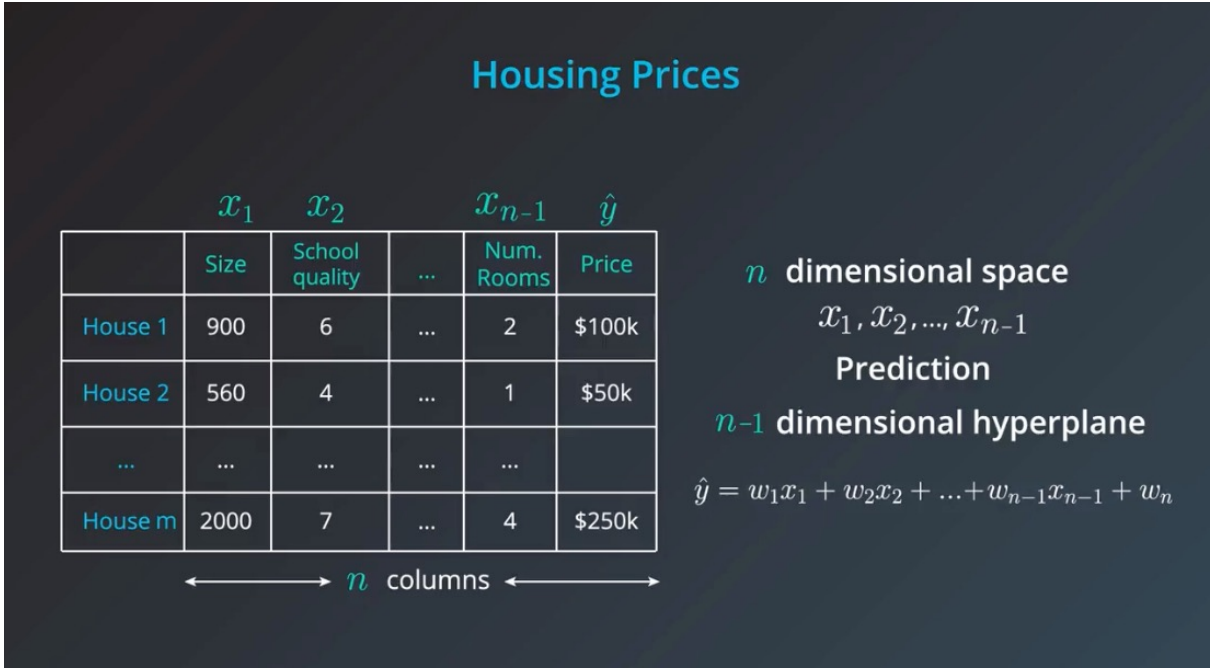
```
# TODO: Add import statements


# Assign the dataframe to this variable.
# TODO: Load the data
bmi_life_data = None

# Make and fit the linear regression model
#TODO: Fit the model and Assign it to bmi_life_model
bmi_life_model = None

# Make a prediction using the model
# TODO: Predict life expectancy for a BMI value of 21.07931
laos_life_exp = None
```

https://github.com/elhidali/EPISEN-2024

# Les hyperplans

# Les hyperplans

# Les hyperplans




Trouver les poids wi revient à minimiser la MAE ou la MSE en utilisant la descente de gradient (même démarche que la régression linéaire)

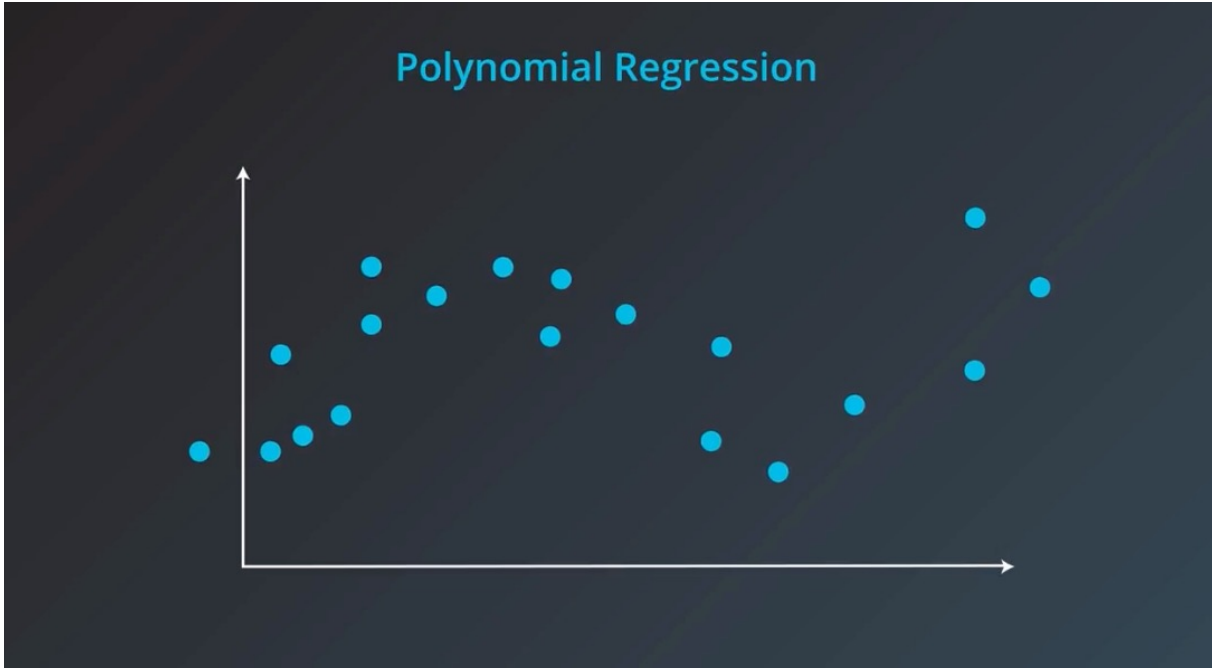
# La Régression linéaire multiple - Exercice

Dans ce quiz, vous utiliserez le jeu de données des prix des maisons de Boston. Le jeu de données comprend 12 caractéristiques de 506 maisons et la valeur médiane des maisons en x 10_000 de dollars.
Vous ajusterez un modèle sur les 12 caractéristiques pour prédire la valeur des maisons.

CRIM - Taux de criminalité par habitant par ville
ZN - Proportion de terrains résidentiels zonés pour des lots de plus de 25 000 pieds carrés
INDUS - Proportion d'acres d'activités non commerciales par ville
CHAS - Variable indicatrice de la rivière Charles (= 1 si la parcelle borde la rivière ; 0 sinon)
NOX - Concentration d'oxydes nitriques (parties par 10 millions)
RM - Nombre moyen de pièces par logement
AGE - Proportion de logements occupés par leur propriétaire construits avant 1940
DIS - Distances pondérées aux cinq centres d'emploi de Boston
RAD - Indice d'accessibilité aux autoroutes radiales
TAX - Taux d'impôt foncier en pleine valeur pour 10 000 dollars
PTRATIO - Ratio élèves-enseignants par ville
LSTAT - Pourcentage de la population de statut inférieur

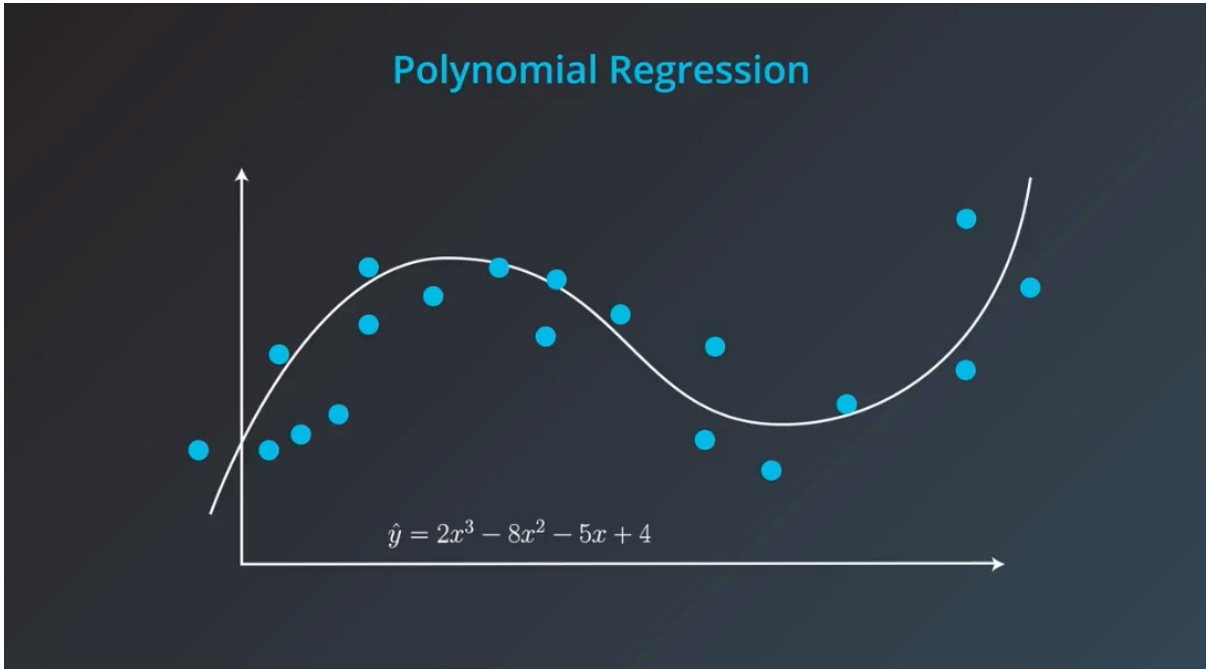
https://github.com/elhidali/EPISEN-2024

# La Régression Polynomiale

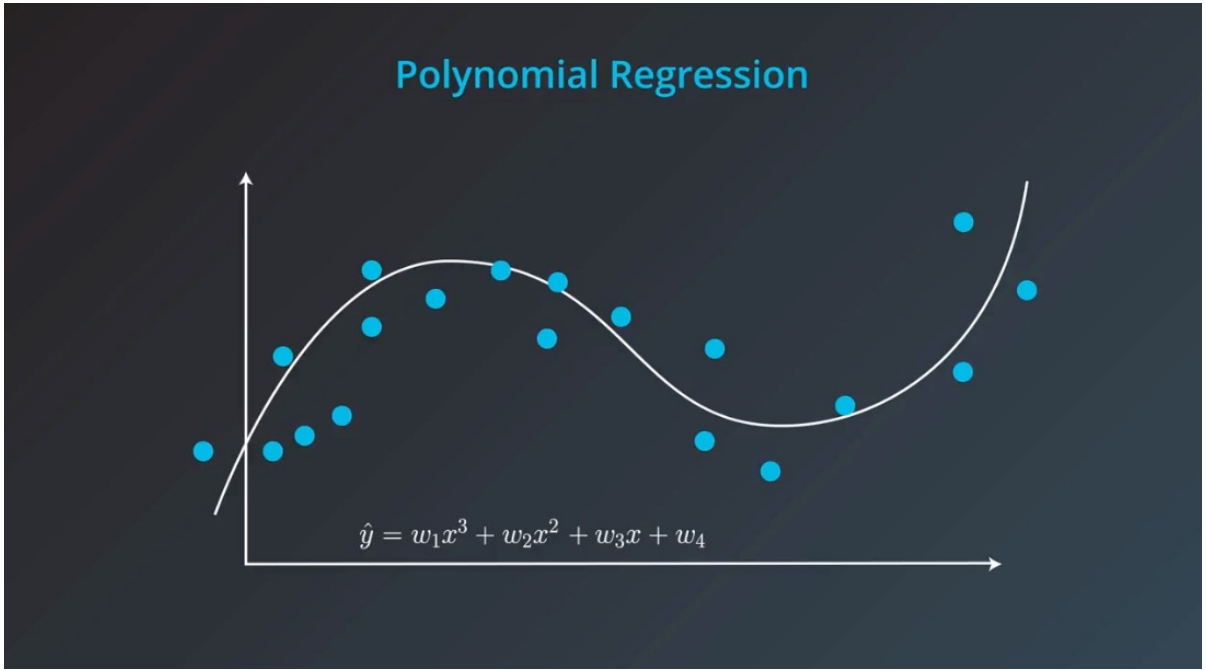



Quelles techniques permettent d'identifier le modèle optimal pour des données non linéaires ?

# La Régression Polynomiale



# La Régression Polynomiale



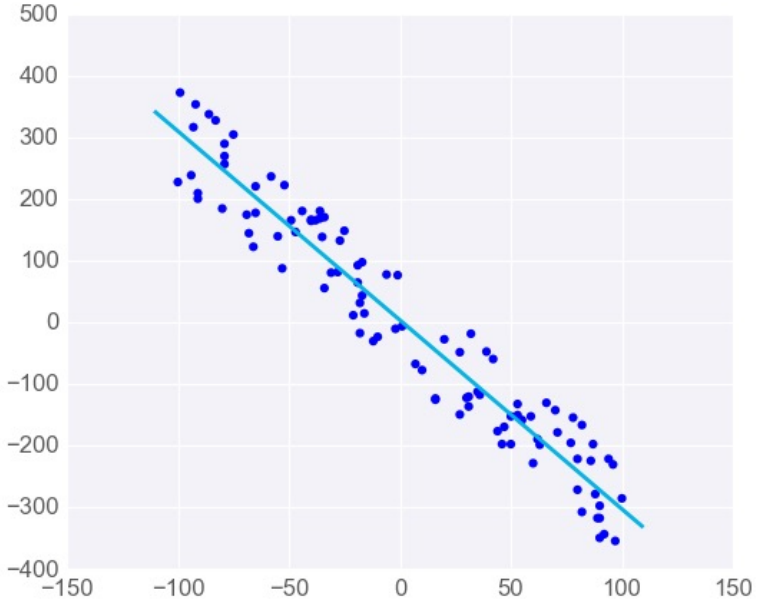

Trouver les poids wi revient à minimiser la MAE ou la MSE en utilisant la descente de gradient (même démarche que la régression linéaire)

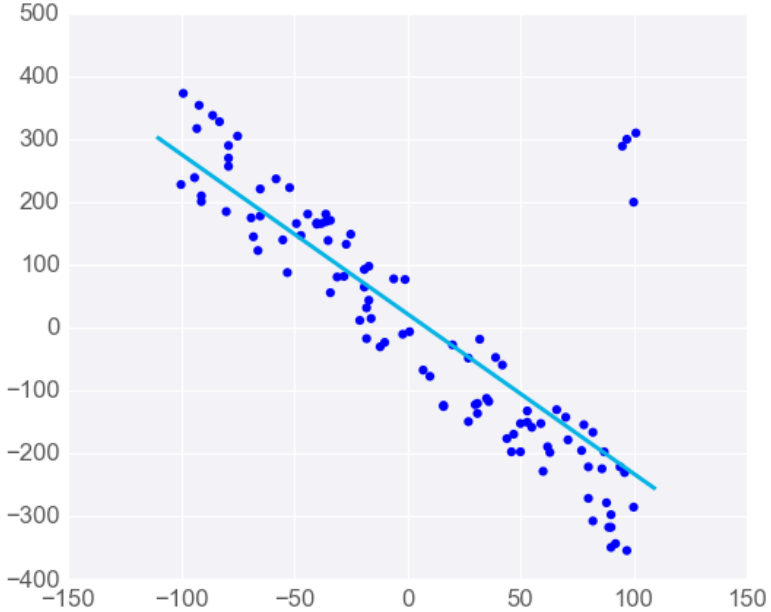
# La Régression Polynomiale – Exercice

```
# TODO: Add import statements

# Assign the data to predictor and outcome variables
# TODO: Load the data
train_data = None
X = None
y = None

# Create polynomial features
# TODO: Create a PolynomialFeatures object, then fit and transform the
# predictor feature
poly_feat = None
X_poly = None

# Make and fit the polynomial regression model
# TODO: Create a LinearRegression object and fit it to the polynomial predictor
# features
poly_model = None

# Once you've completed all of the steps, select Test Run to see your model
# predictions against the data, or select Submit Answer to check if the degree
# of the polynomial features is the same as ours!
```

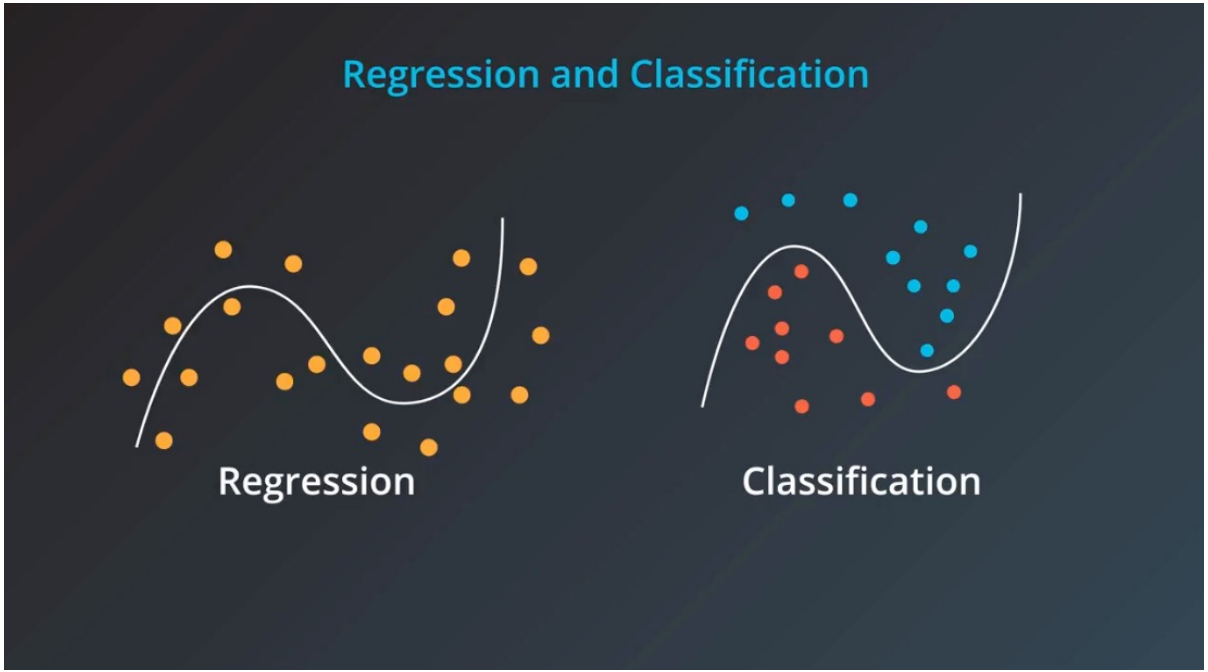
https://github.com/elhidali/EPISEN-2024

# La régression linéaire est sensible aux valeurs aberrantes



# La régularisation




La régularisation est une technique applicable aussi bien aux problèmes de régression qu'aux problèmes de classification. Pour simplifier la compréhension, nous allons expliquer ce concept en utilisant un exemple de classification.

# La régularisation

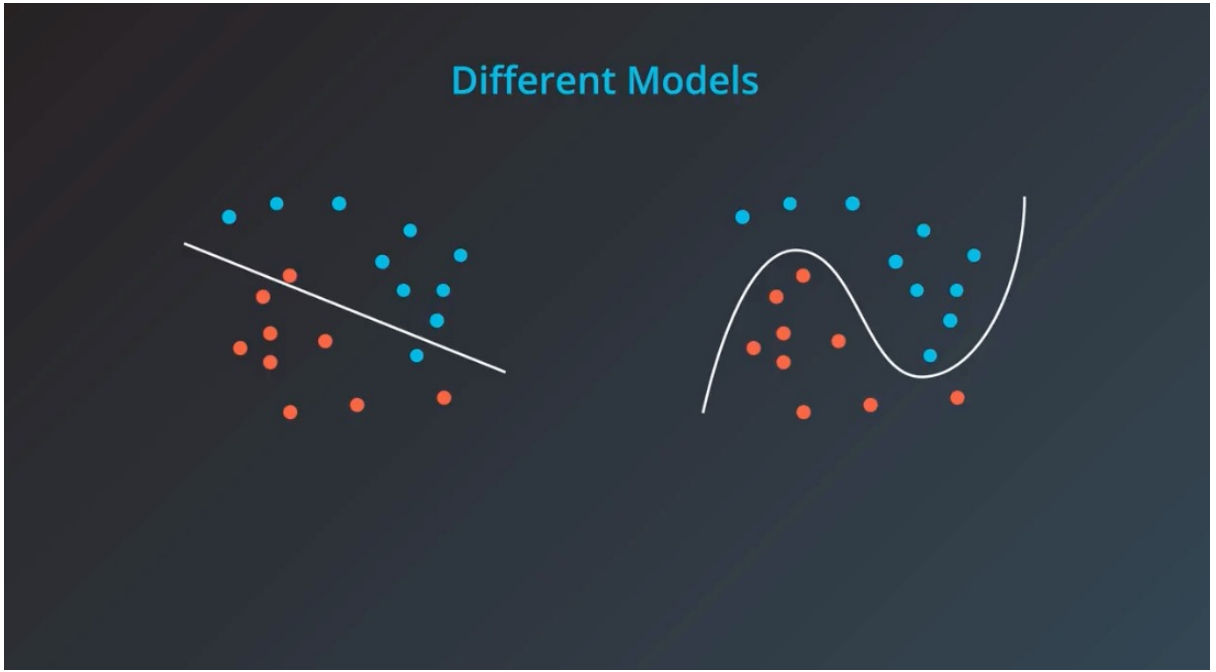
La régularisation est une technique fondamentale en apprentissage automatique et en statistiques, conçue pour prévenir le surapprentissage (**overfitting**) des modèles. Elle joue un rôle crucial dans l'amélioration de la généralisation des modèles, c'est-à-dire leur capacité à bien performer sur des données nouvelles et non vues.

Dans le contexte de l'apprentissage automatique, **la régularisation ajoute une pénalité à la fonction de coût** (ou fonction de perte) lors de l'entraînement du modèle. **Cette pénalité décourage le modèle d'adopter des paramètres trop complexes ou trop grands,** ce qui pourrait le conduire à "mémoriser" les données d'entraînement plutôt que d'apprendre des patterns généralisables.

Il existe plusieurs formes de régularisation, dont les plus courantes sont :

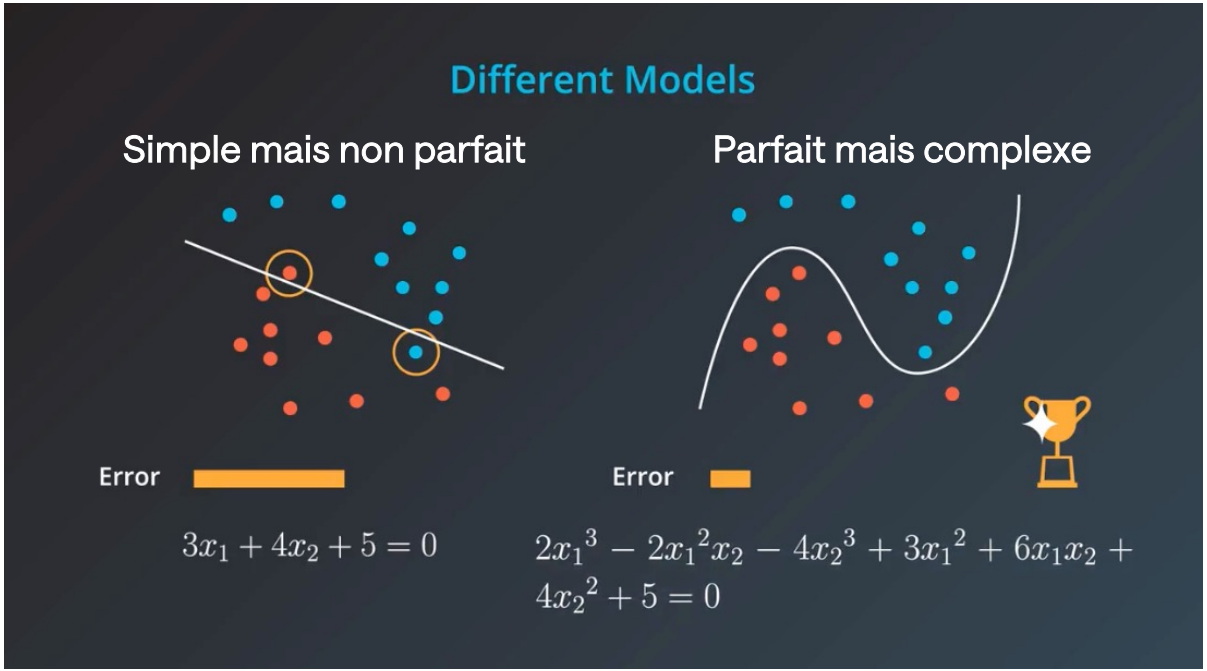
1. La régularisation L1 (Lasso)
2. La régularisation L2 (Ridge)
3. La régularisation Elastic Net (combinaison de L1 et L2)

# La régularisation



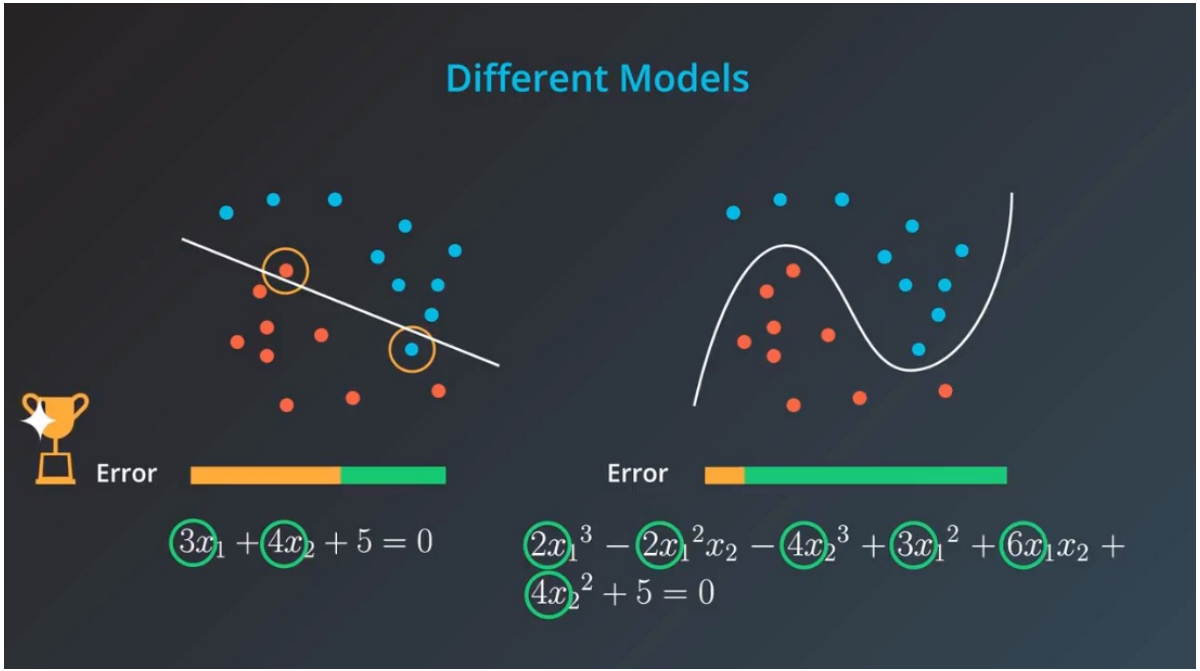

Quel est le meilleur modèle ?

# La régularisation



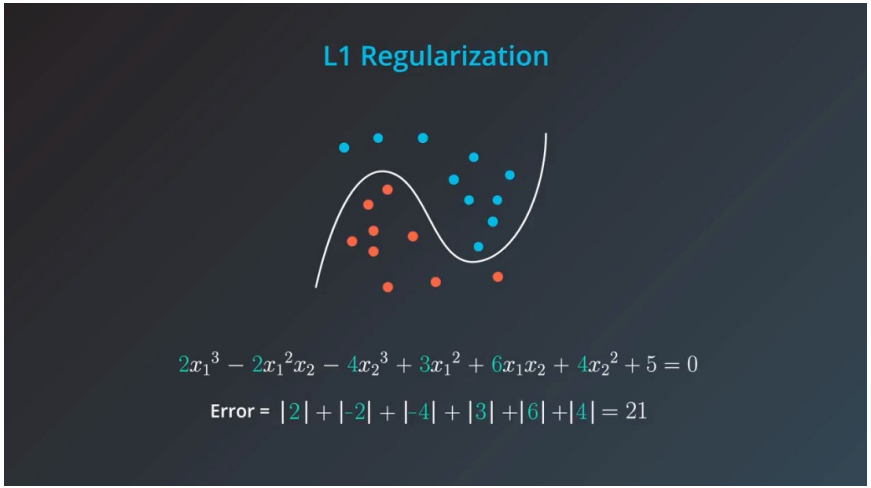

- L'incorporation de la complexité du modèle dans la fonction de coût permet de favoriser des modèles plus simples.
- Cette approche améliore la capacité de généralisation du modèle et réduit le risque de surapprentissage.
- En pénalisant la complexité, on encourage le modèle à capturer les tendances essentielles des données plutôt que d'en mémoriser les détails spécifiques, ce qui conduit à de meilleures performances sur des données nouvelles et non vues.

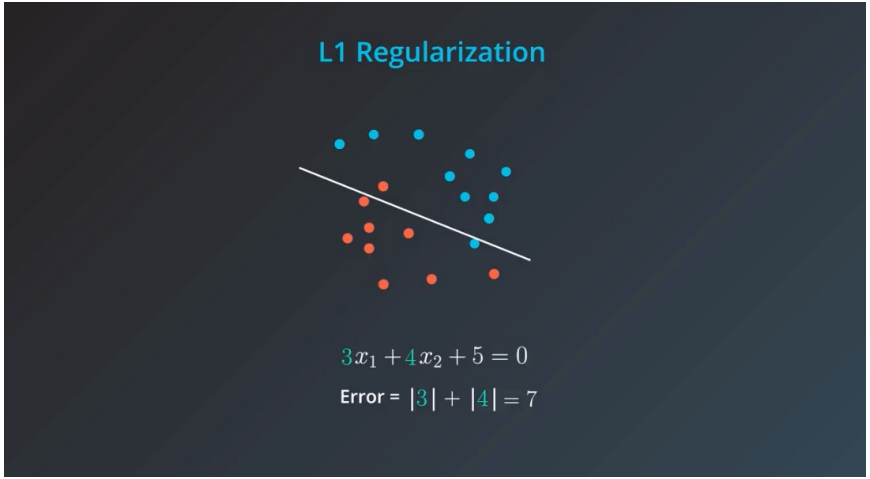
# La régularisation



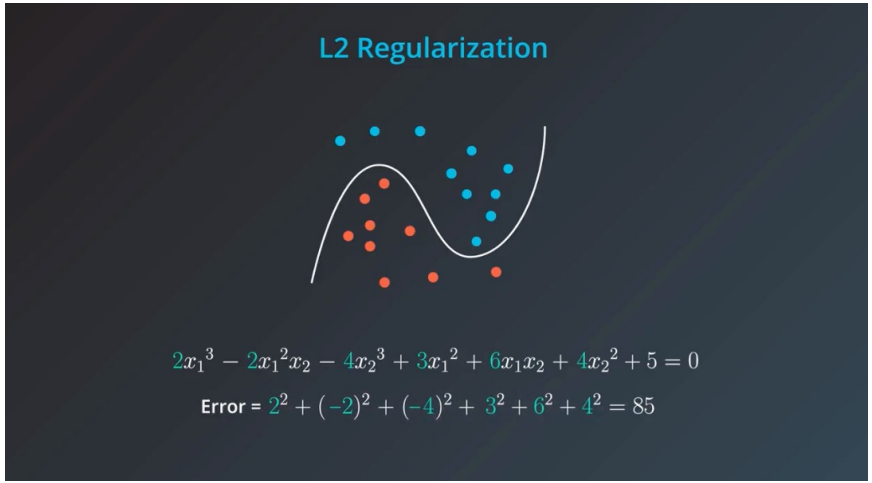

En ajoutant les poids wi à la fonction coût, le modèle linéaire obtient l'erreur la plus faible

# La régularisation L1 (Lasso)

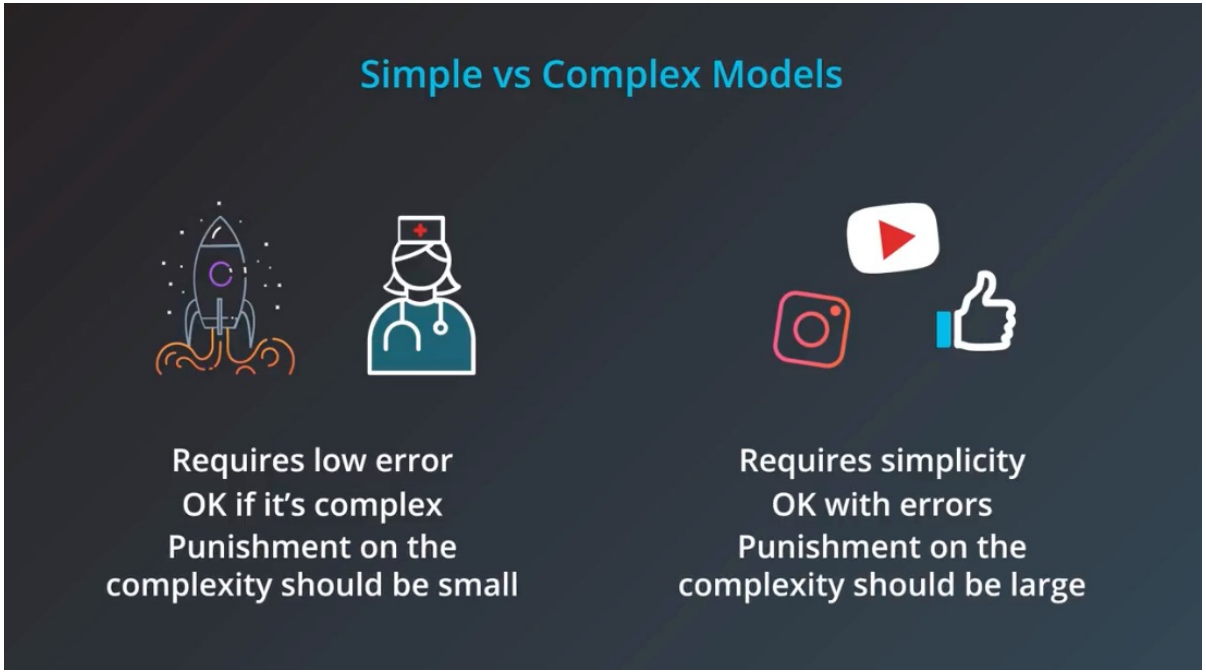


# La régularisation L2 (Ridge)

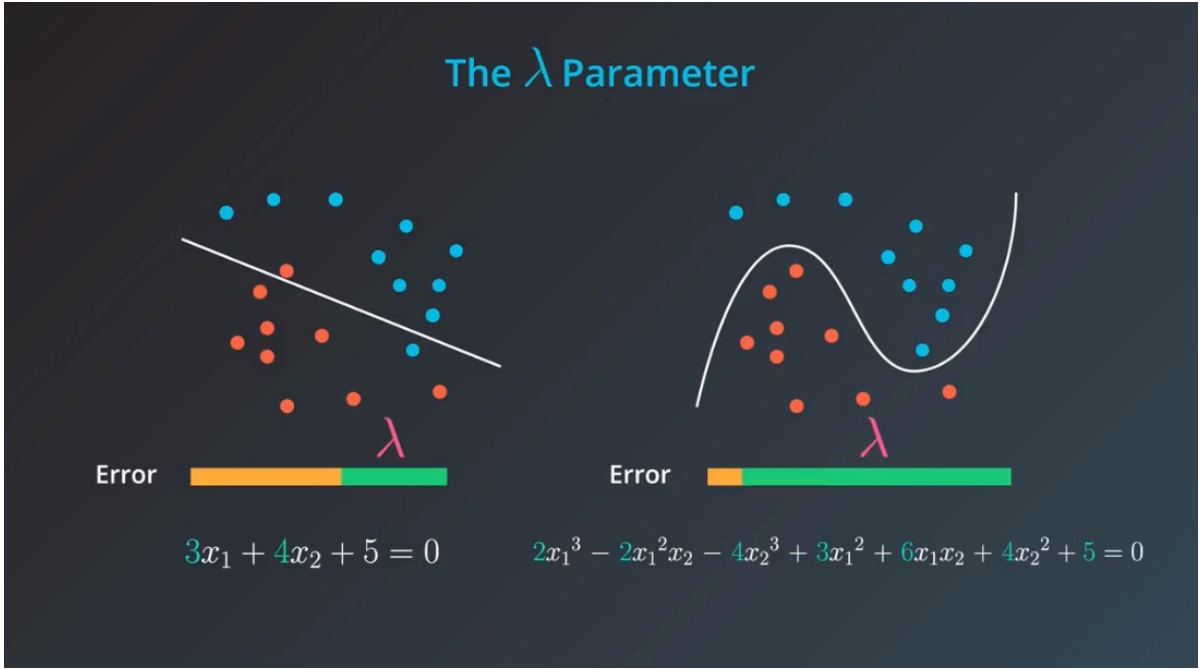



# La régularisation: modèles simples vs modèles complexes



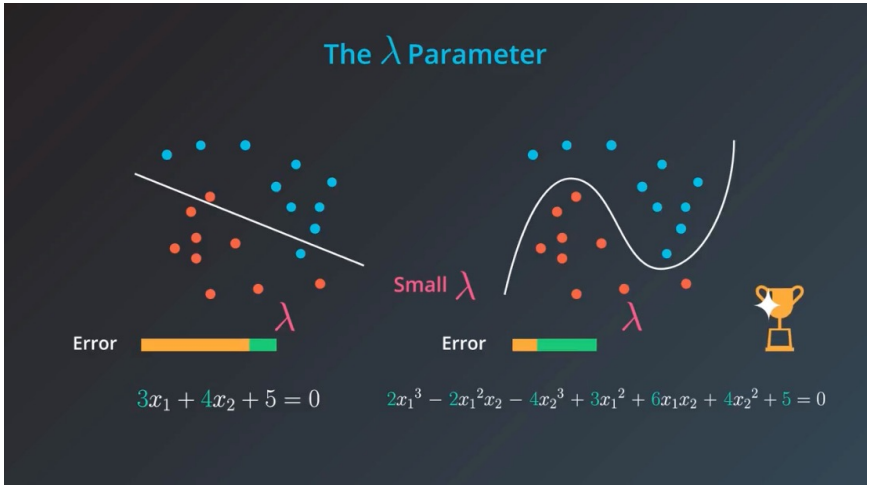

On contrôle la complexité des modèles en utilisant un paramètre ƛ

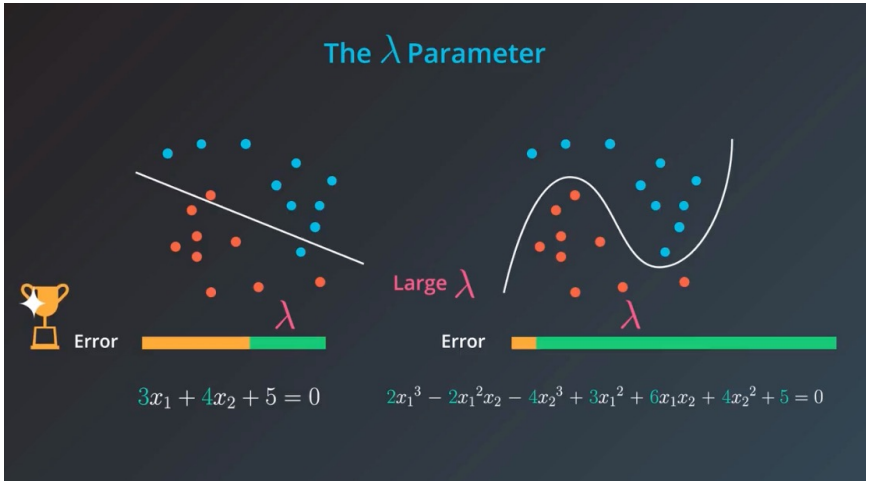
# La régularisation: le paramètre λ




On ajuste l'importance de la pénalité de complexité en la multipliant par un hyperparamètre lambda (λ). Ce coefficient λ permet de contrôler le degré de régularisation appliqué au modèle.

# La régularisation: le paramètre λ



# La régularisation – Exercice

```
# TODO: Add import statements
import numpy as np
import pandas as pd
from sklearn.linear_model import Lasso

# Assign the data to predictor and outcome variables
# TODO: Load the data
train_data = pd.read_csv('data.csv', header = None)
X = train_data.iloc[:,:-1]
y = train_data.iloc[:,-1]

# TODO: Create the linear regression model with lasso regularization.
lasso_reg = Lasso()

# TODO: Fit the model.
lasso_reg.fit(X, y)

# TODO: Retrieve and print out the coefficients from the regression model.
reg_coef = lasso_reg.coef_
print(reg_coef)
```

https://github.com/elhidali/EPISEN-2024