Fouille de Données

Data Mining

Régression Linéaire

Plan du cours

1. Régression / Estimation : Définition et principe
2. Régression linéaire
3. Régression linéaire avec OLS
4. Entrainement et évaluation – Loss functions
5. Régression linéaire avec le Gradient Descent

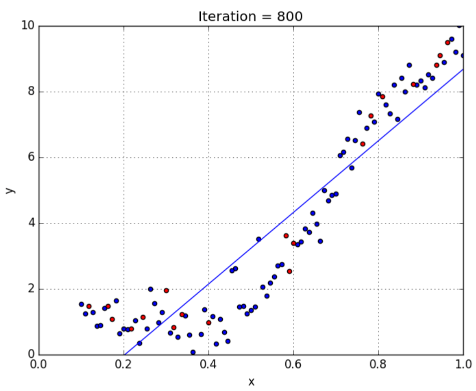
Classification : Algorithmes

SAVOIR - PREDIRE - DECIDER

Données

Linear Regression

$$f(x)=b_1x+b_0$$



Connaissances

La régression

- La **régression** est la méthode utilisée pour l'estimation des valeurs **continues**. Tache supervisée.

- Son objectif est de trouver le meilleur modèle qui décrit la relation entre une variable continue de sortie et une ou plusieurs variables d'entrée.

- Prédire la valeur continue de la sortie Y selon une entrée X ou plusieurs entrées Xi (attributs). = Expliquer une variable *Y* à l'aide d'une variable *X* ou plusieurs variables *X*i.

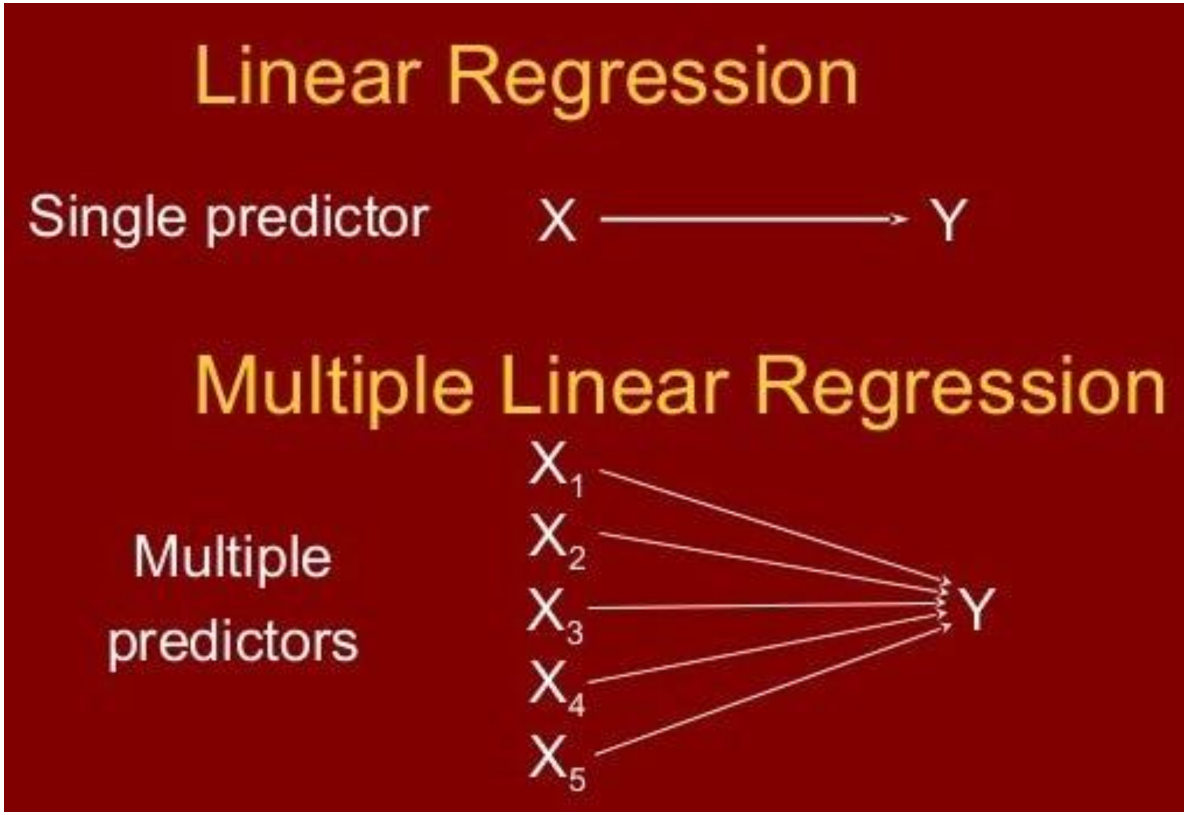
- Ex : prédire le cours de la bourse, le prix d'un appartement, ou bien l'évolution de la température sur Terre.

- Il s'agit donc de trouver une fonction f (=le modèle de régression) qui se rapproche le plus possible d'un scénario donné d'entrées et de sorties.

- Différents types de régression : linéaire, polynomiale, logistique (classification), Lasso, etc.

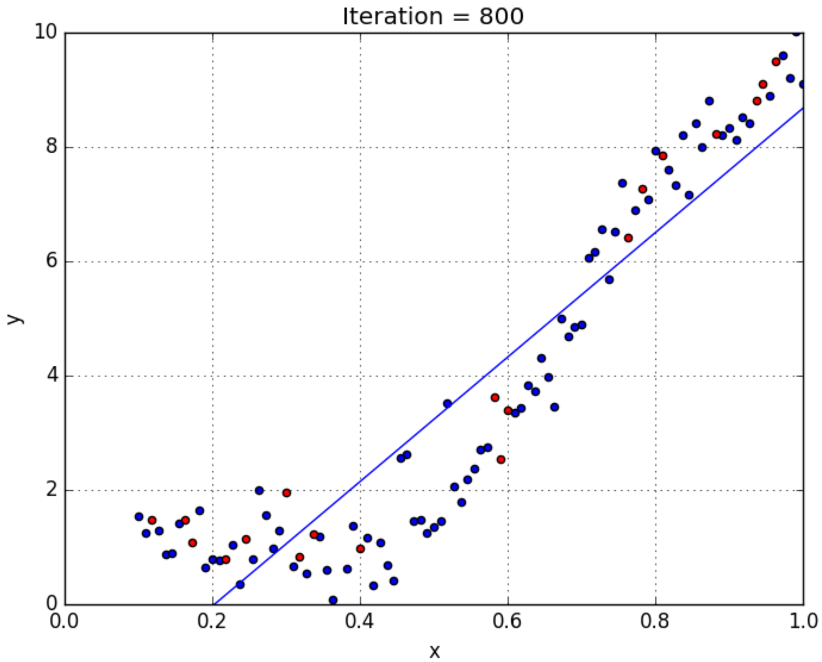
La régression

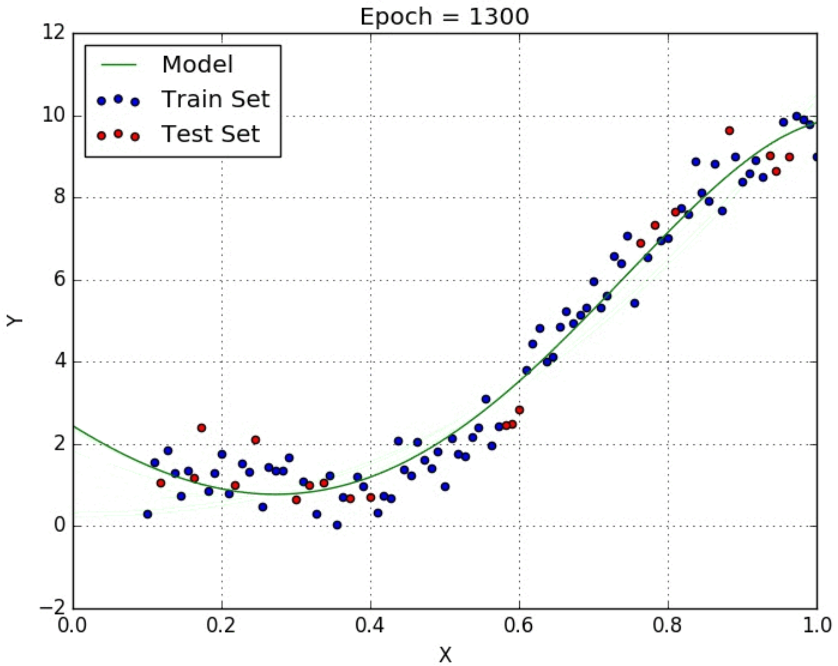
- Trouver la meilleure **fonction** (*modèle de régression*), **f**, qui décrit la relation entre une variable **continue** de **sortie (Y)** et une ou plusieurs variables d'**entrées** **(X ou Xi)**.



La régression

- Différents types de **régression** : **linéaire**, **polynomiale**.

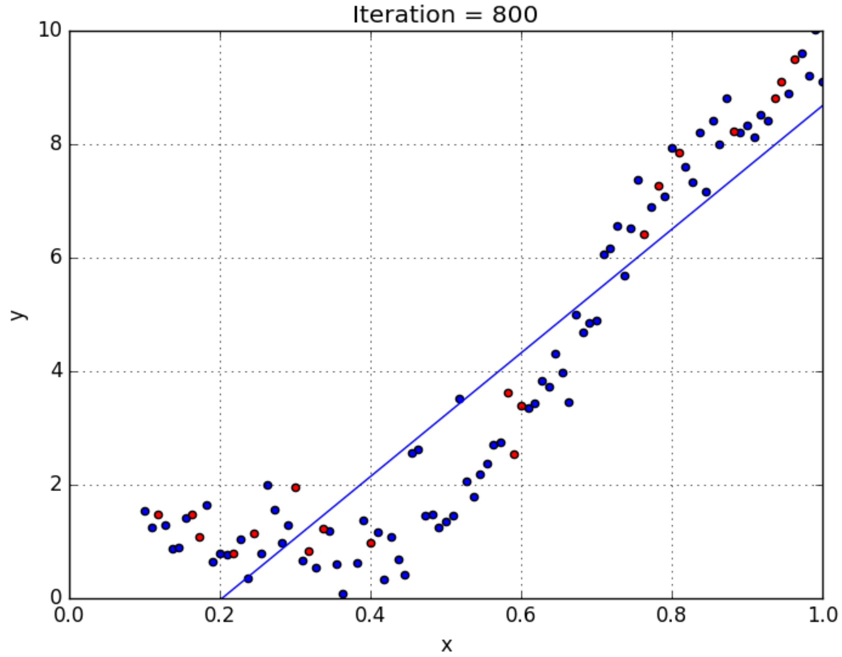


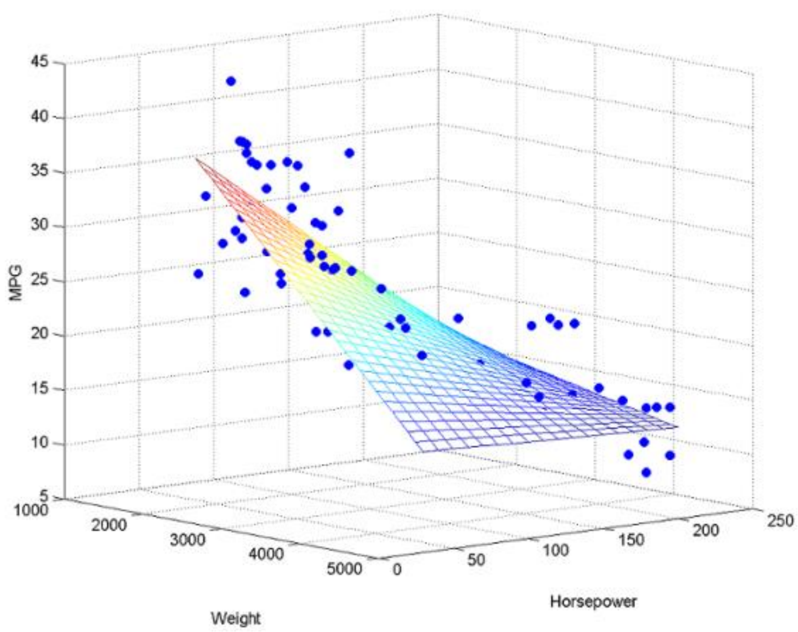


La régression

- Régression **linéaire** : fonction **f** (modèle de régression) **linéaire**.
- Peut être : Simple ou Multiple.

- Linéaire **Simple** : utilisée pour estimer une sortie Y en fonction d'une seule entrée X.
- Linéaire **Multiple** : utilisée pour estimer une sortie Y en fonction de plusieurs entrées Xi.



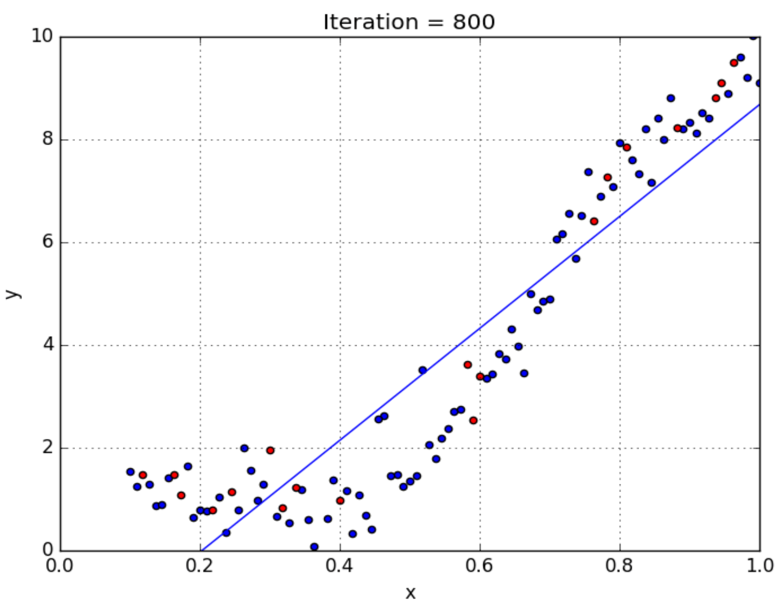


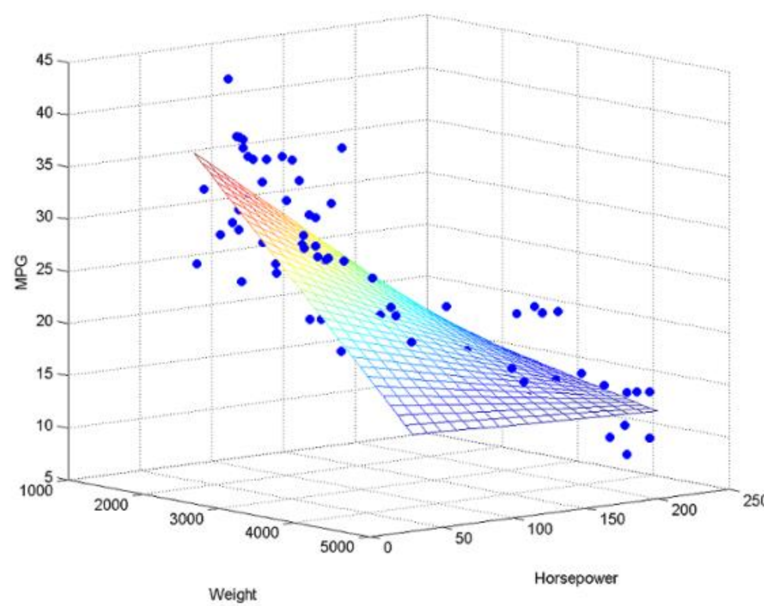
La régression

- Régression **linéaire** : fonction f (modèle) **linéaire**.
- Peut être : Simple ou Multiple.

- Linéaire **Simple** : **Ex** : prédire le prix de vente d'un appartement (Y) en fonction de la surface habitable (X).

- Linéaire **Multiple** : **Ex** : prédire le prix de vente d'un appartement (Y) en fonction de la surface habitable (X_1) et du nombre de pièces (X_2) .

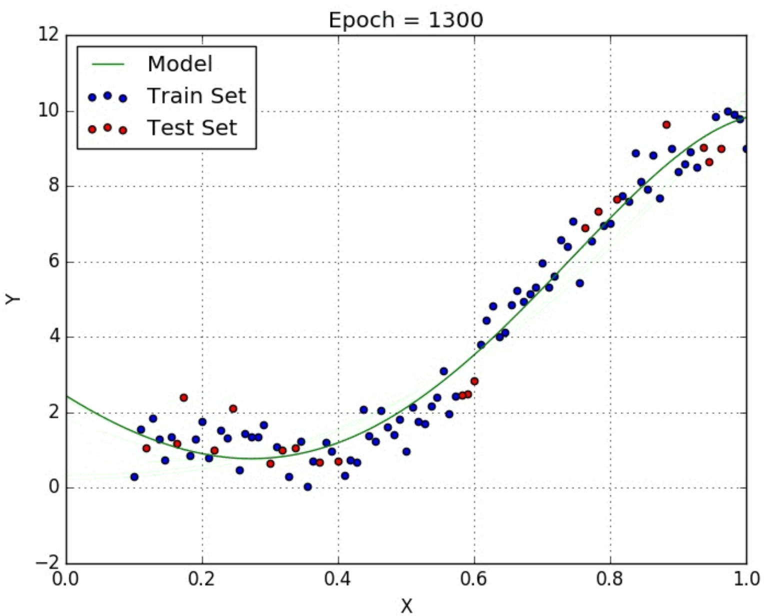


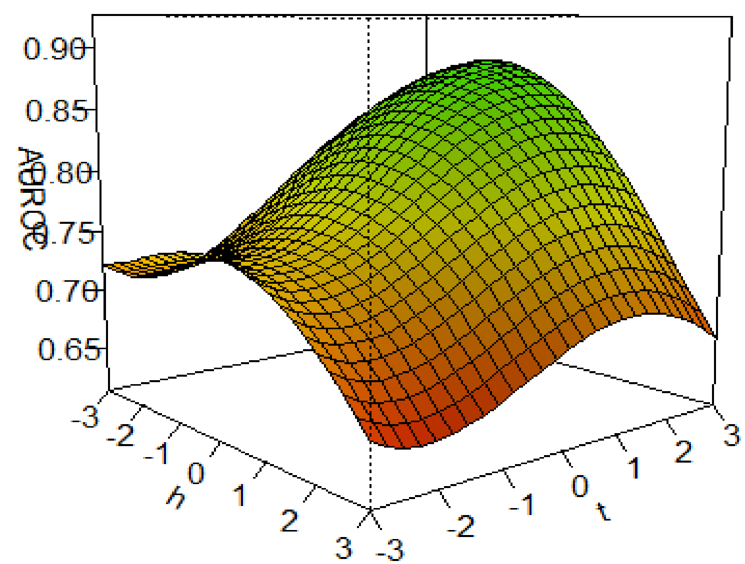


La régression

- Régression **polynomiale**: fonction f (modèle de régression) **non linéaire**.
- Peut être : Simple ou Multiple.

- Polynomiale **Simple** : utilisée pour estimer une sortie Y en fonction d'une seule entrée X.
- Polynomiale **Multiple** : utilisée pour estimer une sortie Y en fonction de plusieurs entrées Xi.





La régression linéaire

- Régression **linéaire** : **fonction f** (modèle) linéaire.

- Simple Linear Regression – one independent variable.

$$y = b_0 + b_1x_1$$

- Multiple Linear Regression – multiple independent variables.

$$y = b_0 + b_1x_1 + b_2x_2 \dots + b_nx_n$$

2nd independent variable and weight (coefficient)

nth independent variable and weight (coefficient)

La régression linéaire

Simple Linear Regression	$y = b_0 + b_1 x_1$
Multiple Linear Regression	$y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n$
Polynomial Linear Regression	$y = b_0 + b_1 x_1 + b_2 x_1^2 + \ldots + b_n x_1^n$

La régression linéaire

- L'objectif de la **régression linéaire** est de trouver les **coefficients optimaux {bo, b1,…, bn}** qui définissent la **droite (ou hyperplan) de meilleur ajustement** (Best fit line/hyperplane) en **minimisant les erreurs** de prédiction de Y.

Simple Linear Regression

$$y = b_0 + b_1 x_1$$

Multiple Linear Regression

$$y = b_0 + b_1 x_1 + b_2 x_2 + \dots + b_n x_n$$

La régression linéaire

- **Linear Regression Model :** find the best-fitting line (or hyperplane) that predicts y from inputs X by **minimizing the errors** between predicted and actual values.

Linear Regression: Single Variable

$$\underbrace{\boxed{\hat{y}}}_{\text{Predicted output}} = \underbrace{\beta_0 + \beta_1}_{\text{Coefficients}} \underbrace{\boxed{x}}_{\text{Input}}$$

Linear Regression: Multiple Variables

$$\boxed{\hat{y}} = \underbrace{\beta_0 + \beta_1}\boxed{x_1} + \dots + \underbrace{\beta_p}\boxed{x_p}$$

La régression linéaire

- **Linear Regression Model :** find the **best-fitting** line (or hyperplane) that predicts y from inputs X by **minimizing the errors** between predicted and actual values.

Linear Regression: Single Variable

$$\underbrace{\boxed{\hat{y}}}_{\text{Predicted output}} = \underbrace{\beta_0 + \beta_1}_{\text{Coefficients}} \underbrace{\boxed{x}}_{\text{Input}} + \underbrace{\boxed{\epsilon}}_{\text{Error}}$$

Linear Regression: Multiple Variables

$$\boxed{\hat{y}} = \underbrace{\beta_0 + \beta_1}_{} \boxed{x_1} + \dots + \underbrace{\beta_p}_{} \boxed{x_p} + \boxed{\epsilon}$$

La régression linéaire

- **Linear Regression Model :** find the **best-fitting** line (or hyperplane) that predicts y from inputs X by **minimizing the errors** between predicted and actual values.

Linear Regression: Single Variable

$$\underbrace{\boxed{\hat{y}}}_{\text{Predicted output}} = \underbrace{\beta_0 + \beta_1}_{\text{Coefficients}} \underbrace{\boxed{x}}_{\text{Input}} + \underbrace{\boxed{\epsilon}}_{\text{Error}}$$

$$y = X\beta + \epsilon$$

Linear Regression: Multiple Variables

$$\boxed{\hat{y}} = \underbrace{\beta_0 + \beta_1}\boxed{x_1} + \dots + \underbrace{\beta_p}\boxed{x_p} + \boxed{\epsilon}$$

La régression linéaire

- **Deux méthodes principales**, parmi d'autres :

Méthode	**Nom Commun**	**Nature de la Solution**	**Points Clés**
Moindres Carrés Ordinaires (Ordinary Least Squares (OLS))	Régression Linéaire	**Analytique** (Forme Fermée - Closed-Form *Direct solution*)	Calcul direct des coefficients via des formules mathématiques (dérivées à zéro).
Descente de Gradient (Gradient Descent)	Régression Linéaire avec *Optimisation*	**Itérative** (Approximation)	Trouver les coefficients en *ajustant* progressivement les valeurs dans la direction du *minimum* de la fonction de coût.

La régression linéaire

- **Deux méthodes principales**, parmi d'autres :

Méthode
Moindres Carrés Ordinaires (Ordinary Least Squares (OLS))

- They are the SAME method mathematically.

✅ 1. Two ways to solve linear regression

Method A — Matrix OLS (Normal Equation)

General formula (works for 1, 2, or 1000 features):

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

Method B — Slope/Intercept Formulas (Simple Linear Regression only)

Only works when you have **ONE** feature x:

$$\hat{\beta}_1 = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

La régression linéaire

- **Deux méthodes principales**, parmi d'autres :

Méthode	**Nom Commun**	**Nature de la Solution**	**Points Clés**
Moindres Carrés Ordinaires (Ordinary Least Squares (OLS))	Régression Linéaire **- Simple**	**Analytique** (Forme Fermée - Closed-Form *Direct solution*)	Calcul direct des coefficients **b1** et **b0** via des formules mathématiques (dérivées à zéro).
Descente de Gradient (Gradient Descent)	Régression Linéaire avec Optimisation	**Itérative** (Approximation)	Trouver les coefficients en ajustant progressivement les valeurs dans la direction du minimum de la fonction de coût.

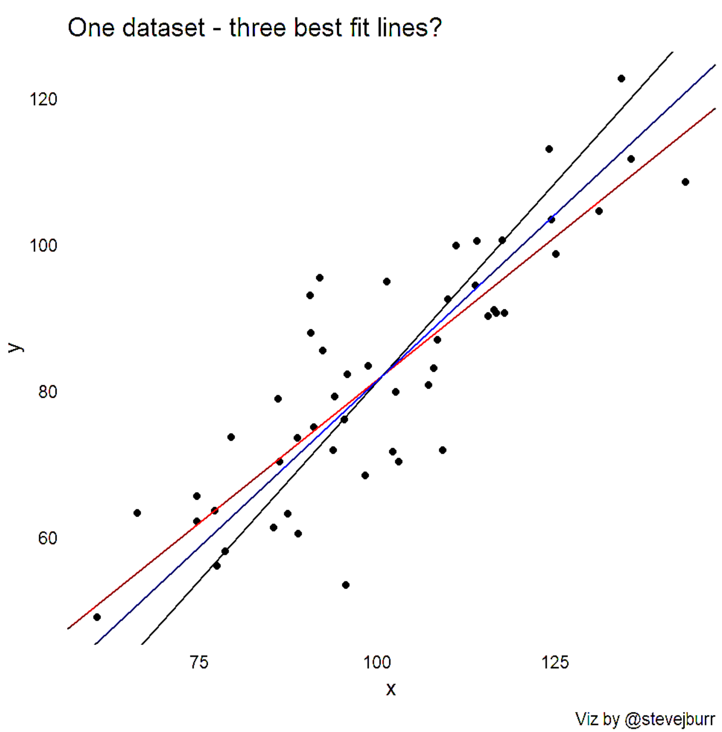
La régression linéaire simple

Régression **linéaire simple**

- **But** : Trouver un **modèle linéaire** $f(x) = b_1 x + b_0$ où b_1 et b_0 sont les **paramètres/coefficients** du modèle.

- **$y = b_1 x + b_0$**

- Trouver le **meilleur** modèle (Best fit line) =>



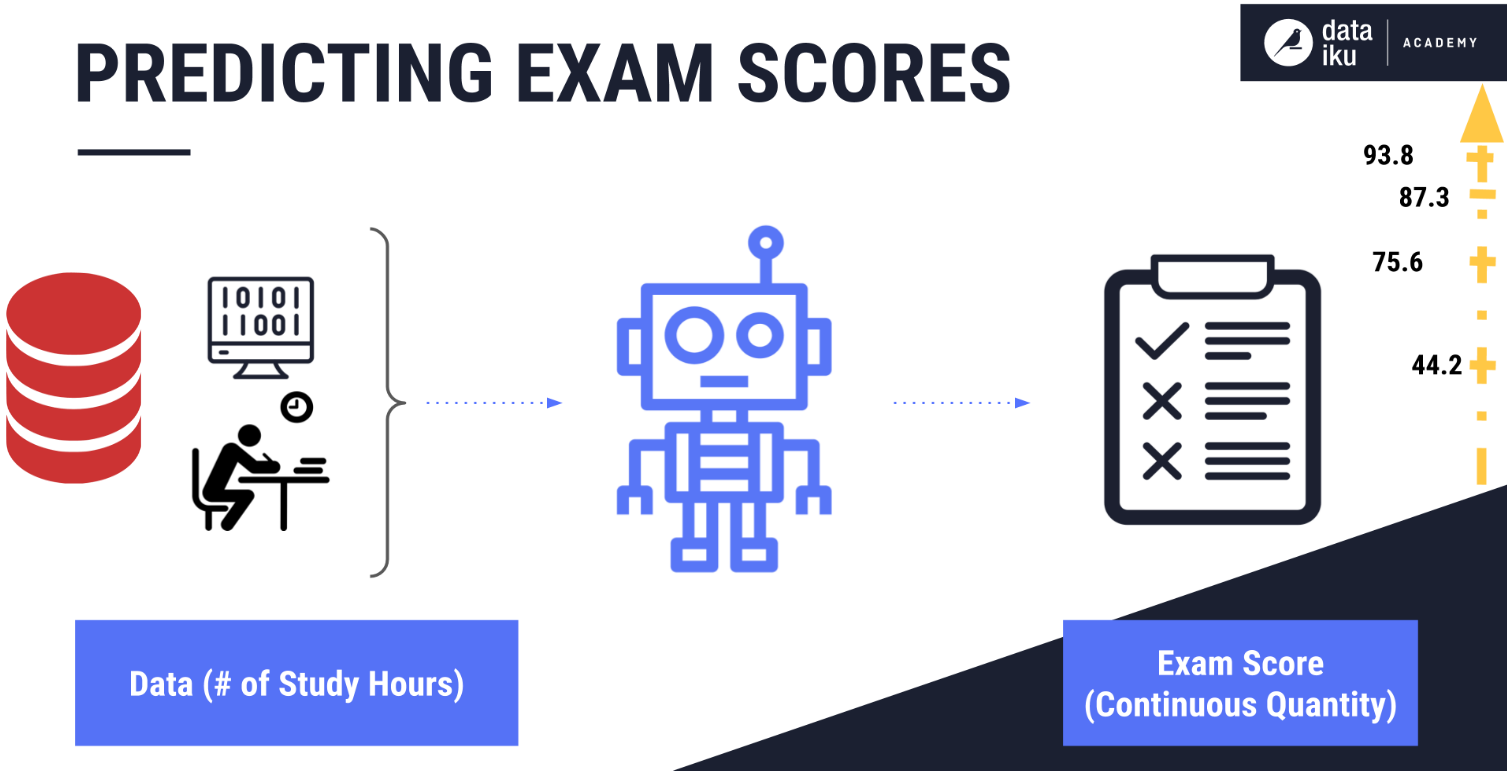
=> Trouver les **meilleures** (optimales) valeurs des paramètres **b_1 (slope)** et **b_0 (intercept)**.

=> Faire le **minimum d'erreurs** possible sur les prédictions de Y.

La régression linéaire simple

Régression **linéaire simple**

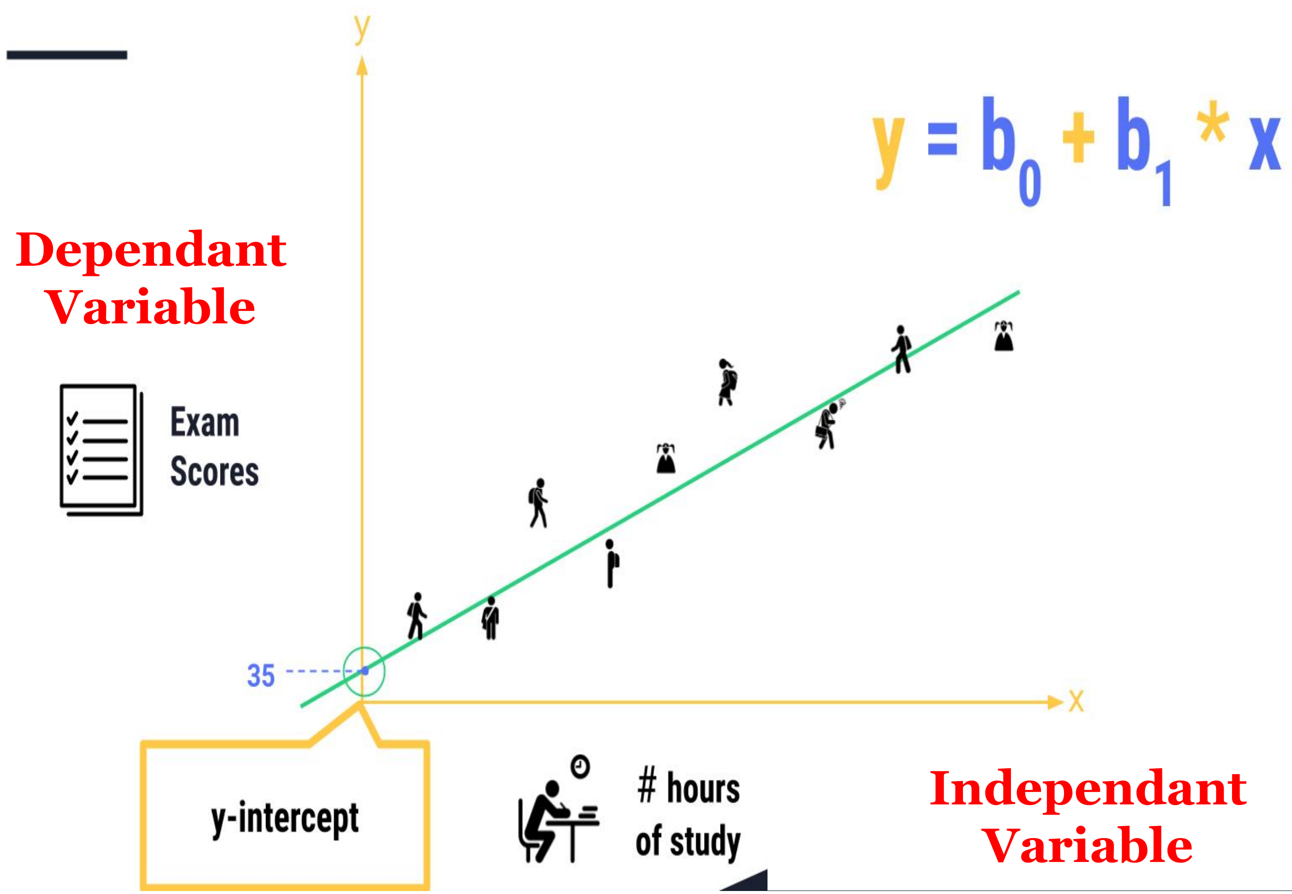
- **But** : Trouver un **modèle linéaire** $f(x) = b1x + b0$
- Exemple :



La régression linéaire simple

Régression **linéaire simple**

- **But** : Trouver un **modèle linéaire** $y = b1x + b0$



La régression linéaire simple

Régression **linéaire simple**

- **Exemple :**

- **X** = nombre d'heures passées à réviser
- **Y** = Note étudiant (/100)

Objectif :

On souhaite savoir si, de façon générale, le nombre d'heures passées à réviser a une **influence** sur la note obtenue et sous quelle forme cette influence peut être exprimée.

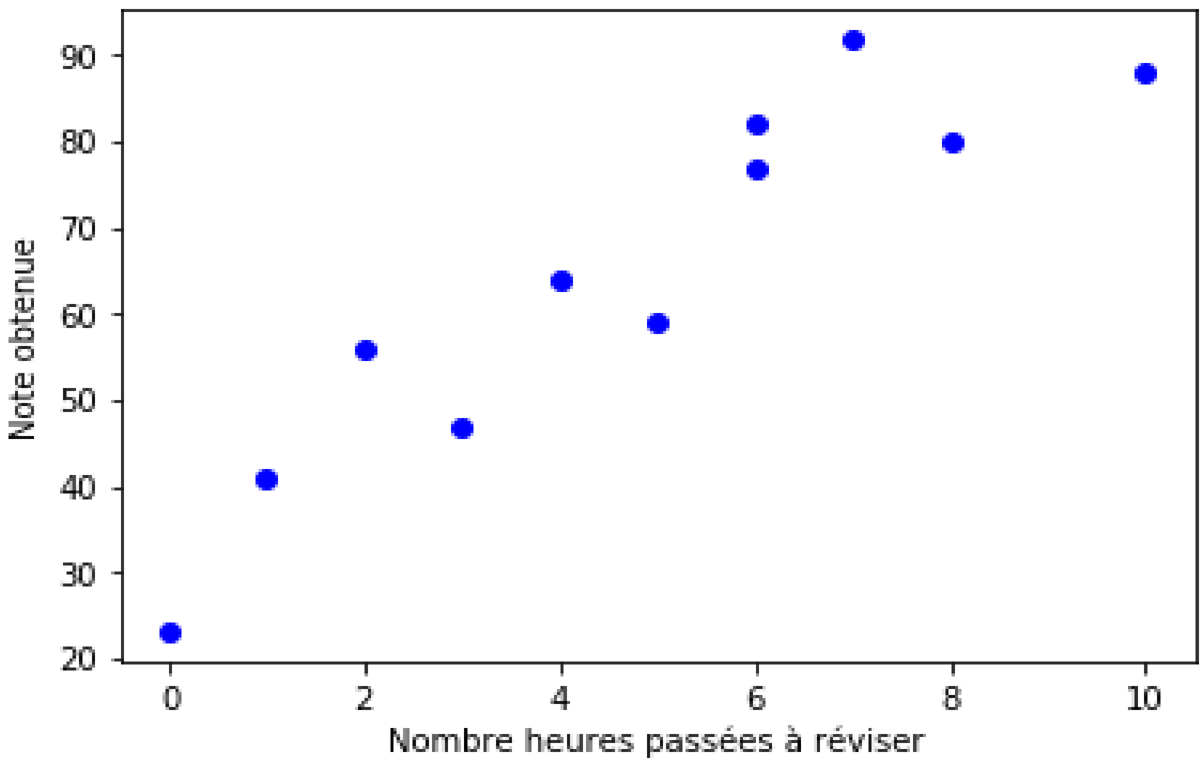
Le but est **d'expliquer** au mieux comment la note d'un étudiant varie en fonction du nombre d'heures de révision et éventuellement de **prédire** la note à partir d'un nombre d'heures donné.

Hours spent on essay	Grade
6	82
10	88
2	56
4	64
6	77
7	92
0	23
1	41
8	80
5	59
3	47

La régression linéaire simple

Régression **linéaire simple**

- **Exemple :** **X** = nombre d'heures de révision ---- **Y** = Note étudiant



Hours spent on essay	Grade
6	82
10	88
2	56
4	64
6	77
7	92
0	23
1	41
8	80
5	59
3	47

La régression linéaire simple

Méthode - Ordinary Least Squares (OLS) :

- **Entrainement :** Training = finding the best coefficients that minimize the difference between predicted and actual values. Exact direct solution, no iterations.

1 - Calculer les valeurs de **b1** (le slope) et **bo** (l'intercept) selon :

$$\begin{cases} b_1 = \dfrac{\sum(x-\bar{x}) * (y-\bar{y})}{\sum(x-\bar{x})^2} = \dfrac{Covariance(x,y)}{Variance(x)} \\ \\ b_0 = \bar{y} - b_1\bar{x} \end{cases}$$

La régression linéaire simple

Méthode - Ordinary Least Squares (OLS) :

- **Entrainement :** Training = finding the best coefficients that minimize the difference between predicted and actual values. Exact solution, no iterations.

1 - Calculer les valeurs de **b1** (le slope) et **bo** (l'intercept) selon :

$$\boxed{\beta_1 = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}}$$

$$\boxed{\beta_0 = \bar{y} - \beta_1 \bar{x}}$$

- $\bar{x}$: mean of all x-values
- $\bar{y}$: mean of all y-values
- β_1: slope — tells how much y changes per unit x
- β_0: intercept — y value when x = 0

La régression linéaire simple

Méthode - Ordinary Least Squares (OLS) :

- **Training - Entrainement :**

1 - Calculer les valeurs de **b1** (le slope) et **bo** (l'intercept) :

$$\begin{cases} b_1 = \dfrac{\sum(x-\bar{x}) * (y-\bar{y})}{\sum(x-\bar{x})^2} = \dfrac{Covariance(x,y)}{Variance(x)} \\ \\ b_0 = \bar{y} - b_1\bar{x} \end{cases}$$

	Hours spent on essay	Grade
	6	82
	10	88
	2	56
	4	64
	6	77
	7	92
	0	23
	1	41
	8	80
	5	59
	3	47
Mean	4.72	64.45
	$\bar{x}$	$\bar{y}$

La régression linéaire simple

Mean	4.72	64.45
	$\bar{x}$	$\bar{y}$

1 - Calculer les valeurs de **b1** (le slope) et **b0** (l'intercept) :

Hours spent on essay	Grade	I
6	82	
10	88	
2	56	
4	64	
6	77	
7	92	
0	23	
1	41	
8	80	
5	59	
3	47	

$$b_1 = \frac{\sum(x - \bar{x}) * (y - \bar{y})}{\sum(x - \bar{x})^2}$$

La régression linéaire simple

$$b_1 = \frac{\sum(x - \bar{x}) * (y - \bar{y})}{\sum(x - \bar{x})^2}$$

1 - Calculer les valeurs de **b1** (le slope) et **b0** (l'intercept) :

Hours spent on essay	Grade	Hours spent – Average Hours Spent $(x - \bar{x})$
6	82	1.27
10	88	5.27
2	56	-2.73
4	64	-0.73
6	77	1.27
7	92	2.27
0	23	-4.73
1	41	-3.73
8	80	3.27
5	59	0.27
3	47	-1.73

Mean	4.72	64.45
	$\bar{x}$	$\bar{y}$

La régression linéaire simple

$$b_1 = \frac{\sum(x - \bar{x}) * (y - \bar{y})}{\sum(x - \bar{x})^2}$$

1 - Calculer les valeurs de **b1** (le slope) et **b0** (l'intercept) :

Hours spent on essay	Grade	Hours spent – Average Hours Spent $(x - \bar{x})$	Grade – Average Grade $(y - \bar{y})$
6	82	1.27	17.55
10	88	5.27	23.55
2	56	-2.73	-8.45
4	64	-0.73	-0.45
6	77	1.27	12.55
7	92	2.27	27.55
0	23	-4.73	-41.45
1	41	-3.73	-23.45
8	80	3.27	15.55
5	59	0.27	-5.45
3	47	-1.73	-17.45

Mean	4.72	64.45
	$\bar{x}$	$\bar{y}$

La régression linéaire simple

$$b_1 = \frac{\sum(x-\bar{x}) * (y-\bar{y})}{\sum(x-\bar{x})^2}$$

1 - Calculer les valeurs de **b1** (le slope) et **b0** (l'intercept) :

Hours spent on essay	Grade	Hours spent – Average Hours Spent $(x-\bar{x})$	Grade – Average Grade $(y-\bar{y})$	$(x-\bar{x}) \times (y-\bar{y})$
6	82	1.27	17.55	22.33
10	88	5.27	23.55	124.15
2	56	-2.73	-8.45	23.06
4	64	-0.73	-0.45	0.33
6	77	1.27	12.55	15.97
7	92	2.27	27.55	62.60
0	23	-4.73	-41.45	195.97
1	41	-3.73	-23.45	87.42
8	80	3.27	15.55	50.88
5	59	0.27	-5.45	-1.49
3	47	-1.73	-17.45	30.15

La régression linéaire simple

Méthode - Ordinary Least Squares (OLS) :

- **Training – Entrainement :**

1 - Calculer les valeurs de **b1** (le slope) et **bo** (l'intercept) :

$$b_1 = \frac{\sum(x-\bar{x}) * (y-\bar{y})}{\sum(x-\bar{x})^2}$$

$$\frac{\sum(x-\bar{x}) * (y-\bar{y}) = 611.36}{\sum(x-\bar{x})^2 = 94.18}$$

➔ **b1 = 611.36/94.18**

= 6.49

La régression linéaire simple

Méthode - Ordinary Least Squares (OLS) :

- **Training – Entrainement :**

1 - Calculer les valeurs de **b1** (le slope) et **bo** (l'intercept) :

$$b_0 = \bar{y} - b_1\bar{x}$$

b1 = 6.49

➔ **bo = 64.45 – (6.49 * 4.72)**

= 33.81

	$\bar{x}$	$\bar{y}$
Mean	4.72	64.45

La régression linéaire simple

Méthode - Ordinary Least Squares (OLS) :

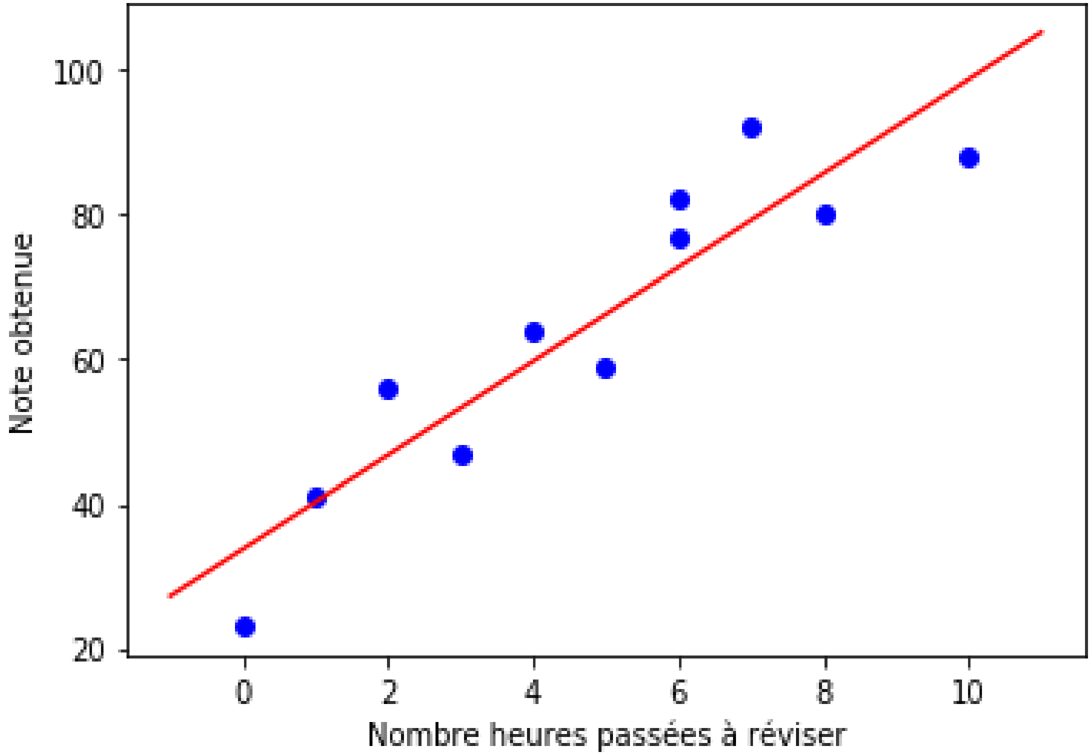
- **Training – Entrainement :**
- Trouver la droite :

$$y = 6.49 * x + 33.81$$

La régression linéaire simple

Méthode - Ordinary Least Squares (OLS) :

- **Training – Entrainement :**
- Trouver la droite : $y = 6.49 * x + 33.81$



La régression linéaire simple

Méthode - Ordinary Least Squares (OLS) :

- **Prédiction - Utilisation :** Prédire la valeur Y (la note) d'un nouvel exemple (**X=6**) selon la fonction trouvée précédemment:

Hours spent on essay	Grade
6	**?**

$$y = 6.49 * x + 33.81$$

➔

$$Y = 6.49 * (6) + 33.81$$

$$= 72.71 \text{ (=Grade)}$$

La régression linéaire simple

Méthode - Ordinary Least Squares (OLS) :

- **Evaluation du modèle :** Prédire les nouvelles valeurs Y des exemples de la **table d'entrainement**, selon la fonction trouvée précédemment:

y = 6.49 * x + 33.81

Hours spent on essay	Grade
6	82
10	88
2	56
4	64
6	77
7	92
0	23
1	41
8	80
5	59
3	47

➔

La régression linéaire simple

Méthode - Ordinary Least Squares (OLS) :

- **Evaluation du modèle :** Prédire les nouvelles valeurs Y des exemples de la **table d'entrainement**, selon la fonction trouvée précédemment:

$$y = 6.49 * x + 33.81$$

Hours spent on essay	Grade
6	82
10	88
2	56
4	64
6	77
7	92
0	23
1	41
8	80
5	59
3	47

➔

Predicted Grade
72.716216
98.681467
46.750965
59.733591
72.716216
79.207529
33.768340
40.259653
85.698842
66.224903
53.242278

La régression linéaire simple

Méthode - Ordinary Least Squares (OLS) :

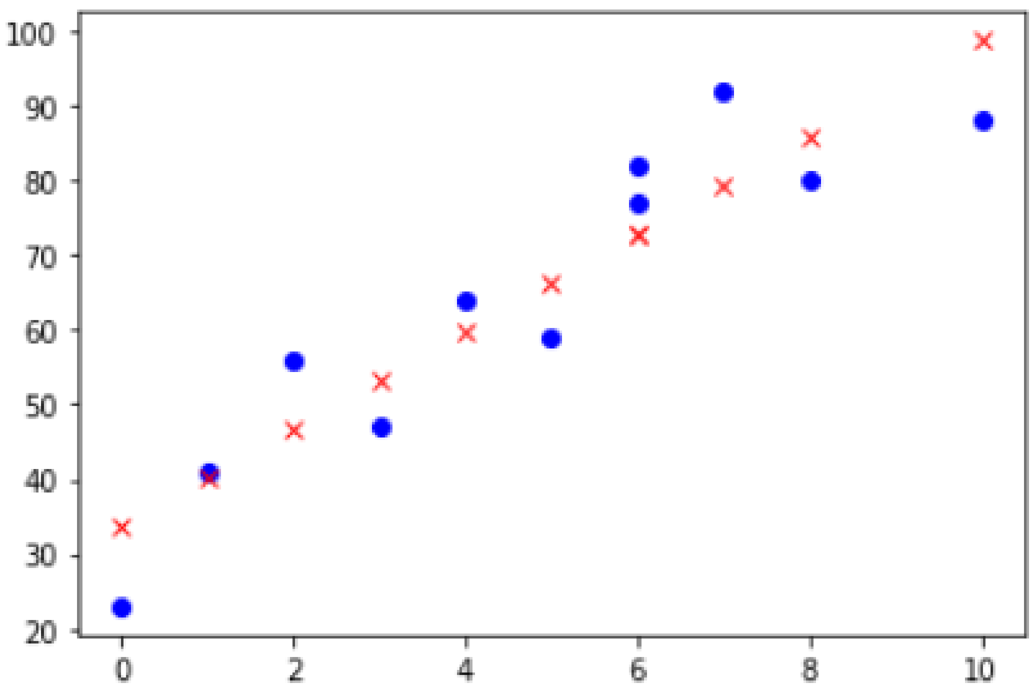
- **Evaluation du modèle :** Prédire les nouvelles valeurs Y des exemples de la **table d'entrainement**, selon la fonction trouvée précédemment:

$$y = 6.49 * x + 33.81$$

En bleu : vraie valeur - **y**
En rouge : valeur prédite – **y_pred**

Hours spent on essay	Grade
6	82
10	88
2	56
4	64
6	77
7	92
0	23
1	41
8	80
5	59
3	47

Predicted Grade
72.716216
98.681467
46.750965
59.733591
72.716216
79.207529
33.768340
40.259653
85.698842
66.224903
53.242278



La régression linéaire simple

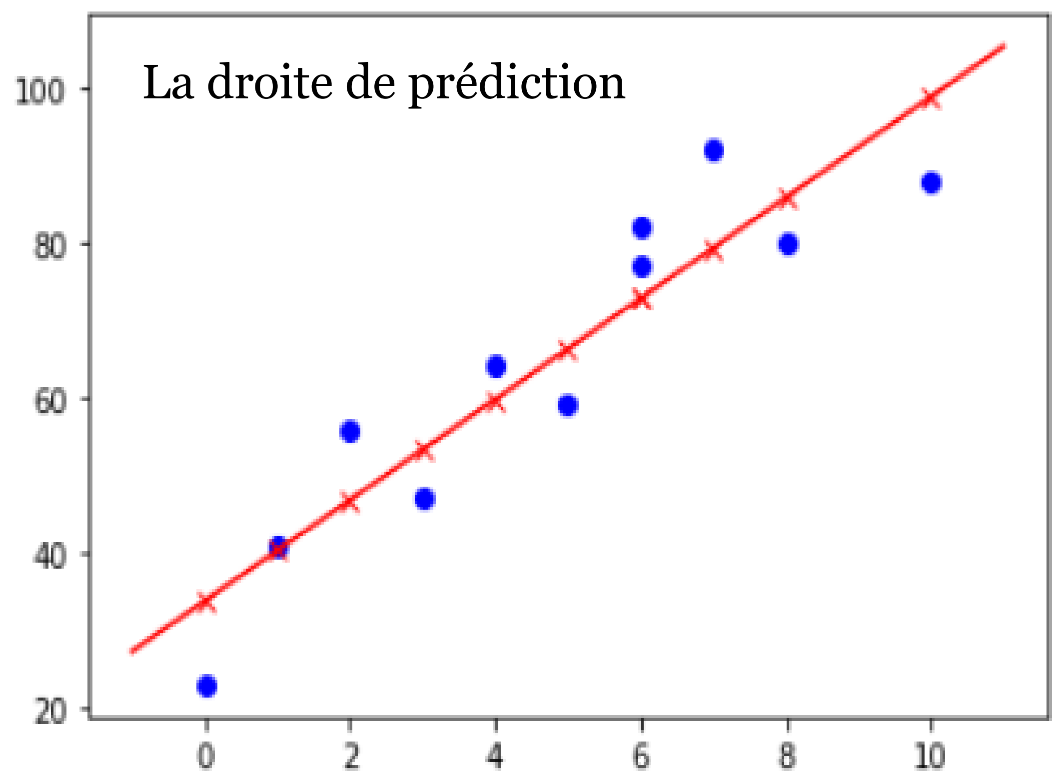
Méthode - Ordinary Least Squares (OLS) :

- **Evaluation du modèle :** Prédire les nouvelles valeurs Y des exemples de la **table d'entrainement**, selon la fonction trouvée précédemment:

$$y = 6.49 * x + 33.81$$

Hours spent on essay	Grade
6	82
10	88
2	56
4	64
6	77
7	92
0	23
1	41
8	80
5	59
3	47

Predicted Grade
72.716216
98.681467
46.750965
59.733591
72.716216
79.207529
33.768340
40.259653
85.698842
66.224903
53.242278



La régression linéaire simple

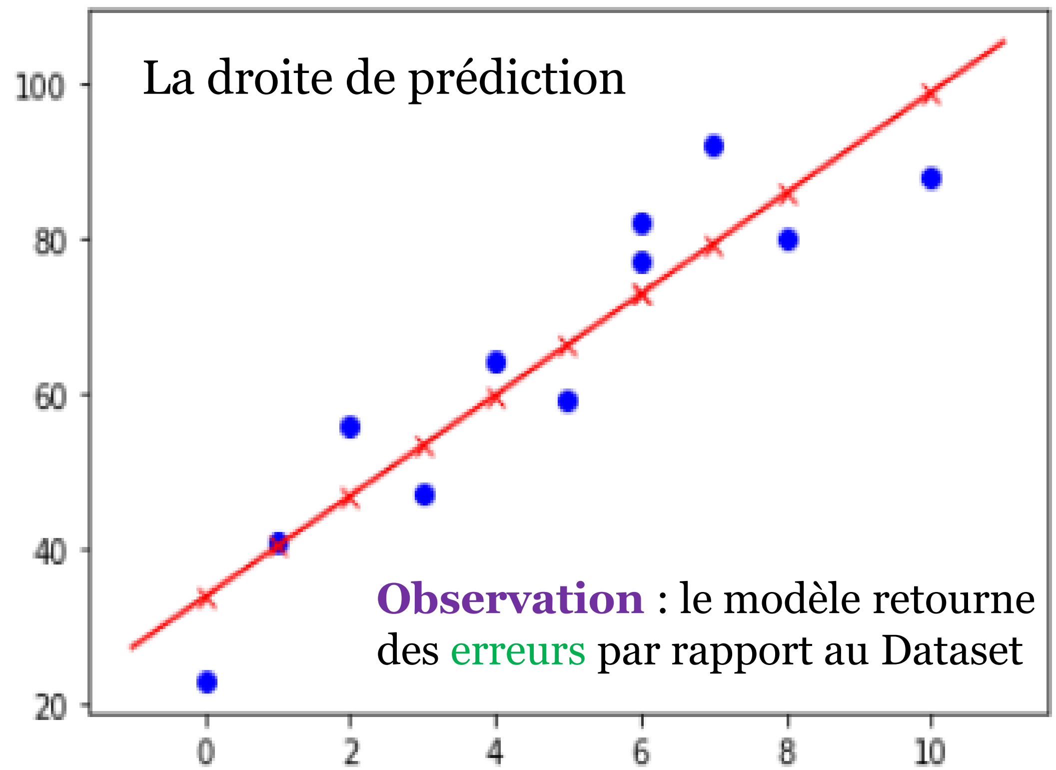
Méthode - Ordinary Least Squares (OLS) :

- **Evaluation du modèle :** Prédire les nouvelles valeurs Y des exemples de la **table d'entrainement**, selon la fonction trouvée précédemment:

$$y = 6.49 * x + 33.81$$

Hours spent on essay	Grade
6	82
10	88
2	56
4	64
6	77
7	92
0	23
1	41
8	80
5	59
3	47

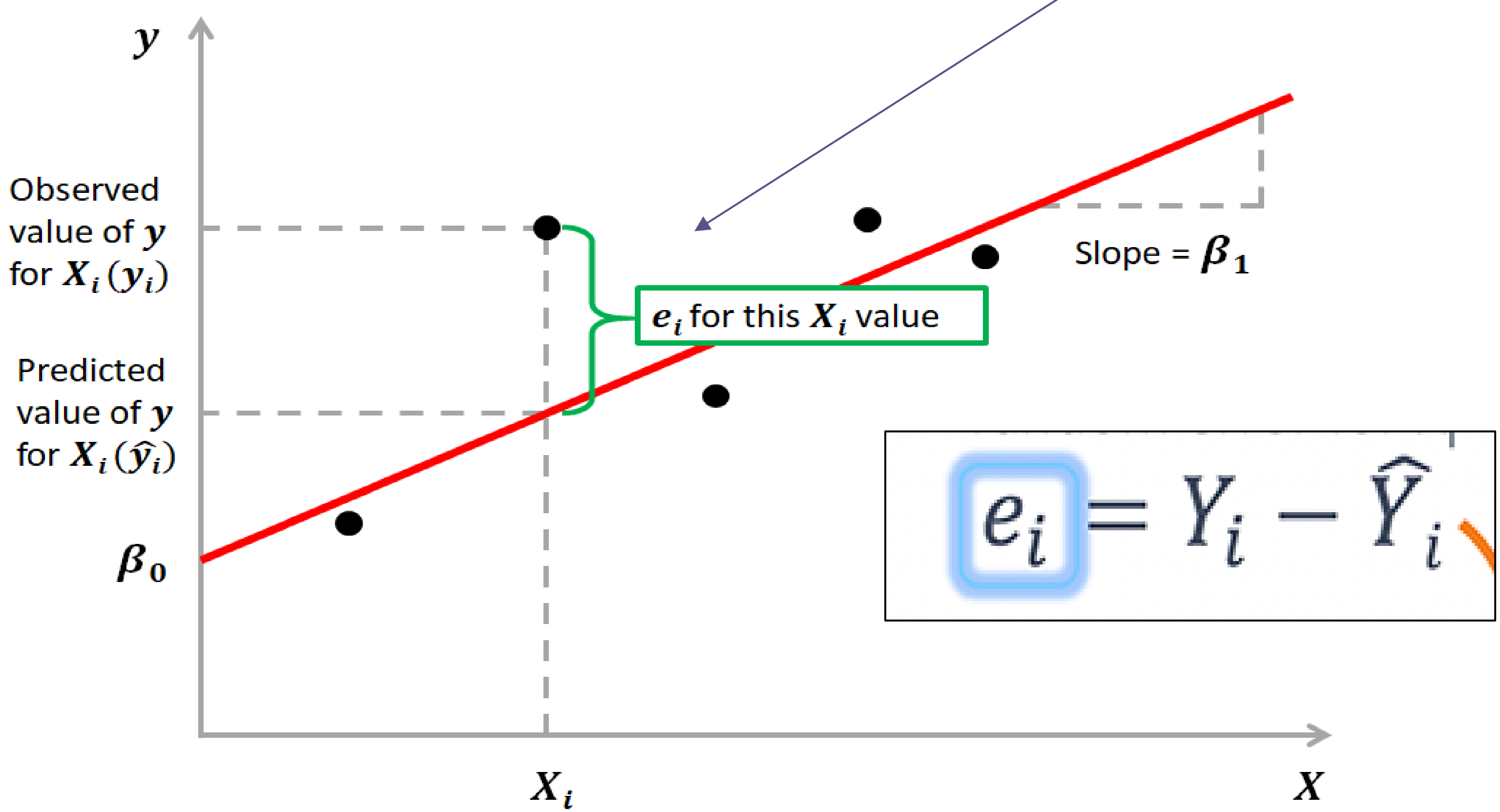
Predicted Grade
72.716216
98.681467
46.750965
59.733591
72.716216
79.207529
33.768340
40.259653
85.698842
66.224903
53.242278



La régression linéaire simple

Méthode - Ordinary Least Squares (OLS) :

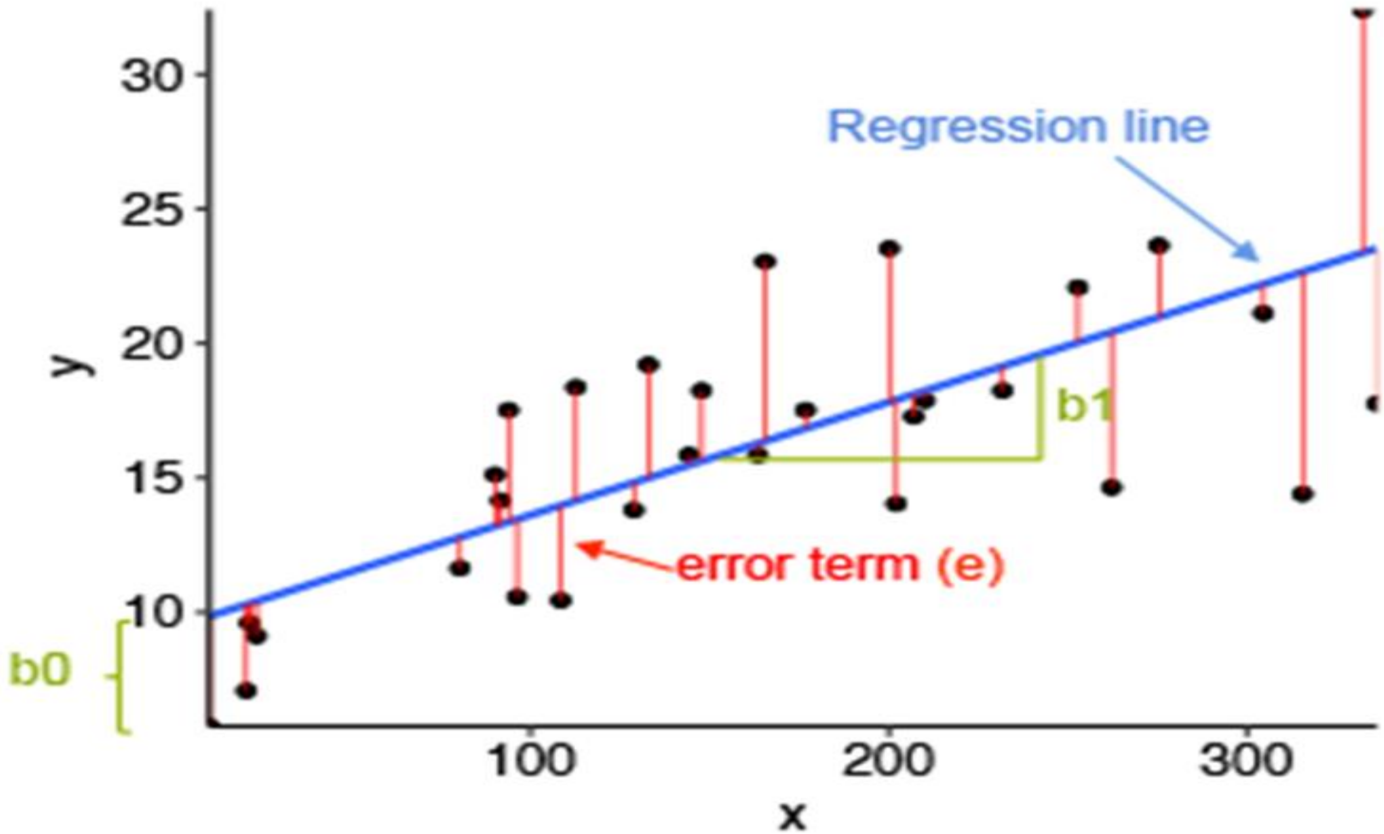
- **Evaluation du modèle** de régression en estimant les **erreurs** sur les prédictions **y_pred** (i.e. f(x)) :



La régression linéaire simple

Méthode - Ordinary Least Squares (OLS) :

- **Evaluation du modèle** de régression en estimant les **erreurs** sur les prédictions **y_pred** (i.e. f(x)) :



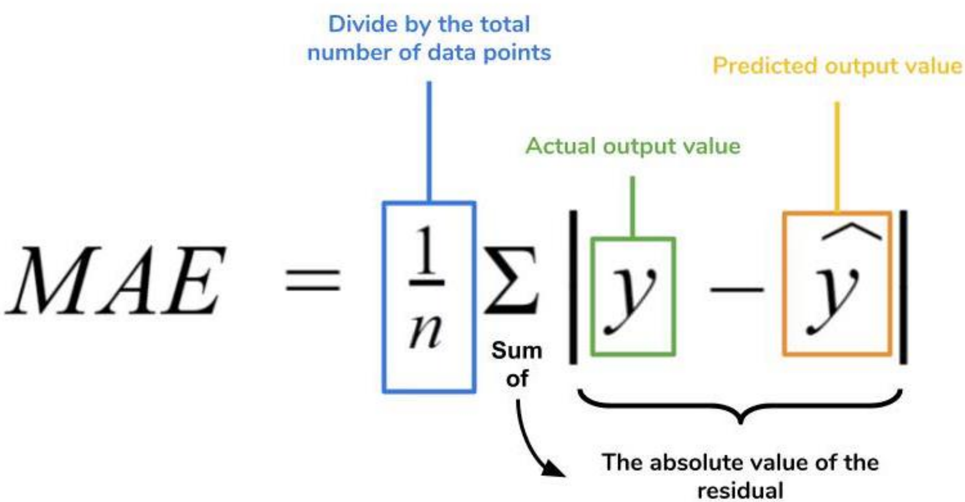
La régression linéaire simple

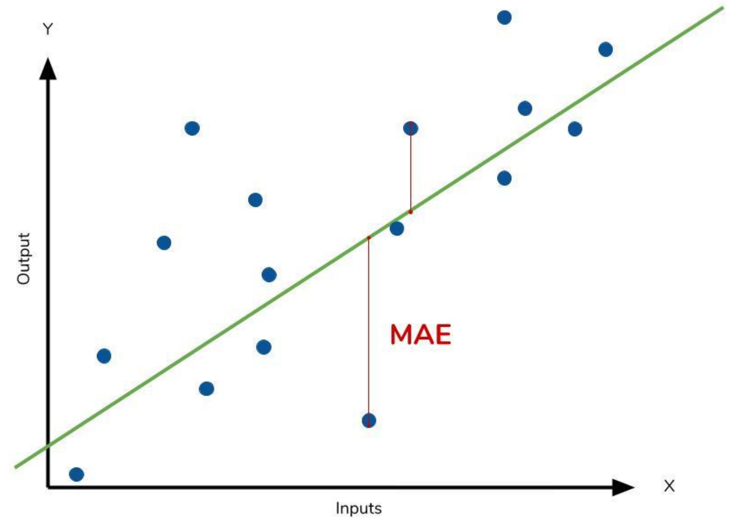
Méthode - Ordinary Least Squares (OLS) :

- **Evaluation du modèle** de régression en estimant les **erreurs** sur les prédictions **y_pred** (i.e. f(x)) :

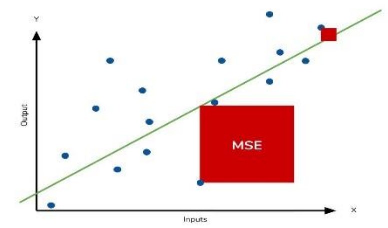
- Différentes **fonctions de coût** permettant d'estimer l'erreur d'un modèle.

- Ex: **MAE** – Mean Absolute Error / l'erreur absolue moyenne.





La régression linéaire simple



Méthode - Ordinary Least Squares (OLS) :

- **Evaluation du modèle** de régression en estimant les **erreurs** sur les prédictions y_pred (i.e. f(x)) :
- Différentes **fonctions de coût** permettant d'estimer l'erreur d'un modèle.
- Ex: **MSE** – Mean Squared Error / l'erreur quadratique moyenne.
- Ex: **RMSE** – Root Mean Squared Error / sqrt (l'erreur quadratique moyenne).

$$MSE = \frac{1}{n} \Sigma \underbrace{\left(y - \widehat{y} \right)^2}_{\text{The square of the difference between actual and predicted}}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

La régression linéaire simple

- **Métriques d'évaluation pour la régression**

Terme	**Contexte**	**But**
Fonction de Perte (Loss)	Mesure l'erreur **d'une seule** prédiction.	Calculer l'écart entre la valeur prédite (y_pred) par le modèle et la valeur réelle (y) pour un seul exemple de données.
Fonction de Coût (Cost)	Mesure l'erreur **moyenne** pour l'ensemble du jeu de données.	Évaluer la performance globale du modèle sur les données d'entraînement. C'est la fonction que l'algorithme d'optimisation (comme la Descente de Gradient) cherche à minimiser.

La régression linéaire simple

- **Métriques d'évaluation pour la régression**

La fonction de coût est généralement la **moyenne** des fonctions de perte calculées sur chaque exemple de l'ensemble d'entraînement.

Terme	**Formule**
Fonction de Perte (Loss)	$\boxed{e_i} = Y_i - \widehat{Y}_i$
Fonction de Coût (Cost)	$J_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$ $J_{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$

La régression linéaire simple

- **Métriques d’évaluation pour la régression**

Classification Metrics	Regression Metrics
Accuracy Correct predictions %	**MSE** Mean Squared Error
Precision Positive predictions % that are actually positive	**RMSE** Root Mean Squared Error
Recall Actual positive cases that were correctly predicted as positive	**MAE** Mean Absolute Error
Confusion Matrix	**R-squared**

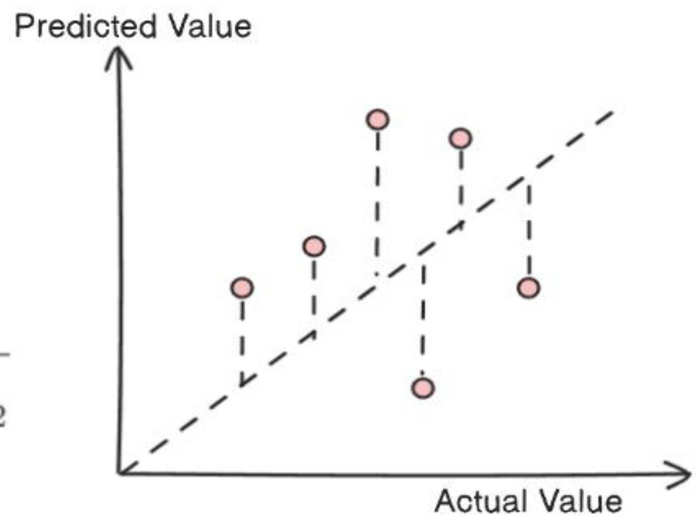
La régression linéaire simple

- **Métriques d'évaluation pour la régression**

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$



$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}$$

$$Adjusted\ R^2 = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1}$$

La régression linéaire simple

- **Evaluation du modèle de régression**

Hours	True Grade	Predicted Grade
6	82	72.716216
10	88	98.681467
2	56	46.750965
4	64	59.733591
6	77	72.716216
7	92	79.207529
0	23	33.768340
1	41	40.259653
8	80	85.698842
5	59	66.224903
3	47	53.242278

$$y = 6.49 * x + 33.81$$

La régression linéaire simple

- **Evaluation du modèle de régression**

Hours	True Grade	Predicted Grade	Error
6	82	72.716216	9.283784
10	88	98.681467	-10.681467
2	56	46.750965	9.249035
4	64	59.733591	4.266409
6	77	72.716216	4.283784
7	92	79.207529	12.792471
0	23	33.768340	-10.768340
1	41	40.259653	0.740347
8	80	85.698842	-5.698842
5	59	66.224903	-7.224903
3	47	53.242278	-6.242278

Fonction de Perte (Loss)

$$\boxed{e_i} = Y_i - \widehat{Y}_i$$

La régression linéaire simple

- **Evaluation du modèle de régression**

Hours	True Grade	Predicted Grade	Error
6	82	72.716216	9.283784
10	88	98.681467	-10.681467
2	56	46.750965	9.249035
4	64	59.733591	4.266409
6	77	72.716216	4.283784
7	92	79.207529	12.792471
0	23	33.768340	-10.768340
1	41	40.259653	0.740347
8	80	85.698842	-5.698842
5	59	66.224903	-7.224903
3	47	53.242278	-6.242278

Fonction de Cout (Cost)

$$MSE = \frac{1}{n} \Sigma \underbrace{\left(y - \widehat{y} \right)^2}_{\text{The square of the difference between actual and predicted}}$$

=> **MSE** = 66.016

La régression linéaire simple

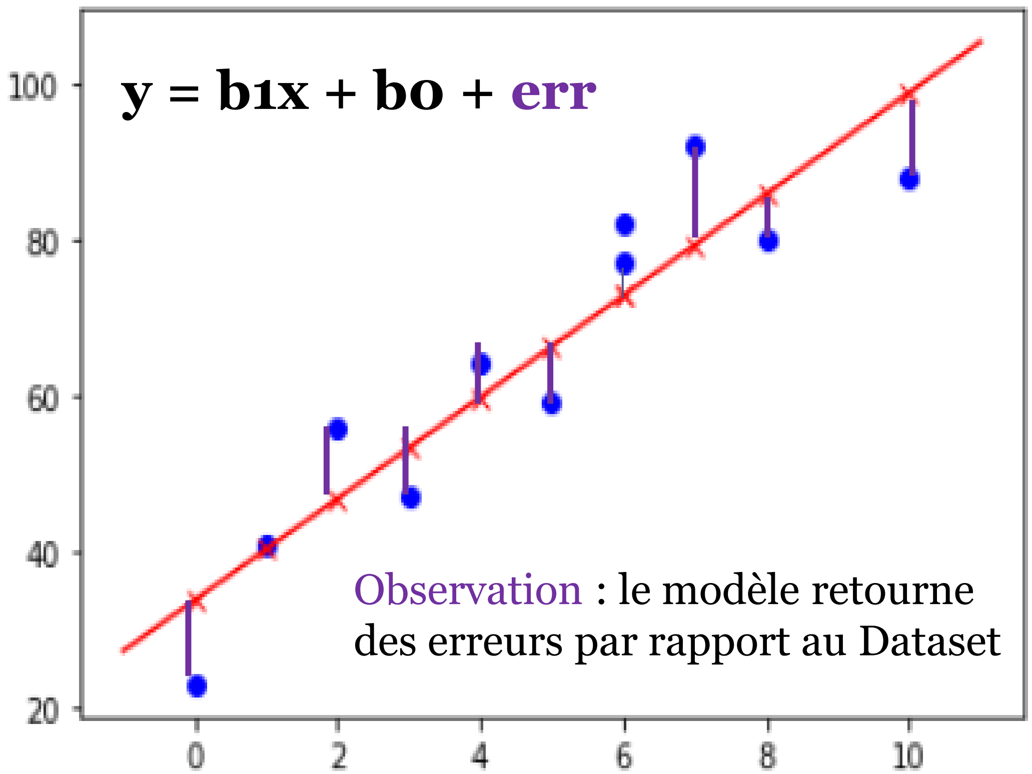
Méthode - Ordinary Least Squares (OLS) :

- **Evaluation du modèle** de régression en estimant les **erreurs (residuals)** sur les prédictions **y_pred** (i.e. f(x)) :

$$MSE = \frac{1}{n}\Sigma\underbrace{\left(y - \hat{y}\right)^2}_{\text{The square of the difference between actual and predicted}}$$

=> **MSE** = 66.016

Chaque prédiction (exemple/datapoint) s'accompagne d'une erreur, on a donc ***n*** **erreurs**.

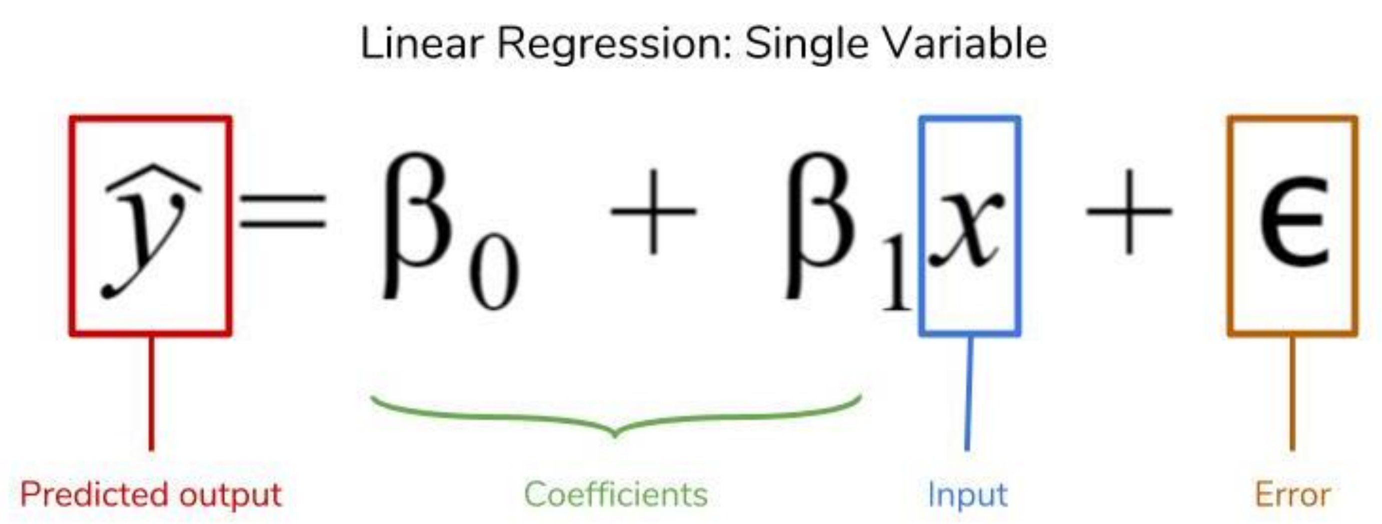


La régression linéaire simple

Méthode - Ordinary Least Squares (OLS) :

- **Evaluation du modèle** de régression en estimant les **erreurs (residuals)** sur les prédictions **y_pred** (i.e. f(x)) :

- **y_pred = b1x + b0 + err**



La régression linéaire simple

- **Evaluation du modèle de régression**

- Les Métriques **à Minimiser** (**Objectif = 0**) : Les fonctions de coût basées sur **l'erreur** mesurent l'écart entre la réalité et la prédiction. Par conséquent, plus l'écart est faible, meilleur est le modèle.

Métrique	**Sigle**	**Objectif**	**Interprétation**
Erreur Absolue Moyenne	**MAE**	Se rapprocher de **0**	**0** signifie qu'en moyenne, l'erreur de prédiction est nulle.
Erreur Quadratique Moyenne	**MSE**	Se rapprocher de **0**	**0** signifie qu'il n'y a aucune erreur quadratique.
Racine de l'Erreur Quadratique Moyenne	**RMSE**	Se rapprocher de **0**	**0** signifie que le modèle est parfait (s'exprime dans l'unité de Y).

La régression linéaire simple

- **Evaluation du modèle de régression**

- Pour les **métriques** d'**erreur** (MAE, MSE, RMSE), la meilleure performance est atteinte lorsque la valeur est **proche de 0**.

Métrique	Sigle	Objectif	Interprétation
Erreur Absolue Moyenne	**MAE**	Se rapprocher de **0**	**0** signifie qu'en moyenne, l'erreur de prédiction est nulle.
Erreur Quadratique Moyenne	**MSE**	Se rapprocher de **0**	**0** signifie qu'il n'y a aucune erreur quadratique.
Racine de l'Erreur Quadratique Moyenne	**RMSE**	Se rapprocher de **0**	**0** signifie que le modèle est parfait (s'exprime dans l'unité de Y).

La régression linéaire simple

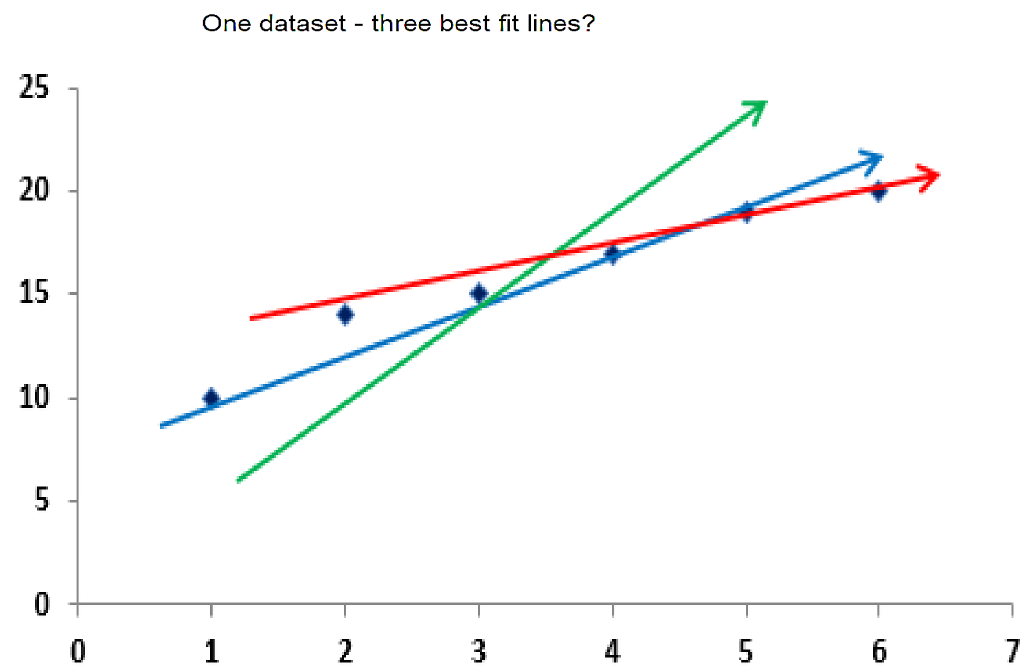
Méthode - Ordinary Least Squares (OLS) :

- Quel est **le meilleur modèle** ? Laquelle de ces droites est la droite la mieux ajustée (performante) ?

- It's called **"least squares"** because it **minimizes the sum of squared residuals** (errors) between the predicted and actual values.

$$MSE = \frac{1}{n}\Sigma\underbrace{\left(y - \widehat{y}\right)^2}_{\text{The square of the difference between actual and predicted}}$$

Chaque prédiction (exemple/datapoint) s'accompagne d'une erreur, on a donc ***n*** **erreurs**.



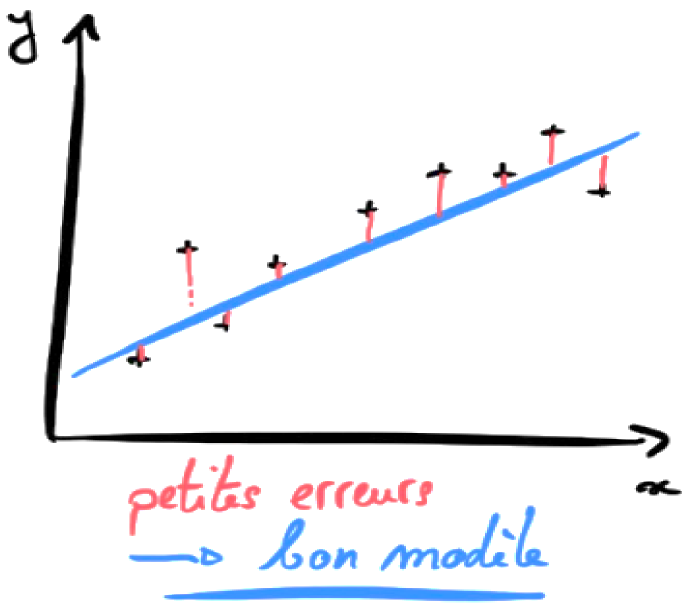
La régression linéaire simple

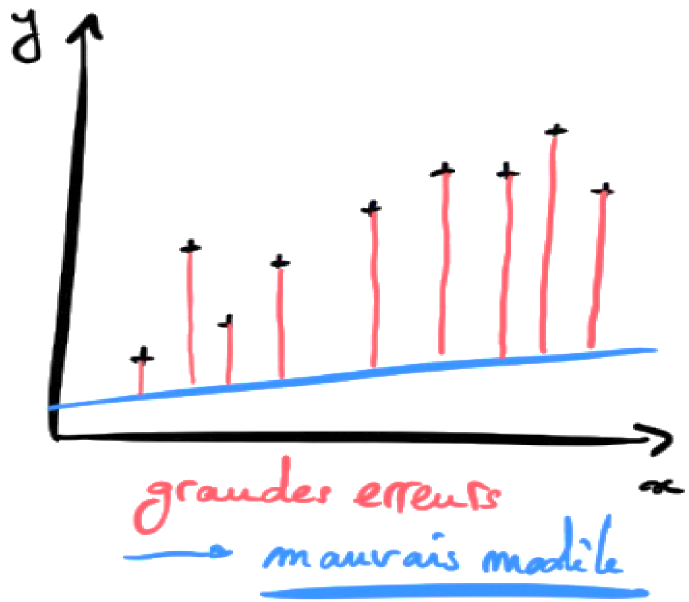
Méthode - Ordinary Least Squares (OLS) :

- **Minimiser l'erreur** en minimisant la fonction coût:

- **y = b1x + b0 + err** => Ramener **err** vers **zéro** => MSE le plus petit possible

- Le but est de trouver les meilleures (**optimales**) estimations des coefficients **b0** et **b1** pour minimiser les erreurs de prédiction de y.





La régression linéaire simple

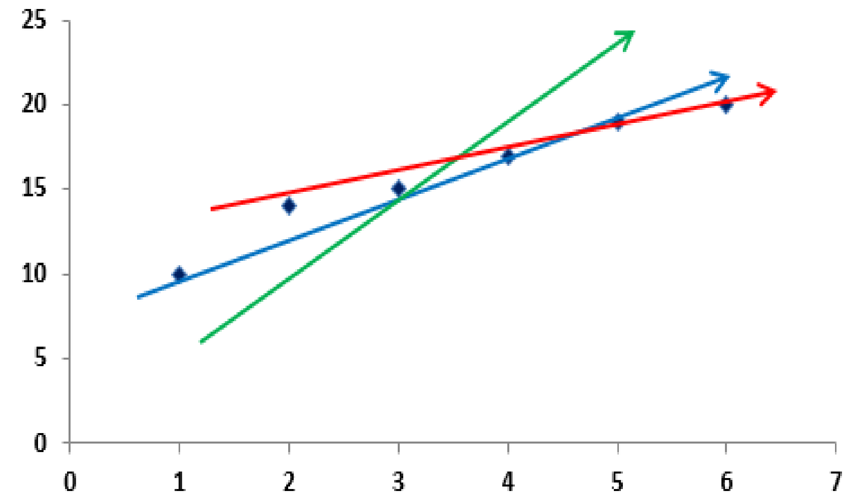
Méthode - Ordinary Least Squares (OLS) :

- **Minimiser l'erreur** en minimisant la fonction coût:

- **y = b1x + b0 + err** => Ramener **err** vers **zéro** => MSE le plus petit possible

  - Le but est de trouver les meilleures (**optimales**) estimations des coefficients **b0** et **b1** pour minimiser les erreurs de prédiction de y.

which of these lines is the best fit line?



Minimiser l'erreur ➔

Problème d'optimisation ➔

Itérer les étapes précédentes en utilisant un **algorithme d'optimisation** cherchant à minimiser la **fonction coût**.

La régression linéaire

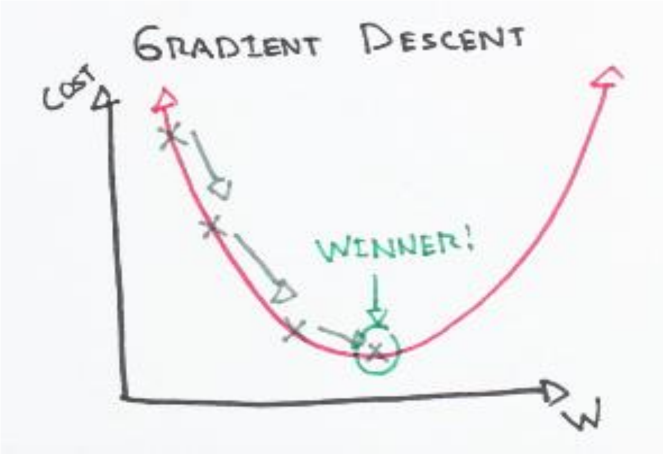
- **Deux méthodes principales**, parmi d'autres :

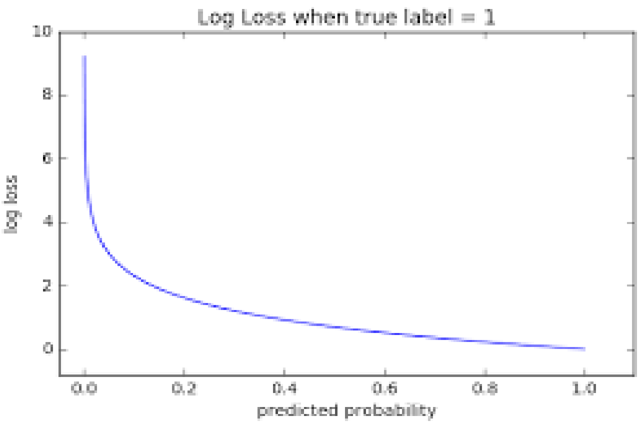
Méthode	**Nom Commun**	**Nature de la Solution**	**Points Clés**
Moindres Carrés Ordinaires (Ordinary Least Squares (OLS))	Régression Linéaire	**Analytique** (Forme Fermée - Closed-Form *Direct solution*)	Calcul direct des coefficients via des formules mathématiques (dérivées à zéro).
Descente de Gradient (Gradient Descent)	Régression Linéaire avec *Optimisation*	**Itérative** (Approximation)	Trouver les coefficients en *ajustant* progressivement les valeurs dans la direction du *minimum* de la fonction de coût.

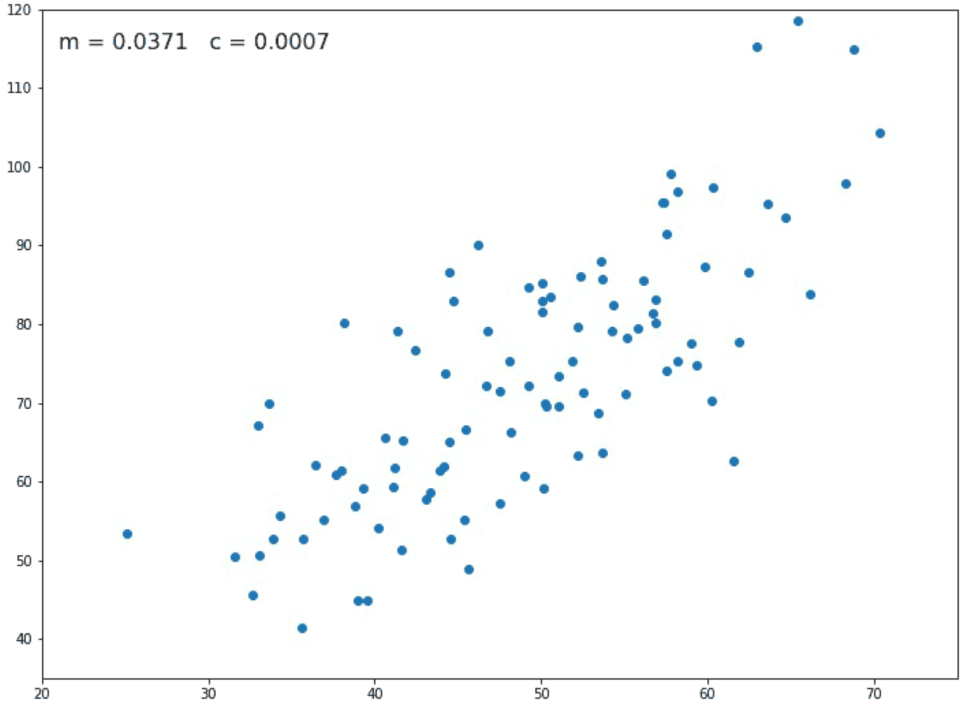
La régression linéaire

Régression linéaire – **Utilisation d'un algorithme d'optimisation**

Algorithme de Gradient Descent (Descente du Gradient)





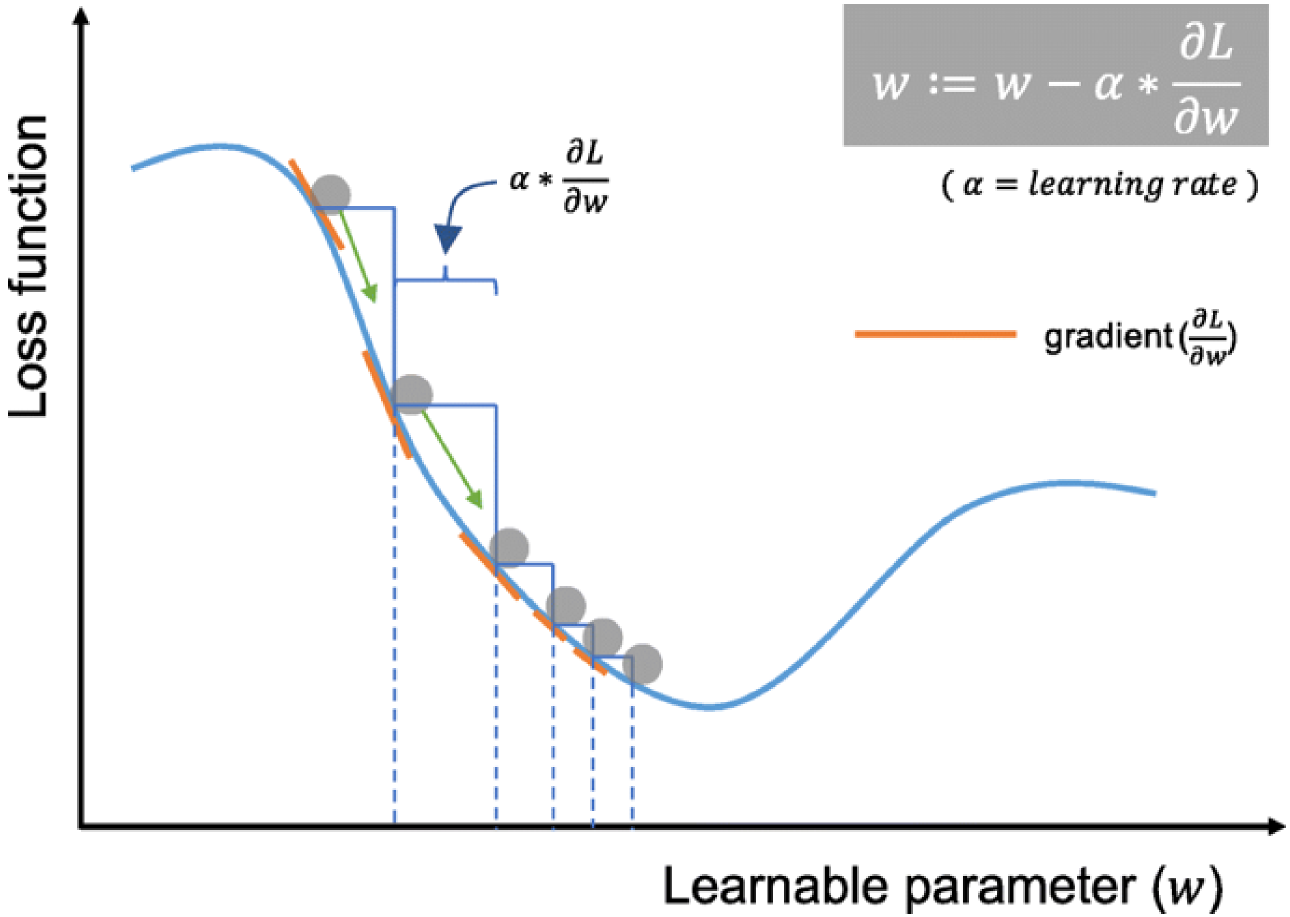


Gif : https://miro.medium.com/max/1400/1*CjTBNFUEI_IokEOXJ0ozKw.gif

La régression linéaire

Régression linéaire – **Utilisation d'un algorithme d'optimisation**

Algorithme de Gradient Descent (Descente du Gradient)



b0 and b1

La régression linéaire

Régression linéaire – **Utilisation d'un algorithme d'optimisation**

Algorithme de Gradient Descent (Descente du Gradient)

Etapes **:**

- Modèle linéaire avec **f(x)** : **y = bj * xj + bo**
- Fonction coût : **MSE** – Mean Squared Error = $\frac{1}{m}\sum_{i=1}^{m}\left(\hat{y}^{(i)} - y^{(i)}\right)^2$

1 - Initialiser les paramètres ***bj*** et ***bo*** à 0

2 - Choisir et fixer le nombre d'itération (***epochs***) et ***learning_rate* α.**

3 – Calculer les prédictions ***y_pred*** pour chaque exemple **i** dans le dataset, selon : **y_pred = bj * xj + bo**

4 – Calculer le gradient : i.e. les dérivées partielles ***D_bj*** et ***D_bo***.

5 - **Mettre à jour** les valeurs de ***bj*** et de ***bo*** en fonction du gradient.

Répéter (3 4 5) *epochs* fois pour minimiser gradient et **optimiser *bj*, *bo***

La régression linéaire

Régression linéaire – **Utilisation d’un algorithme d’optimisation**

Algorithme de Gradient Descent (Descente du Gradient)

Etapes **:**

$$y = b_0 + b_1{*}x_1 + b_2{*}x_2 + \ldots + b_n{*}x_n$$

4 – Calculer le gradient : i.e. les dérivées partielles ***D_bj*** et ***D_bo*** :

- **D_bj** : Dérivée partielle de la fonction coût MSE selon les paramètres bj
- **D_bo** : Dérivée partielle de la fonction coût MSE selon le paramètre bo

$$D\_bj = \frac{\partial}{\partial b_j} MSE = \frac{2}{m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)} \right) x_j^{(i)}$$

La régression linéaire

Régression linéaire – **Utilisation d'un algorithme d'optimisation**

Algorithme de Gradient Descent (Descente du Gradient)

***Etapes* :**

$$y = b_0 + b_1{*}x_1 + b_2{*}x_2 + \dots + b_n{*}x_n$$

4 – Calculer le gradient : i.e. les dérivées partielles ***D_bj*** et ***D_bo*** :

- **D_bj** : Dérivée partielle de la fonction coût MSE selon les paramètres bj
- **D_bo** : Dérivée partielle de la fonction coût MSE selon le paramètre bo

$$\textbf{\textit{D\_bo}} = \frac{\partial}{\partial b_0} MSE = \frac{2}{m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)} \right)$$

La régression linéaire

Régression linéaire – **Utilisation d'un algorithme d'optimisation**

Algorithme de Gradient Descent (Descente du Gradient)

***Etapes* :**

5 – Mettre à jour les valeurs de ***bj*** et de ***bo*** en fonction du gradient et du **learning rate α**, comme suit :

New parameter = old parameter − learning rate × gradient

For b_0:

$$b_0 := b_0 - \alpha \cdot \frac{2}{m} \sum (\hat{y} - y)$$

For each b_j:

$$b_j := b_j - \alpha \cdot \frac{2}{m} \sum (\hat{y} - y) x_j$$

La régression linéaire

Régression linéaire – **Utilisation d'un algorithme d'optimisation**

Algorithme de Gradient Descent (Descente du Gradient)

***Etapes* :**

5 – Mettre à jour les valeurs de ***bj*** et de ***bo*** en fonction du gradient et du **learning rate α**, comme suit :

$$b_j := b_j - \alpha \cdot \frac{\partial}{\partial b_j} MSE$$

Répétez les étapes (3, 4, et 5) ***epochs*** fois; jusqu'à ce que la fonction de coût a une très petite valeur ou idéalement = 0 (ce qui signifie 0 erreur ou 100% de précision).

Les dernières valeurs de **bj** et de **bo** trouvées représentent leurs valeurs optimales.

La régression linéaire

Régression linéaire – **Utilisation d'un algorithme d'optimisation**

Algorithme de Gradient Descent (Descente du Gradient)

```
Initialize b0, b1, ... , bn = 0
Choose learning rate α
repeat until convergence:
    for i in 1..m:
        y_pred[i] = b0 + Σ(bj * xj[i])
    Compute gradients:
        grad_b0 = (2/m) * Σ(y_pred[i] - y[i])
        grad_bj = (2/m) * Σ((y_pred[i] - y[i]) * xj[i]) for each j
    Update parameters:
        b0 = b0 - α * grad_b0
        bj = bj - α * grad_bj for each j
```

1. **Start with random weights**
2. **Predict** using the linear model
3. **Compute error**
4. **Compute gradient**
5. **Update weights**
6. **Repeat until convergence**

La régression linéaire

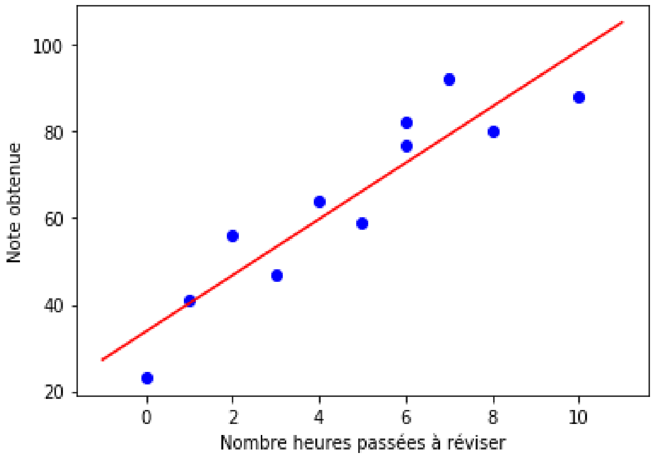
Régression linéaire simple – **Utilisation d'un algorithme d'optimisation**

Algorithme de Gradient Descent (Descente du Gradient)

Entrainement et Exécution sur l'exemple précédent:

On pose : epochs = 11 et l_rate = 0.01

```
>epoch=0, lrate=0.010, m=7.205455, c=1.289091
>epoch=1, lrate=0.010, m=9.834750, c=1.871157
>epoch=2, lrate=0.010, m=10.783632, c=2.192994
>epoch=3, lrate=0.010, m=11.115504, c=2.418682
>epoch=4, lrate=0.010, m=11.220881, c=2.608478
>epoch=5, lrate=0.010, m=11.243171, c=2.784517
>epoch=6, lrate=0.010, m=11.235038, c=2.954926
>epoch=7, lrate=0.010, m=11.215822, c=3.122697
>epoch=8, lrate=0.010, m=11.192622, c=3.288929
>epoch=9, lrate=0.010, m=11.168048, c=3.454030
>epoch=10, lrate=0.010, m=11.143055, c=3.618152
>epoch=11, lrate=0.010, m=11.117996, c=3.781355
```



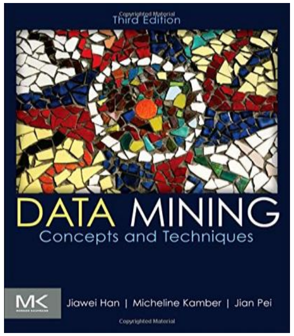
f(x) : $y = m * x + c$

La régression linéaire

- **Autres méthodes principales**, parmi d'autres :

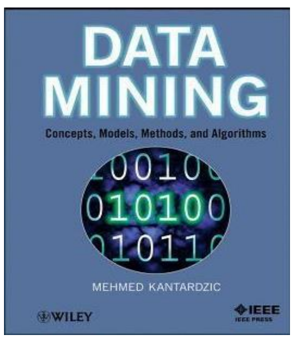
Method	Type	When to Use
Ordinary Least Squares (OLS)	Closed-form solution	When n (samples) is small to medium and X^TX is invertible. Fast and exact.
Gradient Descent (GD)	Optimization	When dataset is large and OLS is too expensive.
Stochastic Gradient Descent (SGD)	Optimization	Very large datasets, streaming data, online learning.
Mini-Batch Gradient Descent	Optimization	Most practical training for ML; balances speed and stability.
Ridge Regression (L2 Regularization)	Regularized Linear Regression (Closed-form or GD)	When features are correlated (multicollinearity). Reduces variance.
Lasso Regression (L1 Regularization)	Regularized Linear Regression (No closed-form; solved via optimization)	When you want automatic feature selection.
Elastic Net	Regularized Linear Regression	When both regularization and feature selection matter.
SVD (Singular Value Decomposition)	Matrix factorization method	Most stable; used in professional numerical computing.

Références



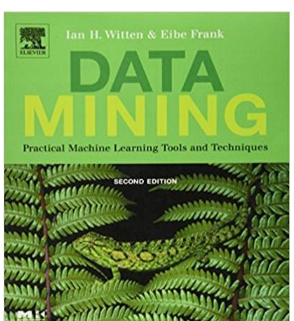
Data Mining : concepts and techniques, 3rd Edition

- ✓ Auteur : Jiawei Han, Micheline Kamber, Jian Pei
- ✓ Éditeur : Morgan Kaufmann Publishers
- ✓ Edition : Juin 2011 - 744 pages - ISBN 9780123814807



Data Mining : concepts, models, methods, and algorithms

- ✓ Auteur : Mehmed Kantardzi
- ✓ Éditeur : John Wiley & Sons
- ✓ Edition : Aout 2011 – 552 pages - ISBN : 9781118029121



Data Mining: Practical Machine Learning Tools and Techniques

- ✓ Auteur : Ian H. Witten & Eibe Frank
- ✓ Éditeur : Morgan Kaufmann Publishers
- ✓ Edition : Juin 2005 - 664 pages - ISBN : 0-12-088407-0

Références

- https://www.technologynetworks.com/informatics/articles/calculating-a-least-squares-regression-line-equation-example-explanation-310265
- https://machinelearningmastery.com/simple-linear-regression-tutorial-for-machine-learning/
- https://towardsdatascience.com/linear-regression-using-gradient-descent-97a6c8700931
- https://machinelearningmastery.com/implement-simple-linear-regression-scratch-python/