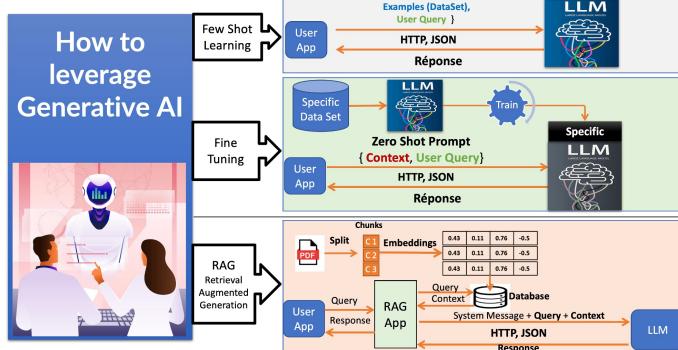


Intelligence Artificielle & IA Générative

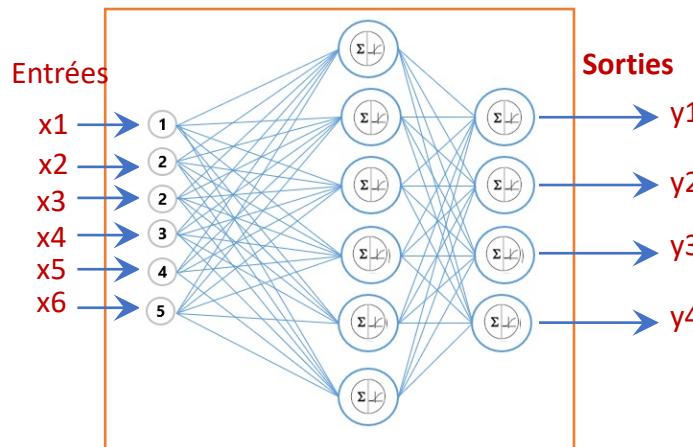


Mohamed Youssfi, Enseignant Chercheur, ENSET Mohammedia, Directeur Laboratoire Informatique, Intelligence Artificielle et Cyber Sécurité

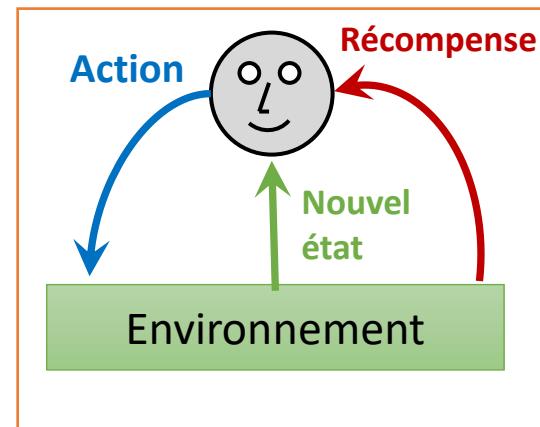
Leverage Generative AI



Apprentissage Supervisé



Apprentissage Par Renforcement



Artificial Intelligence Machine Learning

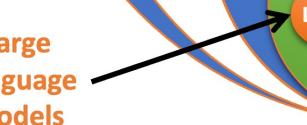
- Data-driven learning**
 - Supervised Learning
 - Unsupervised Learning
 - Self Supervised Learning
- Experiential learning**
 - Reinforcement Learning

Deep Learning

Neural Network
Computer Vision, NLP
CNN, RNN, LSTM, GRU

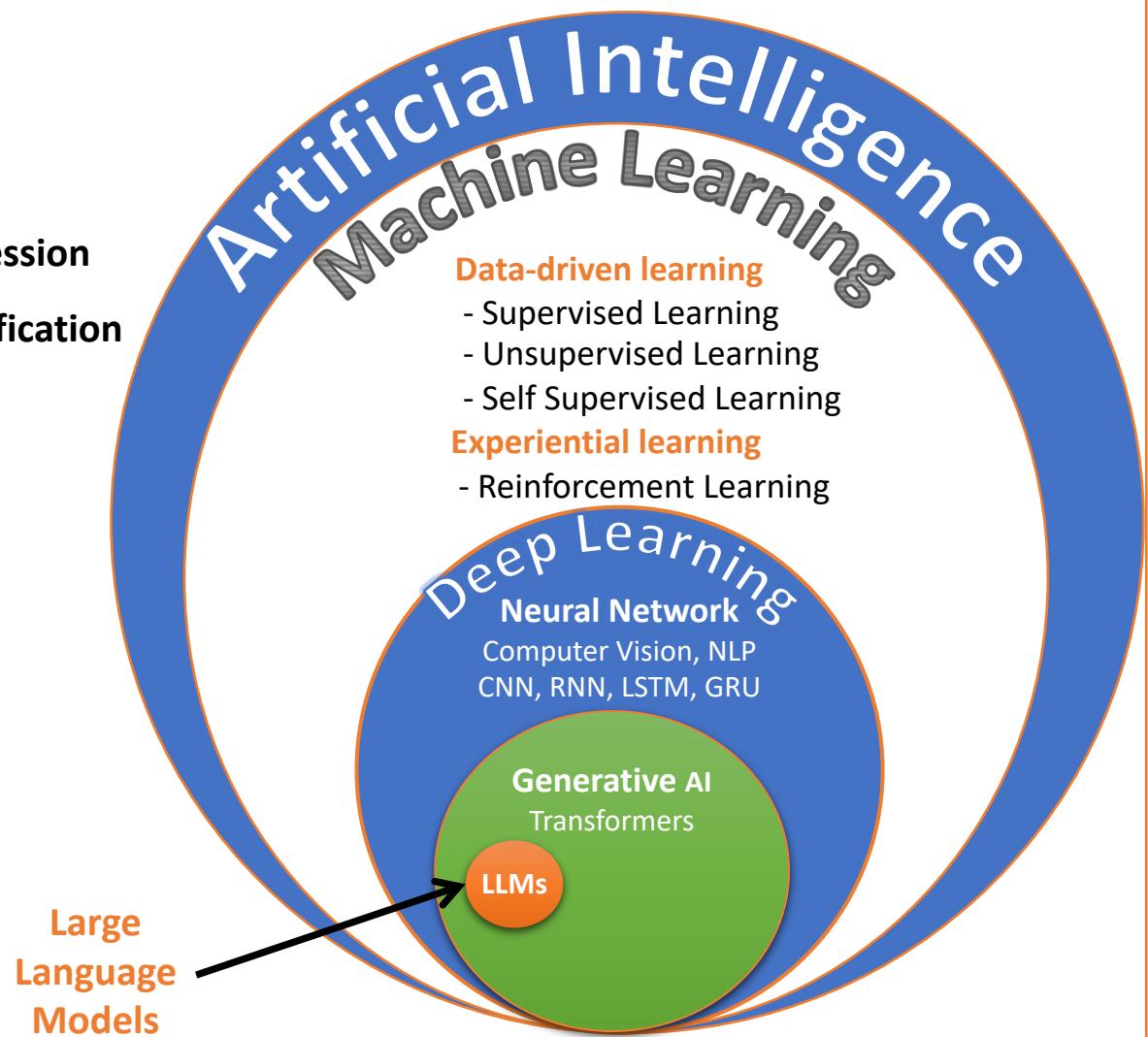
Generative AI
Transformers

Large Language Models



Intelligence Artificielle

- L'intelligence artificielle
- Intelligence Artificielle Distribuée =
 - IA : Pour des agents intelligent (Modéliser le savoir et le comportement)
 - + Distribuée : Modéliser leurs interactions => **Intelligence Collective**
- IA = **IA symbolique (5%) + Machine Learning (95%)**
- Techniques d'apprentissage :
 - Piloté par les données
 - Apprentissage Supervisé
 - Apprentissage Non Supervisé
 - Apprentissage Auto Supervisé
 - Piloté par l'expérience
- Deep Learning
- IA Générative



Intelligence Artificielle

- L'intelligence artificielle
- Intelligence Artificielle Distribuée =
 - IA : Pour des agents intelligent (Modéliser le savoir et le comportement)
 - + Distribuée : Modéliser leurs interactions => **Intelligence Collective**
- IA = IA symbolique (5%) + Machine Learning (95%)
- Techniques d'apprentissage :
 - Piloté par les données
 - Piloté par l'expérience
- Deep Learning
- IA Générative



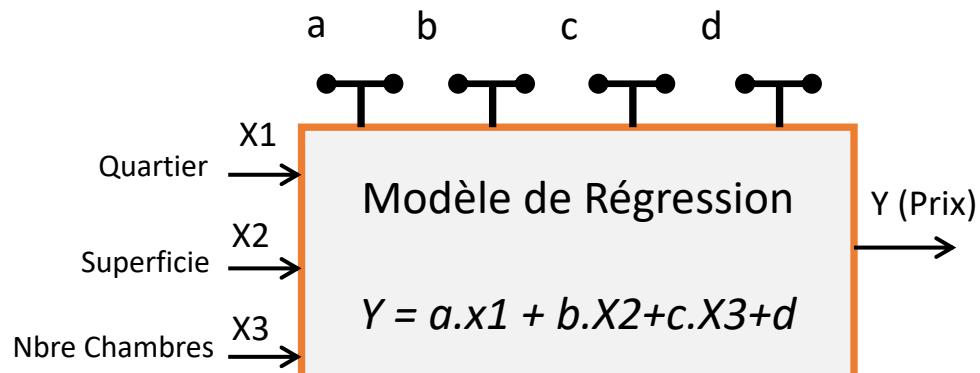
Apprentissage Supervisé

Régression

Exemple : Prédiction du prix d'un appartement

Data Set (Données étiquetées)

Input		Output	
Quartier X1	Superficie X2	Nbre chambres	Prix (MDH) Y
Maarif	100	3	2
SBATA	120	3	1.1
Maarif	100	4	2.1
Bourgogne	80	3	1.44
Maarif	200	5	???????



Modèle à 4 Paramètres a, b, c et d

Algorithmes : Linéaire, Polynomiale, Ridge, Lasso, etc..

Intelligence Artificielle

- L'intelligence artificielle
- Intelligence Artificielle Distribuée =
 - IA : Pour des agents intelligent (Modéliser le savoir et le comportement)
 - + Distribuée : Modéliser leurs interactions => **Intelligence Collective**
- IA = IA symbolique (5%) + Machine Learning (95%)
- Techniques d'apprentissage :
 - Piloté par les données
 - Piloté par l'expérience
- Deep Learning
- IA Générative



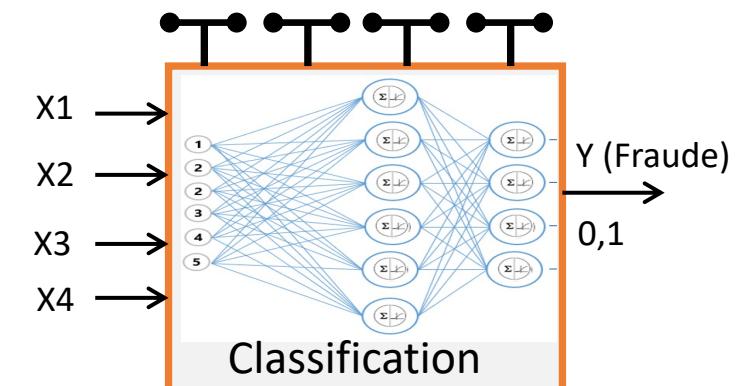
Apprentissage Supervisé

Classification

Exemple : Prédiction du prix d'un appartement

Data Set (Données étiquetées)

	Input		Output		
	Heure X1	Montant X2	Long X3	Lat X4	Fraude
12:20	4500	-1.2	3.2	0	
01:45	1,60	0.6	4.3	1	
10:08	100	-2.45	1.3	0	
11:55	80	-1.2	3.2	1	
00:00	3200	-1	3	???????	



Algorithmes : KNN, SVM, DT, RF, Réseaux de neurones, etc...

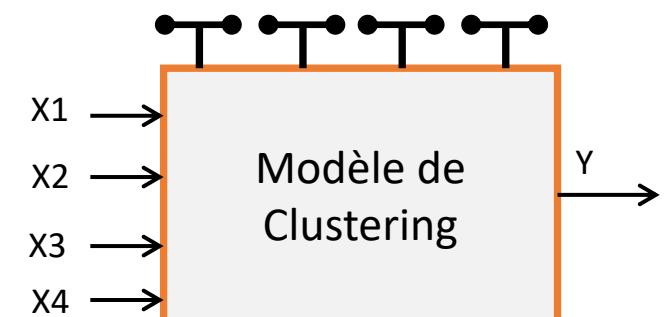
Intelligence Artificielle

- L'intelligence artificielle
- Intelligence Artificielle Distribuée =
 - IA : Pour des agents intelligent (Modéliser le savoir et le comportement)
 - + Distribuée : Modéliser leurs interactions => **Intelligence Collective**
- IA = IA symbolique (5%) + Machine Learning (95%)
- Techniques d'apprentissage :
 - Piloté par les données
 - Apprentissage Supervisé
 - Régression
 - Classification
 - Apprentissage Non Supervisé
 - Apprentissage Auto Supervisé
 - Piloté par l'expérience
- Deep Learning
- IA Générative

Apprentissage Non Supervisé

Clustering

Data Set (Données Non étiquetées)



Algorithmes : K-Means

Intelligence Artificielle

- L'intelligence artificielle
- Intelligence Artificielle Distribuée =
 - IA : Pour des agents intelligent (Modéliser le savoir et le comportement)
 - + Distribuée : Modéliser leurs interactions => **Intelligence Collective**
- IA = IA symbolique (5%) + Machine Learning (95%)
- Techniques d'apprentissage :
 - Piloté par les données
 - Apprentissage Supervisé
 - Apprentissage Non Supervisé
 - Apprentissage Auto Supervisé
 - Piloté par l'expérience
- Deep Learning
- IA Générative

Apprentissage Auto Supervisé

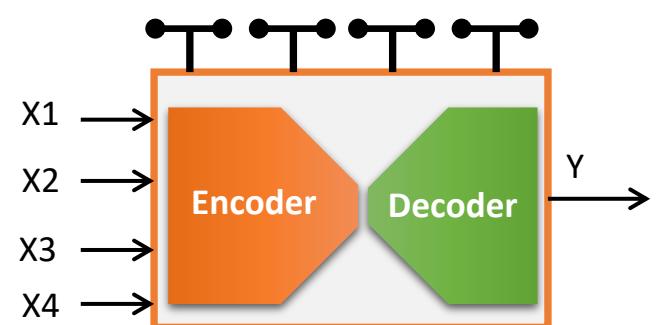
Self-Supervised Learning

Encodeur - Décodeur

Data Set (Données Non étiquetées)

Cacher une partie des données et entraîner le modèle à prédire ce parties cachées en utilisant le principe des encodeurs et décodeurs

Pae exemple en NLP, On cache des mot du texte et on entraîne le modèle pour deviner ces mots cachés



Algorithmes : Transformers (BERT, GPT)

Intelligence Artificielle

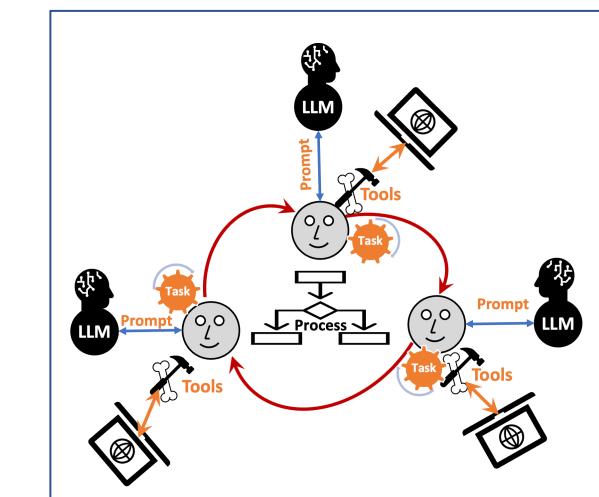
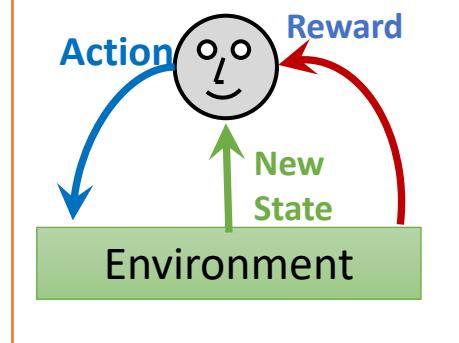
- L'intelligence artificielle
- Intelligence Artificielle Distribuée =
 - IA : Pour des agents intelligent (Modéliser le savoir et le comportement)
 - + Distribuée : Modéliser leurs interactions => **Intelligence Collective**
- IA = IA symbolique (5%) + Machine Learning (95%)
- Techniques d'apprentissage :
 - Piloté par les données
 - Apprentissage Supervisé
 - Régression
 - Classification
 - Apprentissage Non Supervisé
 - Apprentissage Auto Supervisé
- Piloté par l'expérience
- Deep Learning
- IA Générative

Apprentissage Par Renforcement

Reinforcement Learning

Embodiment

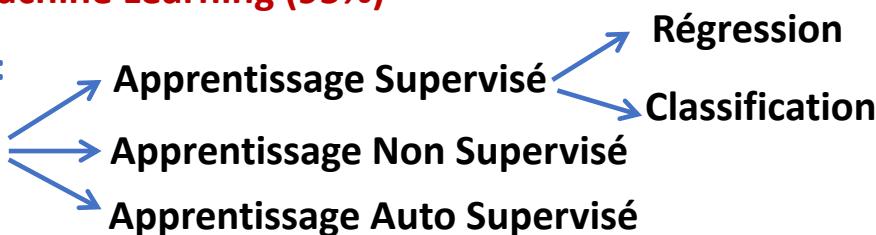
Reinforcement learning



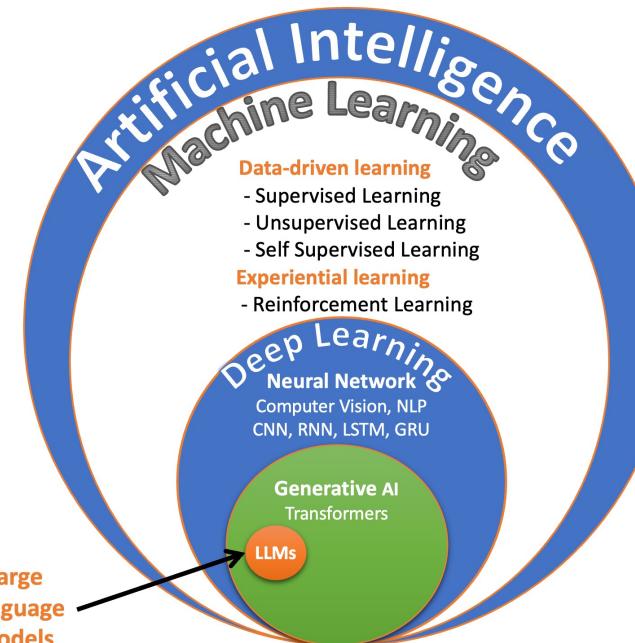
Algorithmes : Q-Learning, PPO

Intelligence Artificielle

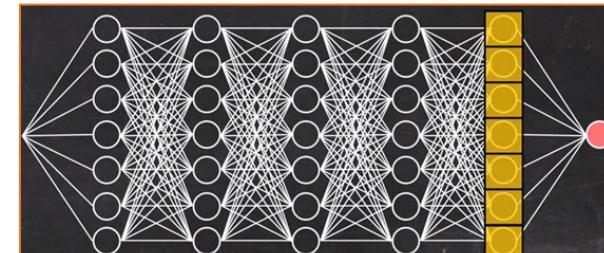
- L'intelligence artificielle
- Intelligence Artificielle Distribuée =
 - IA : Pour des agents intelligent (Modéliser le savoir et le comportement)
 - + Distribuée : Modéliser leurs interactions => **Intelligence Collective**
- IA = IA symbolique (5%) + Machine Learning (95%)
- Techniques d'apprentissage :
 - Piloté par les données
 - Apprentissage Supervisé
 - Apprentissage Non Supervisé
 - Apprentissage Auto Supervisé
 - Piloté par l'expérience
- Deep Learning
- IA Générative



Deep Learning



500 x 400 = 200 000 Pixels



CNN : [Convolution, RELU, MAX PULLING] [Fully Connected]

RNN : [Recurrent Neural Network]

Intelligence Artificielle

- L'intelligence artificielle
- Intelligence Artificielle Distribuée =
 - IA : Pour des agents intelligent (Modéliser le savoir et le comportement)
 - + Distribuée : Modéliser leurs interactions => **Intelligence Collective**

- IA = IA symbolique (5%) + Machine Learning (95%)

- Techniques d'apprentissage :



- Piloté par les données

- Piloté par l'expérience

- Deep Learning

- IA Générative

Text Prompt

LLM

GPT-4
LIAMA 3

(Text + Images) Prompt

LLM

GPT-4o
LIAMA 3.2

(Text + Images) Prompt

LLM

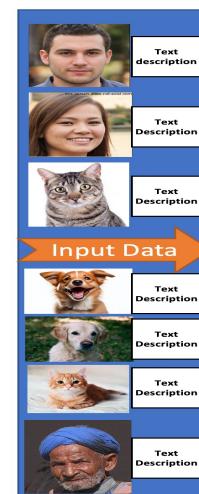
DALL-E
Stable Diffusion

Text Generation

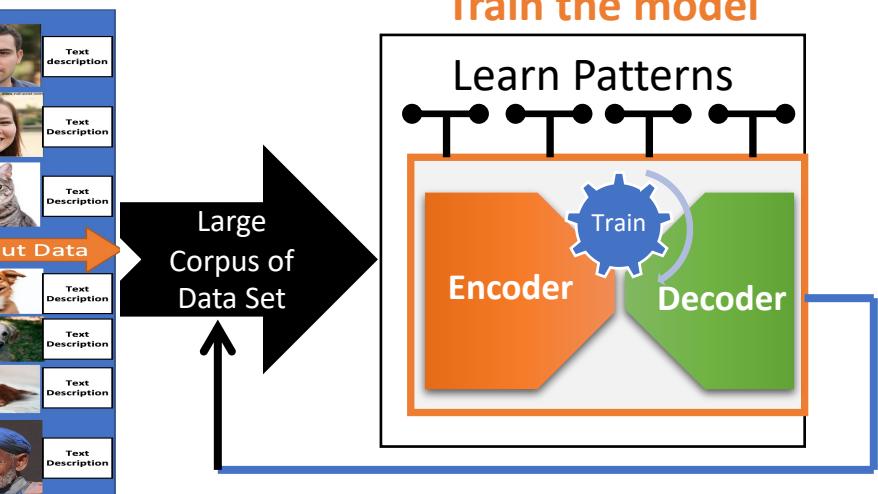
Text Generation

Image Generation

L'IA Générative est une catégorie de l'IA capable de générer des nouvelles données en conséquence d'un raisonnement conditionné par un prompt utilisateur en utilisant les LLMs (Large Language Models).

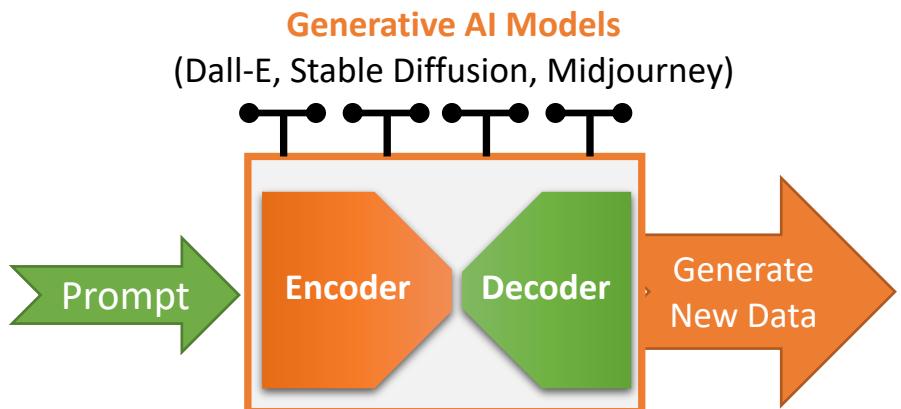


Large Corpus of Data Set



Generative AI Models

(Dall-E, Stable Diffusion, Midjourney)



Concepts fondamentaux de l'IA

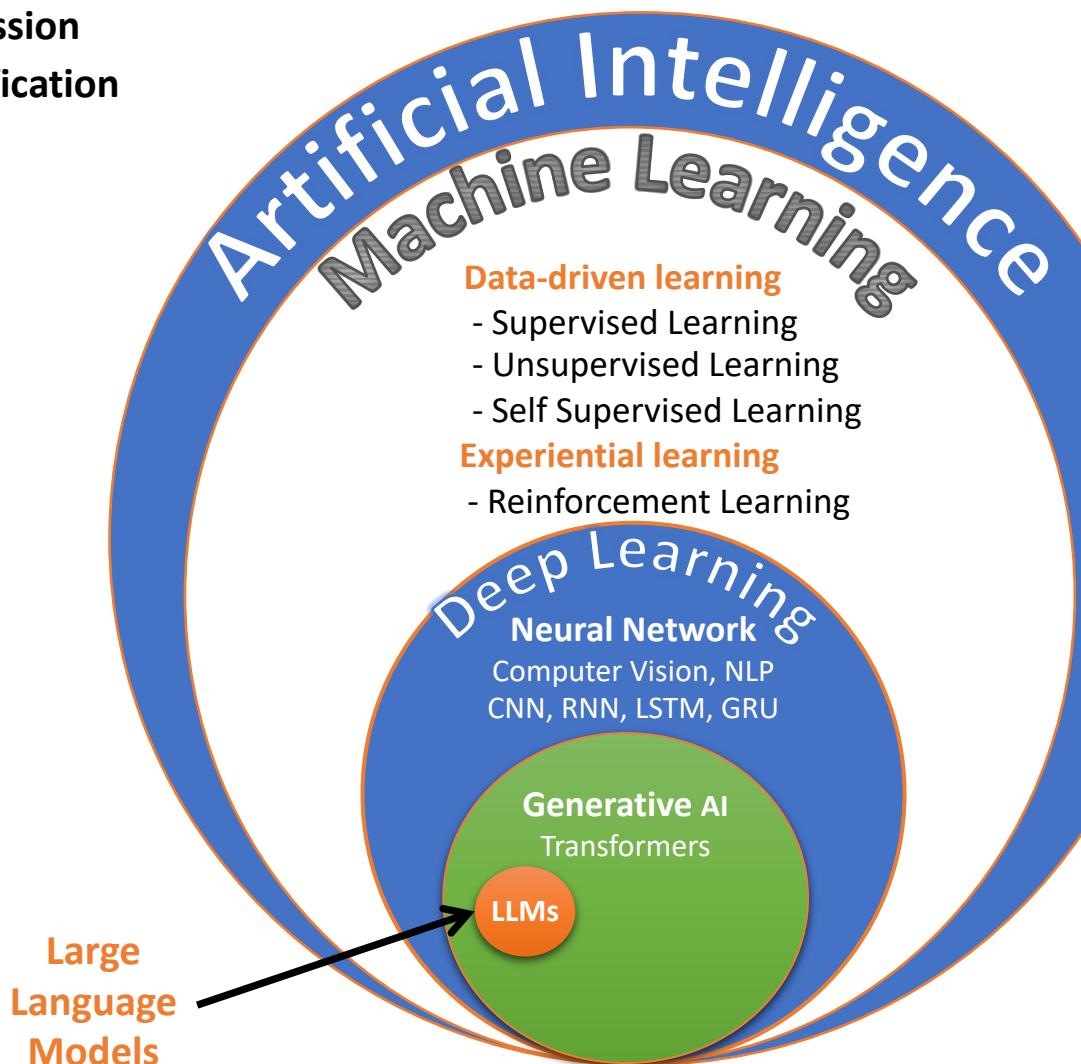
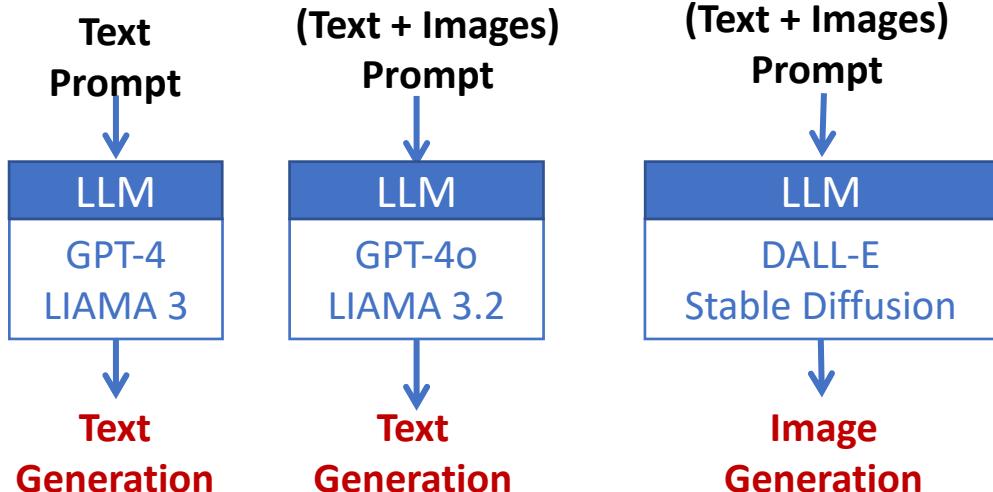
- L'intelligence artificielle
- IA = IA symbolique (5%) + Machine Learning (95%)
- Techniques d'apprentissage :
 - Piloté par les données
 - Piloté par l'expérience (Apprentissage par Renforcement)



- Intelligence Artificielle Distribuée =
 - IA : Pour des agents intelligent (Modéliser le savoir et le comportement)
 - + Distribuée : Modéliser leurs interactions => **Intelligence Collective**

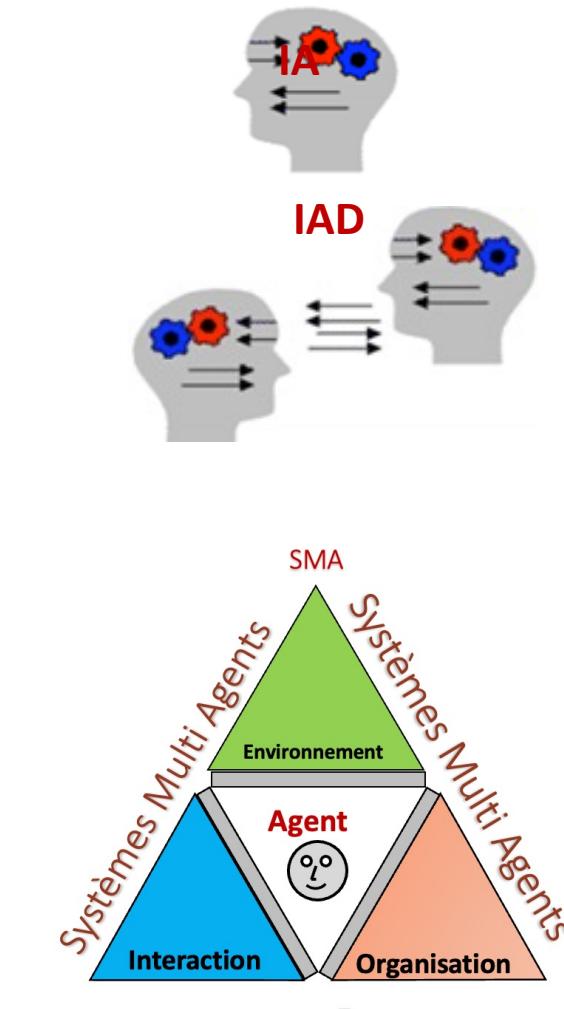
Generative AI

Large Language Models (LLMs) : Transformers (GPT, BERT)



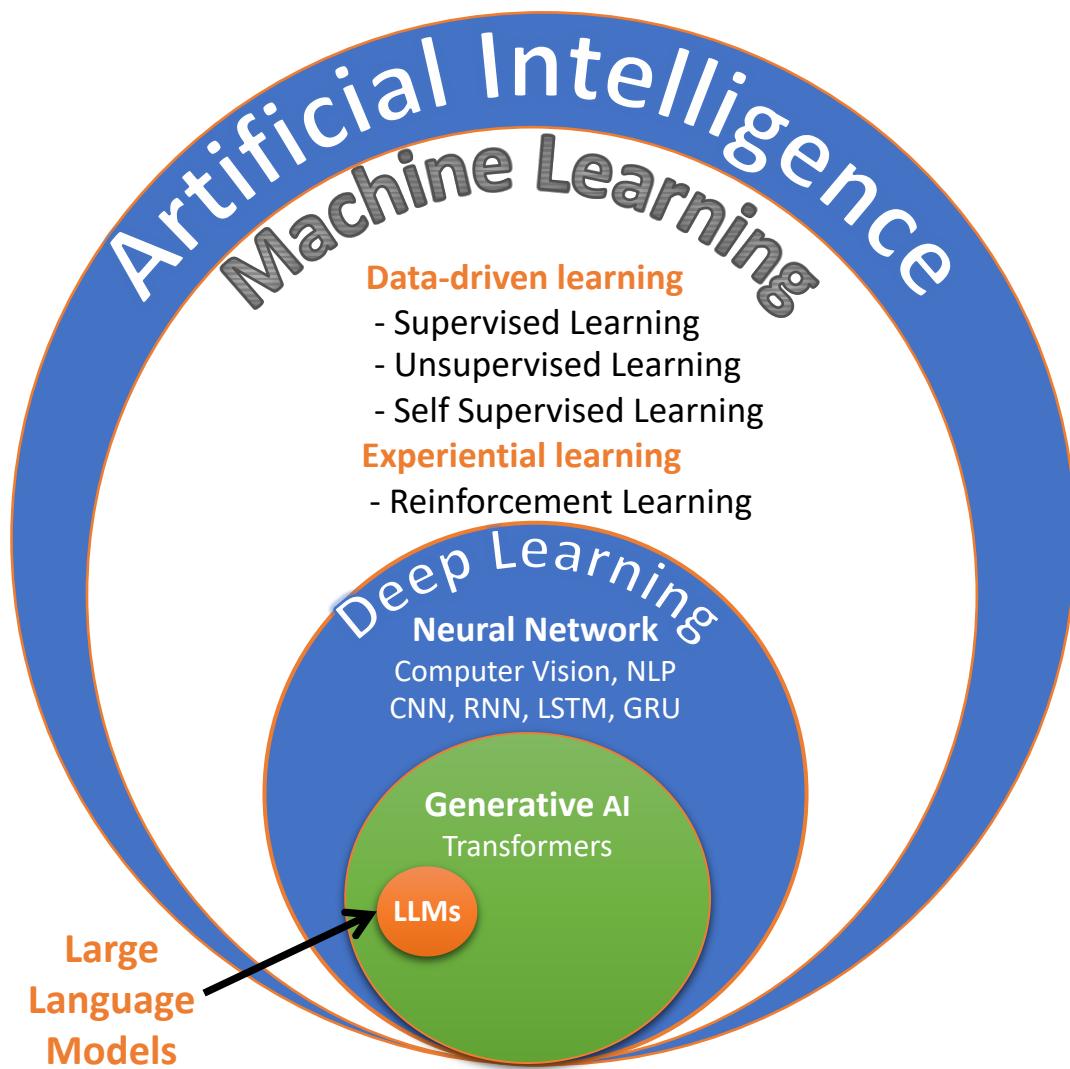
Intelligence Artificielle Distribuée & AI Agents

- **L'intelligence artificielle** est une discipline qui cherche doter les systèmes informatiques avec des capacités intellectuelles semblables à celle des êtres humains et des animaux en utilisant des algorithmes
- **IA et IAD :**
 - L'Intelligence Artificielle permet de modéliser un penseur isolé en exploitant l'intelligence individuelle d'un agent.
 - L'Intelligence Artificielle Distribuée permet d'exploiter l'intelligence collective de plusieurs agents qui vont collaborer selon une planification et une organisation pour participer ensemble à résoudre des problèmes complexes. Cette discipline est connue par les Systèmes Multi Agents. Un Agent est une entité autonome entraînée pour prendre des décisions pour atteindre un but pour lequel il a été créé.



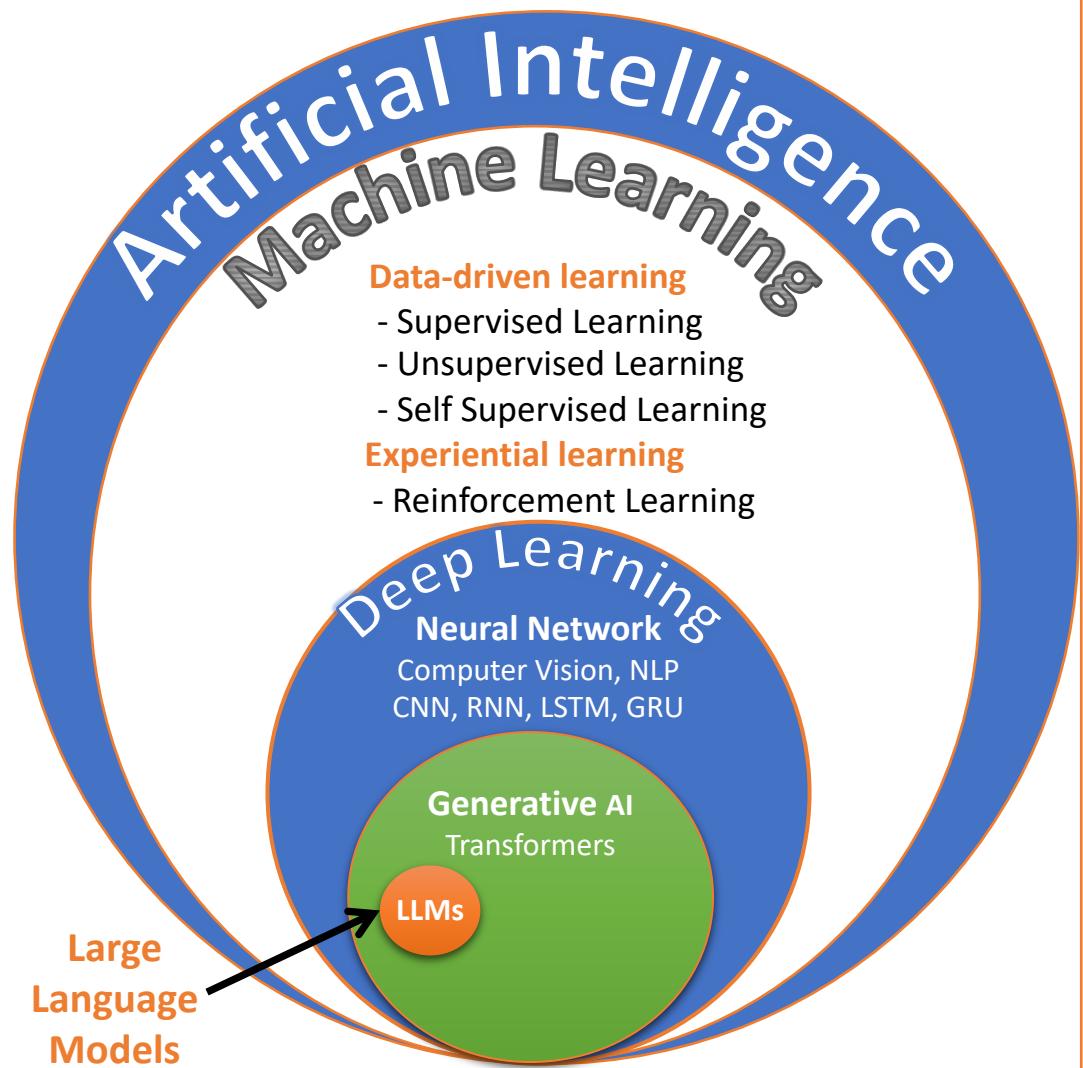
IA Symbolique et Machine Learning

- Il existe deux **familles d'algorithmes de l'IA** :
 - l'IA Symbolique**: Des techniques qui n'ont pas pu décoller et donc rarement utilisées aujourd'hui dans les applications de l'IA. Elle couvre moins de 5% des applications industrielles.
 - Machine Learning** (Apprentissage automatique) (95%): Ce sont les techniques qui sont les plus exploitées aujourd'hui dans les applications de l'IA



Machine Learning

- Dans le domaine du Machine Learning, il existe deux façons pour entraîner les algorithmes :
 - Apprentissage piloté par les données
 - Apprentissage piloté par l'expérience (Apprentissage par renforcement)
- Apprentissage piloté par les données :
 - Consiste à entraîner des algorithmes à effectuer des prédictions en utilisant un ensemble de données collectées du domaine réel étudié.
 - Il existe trois types d'apprentissage pilotés par les données** :
 - Apprentissage supervisé
 - Apprentissage non supervisé (Clustering)
 - Apprentissage Auto Supervisé (Self Supervised Learning)
 - Apprentissage piloté par l'expérience (Apprentissage par renforcement)



Apprentissage Supervisé

- Consiste à utiliser un data set étiqueté. C'est-à-dire des données dont on connaît les inputs et les outputs. Les outputs représentent des étiquettes ou des valeurs fournis par les experts du métier.
- Dans l'apprentissage supervisé on distingue deux types de problèmes :
 - **Régression** : Consiste à prédire, en sortie, une valeur continue comme le prix d'un appartement ou la durée de vie d'une pièce mécanique ou la durée de guérison d'un patient : Exemple Régression Linéaire
 - **Classification** : Consiste à prédire des classes d'appartenance parmi un ensemble de classe finies. Par exemple prédire si une transaction est frauduleuse ou encore prédire si un animal est un chien, un chat, un tigre ou un lapin.
 - Exemples d'algorithmes : Régression logistique, Support Vector Machine (SVM), Multi Layer Perceptron (MLP), KNN (K plus proches voisins), Decision Tree, Random Forest, etc.

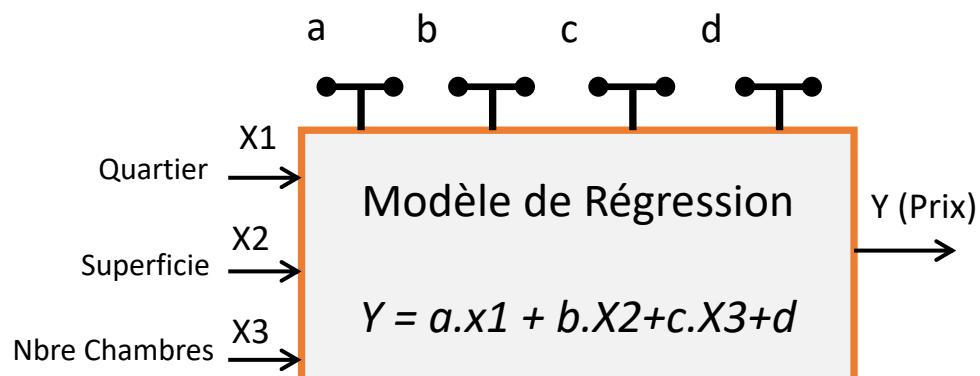
Apprentissage Supervisé

Régression

Exemple : Prédiction du prix d'un appartement

Data Set (Données étiquetées)

Input			Output
Quartier X1	Superficie X2	Nbre chambres	Prix (MDH) Y
Maarif	100	3	2
SBATA	120	3	1.1
Maarif	100	4	2.1
Bourgogne	80	3	1.44
Maarif	200	5	???????



Modèle à 4 Paramètres a, b, c et d

Algorithmes : Linéaire, Polynomiale, Ridge, Lasso, etc..

Apprentissage Supervisé

- Consiste à utiliser un data set étiqueté. C'est-à-dire des données dont on connaît les inputs et les outputs. Les outputs représentent des étiquettes ou des valeurs fournies par les experts du métier.
- Dans l'apprentissage supervisé on distingue deux types de problèmes :
 - Régression : Consiste à prédire, en sortie, une valeur continue comme le prix d'un appartement ou la durée de vie d'une pièce mécanique ou la durée de guérison d'un patient : Exemple Régression Linéaire
 - Classification : Consiste à prédire des classes d'appartenance parmi un ensemble de classes finies. Par exemple prédire sur une transaction est frauduleuse ou encore prédire si un animal est un chien, un chat, un tigre ou un lapin. Exemples : Régression logistique, Support Vector Machine (SVM), Multi Layer Perceptron (MLP), KNN (K plus proches voisins), Decision Tree, Random Forest, etc.

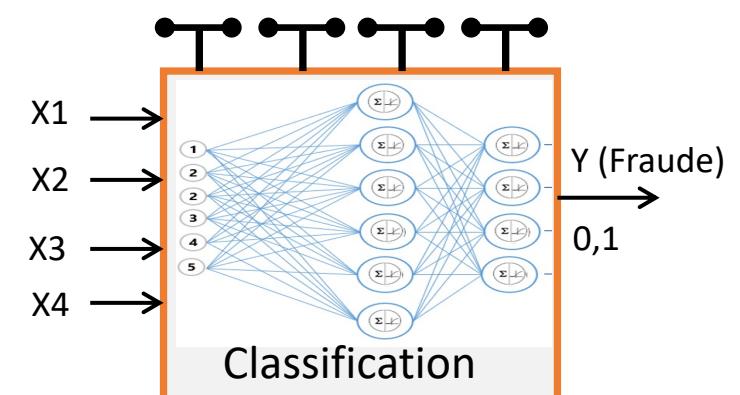
Apprentissage Supervisé

Classification

Exemple : Prédiction du prix d'un appartement

Data Set (Données étiquetées)

Input				Output
Heure X1	Montant X2	Long X3	Lat X4	Fraude
12:20	4500	-1.2	3.2	0
01:45	1,60	0.6	4.3	1
10:08	100	-2.45	1.3	0
11:55	80	-1.2	3.2	1
00:00	3200	-1	3	??????



Algorithmes : KNN, SVM, DT, RF, Réseaux de neurones, etc...

Apprentissage Non Supervisé

- **Apprentissage Non Supervisé :**

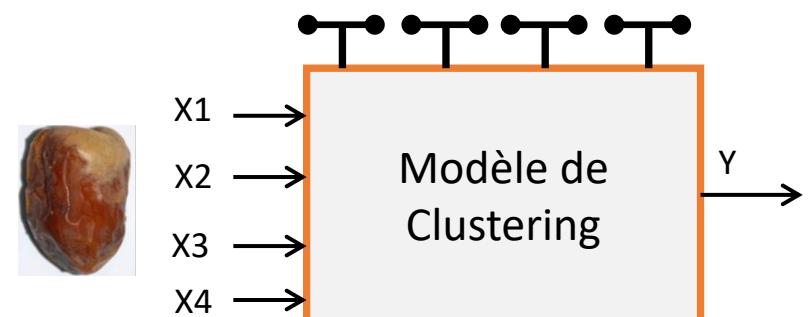
- Consiste à utiliser des données non étiquetées et utiliser des algorithmes de clustering qui vont fouiller dans les données pour chercher à les segmenter en un ensemble de groupes homogènes en se basant sur des mesures de similarités.

- Exemple : KMeans

Apprentissage Non Supervisé

Clustering

Data Set (Données Non étiquetées)



Algorithmes : K-Means

Apprentissage Auto Supervisé

- Apprentissage Auto Supervisé :
 - Utilisé dans le domaine de NLP (Natural Language Processing),
 - Ce type d'apprentissage consiste à exploiter des données non structurées et non étiquetées comme le texte d'un livre.
 - Pendant l'entraînement d'un algorithme, on cache des parties du texte et puis on entraîner le modèle à prédire les parties cachées du texte en utilisant des encodeurs et des décodeurs qui sont basés sur un mécanisme d'attention qui est la base des Transformers qui sont les plus utilisés dans le domaine de l'IA générative sur les LLMs (Large Language Models) : Exemple BERT, GPT

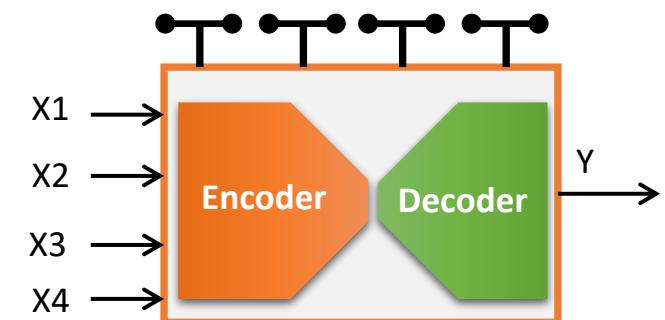
Apprentissage Auto Supervisé

Self-Supervised Learning

Encodeur - Décodeur

Data Set (Données Non étiquetées)

Cacher une partie des données et entraîner le modèle à prédire ces parties cachées en utilisant le principe des encodeurs et décodeurs
Pae exemple en NLP, On cache des mots du texte et on entraîne le modèle pour deviner ces mots cachés



Algorithmes : Transformers (BERT, GPT)

Apprentissage Par renforcement

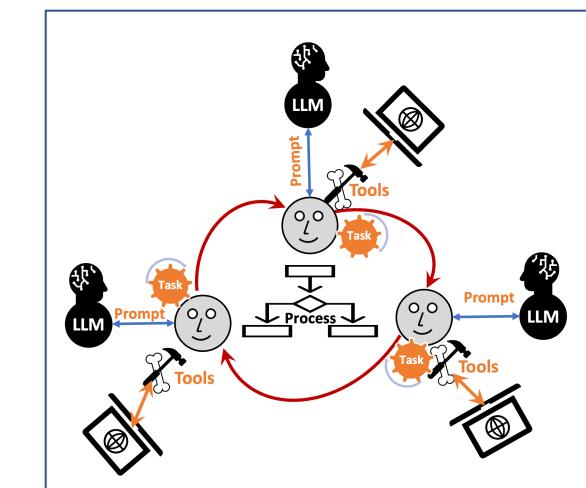
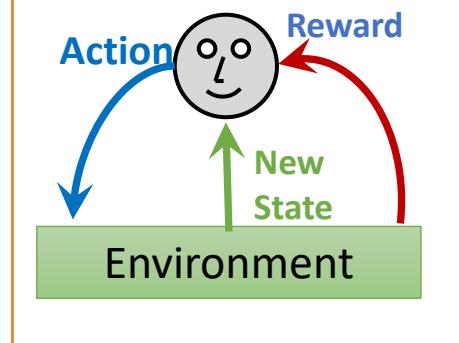
- **Apprentissage piloté par l'expérience (Apprentissage par renforcement) :**
 - Dans l'apprentissage par renforcement, on modélise l'environnement où un agent peut évoluer en agissant sur l'environnement en utilisant des actions.
 - Pendant la phase d'entraînement, l'agent qui possède un ensemble de Tools (Actions) explore l'environnement en agissant de manière itérative avec des actions disponibles.
 - Ensuite l'environnement lui procure des récompenses ou des observations. Ce qui permet de changer l'état de l'agent.
 - Ce processus répétitif d'exploration permet de construire une table de raisonnement.
 - Une fois que la phase d'entraînement et d'exploration est terminée, l'agent devient capable d'évoluer de manière autonome dans l'environnement en prenant les actions optimales selon son état actuel qui encapsule une abstraction et une représentation de l'environnement. Exemple : QLearning, PPO (Proximal Policy Optimisation)

Apprentissage Par Renforcement

Reinforcement Learning

Embodiment

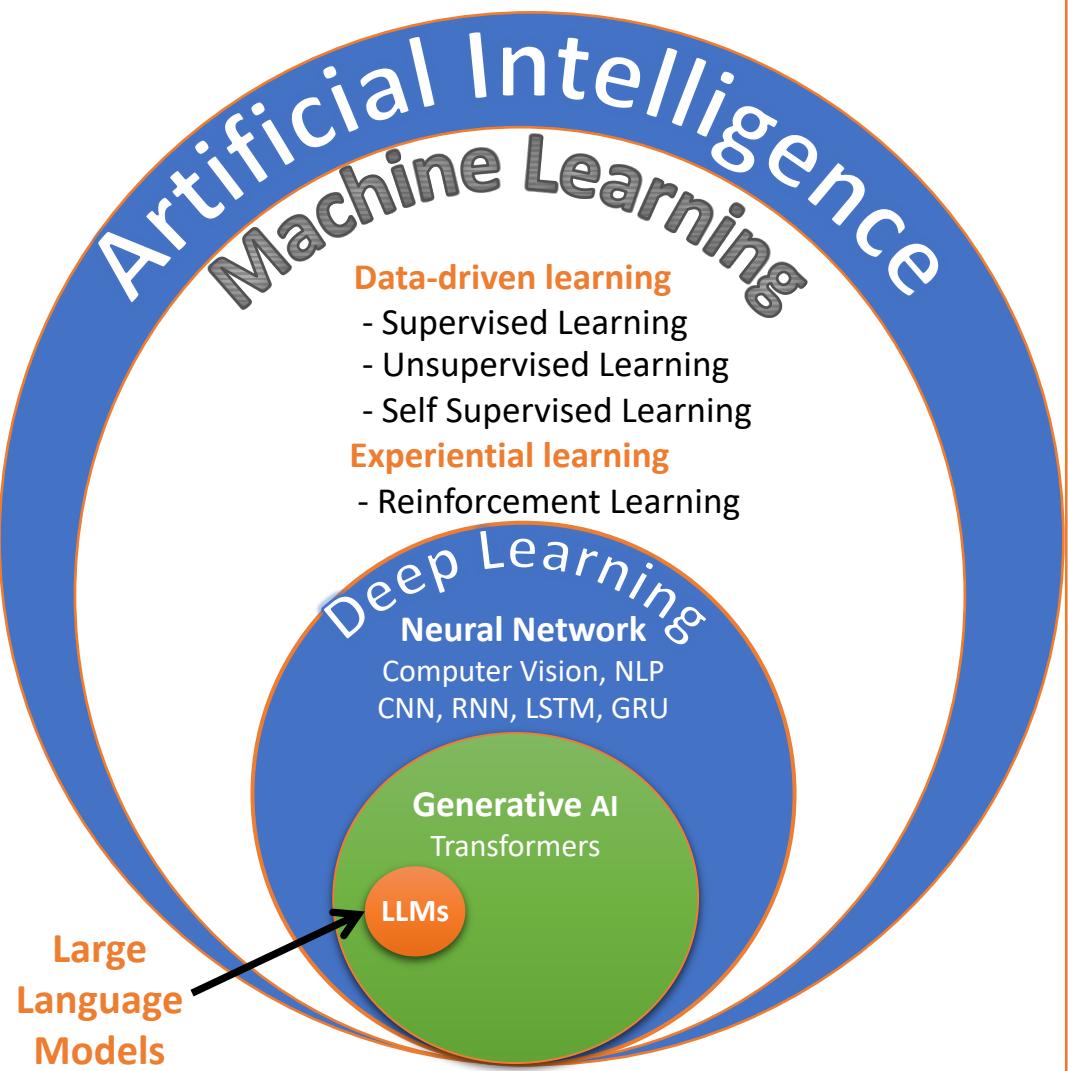
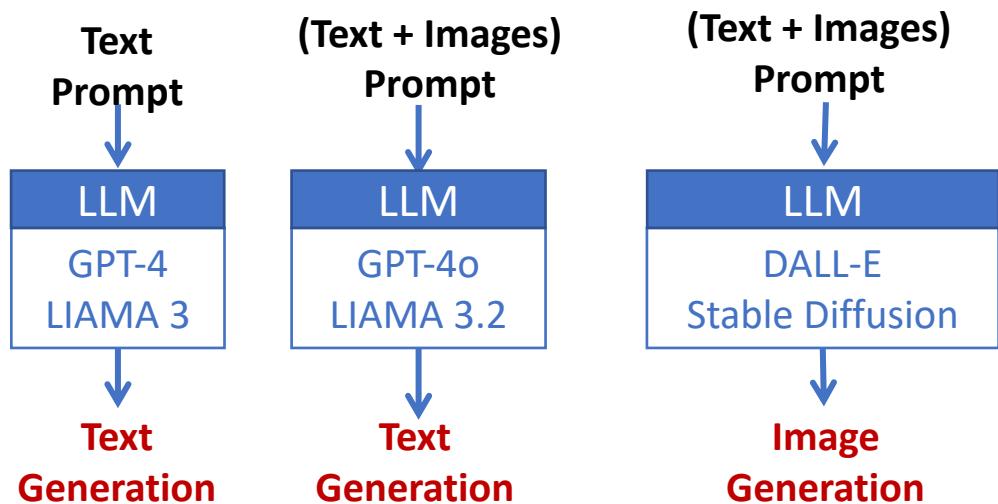
Reinforcement learning



Algorithmes : Q-Learning, PPO

IA Générative

- L'IA Générative est une catégorie de l'IA capable de générer des nouvelles données en conséquence d'un raisonnement conditionné par un prompt utilisateur en utilisant les LLMs (Large Language Models).
- Ces LLMs, basés sur les algorithmes dits « Transformers » sont entraînés en utilisant un large corpus de données.
- Ces LLMs sont des réseaux de neurones de très grandes tailles avec des Billions de paramètres et sont capables de générer des nouvelles données n'ayant jamais été vues de différentes modalités : Texte, Images, Son, Vidéo, Musique. Exemples de LLMs : Gpt GPT-4o, DeepSeek, Claude, Gemini, GPT-03, Dall-e, CLIP, Stable Diffusion



Apprentissage supervisé

Régression

Apprentissage Supervisé

- Consiste à utiliser un dataset étiqueté. C'est-à-dire des données dont on connaît les inputs et les outputs. Les outputs représentent des étiquettes ou des valeurs fournis par les experts du métier.
- Dans l'apprentissage supervisé on distingue deux types de problèmes :
 - **Régression** : Consiste à prédire, en sortie, une valeur continue comme le prix d'un appartement ou la durée de vie d'une pièce mécanique ou la durée de guérison d'un patient : Exemple Régression Linéaire
 - **Classification** : Consiste à prédire des classes d'appartenance parmi un ensemble de classe finies. Par exemple prédire si une transaction est frauduleuse ou encore prédire si un animal est un chien, un chat, un tigre ou un lapin.
 - Exemples d'algorithmes : Régression logistique, Support Vector Machine (SVM), Multi Layer Perceptron (MLP), KNN (K plus proches voisins), Decision Tree, Random Forest, etc.

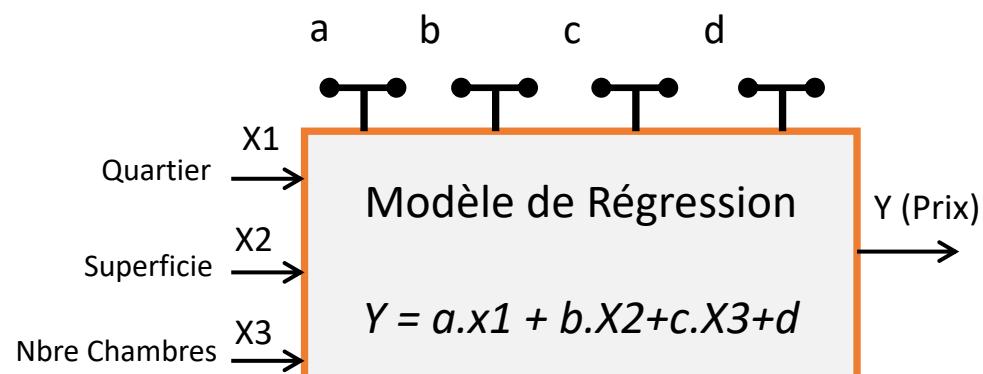
Apprentissage Supervisé

Régression

Exemple : Prédiction du prix d'un appartement

Data Set (Données étiquetées)

Input			Output
Quartier X1	Superficie X2	Nbre chambres	Prix (MDH) Y
Maarif	100	3	2
SBATA	120	3	1.1
Maarif	100	4	2.1
Bourgogne	80	3	1.44
Maarif	200	5	???????



Modèle à 4 Paramètres a, b, c et d

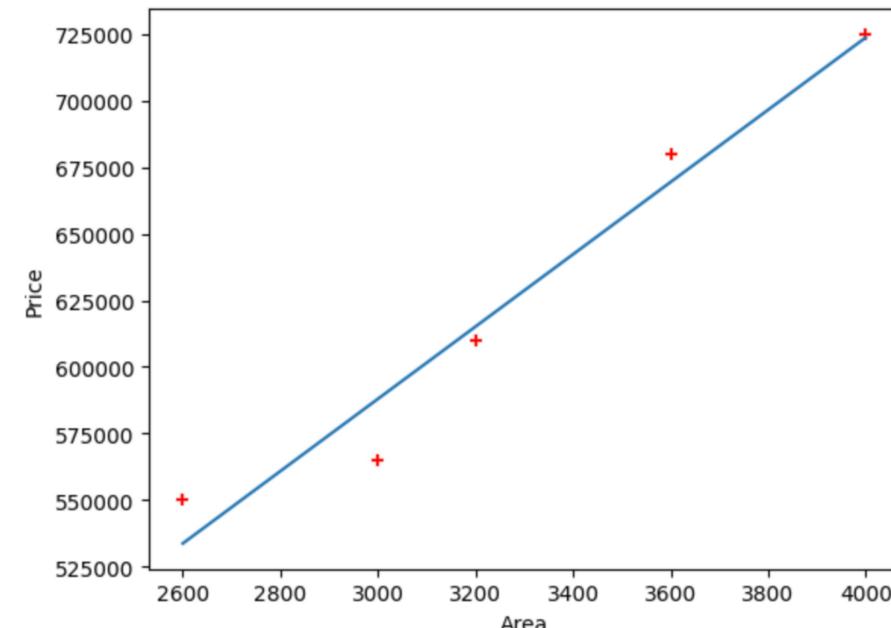
Algorithmes : Linéaire, Polynomiale, Ridge, Lasso, etc..

Apprentissage Supervisé : Régression

- La régression consiste à déterminer une relation entre la ou les variables indépendantes et la variable dépendante.
- La régression linéaire suppose que la relation entre les variables peut être modélisée par une équation linéaire ou une équation de droite.
- La variable utilisée pour la prédiction est appelée **variable indépendante** (Features), tandis que la variable prédite est appelée **variable dépendante** (Target or Outcome).
- Dans le cas d'une régression linéaire avec une seule variable explicative, la combinaison linéaire peut s'exprimer comme suit :
- **Y = intercept + (Coefficient * X)**
 - X est la variable indépendante (Input)
 - Y est la variable dépendante (Output)

Line of Best Fit

- L'objectif est de trouver la droite de régression qui s'ajuste le mieux aux données.
- Meilleur ajustement => que la ligne sera telle que la distance cumulative de tous les points par rapport à la ligne est minimisée.
- Mathématiquement, la ligne qui minimise la somme des carrés des erreurs résiduelles est appelée la droite de régression ou la ligne de meilleur ajustement.



Apprentissage Supervisé : Evaluation de la Régression Linéaire

• R-Squared :

- Mesure du % de variance dans la variable cible expliquée par le modèle
- Généralement la première métrique à examiner pour évaluer la performance d'un modèle linéaire
- Valeur entre 0 et 1. Plus elle est élevée, mieux c'est

• Mean Absolute Error:

- Métrique la plus simple pour vérifier la précision des prédictions
- Même unité que la variable dépendante
- Non sensible aux valeurs aberrantes (outliers), c'est-à-dire que les erreurs n'augmentent pas trop en présence de valeurs aberrantes
- Difficile à optimiser d'un point de vue mathématique (logique purement mathématique)
- Plus la valeur est faible, mieux c'est

• Root Mean Square Error:

- Une autre métrique pour mesurer la précision des prédictions
- Même unité que la variable dépendante
- Sensible aux valeurs aberrantes (outliers) : les erreurs sont amplifiées en raison de la fonction carrée
- Mais présente d'autres avantages mathématiques
- Plus la valeur est faible, mieux c'est

Line of Best Fit

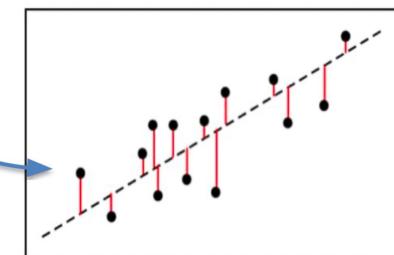
$$R^2 = 1 - \frac{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}{\frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2}$$

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

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

Obs	Height in Inches, X	Act Weight in Pounds, Y	Predicted Weight \hat{Y}	Residual / Error $e_i = Y_i - \hat{Y}_i$	Residual ² / Error ² $e_i^2 = (Y_i - \hat{Y}_i)^2$
1	63	127	120.1	6.900	47.61
2	64	121	126.3	-5.300	28.09
3	66	142	138.5	3.500	12.25
4	69	157	157.0	0.000	0
5	69	162	157.0	5.000	25
6	71	156	169.2	-13.200	174.24
7	71	169	169.2	-0.200	0.04
8	72	165	175.4	-10.400	108.16
9	73	181	181.5	-0.500	0.25
10	75	208	193.8	14.200	201.64
				0.000	597.28

Sum of Squared Residuals :



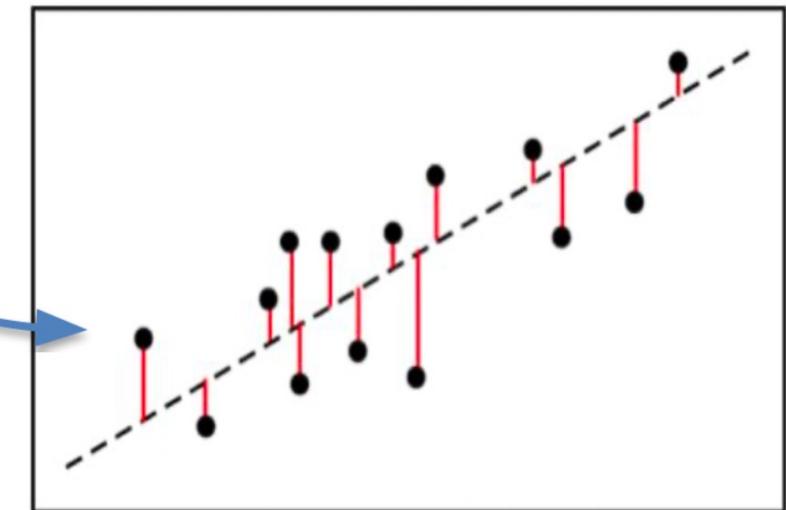
Apprentissage Supervisé : Evaluation de la Régression Linéaire

Obs	Height in Inches, X	Act Weight in Pounds, Y	Predicted Weight \hat{Y}	Residual/Error $e_i = Y_i - \hat{Y}_i$	Residual ² /Error ² $e_i^2 = (\hat{Y}_i - Y_i)^2$
1	63	127	120.1	6.900	47.61
2	64	121	126.3	-5.300	28.09
3	66	142	138.5	3.500	12.25
4	69	157	157.0	0.000	0
5	69	162	157.0	5.000	25
6	71	156	169.2	-13.200	174.24
7	71	169	169.2	-0.200	0.04
8	72	165	175.4	-10.400	108.16
9	73	181	181.5	-0.500	0.25
10	75	208	193.8	14.200	201.64
				0.000	597.28

Sum of Squared Residuals :

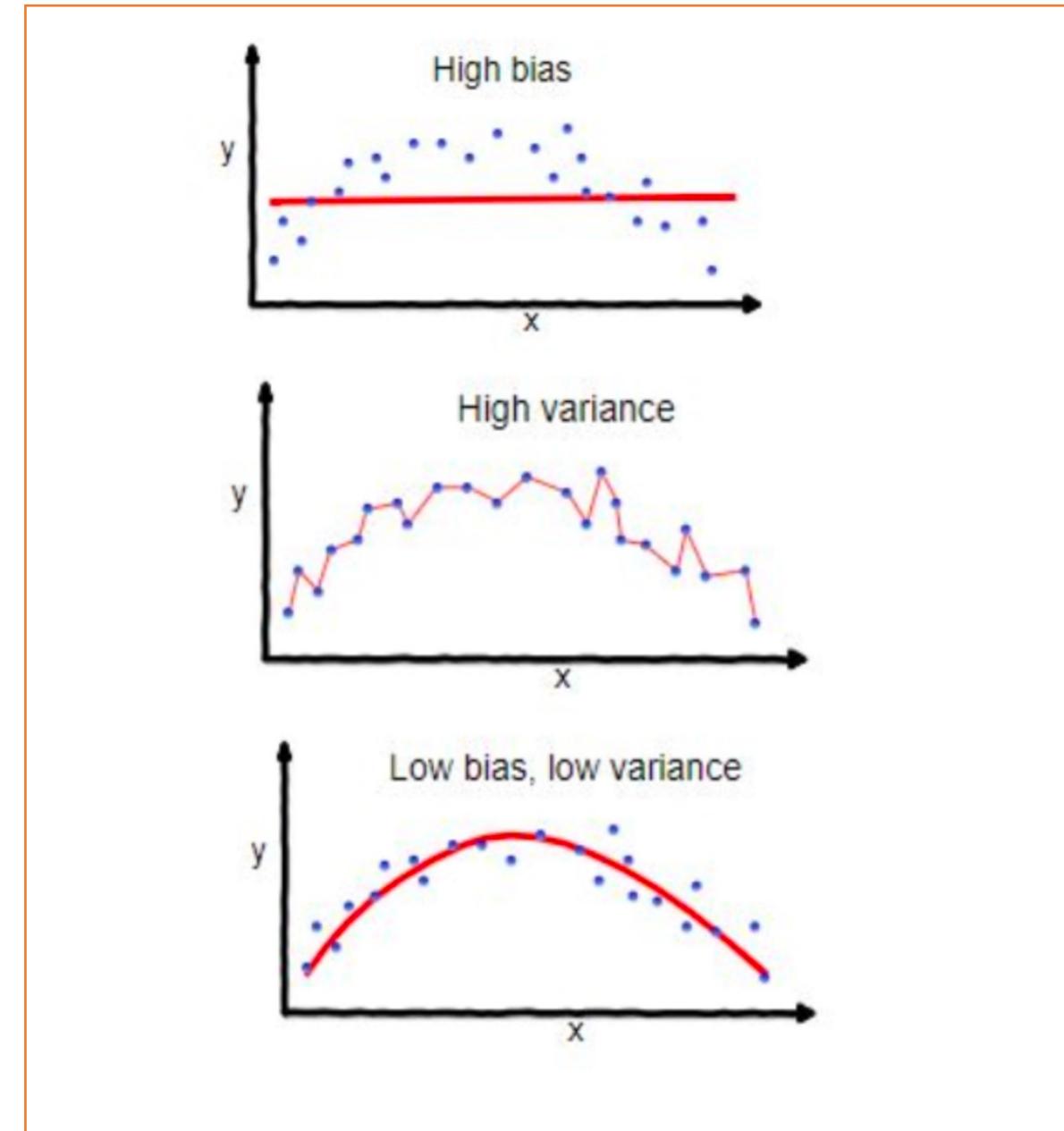
$$R^2 = 1 - \frac{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}{\frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2}$$

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



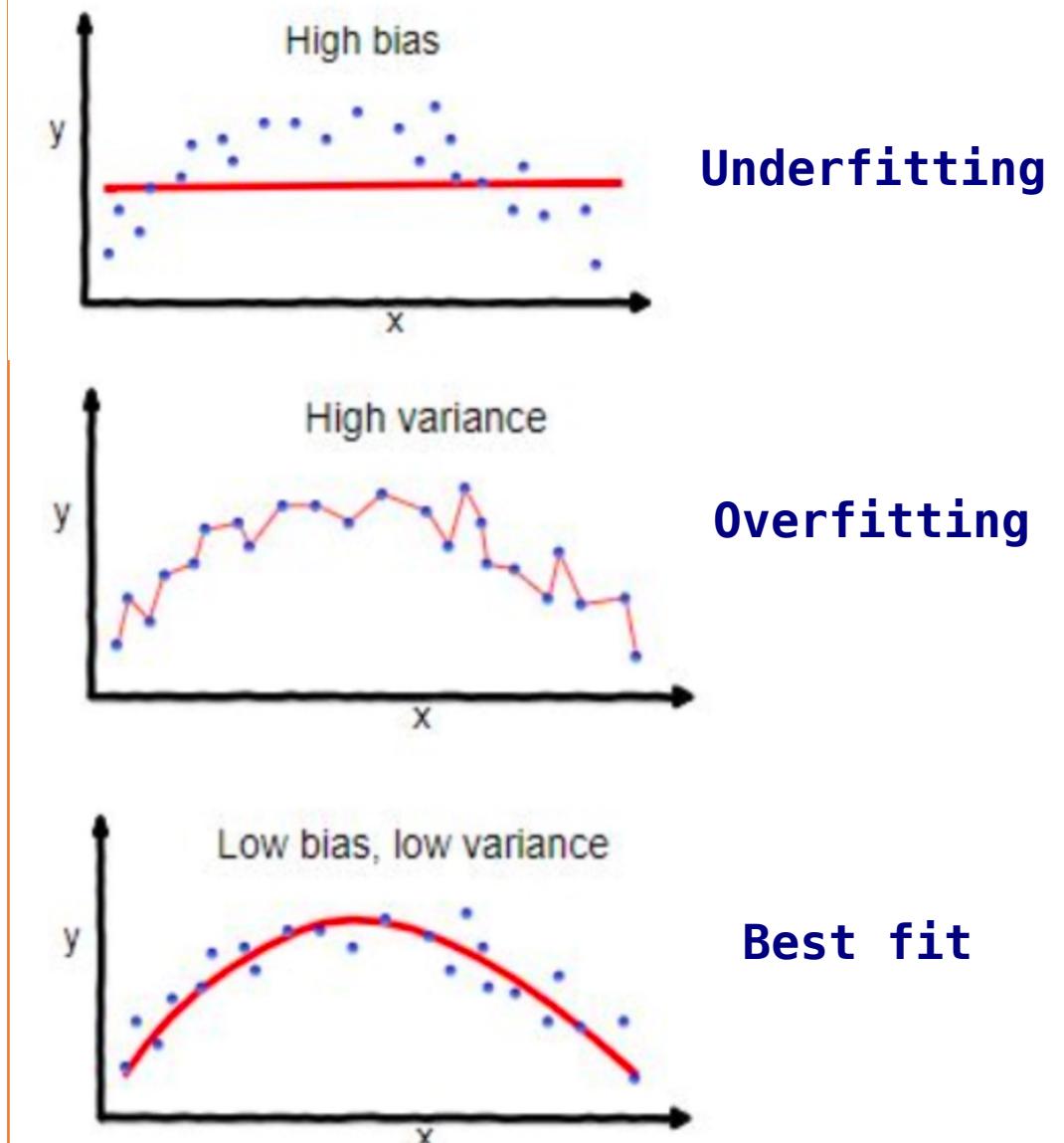
Bias-Variance: Underfitting and Overfitting

- **Le biais et la variance** sont deux sources d'erreur clés dans les modèles d'apprentissage automatique qui impactent directement leurs performances et leur capacité de généralisation.
- **Biais** : Le biais est la différence entre la prédiction de notre modèle et la valeur correcte que nous essayons de prédire. Un modèle avec un biais élevé accorde moins d'attention aux données d'entraînement et sur-généralise, ce qui entraîne une erreur élevée sur les données d'entraînement et de test.
- **Variance** : La variance est la valeur qui indique la dispersion des données. Un modèle avec une variance élevée accorde beaucoup d'attention aux données d'entraînement en captant aussi le bruit. Ces modèles ne se généralisent pas bien sur les données de test. Par conséquent, ces modèles sont performant très bien sur les données d'entraînement mais ont une erreur élevée sur les données de test.



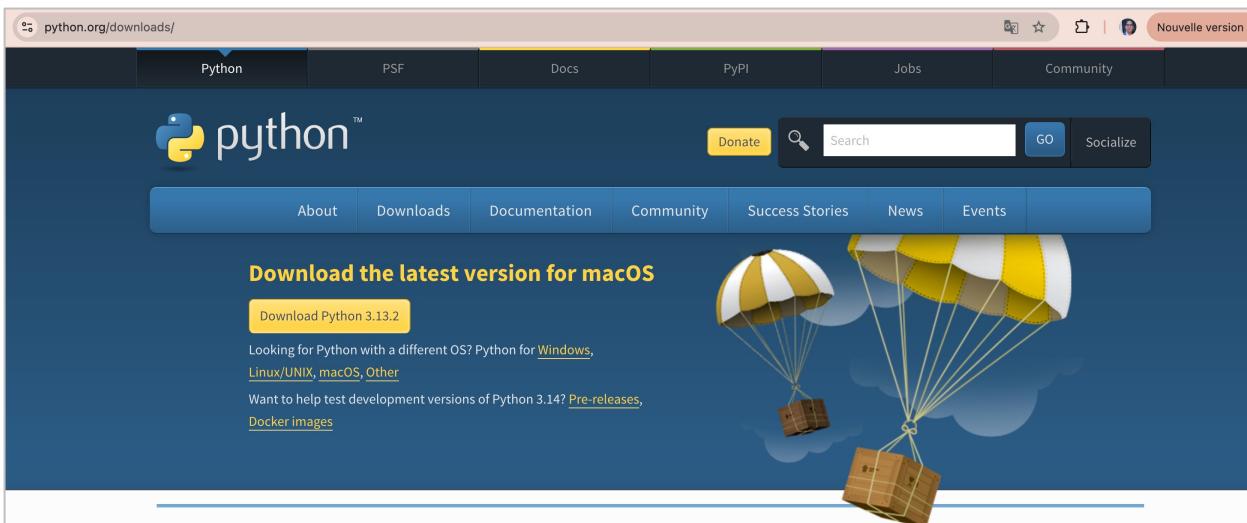
Bias-Variance: Underfitting and Overfitting

- **Underfitting (Sous-ajustement)** : En apprentissage supervisé, le sous-ajustement se produit lorsqu'un modèle n'est pas capable de capturer la tendance sous-jacente des données. Ces modèles ont généralement un biais élevé et une faible variance.
- **Overfitting (Sur-ajustement)** : se produit lorsque notre modèle capture le bruit (ou les fluctuations aléatoires) en plus de la tendance sous-jacente des données. Ces modèles ont généralement un faible biais et une variance élevée.



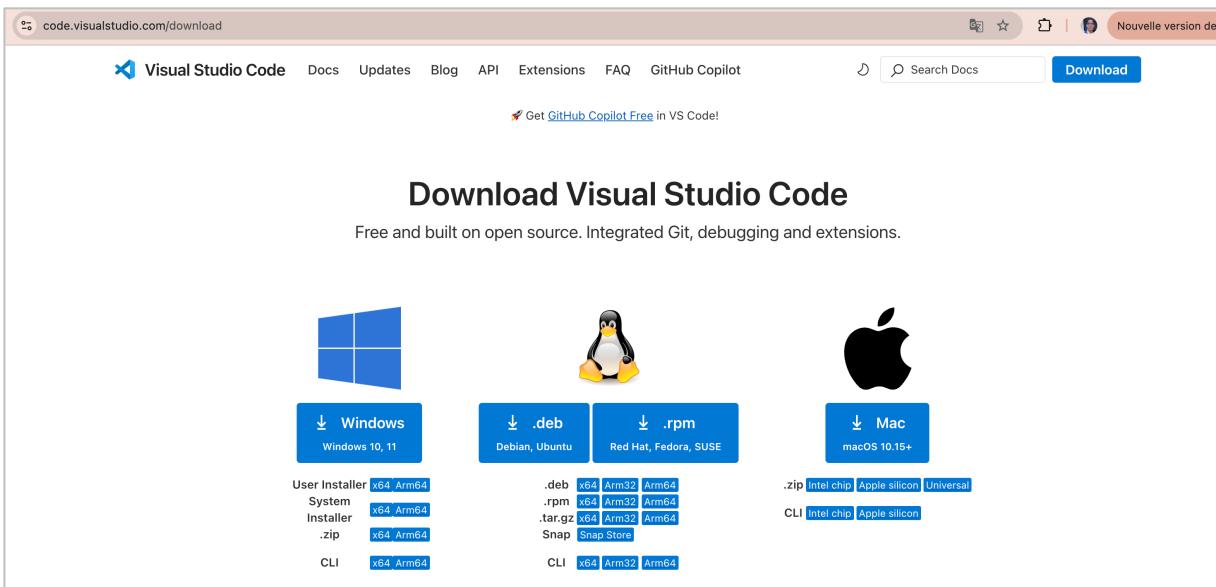
AI Tools

• INSTALLATION DE PYTHON

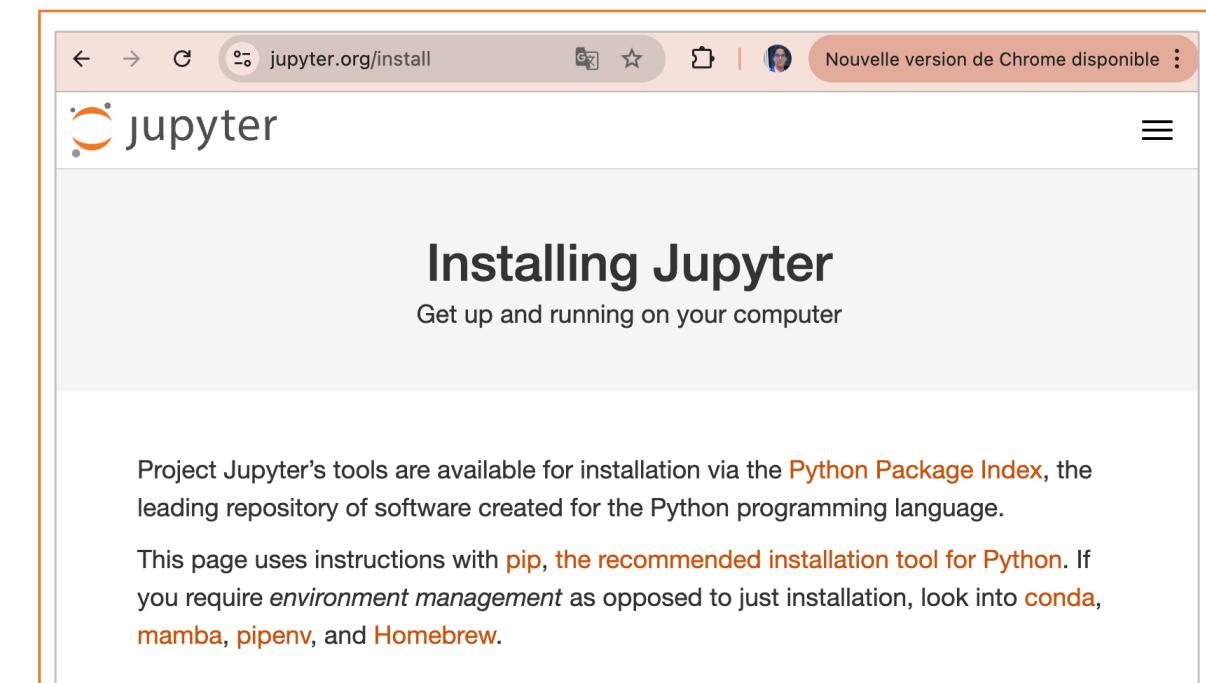


The screenshot shows the Python.org Downloads page. At the top, there are tabs for Python, PSF, Docs, PyPI, Jobs, and Community. Below the tabs, the Python logo is displayed. A search bar with a magnifying glass icon and a 'GO' button are present. A yellow 'Donate' button is also visible. The main content features a large illustration of two parachutes descending from the sky, each carrying a box. Text on the page includes 'Download the latest version for macOS' and 'Download Python 3.13.2'. It also mentions 'Looking for Python with a different OS? Python for Windows, Linux/UNIX, macOS, Other' and 'Want to help test development versions of Python 3.14? Pre-releases, Docker images'.

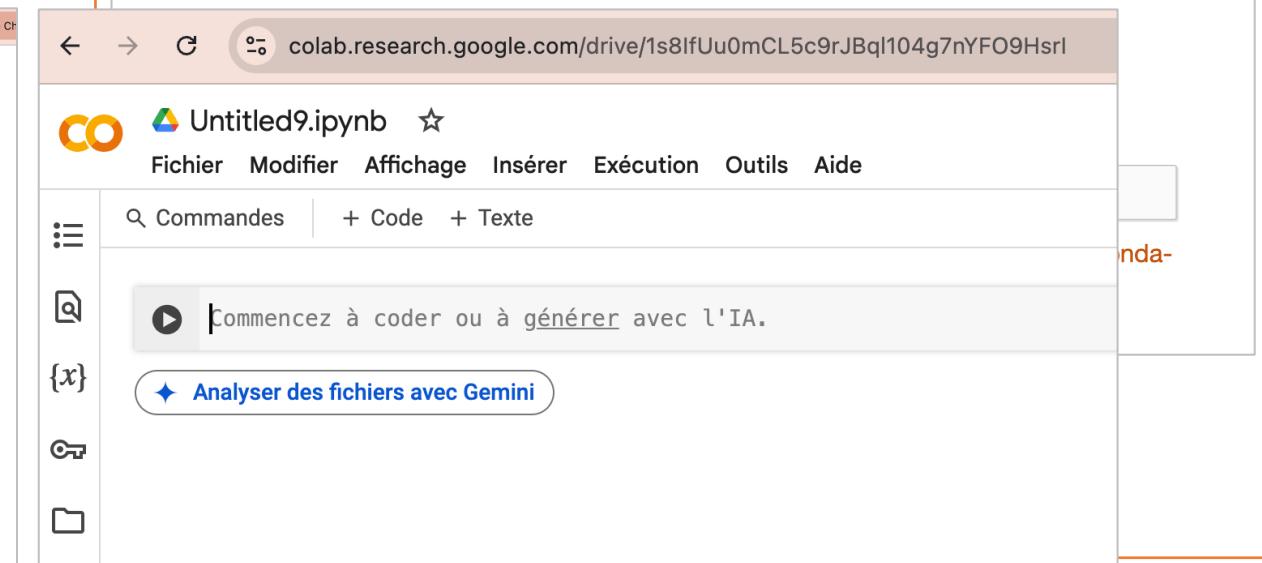
• INSTALLATION DE VISUAL STUDIO CODE (VS CODE)



The screenshot shows the Visual Studio Code download page. The top navigation bar includes links for Visual Studio Code, Docs, Updates, Blog, API, Extensions, FAQ, and GitHub Copilot. A 'Download' button is prominently displayed. Below the navigation, there's a link to 'Get GitHub Copilot Free in VS Code!'. The main section is titled 'Download Visual Studio Code' with the subtext 'Free and built on open source. Integrated Git, debugging and extensions.' It features icons for Windows, Linux (Ubuntu), and macOS. Below these are detailed download links for each operating system, including User Installer, System Installer, .zip, and CLI options for various architectures like x64, Arm64, and Intel chip.



The screenshot shows the Jupyter.org/install page. The title 'jupyter.org/install' is at the top. The main heading is 'Installing Jupyter' with the subtitle 'Get up and running on your computer'. A paragraph explains that Project Jupyter's tools are available via the Python Package Index. Another paragraph discusses the use of pip for installation, mentioning conda, mamba, pipenv, and Homebrew as alternatives for environment management.



The screenshot shows a Google Colab notebook titled 'Untitled9.ipynb'. The interface includes a menu bar with Fichier, Modifier, Affichage, Insérer, Exécution, Outils, and Aide. A search bar at the top right contains 'Commandes', 'Code', and 'Texte'. The main workspace has a play button icon and the text 'Commencez à coder ou à générer avec l'IA.' A callout bubble says 'Analyser des fichiers avec Gemini'. On the left, there are icons for Commandes, Code, Texte, and a file folder.

Application 1 – Régression linéaire

➤ Data set: `prix_maison.csv`

```
area,price  
100,1218000  
120,1440000  
150,1800000  
260,2860000  
320,2880000
```

➤ pip install pandas

```
import pandas as pd
```

```
[181] df = pd.read_csv('prix_maisons.csv')  
  
[182] df  
  
...  
area price  
0 100 1218000  
1 120 1440000  
2 150 1800000  
3 260 2860000  
4 320 2880000  
  
df.shape  
  
... (5, 2)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 5 entries, 0 to 4  
Data columns (total 2 columns):  
 #   Column  Non-Null Count  Dtype    
---  --      --          --  
 0   area    5 non-null      int64  
 1   price   5 non-null      int64  
dtypes: int64(2)  
memory usage: 212.0 bytes
```

Resumé statistique

```
df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
area	5.0	190.0	95.393920	100.0	120.0	150.0	260.0	320.0
price	5.0	2039600.0	786015.775923	1218000.0	1440000.0	1800000.0	2860000.0	2880000.0

- Les deux variables area et price sont de type numérique
- Il y a 5 valeurs pour chaque variable. Ce qui montre qu'il n'y a pas de valeurs manquante

- Le dataset contient 5 lignes et 2 colonnes

Application 1 – Régression linéaire

Vérification de l'existence des valeurs manquantes :

```
df.isna().sum()
```

✓ 0.0s

```
area      0  
price     0  
dtype: int64
```

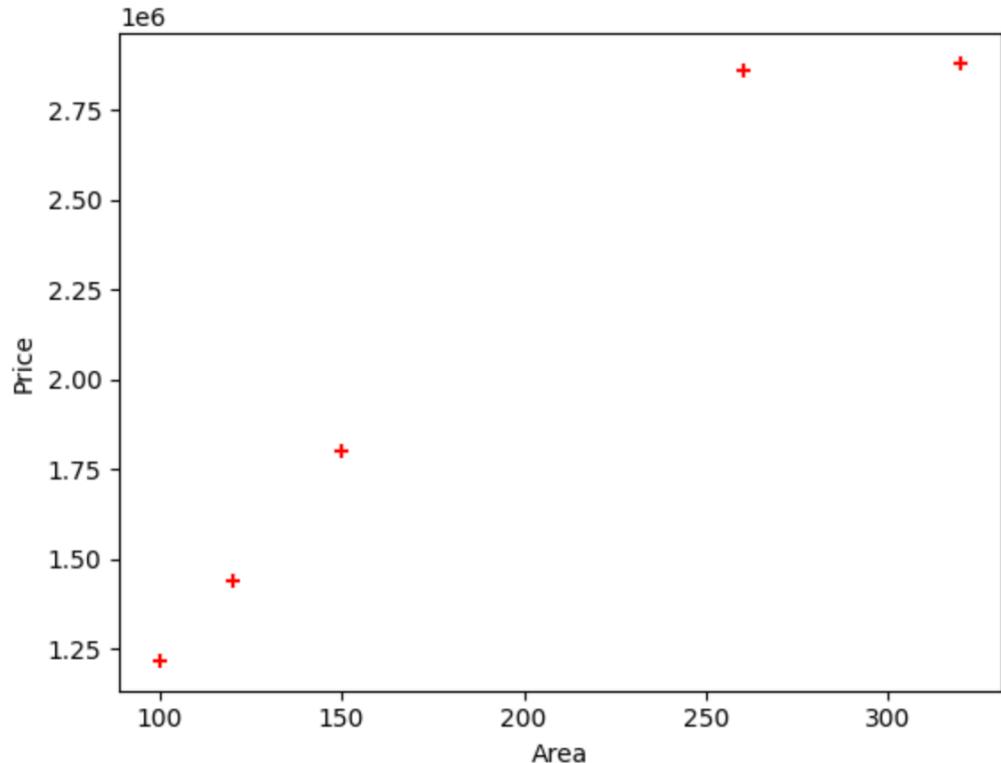
- Il n'y a aucune valeur manquante pour les deux variables

Nuage de points avec Scatter Plot

```
plt.xlabel("Area")  
plt.ylabel("Price")  
plt.scatter(df['area'], df['price'], color='red', marker="+")
```

✓ 0.1s

```
<matplotlib.collections.PathCollection at 0x16dc3cd70>
```



Application 1 – Régression linéaire

```
from sklearn import linear_model  
from sklearn.metrics import root_mean_squared_error, r2_score
```

Création du modèle de régression linéaire

```
lr = linear_model.LinearRegression()
```

Prédiction :

```
price = lr.predict([[2300]])  
print(price.tolist()[0])
```

✓ 0.0s
18976407.692307692

/Users/mohamedyoussfi/Documents/AI/Supervised Learning/Regression/.venv/lib/python3.13/s
warnings.warn(

+ Code + Marquege

```
predicted = lr.predict(X)
```

```
print("Predicted => ",predicted)  
print ("Target => ", y.tolist())
```

Predicted => [1317176.92307692 1477715.38461538 1718523.07692308 2601484.61538462
3083100.]
Target => [1218000, 1440000, 1800000, 2860000, 2880000]

Entraîner le modèle :

```
y = df['price']  
X = df.drop(columns=['price'])
```

```
9]  
lr.fit(X,y)
```

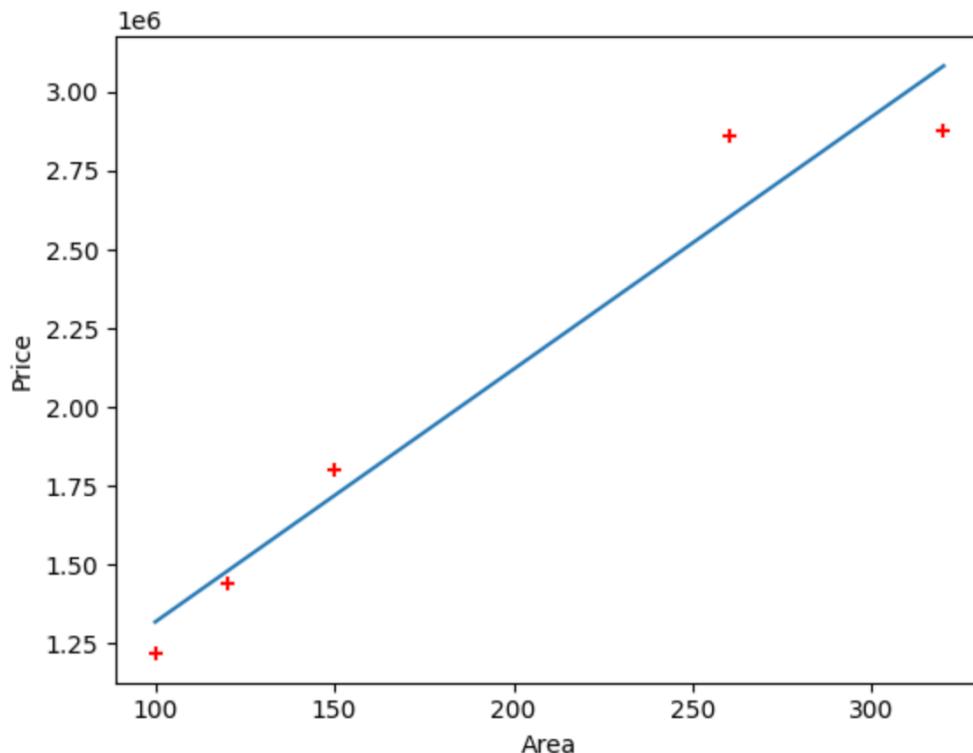
0]
▼ LinearRegression ⓘ ?
LinearRegression()

Application 1 – Régression linéaire

Visualiser le modèle avec les nuages de points

```
plt.xlabel("Area")
plt.ylabel("Price")
plt.scatter(df['area'], df['price'], color='red', marker="+")
plt.plot(df['area'], lr.predict(df[['area']]))

[<matplotlib.lines.Line2D at 0x29db72710>]
```



Evaluation du modèle

```
r2_score(y, predicted)
```

```
[0] 0.9490237236328821
```

```
root_mean_squared_error(y, predicted)
```

```
[1] 158730.47305707596
```

Expression mathématique du modèle

```
intercept= lr.intercept_
coef = lr.coef_
```

```
price1 = intercept + coef * 1000
price2 = lr.predict([[1000]])
print(price1)
print(price2)
```

```
[8541407.69230769]
[8541407.69230769]
```

Application 2 – Régression linéaire

➤ Data set: prix_maison2.csv

```
area,rooms,age,city,price  
100,3,4,Casablanca,1500000  
120,4,10,Casablanca,1680000  
260,,20,Casablanca,3380000  
190,5,1,Casablanca,3040000  
260,5,4,Casablanca,3900000  
60,2,3,Casablanca,1080000  
100,3,4,Marrakech,1400000  
120,4,10,Marrakech,1580000  
260,5,20,Marrakech,3280000  
190,5,1,Marrakech,2900000  
260,,4,Marrakech,3000000  
60,2,3,Marrakech,860000  
100,3,4,Tanger,1480000  
120,4,10,Tanger,1600000  
260,5,20,Tanger,3300000  
190,5,1,Tanger,3000000  
260,,4,Tanger,3000000  
60,2,3,Tanger,1000000
```

```
df2 = pd.read_csv('prix_maisons2.csv')
```

df2

	area	rooms	age	city	price
0	100	3.0	4	Casablanca	1500000
1	120	4.0	10	Casablanca	1680000
2	260	NaN	20	Casablanca	3380000
3	190	5.0	1	Casablanca	3040000
4	260	5.0	4	Casablanca	3900000
5	60	2.0	3	Casablanca	1080000
6	100	3.0	4	Marrakech	1400000
7	120	4.0	10	Marrakech	1580000
8	260	5.0	20	Marrakech	3280000
9	190	5.0	1	Marrakech	2900000
10	260	NaN	4	Marrakech	3000000
11	60	2.0	3	Marrakech	860000
12	100	3.0	4	Tanger	1480000
13	120	4.0	10	Tanger	1600000
14	260	5.0	20	Tanger	3300000
15	190	5.0	1	Tanger	3000000
16	260	NaN	4	Tanger	3000000
17	60	2.0	3	Tanger	1000000

df2.shape

(18, 5)

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 18 entries, 0 to 17  
Data columns (total 5 columns):  
 #   Column   Non-Null Count  Dtype    
---  --      --          --       --  
 0   area     18 non-null    int64    
 1   rooms    15 non-null    float64  
 2   age      18 non-null    int64    
 3   city     18 non-null    object   
 4   price    18 non-null    int64    
dtypes: float64(1), int64(3), object(1)  
memory usage: 852.0+ bytes
```

- 18 Lignes et 5 Colonnes

- Toutes les variables sont numériques sauf city

```
df2.describe()
```

	area	rooms	age	price
count	18.000000	15.000000	18.000000	1.800000e+01
mean	165.000000	3.800000	7.000000	2.276667e+06
std	79.64997	1.207122	6.61549	9.947450e+05
min	60.000000	2.000000	1.000000	8.600000e+05
25%	100.000000	3.000000	3.000000	1.485000e+06
50%	155.000000	4.000000	4.000000	2.290000e+06
75%	260.000000	5.000000	10.000000	3.030000e+06
max	260.000000	5.000000	20.000000	3.900000e+06

Application 2 – Régression linéaire

```
df2.isna().sum()
```

✓ 0.0s

```
area      0  
rooms     3  
age       0  
city      0  
price     0  
dtype: int64
```

- Il y a des valeurs manquantes pour la variable bedrooms
- Nous allons remplacer la valeur manquante par la médiane

```
median = df2['rooms'].median()  
print(median)
```

✓ 0.0s

4.0

```
df2['rooms']=df2['rooms'].fillna(median)
```

df2

	area	rooms	age	city	price
0	100	3.0	4	Casablanca	1500000
1	120	4.0	10	Casablanca	1680000
2	260	4.0	20	Casablanca	3380000
3	190	5.0	1	Casablanca	3040000
4	260	5.0	4	Casablanca	3900000
5	60	2.0	3	Casablanca	1080000
6	100	3.0	4	Marrakech	1400000
7	120	4.0	10	Marrakech	1580000
8	260	5.0	20	Marrakech	3280000
9	190	5.0	1	Marrakech	2900000
10	260	4.0	4	Marrakech	3000000
11	60	2.0	3	Marrakech	860000
12	100	3.0	4	Tanger	1480000
13	120	4.0	10	Tanger	1600000
14	260	5.0	20	Tanger	3300000
15	190	5.0	1	Tanger	3000000
16	260	4.0	4	Tanger	3000000
17	60	2.0	3	Tanger	1000000

```
df2['city'].value_counts()
```

city

```
Casablanca    6  
Marrakech    6  
Tanger        6  
Name: count, dtype: int64
```

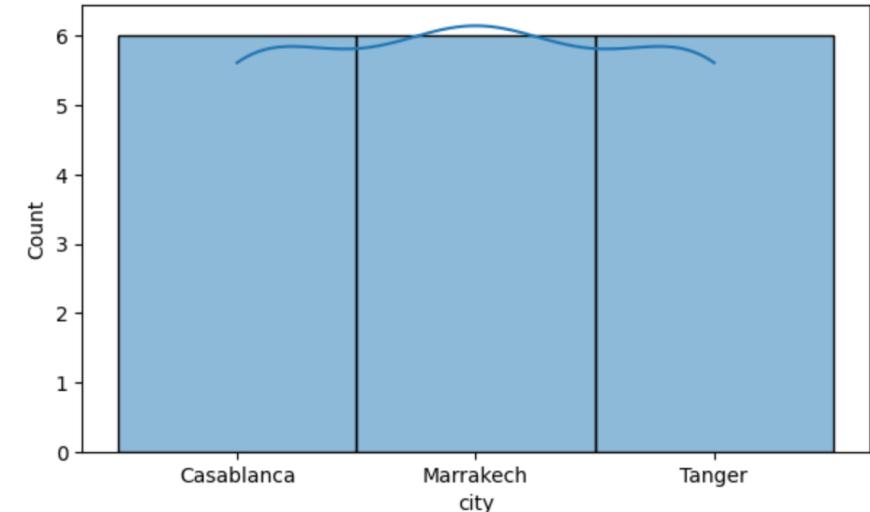
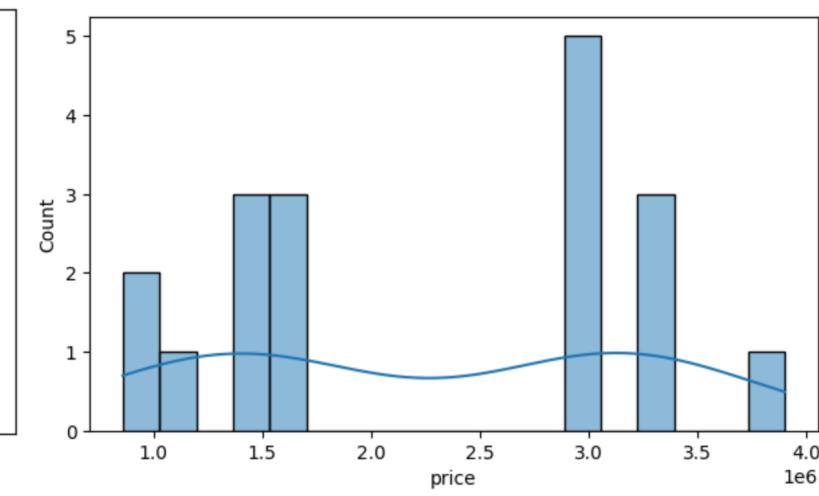
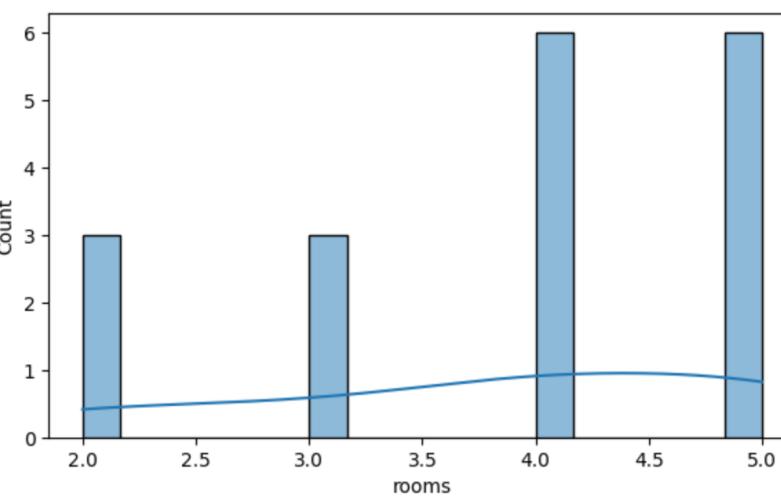
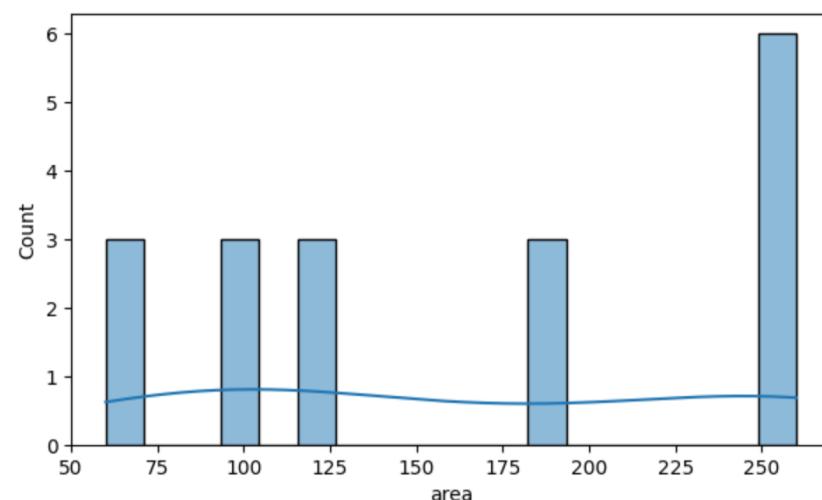
Application 2 – Régression linéaire

```
# For Data Visualisation  
import seaborn as sns
```

Univariate Analysis

Check the distribution of the variables

```
[8] for col in df2.columns:  
    plt.figure(figsize = (7, 4))  
    sns.histplot(data = df2, x = col, kde = True, bins=18)  
    plt.show()
```



Application 2 – Régression linéaire

Categorical Variables Encoding

```
label_encoder = LabelEncoder()  
dummy_df = pd.get_dummies(df2['city'], drop_first=False, prefix='city')
```

```
#df2['city_label'] = label_encoder.fit_transform(df2['city'])  
df2 = pd.concat([df2, dummy_df], axis=1)
```

df2

	area	rooms	age	city	price	city_Casablanca	city_Marrakech	city_Tanger
0	100	3.0	4	Casablanca	1500000	True	False	False
1	120	4.0	10	Casablanca	1680000	True	False	False
2	260	4.0	20	Casablanca	3380000	True	False	False
3	190	5.0	1	Casablanca	3040000	True	False	False
4	260	5.0	4	Casablanca	3900000	True	False	False
5	60	2.0	3	Casablanca	1080000	True	False	False
6	100	3.0	4	Marrakech	1400000	False	True	False
7	120	4.0	10	Marrakech	1580000	False	True	False
8	260	5.0	20	Marrakech	3280000	False	True	False
9	190	5.0	1	Marrakech	2900000	False	True	False
10	260	4.0	4	Marrakech	3000000	False	True	False
11	60	2.0	3	Marrakech	860000	False	True	False
12	100	3.0	4	Tanger	1480000	False	False	True
13	120	4.0	10	Tanger	1600000	False	False	True
14	260	5.0	20	Tanger	3300000	False	False	True
15	190	5.0	1	Tanger	3000000	False	False	True
16	260	4.0	4	Tanger	3000000	False	False	True
17	60	2.0	3	Tanger	1000000	False	False	True

Application 2 – Régression linéaire

Bivariate Analysis

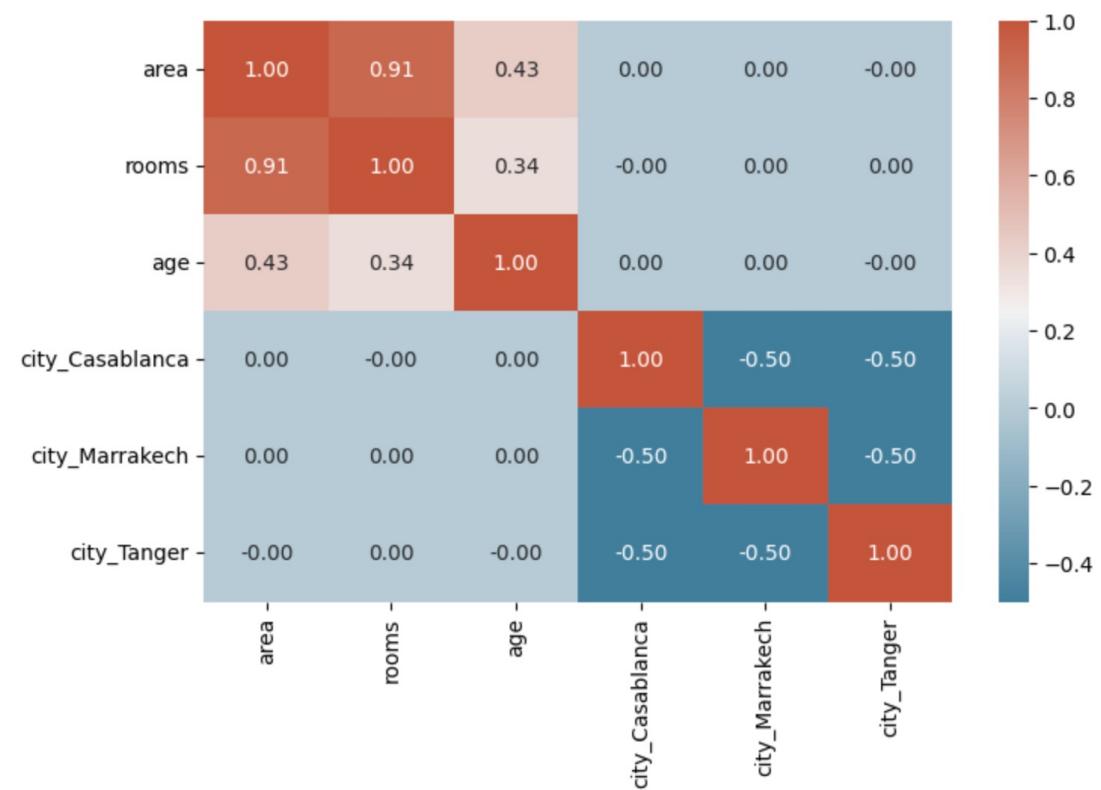
```
outcome = df2['price']
features = df2.drop(columns=['price', 'city'])
```

✓ 0.0s

Let's check the correlation using the heatmap

```
plt.figure(figsize = (8, 5))
cmap = sns.diverging_palette(230, 20, as_cmap = True)
sns.heatmap(features.corr(), annot = True, fmt = '.2f', cmap = cmap )
plt.show()
```

✓ 0.0s



- On note une forte corrélation entre area et bedrooms. Ce qui semble logique

Application 2 – Régression linéaire

Model Building - Approach

1. Data preparation
2. Partition the data into train and test set
3. Build model on the train data
4. Test the data on test set

Split the dataset

Let's split the data into the dependent and independent variables and further split it into train and test set in a ratio of 70:30 for train and test set.

```
# Splitting the data  
X_train, X_test, y_train, y_test = train_test_split(features, outcome, test_size = 0.3, random_state = 1)
```

31]

Python

Create the model

```
regressor = linear_model.LinearRegression()
```

32]

Python

Train the model

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

33]

Python

▼ LinearRegression ⓘ ⓘ
LinearRegression()

Application 2 – Régression linéaire

Prediction

```
preds_train = regressor.predict(X_train)
```

34]

Model Evaluation

```
r2_score(y_train, preds_train)
```

35]

```
0.9760647063979758
```

```
preds_test = regressor.predict(X_test)
```

36]

```
r2_score(y_test, preds_test)
```

37]

```
0.9324022925175757
```

- Le R2 Score pour les données de'entrainement et pour les données de Tests sont proches. Ce qui signifie qu'il n'y a pas d'overfitting

```
regressor.coef_
```

38]

```
array([ 8626.11742769, 310819.43745123, -19758.08760919, 87415.85653553,
       -59418.96096589, -27996.89556964])
```

```
regressor.predict([[2600,3,20,False, False, True]])
```

```
regressor.intercept_
```

40]

```
np.float64(-196079.7804040648)
```

```
array([22741125.19617736])
```