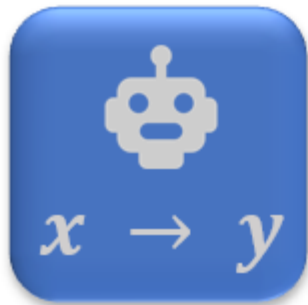


« Donner à une machine la **capacité d'apprendre** sans la programmer de façon explicite. » Arthur Samuel

→ Développer un modèle à partir de données



Apprentissage  
Supervisé

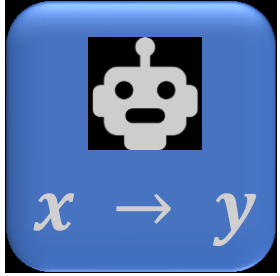


Apprentissage  
Non-supervisé



Apprentissage  
Par renforcement

# SUPERVISED LEARNING <https://machinelearningia.com/>



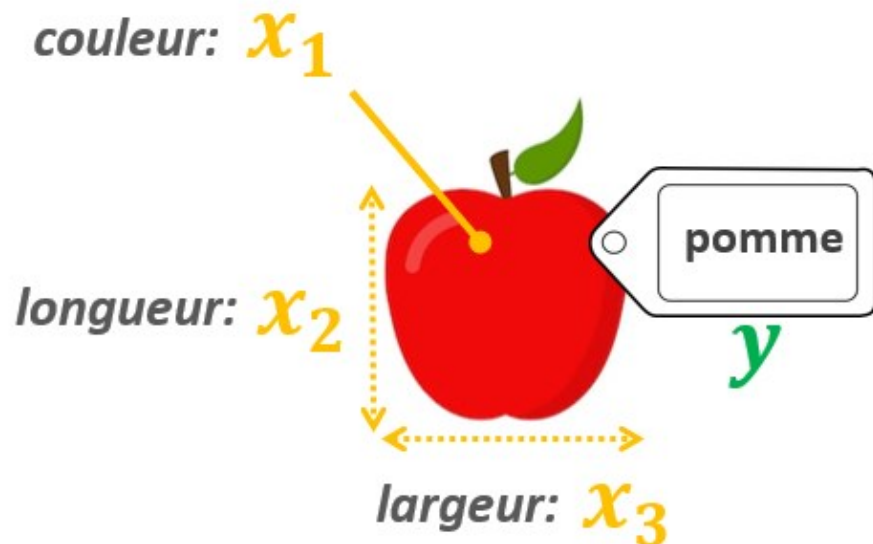
La machine reçoit des données  
**caractérisées** par des variables  $x$   
et **annotées** d'une variable  $y$

$x$

Features

$y$

Label / Target

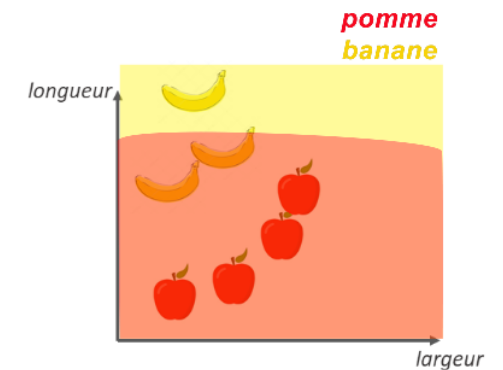
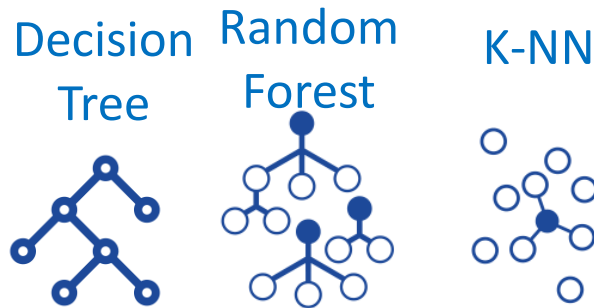
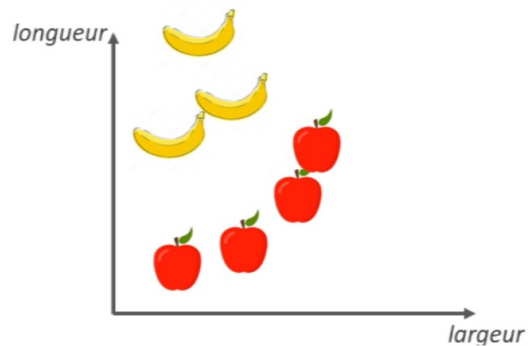


# SUPERVISED LEARNING <https://machinelearnia.com/>

1. En fournissant beaucoup de données à la machine, on constitue un **Dataset**.

2. On spécifie quel **type de modèle** la machine doit apprendre, en précisant les **hyperparamètres** du modèle.

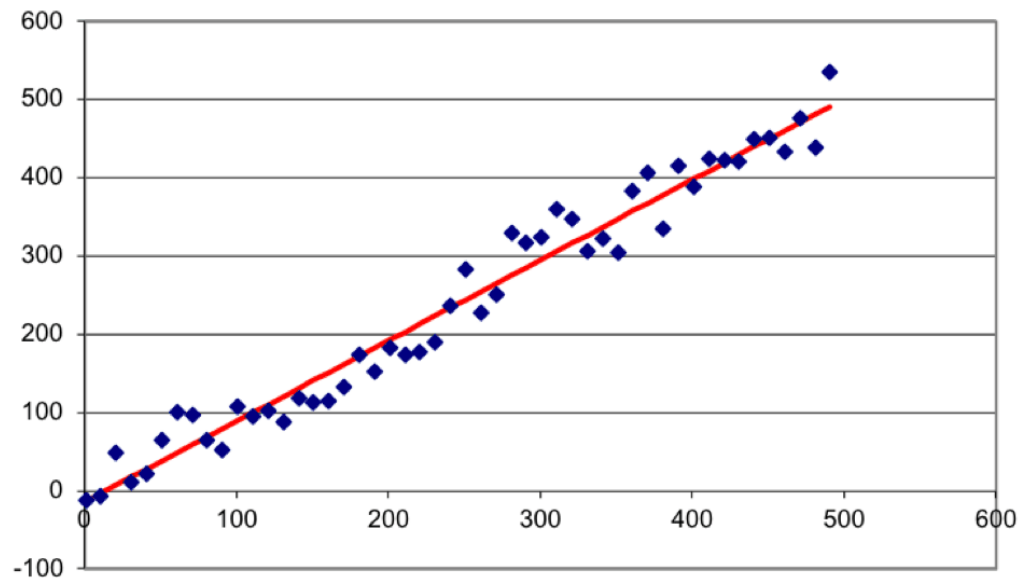
3. Grâce a un **algorithme d'optimisation**, la machine trouve les paramètres qui fournissent les **meilleures performances sur le Dataset**.



# SUPERVISED LEARNING <https://machinelearning.com/>

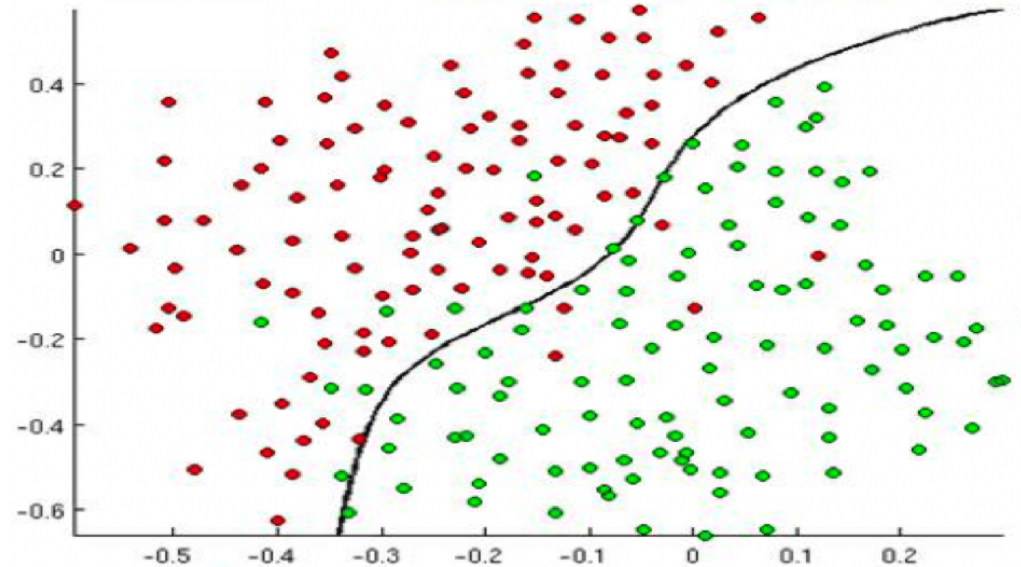
Permet de résoudre 2 types de problèmes:

## Régression



*Prix d'un appartement*  
(*y* est continue)

## Classification



*Email Spam / non Spam*  
(*y* est discrète)

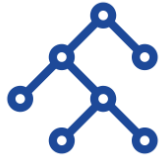
# INTERFACE DE SKLEARN

<https://machinelearningia.com/>

Linear  
Regression



Decision  
Tree



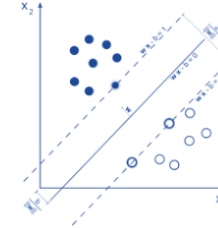
Random  
Forest



K-NN



SVM



Neural  
Network



*Différents mécanismes*

*Mais la même interface*



*fit*



*score*



*predict*

# INTERFACE DE SKLEARN

<https://machinelearningia.com/>

1. Sélectionner un **estimateur** et préciser ses **hyperparamètres** :

*model* = *LinearRegression*(.....)  
objet                      Constructeur                      Hyperparamètres

2. **Entraîner** le modèle sur les données **X, y**  
(divisées en 2 tableaux **Numpy**)

*model.fit(X, y)*

3. **Évaluer** le modèle

*model.score(X, y)*

4. **Utiliser** le modèle

*model.predict(X)*

1. Sélectionner un **estimateur** et préciser ses **hyperparamètres** :

*model = LinearRegression(.....)*

*objet*      *Constructeur*      *Hyperparamètres*

## Exemples:

```
model = SGDRegressor(eta0 = 0.3) # learning_rate = 0.3
```

```
model = RandomForestClassifier(n_estimators=100)
```

1. Sélectionner un **estimateur** et préciser ses **hyperparamètres** :

*model = LinearRegression(.....)*

2. **Entraîner** le modèle sur les données **X, y**  
(divisées en 2 tableaux **Numpy**)

*model.fit(X, y)*



# INTERFACE DE SKLEARN

<https://machinelearningia.com/>

1. Sélectionner un **estimateur** et préciser ses **hyperparamètres** :

```
model = LinearRegression(.....)
```

2. Entrainer le modèle sur les données **X, y**  
(divisées en 2 tableaux **Numpy**)

```
model.fit(X, y)
```

⇒ **X** et **y** doivent avoir **2 dimensions** !

*[n\_samples, n\_features]*

Feature Matrix (X)

n\_features →


← n\_samples

Target Vector (y)


← n\_samples

1. Sélectionner un **estimateur** et préciser ses **hyperparamètres** :

*model = LinearRegression(.....)*

2. **Entraîner** le modèle sur les données **X, y**  
(divisées en 2 tableaux **Numpy**)

*model.fit(X, y)*

3. **Évaluer** le modèle

*model.score(X, y)*

# INTERFACE DE SKLEARN

<https://machinelearningia.com/>

1. Sélectionner un estimateur et préciser ses hyperparamètres :

*model = **LinearRegression**(.....)*

2. Entraîner le modèle sur les données  $X, y$   
(divisées en 2 tableaux *Numpy*)

*model.**fit**( $X, y$ )*

3. Évaluer le modèle

*model.**score**( $X, y$ )*

4. Utiliser le modèle

*model.**predict**( $X$ )*

# INTERFACE DE SKLEARN

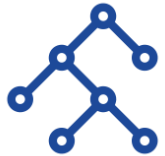
<https://machinelearnia.com/>

Linear  
Regression



```
model = LinearRegression()  
model.fit(X, y)  
model.score(X, y)  
model.predict(X)
```

Decision  
Tree



```
model = DecisionTreeClassifier()  
model.fit(X, y)  
model.score(X, y)  
model.predict(X)
```