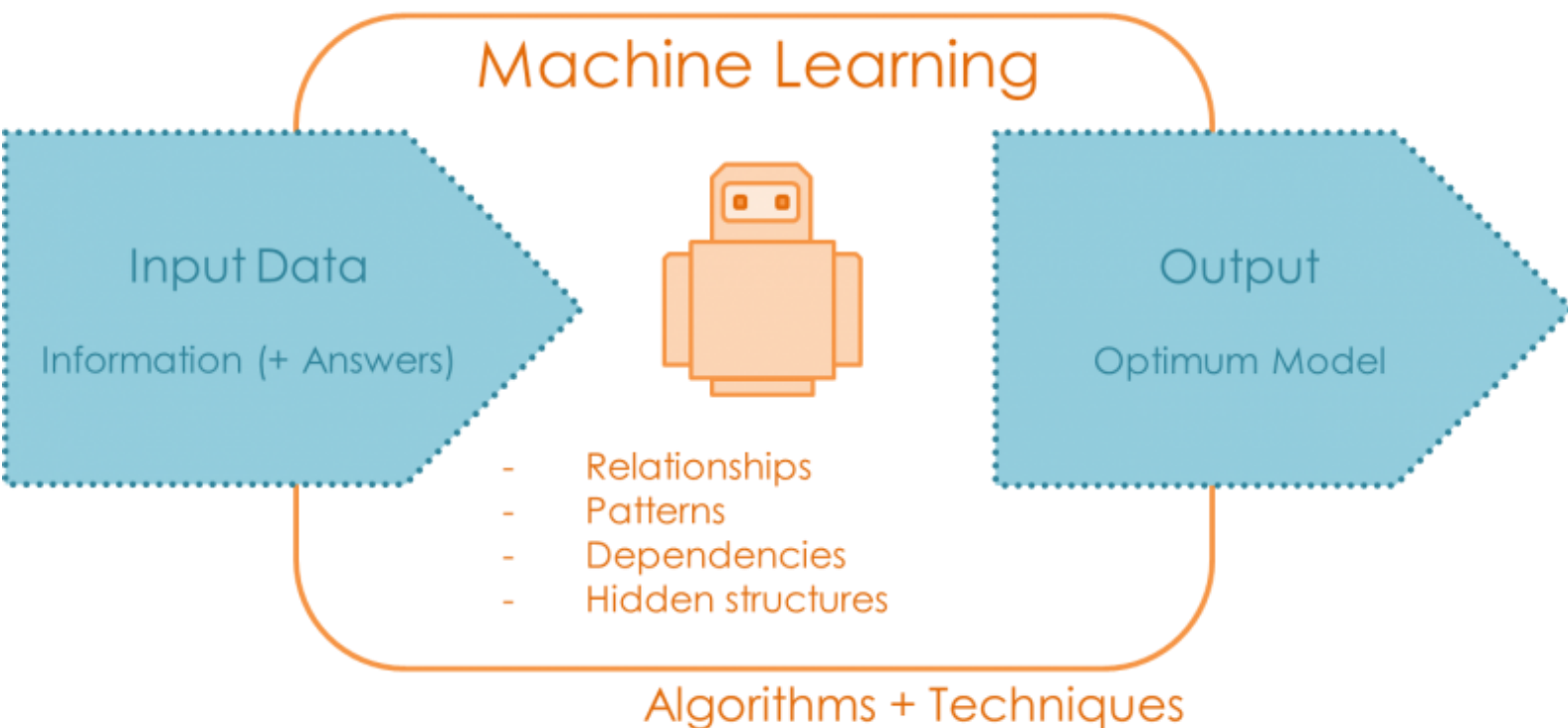


# MACHINE LEARNING

## Prévision de la demande (la quantité)



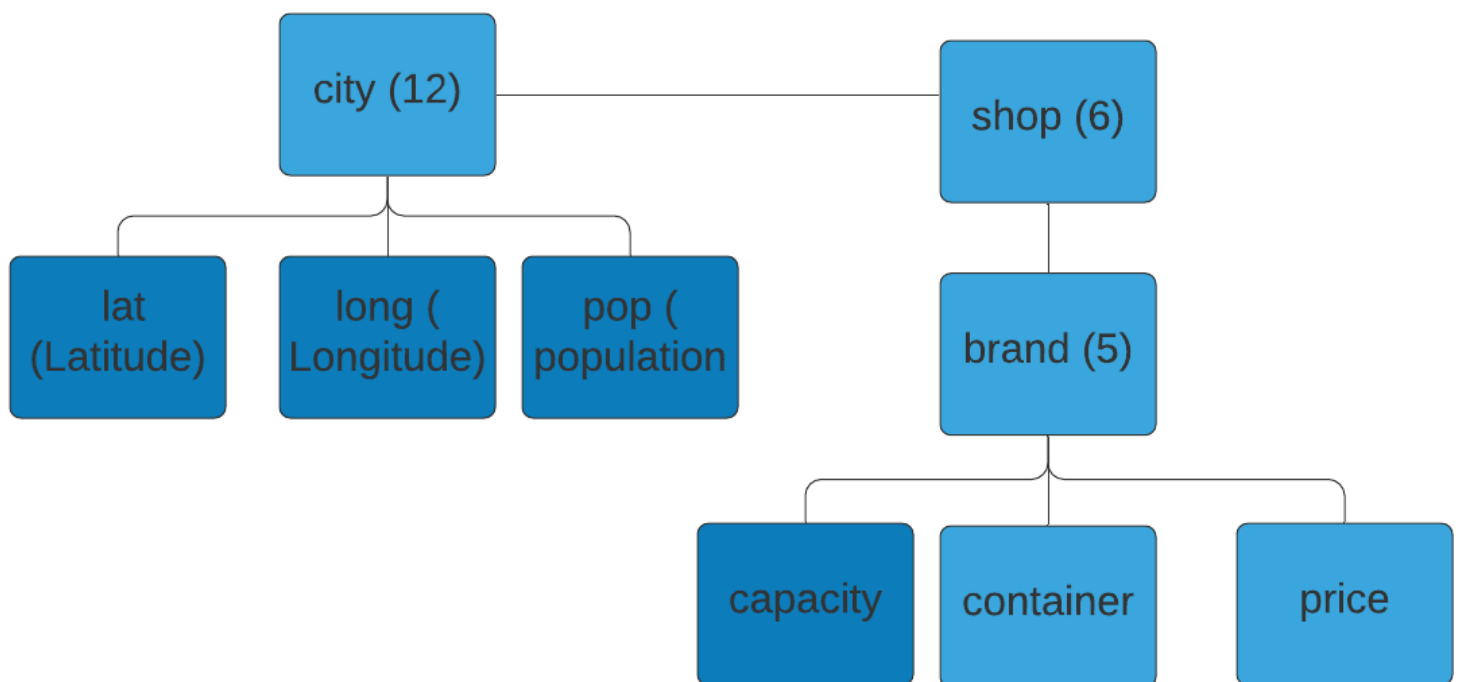
## Description du projet:

### Prévision de la demande (la quantité)

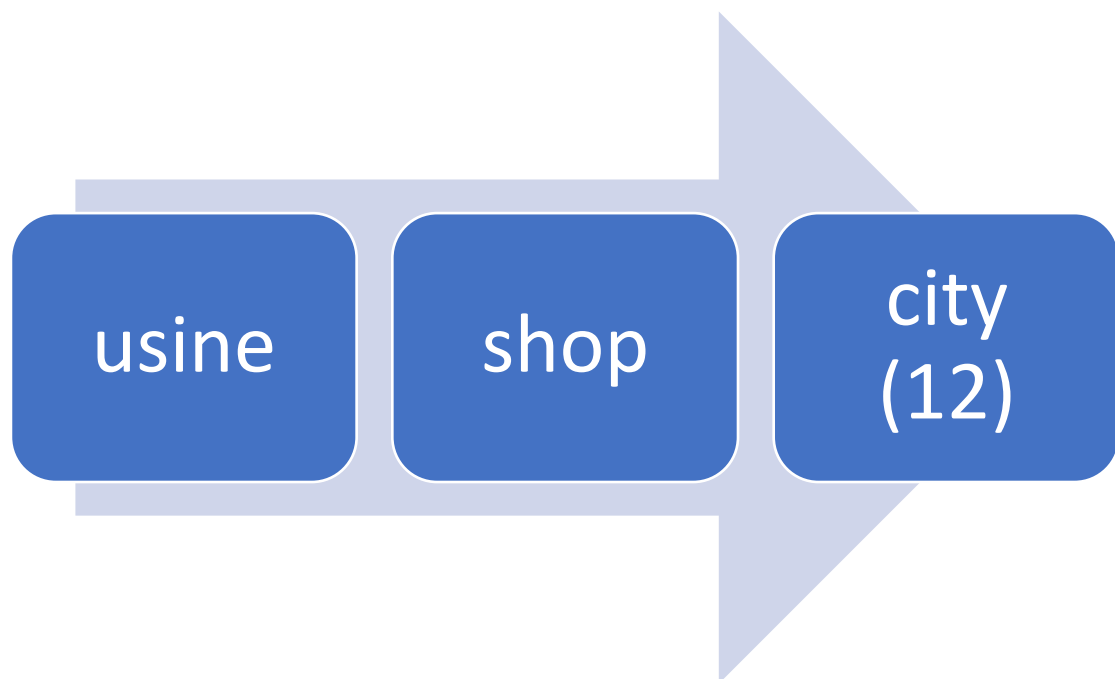
dans la ville grecque, nous devons prévoir la quantité exigeante de nos produits pour chaque ville.

Et 6 boutiques dans 12 villes et chaque boutique contient 5 marques, avec leurs capacités et leurs prix (boissons)

nous espérons connaître **la quantité** exacte de la demande pour chaque boutique pour résumer cela voir l'image ci-dessous.

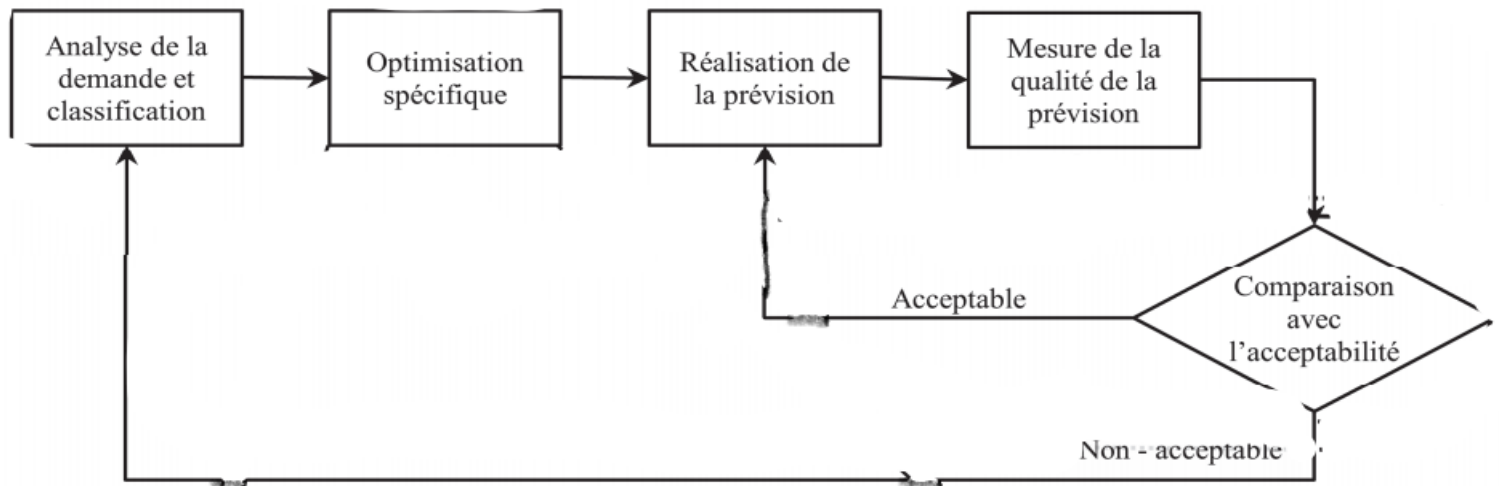


**Prévision de la demande (la quantité) nécessaire pour  
chaque marque(brand)**

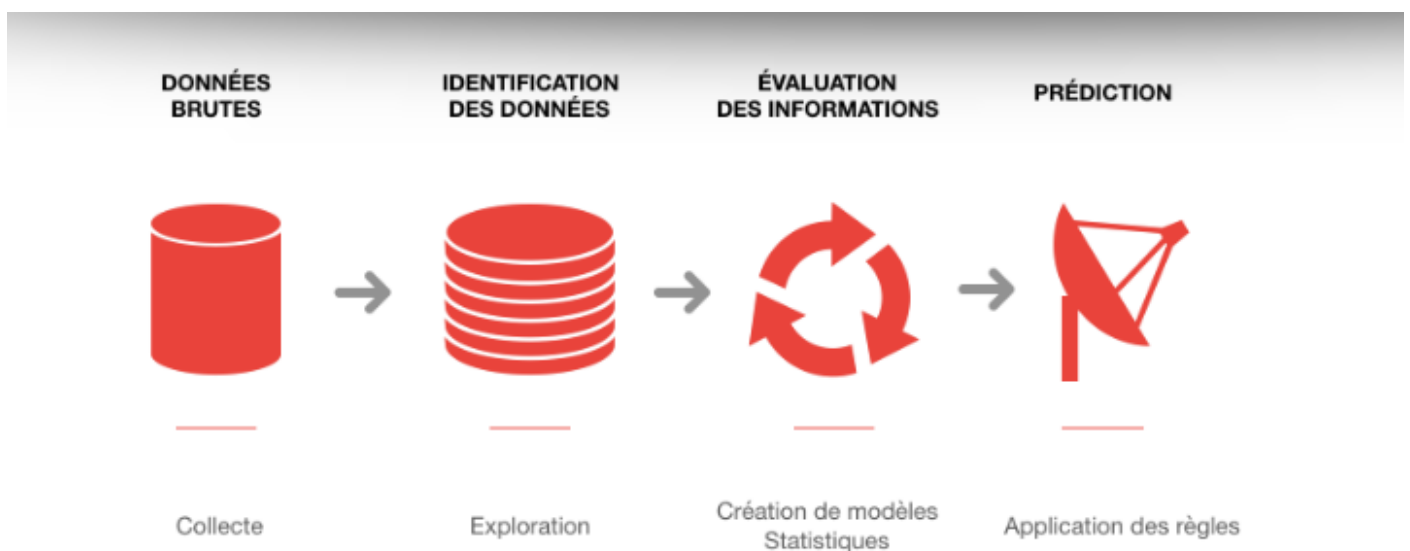


## Objectifs :

- Optimiser la gestion des stocks
- Réduire les erreurs de prévision
- Minimiser le stock inactif
- Améliorer la satisfaction client



**Après la définition du problème maintenant on a besoin de collecter les informations**



On a besoin de 12 informations (11 informations + 1réponse)

id  
date  
city(12)  
latitude  
longitude  
population  
shop(6)  
brand(5)  
container  
capacity  
price  
quantity



MACHINE  
LEARNING



PREDICT QUANTITY

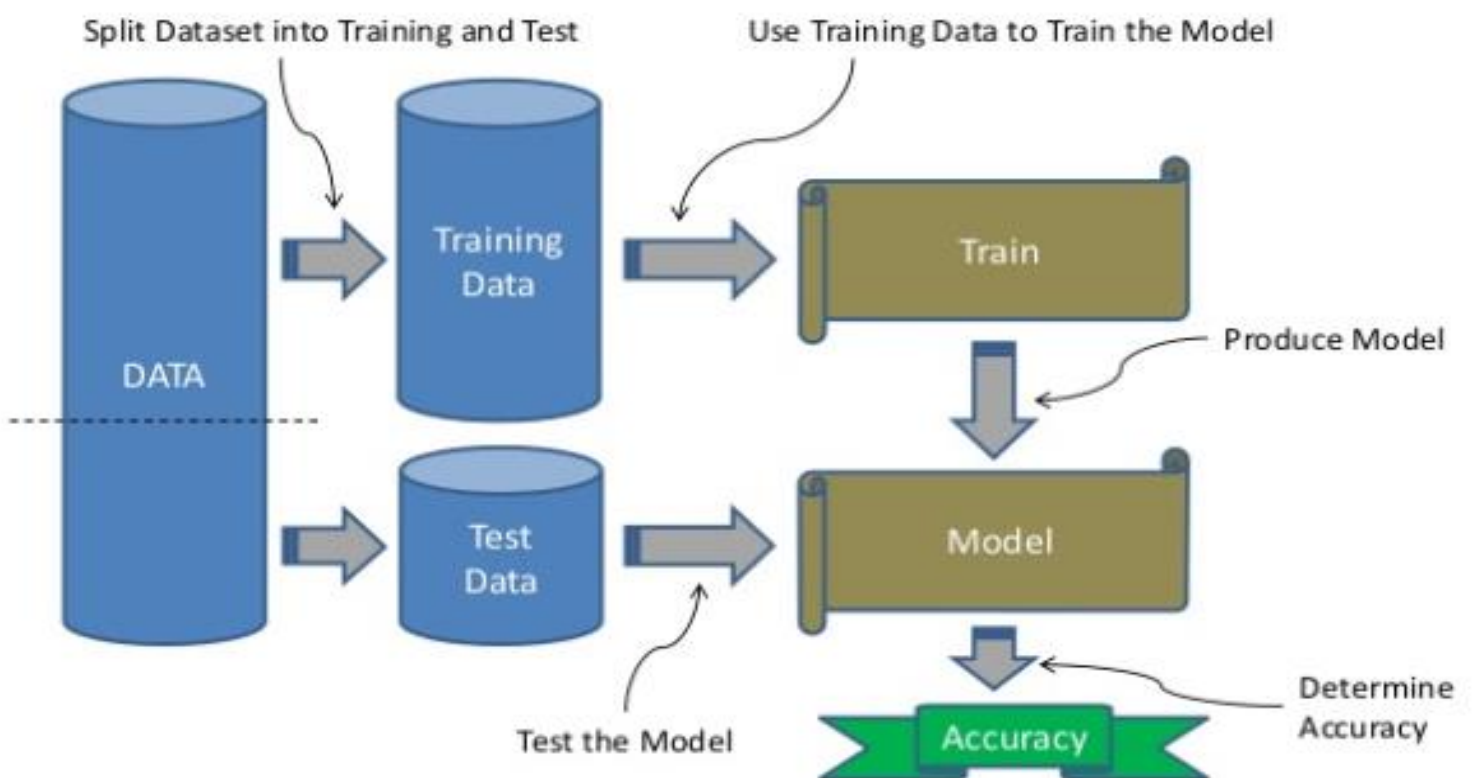
## Identification des données

## Machine Learning TRAIN AND TEST

**Data se compose en deux :**



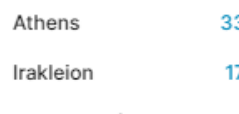



**Training for train the model 80%**



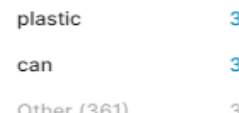
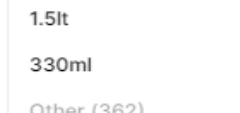


**Test data to test the model 20%**



## Gather data train.csv and test.csv by KAGGLE

Comme c'est un projet de « Machine Learning », on a besoin de suffisamment de dataset pour nous aider à comprendre les échecs précédents.

🔍 id	📅 date	🏙️ city	📍 lat	# long	# pop
ID	Date	City	Latitude	Longitude	Population
 64807559	 12 unique values	 Athens 33% Irakleion 17% Other (540) 50%	 35.340.6	 21.725.1	 137k
6480	31/01/18	Athens	37.97945	23.71622	664046
6481	31/01/18	Athens	37.97945	23.71622	664046
6482	31/01/18	Athens	37.97945	23.71622	664046
6483	31/01/18	Athens	37.97945	23.71622	664046
6484	31/01/18	Athens	37.97945	23.71622	664046
6485	31/01/18	Athens	37.97945	23.71622	664046
6486	31/01/18	Athens	37.97945	23.71622	664046

🏪 shop	🏷️ brand	📦 container	📏 capacity	# price	# qua
Shop	Brand Name	Container	Capacity	Price	Quantity
 6 unique values	 5 unique values	 plastic 33% can 33% Other (361) 33%	 1.5lt 33% 330ml 33% Other (362) 34%	 0.234.79	 3523
shop_1	kinder-cola	plastic	1.5lt	3.1	7056
shop_1	kinder-cola	can	330ml	0.85	12490
shop_1	adult-cola	glass	500ml	0.83	26640
shop_1	orange-power	glass	500ml	0.54	41892
shop_1	orange-power	plastic	1.5lt	0.83	22923
shop_1	gazoza	glass	500ml	0.77	24297

## Interpreting DATA

Machine Learning c'est informatique +statistique alors on doit besoin d'abord de manipuler les données et l'interpréter .

1. Pour prendre une idée sur nos données on a besoin de savoir la moyennes (mean)de chaque colonnes et le standard de déviation (std) min et le max

Entrée [5]: `import pandas as pd`

```
train_file_path = 'C:/Users/adamd/Desktop/ESITH 2.0/MR fri/projets FRI/train.csv'
train_data = pd.read_csv(train_file_path)
train_data.describe()
```

Out[5]:

	id	lat	long	pop	price	quantity
count	6480.000000	6429.000000	6434.000000	6480.000000	6480.000000	6480.000000
mean	3239.500000	38.301238	23.270246	355141.416667	1.188981	29431.855093
std	1870.759204	1.649771	1.086960	232828.796289	0.814567	17901.328301
min	0.000000	35.327870	21.734440	134219.000000	0.110000	2953.000000
25%	1619.750000	37.962450	22.417610	141439.750000	0.610000	16441.750000
50%	3239.500000	38.244440	22.930860	257501.500000	0.920000	25184.000000
75%	4859.250000	39.636890	23.716220	665871.000000	1.500000	37866.000000
max	6479.000000	40.643610	25.143410	672130.000000	4.540000	145287.000000

2. Après on a besoin de supprimer quelques valeurs manquantes

```
train_data = train_data.dropna(axis=0)
```



## Selecting The Prediction Target

Sélection de la cible de prédiction ( **quantity** ) l'élément de sortie y

```
y=train_data.quantity
```

```
Out[15]: 0    13280.0  
        1     6727.0  
        2     9848.0  
        3    20050.0  
        4    25696.0  
        Name: quantity, dtype: float64
```

## Choosing "Features"

Les colonnes qui sont entrées dans notre modèle (et utilisées plus tard pour faire des prédictions) sont appelées «Features». Ces les éléments d'entrées

Entrées = X= id , latitude et longitude et population de chaque ville et prix

```
Entrée [29]: import pandas as pd
```

```
train_file_path = 'C:/Users/adamd/Desktop/ESITH 2.0/MR fri/projets FRI/train.csv'  
train_data = pd.read_csv(train_file_path)  
  
train_data = train_data.dropna(axis=0)  
y=train_data.quantity  
train_features = ['id', 'lat', 'long', 'pop', 'price',]  
X = train_data[train_features]  
X.describe()  
X.head()
```

```
Out[29]:
```

	id	lat	long	pop	price
0	0.0	37.97945	23.71622	672130.0	0.96
1	1.0	37.97945	23.71622	672130.0	2.86
2	2.0	37.97945	23.71622	672130.0	0.87
3	3.0	37.97945	23.71622	672130.0	1.00
4	4.0	37.97945	23.71622	672130.0	0.39

## Building Model

Après définir les entrées et la sortie de notre model on passe au réaliser notre model

On doit d'abord :

- **Définir**: de quel type de modèle s'agira-t-il?
- **Fit**: capturez des modèles à partir des données fournies.  
C'est le cœur de la modélisation.
- **Prédire**
- **Évaluer**: déterminer la précision des prévisions du modèle.

## Les éléments d'entrées et les éléments de sorties

+ Code












+ Markdown

[9]:

```
nRowsRead = 1000 # specify 'None' if want to read whole file
# test.csv may have more rows in reality, but we are only loading/previewing the first 1000 rows
df1 = pd.read_csv('/kaggle/input/test.csv', delimiter=',', nrows = nRowsRead)
df1.dataframeName = 'test.csv'
nRow, nCol = df1.shape
print(f'There are {nRow} rows and {nCol} columns')
```

There are 1000 rows and 12 columns

Inputs :

Columns	
	id ID
	date Date
	city City
	lat Latitude
	long Longitude
	pop Population
	shop Shop
	brand Brand Name
	container Container
	capacity Capacity
	price Price

Output : Target Variable = "quantity"

## **SOLUTION**

apprendre à une fonction à faire correspondre une entrée à une sortie en se basant sur des exemples connus (des paires entrée-sortie).

alors le modèle le plus performant pour ce projet de previsions c'est :

**L'apprentissage supervisé**



plotScatterMatrix(df1, 18, 10) :

Scatter and Density Plot

