

## Apprentissage Automatique 1

### TP N°1 – Régression linéaire

#### Objectifs du TP :

- Appliquer la régression linéaire
- Calculer la prediction
- Calculer la performance
- Calculer l'erreur

#### Outils et version :

- Anaconda, Jupyter
- Python

#### Exercice N°1 : Prédiction des prix des maisons à BOSTON

Créer un classifieur de la régression linéaire pour prédire les prix des maisons à Boston. Nous avons un ensemble de données des caractéristiques des maisons et de prix.

Appliquer les étapes nécessaires pour créer le classificateur de la Régression linéaire afin d'analyser les prix des maisons.

1. Importer les bibliothèques nécessaires : pandas, numpy, matplotlib, et sklearn.

```
import pandas as pd
```

```
import numpy as np
```

```
from sklearn import linear_model
```

```
from sklearn.model_selection import train_test_split
```

2. Importer la base de données boston house

```
from sklearn.datasets import load_boston
```

```
from sklearn.metrics import mean_squared_error
```

```
boston=load_boston()
```

3. Afficher les clés et les targets de la base de données boston par : **keys(),target()**

4. Affecter df\_x , df\_y respectivement au deux éléments de la base de données Feature-name et Target par :  
**df\_x=pd.DataFrame(boston.data,columns=boston.feature\_names)**  
**df\_y=pd.DataFrame(boston.target)**
5. Afficher les 5 premières lignes des caractéristiques et de prix.
6. Créer le modèle de la régression linéaire : **regL=linear\_model.LinearRegression()**
7. Diviser la base de données en 2 parties : une partie de l'apprentissage qui contient 80% et une partie de test qui contient 20% :  
**x\_train,x\_test,y\_train,y\_test=train\_test\_split(df\_x,df\_y, test\_size=0.2)**
8. Afficher les données de la base d'apprentissage et celles de test.
9. Faire l'entraînement du modèle sur les données de la base d'apprentissage en utilisant le classifieur de la régression linéaire par la fonction suivante : **reg.fit(x\_train,y\_train)**
10. Faire des prévisions en utilisant le modèle entraîné (question 8) :  
**y\_pred=reg.predict(x\_test)**
11. Afficher les valeurs de la prédiction
12. Afficher les valeurs de test
13. Faire une comparaison et commenter les résultats
14. Calculer la performance du modèle utilisé pour prédire les prix des maisons à boston tout en utilisant la moyenne des erreurs MSE : **np.mean((y\_test-y\_pred)\*\*2)**
15. Afficher les résultats
16. Calculer l'erreur quadratique par : **mean\_squared\_error(y\_test,y\_pred)**

## Exercice N°2 : Evolution des loyers

Examiner les prix des loyers des maisons de 10 dernières Année à partir des trimestres qui représente la période et l'indice des prix pour pouvoir prédire le prix de loyer dans le futur.

Créer un classifieur de la régression linéaire pour analyser une base de données des trimestres et des indices de prix pour prédire la période et l'indice de prix dans le futur. Nous avons un ensemble de données qui contient les périodes et les indices.

Appliquer les étapes nécessaires pour créer le classificateur de la Régression linéaire afin d'analyser l'évolution des prix.

1. Importer les bibliothèques nécessaires : **pandas, matplotlib, et sklearn**  
**import matplotlib.pyplot as plt**  
**import pandas as pd**  
**from sklearn.model\_selection import train\_test\_split**  
**from sklearn.linear\_model import LinearRegression**
2. Importer la base de données de type csv : dataset =  
**pd.read\_csv('valeurs\_trimestrielles.csv',sep=';')**
3. Extraire les variables utilisées par la fonction :  
**iloc[:, :-1].values & iloc[:, -1].values**
4. Afficher les 5 premières lignes du dataset.
5. Diviser la base de données en 2 parties : une partie de l'apprentissage qui contient 70% et une partie de test qui contient 30%.
6. Créer le modèle de la régression linéaire : **regL=linear\_model.LinearRegression()**
7. Afficher les données de la base d'apprentissage et celles de test.
8. Faire l'entraînement du modèle sur les données de la base d'apprentissage en utilisant le classifieur de la régression linéaire.
9. Appliquer la fonction de la prédiction.
10. Visualiser les résultats obtenus par :  
**plt.scatter(X\_test, y\_test, color = 'red')**  
**plt.plot(X\_train, regressor.predict(X\_train), color = 'blue')**  
**plt.title('Evolution des loyers')**  
**plt.xlabel('Trimestre')**  
**plt.ylabel('Loyer')**  
**plt.show()**