**Etapes à suivre et codes concernant l’application de la régression linéaire pour l’apprentissage automatique**

**# Importer les libérerais**

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

**#Importer le dataset**

Data=pd.read\_csv('Boston.csv', sep=";")

**#Afficher le nuage de points**

plt.scatter(Data['RM'], Data['MEDV'],color='red')

**# Exclure les données dont les valeurs de la variable à expliquer sont <= 100**

Data = Data[Data[‘gen’]<=100

plt.scatter(Data['RM'], Data['MEDV'],color='red')

**#Déterminer les données de la variable d’entrée (explicative)**

x=Data.iloc[:,:-1]. values

**#Afficher x**

x

**#Déterminer les données de la variable de sortie (à expliquer)**

Y=Data.iloc[:,-1]. values

**#Afficher Y**

Y

**#Afficher la taille de x**

len(x)

**#Afficher la taille de Y**

len(Y)

**#Division de dataset (training data, testing data)**

x\_train, x\_test, Y\_train, Y\_test= train\_test\_split(x, Y, test\_size=1.0/3)

**#Afficher la taille de training data**

len(x\_train)

**#Afficher la taille de testing data**

len(x\_test)

**#Construire le modèle d’entrainement**

regressor=LinearRegression()

regressor.fit(x\_train, Y\_train)

**#Faire des prédictions**

Y\_pred=regressor.predict(x\_test)

**#Affichage de Y\_pred**

Y\_pred

**#Prédire une valeur de Y pour une valeur quelconque de x**

regressor.predict([[130]])

**#visualiser les résultats sous forme de nuage de points et d’adroite d’ajustement**

plt.scatter(x\_test, Y\_test, color =`red’)

plt.plot(x\_train, regressor.predict(x\_train), color=’bleue’)

plt.title(‘evaliation des energy’)

plt.x label(‘RM’)

plt.Y label (‘MEDV’)