Correction Activité 2

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
#Q1-Importer les données à partir du fichier.
dataset = pd.read_csv('./Weather.csv')
#Q2- afficher la dimension de l'ensemble de données
print('Taille de l''ensemble de données',dataset.shape)
#Q3-afficher les noms des colonnes (les variables)
print(dataset.columns)
#Q4-afficher les observations sur un plan 2D avec x=MinTemp et y=MaxTemp
dataset.plot(x='MinTemp', y='MaxTemp', style='o')
plt.title('MinTemp vs MaxTemp')
plt.xlabel('MinTemp')
plt.ylabel('MaxTemp')
plt.show()
#Q5-Construction du nouvel ensemble de données,composé
#seulement des deux variables "MinTemp" et "MaxTemp".
#la fonction reshape(-1,1) permet de créer une matrice 2D avec plusieurs lignes et une seule colonne
X = dataset['MinTemp'].values.reshape(-1,1)
y = dataset['MaxTemp'].values.reshape(-1,1)
# Maxtemp = a MinTemp + b
#Q6-Division des données en un ensemble d'apprentissage (80%) et un ensemble de test(20%)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
#Q7-Effectuer la régression linéaire en utilisant l'ensemble d'apprentissage.
regressor  = LinearRegression()
regressor.fit(X_train, y_train)
#Q8-Affichage des paramètres du modèle
print('Pente a: ', regressor.intercept_) # la pente a
print('Ordonnée à l''origine b :',regressor.coef_) # l'ordonnée à l'origine b
```

```
#Q9-Affichage du coefficient  de détermination R_2
#Conclsion:  R_2= 0.77 ==> pouvoir de prédiction est proche de 1 ==> droite de régression
# est capable de déterminer 77% de la distibution des observations==> assez bon
r_sq = regressor.score(X_train, y_train)
print('coefficient of determination:', r_sq)
#Q10-Prédire les températures maximales en utilisant
#les températures minimales de l'ensemble de test
y_pred = regressor.predict(X_test)
#Q11-Affichage du résultat
plt.scatter(X_test, y_test,  color='gray') #Valeurs réelles observées
plt.scatter(X_test, y_pred,  color='blue') #Valeurs prédites
plt.plot(X_test, y_pred, color='red', linewidth=2) #Valeurs prédites
plt.show()
```