TP3 - AIGLE

1)

```
with open('PRSA.csv', newline='') as csvfile:
    reader = csv.DictReader(csvfile)
    FinTrou = []
valcbwd = []
                  for val in range(len(valcbwd)):
                      if(valcbwd[val] == row['cbwd']):
   test = True
                  valcbwd.append(row['cbwd'])
if(row['pm2.5'] == 'NA'):
                                PrevTrou = int(row['No']) - int(nbVal+1)
PrevTrouData = df.loc[(df['No'] == PrevTrou)]
ValPrev = int(PrevTrouData['pm2.5'])
                                ValSuiv = int(row['pm2.5'])
i = (ValSuiv-ValPrev)/(nbVal+1)
                  elif (row['pm2.5'] !='NA' and trou == True):

a = int(row['No'])-1
    trou = False

f = open('AIGLE_DonnéePRSA.csv', 'w')
entetes = ["Trou", "Debut", "Fin", "ValManq", "Incré"]
ligneEntete = ",".join(entetes) + "\n"
f.write(ligneEntete)
    print(" Trou | Debut | Fin | ValM | Incrément")
     for i in range(len(valI)) :
        elif (i > 8 & i < 99):
    print(" ", i + 1, " |", DebTrou[i], " |", FinTrou[i], " |", NbValManquante[i], " |",</pre>
             print(" ", i + 1, "|", DebTrou[i], " |", FinTrou[i], " |", NbValManquante[i], " |",
print("Retrouvez ces données dans le fichier : DonnéePRSa.csv")
     for inc in range(len(valI)):
             for j in range(NbValManquante[inc]):
             for j in range(NbValManquante[inc]):
    b = df.loc[df["No"] == DebTrou[inc] - 1]
                  df.loc[df["No"] == DebTrou[inc] + j, "pm2.5"] = float(b["pm2.5"]+valI[inc]*(j+1))
    print("Save on file AIGLE_tp2_1.csv")
        print("plus aucune trou de plus de 15 valeurs")
         print("il existe des trou toujours")
    print(valcbwd)
```

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/3942a721-00f 1-49f5-8334-d3079e4d6a28/main.py

Il crée un fichier AIGLE_DonnéesPRSA.csv recenser l'emplacement du trou son début, sa fin et sont incrément :

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/5c040f8d-2c5 1-4cc2-9255-cbb0198b8b9e/AIGLE DonnePRSA.csv

Il crée un fichier AIGLE_tp3_1.csv dans lequel les trou de moins de 15 valeur manquantes sont modifier par la méthode de l'incrémentation et les plus de 15 valeurs par un : "?"

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/17d4763a-e29 7-48e8-bb8f-c6dfc6e65b80/AIGLE_tp3_1.csv

2)

a)

Multilayer Perceptrion

cross validation: 10

Time taken to build model: 43.59	seconds
=== Cross-validation ===	
=== Summary ===	
Correlation coefficient	0.5217
Mean absolute error	59.4473
Root mean squared error	80.37
Relative absolute error	86.3737 %
Root relative squared error	87.2767 %
Total Number of Instances	42220
Ignored Class Unknown Instances	1580

percentage split 66% et preserve order for%split

Correlation coefficient	0.5087
Mean absolute error	87.7006
Root mean squared error	112.7966
Relative absolute error	132.1985 %
Root relative squared error	129.1918 %
Total Number of Instances	14859
Ignored Class Unknown Instances	33

RandomTree

cross validation: 10

Correlation coefficient	0.8236
Mean absolute error	29.2853
Root mean squared error	54.6016
Relative absolute error	42.5499 %
Root relative squared error	59.2938 %
Total Number of Instances	42220
Ignored Class Unknown Instances	1580

percentage split 66% et preserve order for%split

=== Summary ===		
Correlation coefficient	0.2652	
Mean absolute error	70.0144	
Root mean squared error	98.6312	
Relative absolute error	105.5386	8
Root relative squared error	112.9675	8
Total Number of Instances	14859	
Ignored Class Unknown Instances		33

b)

code permettant de crée les colonnes et ajouter les valeurs a -3heures,-4heures,-5heures

```
import pandas
df = pandas.read_csv("AIGLE_tp3_1.csv")

df["pm2.5_3"] = ['?'] * len(df)
df["pm2.5_4"] = ['?'] * len(df)
df["pm2.5_5"] = ['?'] * len(df)

for i, ligne in enumerate(df.iterrows()):
    if(int(df["No"][i])>29):
        df["pm2.5_3"][i] = df["pm2.5"][i-3]
        df["pm2.5_4"][i] = df["pm2.5"][i-4]
        df["pm2.5_5"][i] = df["pm2.5"][i-5]

df = df.drop(range(0,5))

df.to_csv("AIGLE_tp3_2c.csv", index=False)
```

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/0fecec1a-6b8 8-4c68-ada5-9a49f2610d9a/tp3 main3.py

le fichier contenant le nouveau de donnés avec les données avec :

```
pm2.5 5 heures plus tot (5 lignes avant)
pm2.5 4 heures plus tot (4 lignes avant)
pm2.5 3 heures plus tot (3 lignes avant)
```

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/ef9c6e0f-0d45 -45e1-a060-13b132f305c1/AIGLE tp3 2c.csv

jeux de donnée + pm2.5 | pm2.5_3 | pm2.5_4 | pm2.5_5

percentage split 66% et preserve order for%split

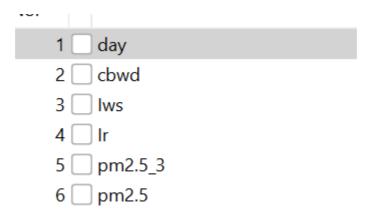
=== Summary ===		
Correlation coefficient Mean absolute error	0.8891 25.4132	
Root mean squared error	39.9627	
Relative absolute error	38.3062 45.7689	
Root relative squared error Total Number of Instances	14857	70
Ignored Class Unknown Instances		33

Random Tree

=== Summary ===		
Correlation coefficient	0.7335	
Mean absolute error	39.3566	
Root mean squared error	62.8407	
Relative absolute error	59.3235	8
Root relative squared error	71.971	8
Total Number of Instances	14857	
Ignored Class Unknown Instances		33

c)

attribut selection



Random Tree

0.7584
37.5635
60.1211
56.6206 %
68.8563 %
14857
33

Multilayer perceptron

=== Summary ===	
Correlation coefficient	0.7584
Mean absolute error	37.5635
Root mean squared error	60.1211
Relative absolute error	56.6206 %
Root relative squared error	68.8563 %
Total Number of Instances	14857
Ignored Class Unknown Instances	33

d)

code permettant de crée un nouveau fichier uniquement avec les colones pm2.5

```
import pandas
df = pandas.read_csv("AIGLE_tp3_2c.csv")
df = df.drop(columns=
["No","year","month","day","hour","DEWP","TEMP","PRES","cbwd","Iws","Is","Ir"])
df.to_csv("AIGLE_tp3_2d.csv", index=False)
```

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/4c6adbf8-6a8 e-48bd-aaec-198e1063c546/Tp3_d.py

fichier contenant uniquement les colonnes pm2.5 :

 $\underline{https://s3-us-west-2.amazonaws.com/secure.notion-static.com/76b0cc59-019}\\ \underline{f-4e48-af59-d789c9f301db/AIGLE_tp2_2d.csv}$

```
1 pm2.5
2 pm2.5_3
3 pm2.5_4
4 pm2.5_5
```

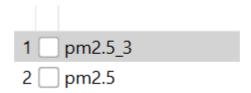
Multilayer perceptron

=== Summary ===	
Correlation coefficient	0.8812
Mean absolute error	25.9237
Root mean squared error	41.4716
Relative absolute error	39.0756 %
Root relative squared error	47.4972 %
Total Number of Instances	14857
Ignored Class Unknown Instances	33

Random Tree

=== Summary ===		
Correlation coefficient	0.7775	
Mean absolute error	36.5356	
Root mean squared error	59.0848	
Relative absolute error	55.0714	8
Root relative squared error	67.6694	8
Total Number of Instances	14857	
Ignored Class Unknown Instances		33

Attribut selection



Multilayer perceptron

=== Summary ===	
Correlation coefficient	0.8792
Mean absolute error	26.2866
Root mean squared error	41.8917
Relative absolute error	39.6227 %
Root relative squared error	47.9782 %
Total Number of Instances	14857
Ignored Class Unknown Instances	33

RandomTree

=== Summary ===		
Correlation coefficient	0.8734	
Mean absolute error	26.0624	
Root mean squared error	42.4244	
Relative absolute error	39.2848	8
Root relative squared error	48.5884	8
Total Number of Instances	14857	
Ignored Class Unknown Instances		33