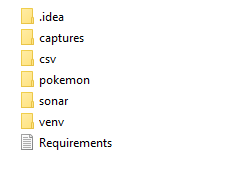
**TP : Apprentissage Automatique**

Afin de valider les compétences du cours, nous avons reçu un projet de fin de module. Ce projet est divisé en deux parties et ce rapport va détailler les différentes étapes qui nous ont permis de résoudre ces différents exercices.

Pour se faire, afin de mieux comprendre le fonctionnement, nous avons décidé d’utiliser directement python dans notre éditeur PYCHARM au lieu d’utiliser ANACONDA. Nous avons donc créé un projet python nommé IA\_M1\_2021\_12\_28. La capture suivante va permettre de détailler la structure du projet.



Le dossier capture contient les différentes captures qui montrent les résultats obtenus.

Le dossier csv regroupe les différents fichiers Excel aves lesquels nous allons travailler.

Le dossier pokemon contient le fichier python qui va contenir les codes pour traiter le tp2.

Le dossier sonar contient le fichier python qui va contenir les codes pour traiter le tp1.

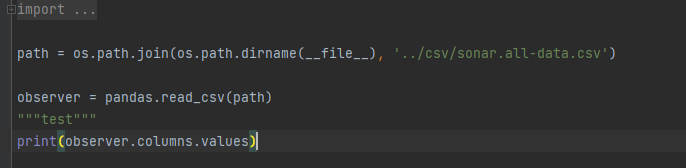
Le dossier venv est notre environnement virtuel. Requirements est un fichier texte qui contient les différents modules que nous devrons installer pour effectuer le travail.

Cette façon de travailler va nous permettre de mieux comprendre ce qui se passe avec les différentes fonctions que nous utilisons.

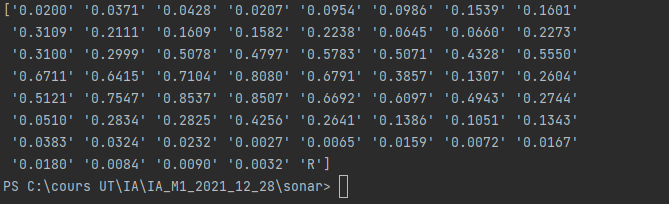
**TP1 : Les k plus proches voisins Classification**

Pour ce TP, nous allons travailler sur la base sonar.all-data.

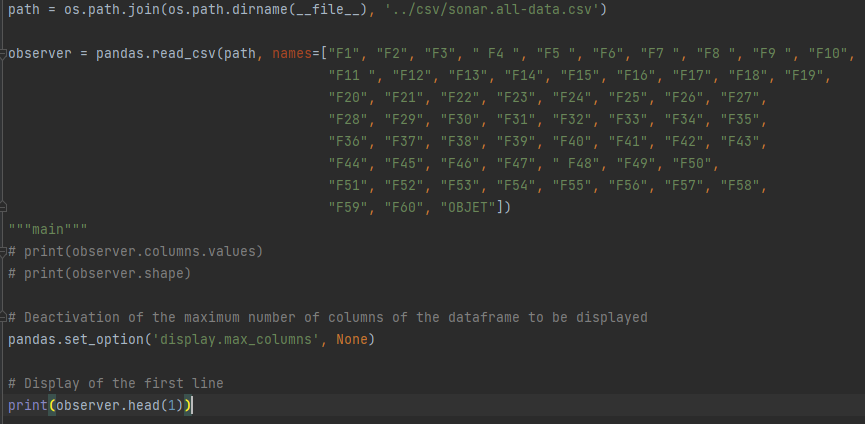
Nous allons d’abord charger la base de données sonar :



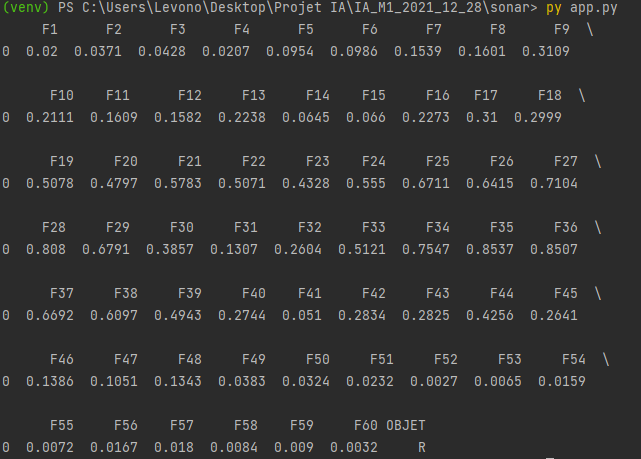
On affiche les entrées du fichier avec la commande print(fichier.columns.values) :



On renomme ensuite les variables avec la commande observer = pandas.read\_csv(path, names=["F1", "F2", "F3", "F4", "F5", "F6", "F7", "F8", "F9", "F10",  
 "F11 ", "F12", "F13", "F14", "F15", "F16", "F17", "F18", "F19",  
 "F20", "F21", "F22", "F23", "F24", "F25", "F26", "F27",  
 "F28", "F29", "F30", "F31", "F32", "F33", "F34", "F35",  
 "F36", "F37", "F38", "F39", "F40", "F41", "F42", "F43",  
 "F44", "F45", "F46", "F47", " F48", "F49", "F50",  
 "F51", "F52", "F53", "F54", "F55", "F56", "F57", "F58",  
 "F59", "F60", "OBJET"])



On affche ensuite les premières lignes de la base de donnée : print(observer.head())



**Combien de classes ?**

Nous avons deux classes : les Rochers (R) et les mines (M).

**Combien de caractéristiques descriptives ? De quels types ?**

Pour les statistiques descriptives, nous en avons 4 :

Mean : qui nous donne la moyenne

Std : qui nous donne l’écart-type

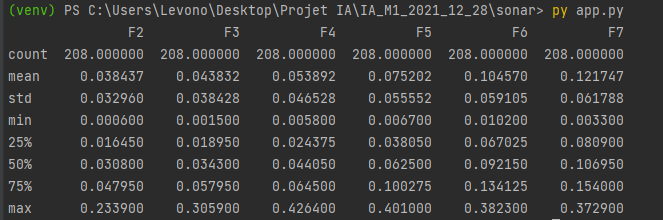
Min, Max qui nous donnent les minimum et maximum

Les quartiles qui nous donnent les valeurs à 25, 50 et 75%.

**Calculer les statistiques de base des variables 2 à 7**

Nous avons eu le résultat en faisant :

**stat\_base = observer[["F2", "F3", "F4", "F5", "F6", "F7"]]  
print(stat\_base.describe())**



**Combien d’exemples ?**

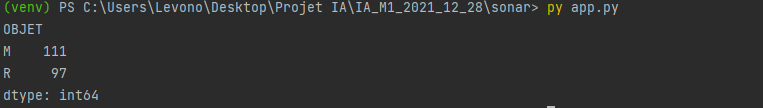


Nous avons 61 features avec 208 entrées chacune numérotées de 0 à 207.

**Combien d’exemples de chaque classe ?**

Nous allons utiliser la commande suivante :

**values\_expl = observer['OBJET'].value\_counts()  
print("Exemples de chaque classe : " + str(values\_expl))**

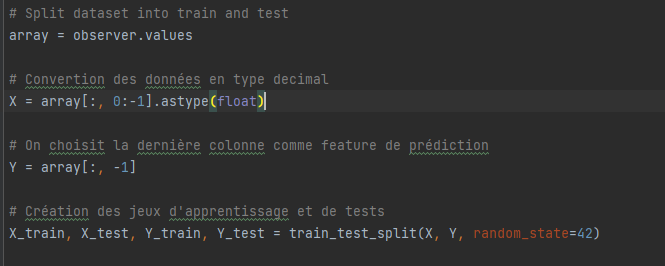


Nous avons 11 objets de type Mine et 97 de type Rocher.

**Comment sont organisés les exemples ?**

**Apprentissage et test**

Créons le modèle knn en python et entrainons-le sur la base d’apprentissage :



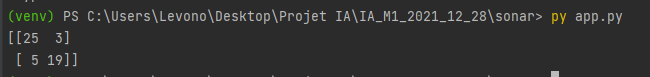
Le score obtenu en apprentissage :



Le score obtenu en test :



Affichons la matrice de confusion en utilisant la commande : **print(confusion\_matrix(predictions,Y\_test))**



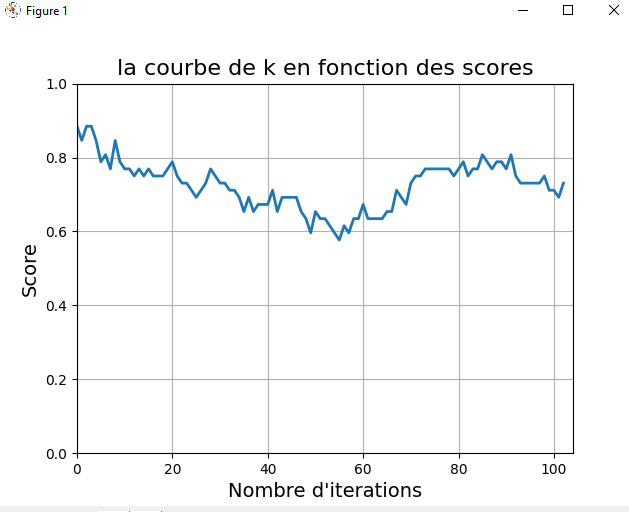
Dans la ligne 0 il y’a 28 données. 25 sont correctement attribuées à 0 et 3 sont attribuées à tort à 1.

Dans la ligne 1 il y’a 24 données. 5 sont incorrectement attribuées à 0 et 19 sont correctement attribuées à 1.

**Jouer avec le paramètre k**

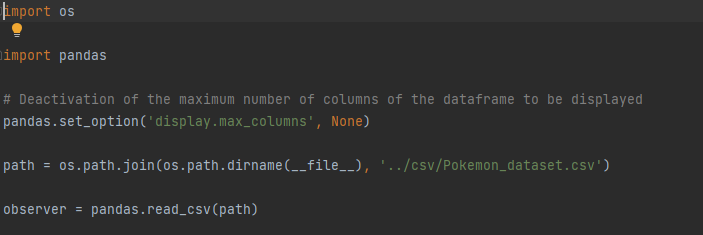
K est influant par le fait que la modification de la valeur de K entraine un changement des résultats obtenus. Si K augmente, les valeurs obtenues en apprentissage et en test diminuent et inversement.

**Traçons la courbe de k en fonction des scores**

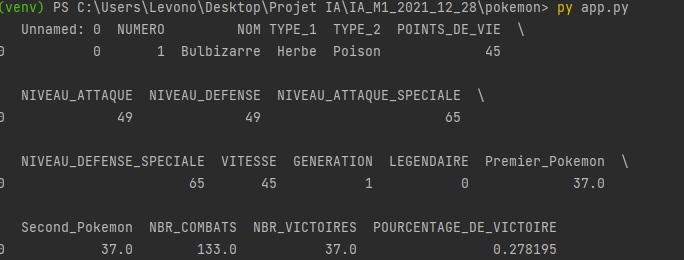


**TP2 : La régression linéaire**

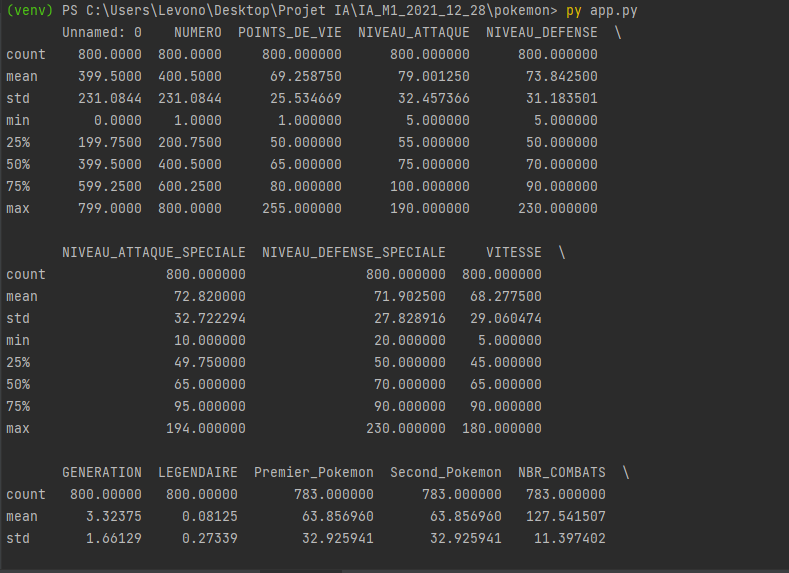
Pour ce TP, nous allons travailler sur la base Pokemon\_dataset. Commençons d’abord par l’importer.



**Afficher les premières observations de la base de données ?**



**Combien de caractéristiques descriptives ? De quels types ?**



Pour les statistiques descriptives, nous en avons 4 :

Mean : qui nous donne la moyenne

Std : qui nous donne l’écart-type

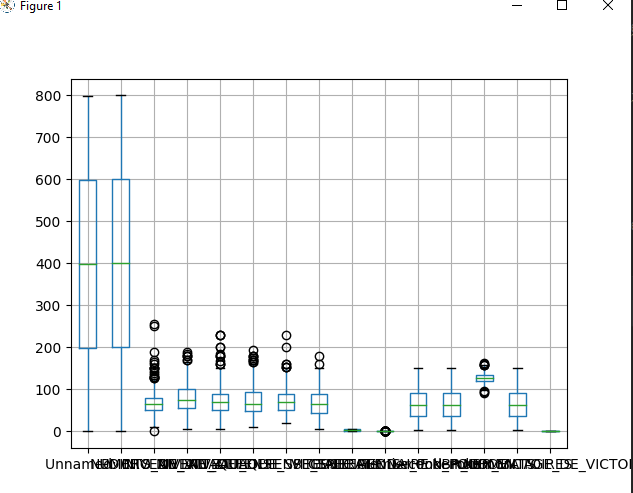
Min, Max qui nous donnent les minimum et maximum

Les quartiles qui nous donnent les valeurs à 25, 50 et 75%.

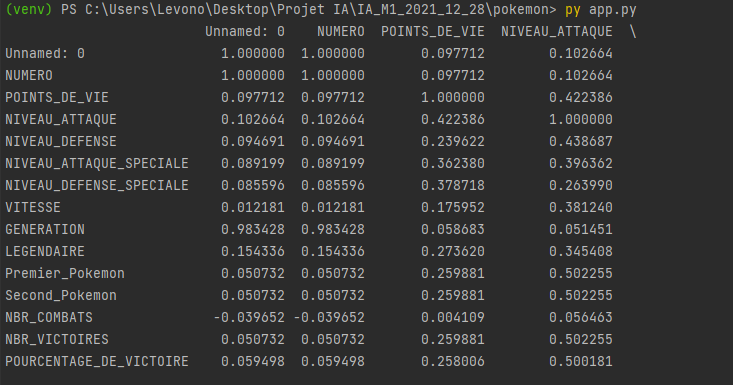
**Faire une analyse descriptive des données**

**Comment sont organisés les exemples ?**

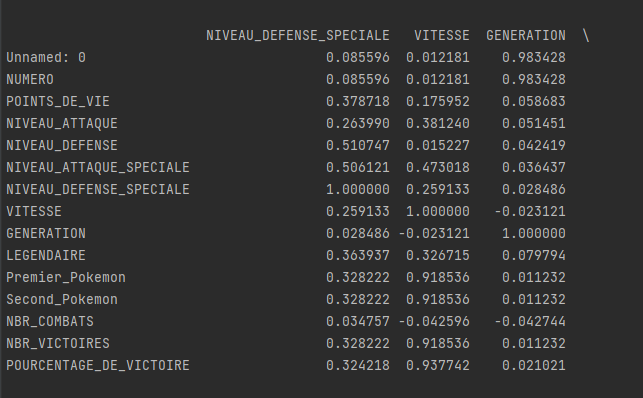
**Tracer les boxplots de toutes les variables**

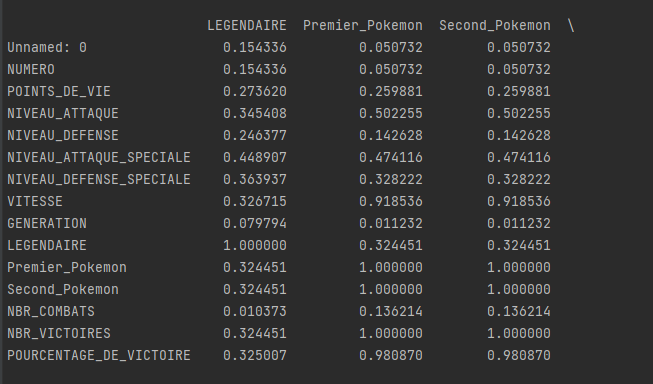
****

**Calculer et tracer la matrice de corrélation des différentes features**



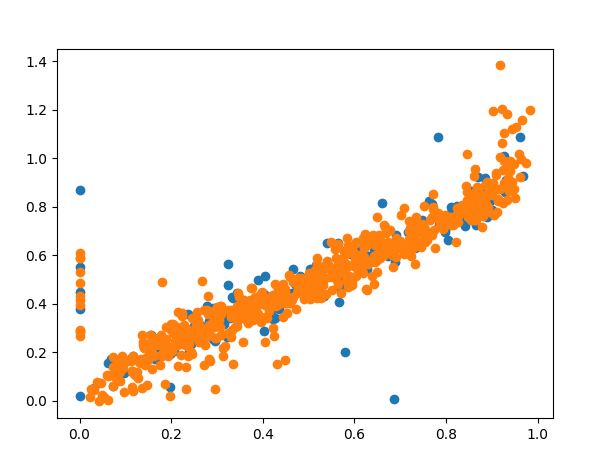




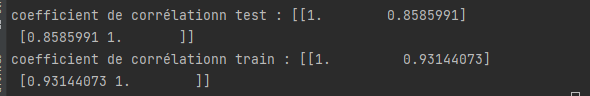


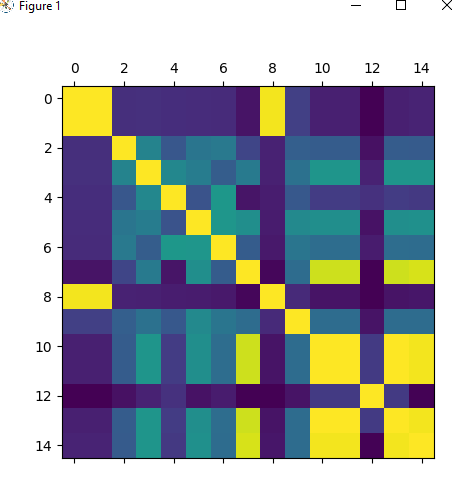


Traçons la matrice de corrélation :

Le

Le coefficient de corrélation :





Les valeurs obtenues en apprentissage :



Les valeurs obtenues en test :

