Université des Sciences et de la Technologie Houari Boumediene
Faculté d'Informatique

TP Fouille de données

Rapport TP : Les algorithmes de l'apprentissage automatique (Machine Learning)

Fait par :

Nom et prénom : ABDELMALEK BENMEZIANE

Matricule : 171731046778

Spécialité : M2 BIOINFO

Section : A

Contents

List of Figures

Chapter 1

Introduction générale

1.1 Introduction générale

Le data mining désigne le processus d'analyse de volumes massifs de données et du Big Data sous différents angles afin d'identifier des relations entre les data et de les transformer en informations exploitables. Ce dispositif rentre dans le cadre de la Business Intelligence et a pour but d'aider les entreprises à résoudre des problèmes, à atténuer des risques et à identifier et saisir de nouvelles opportunités business.

En français, ce processus porte différents noms :

- Exploration de données.
- Fouille de données.
- Forage de données.
- Ou encore extraction de connaissances à partir de données.

Le data mining n'est pas un concept récent. Déjà au XVIIème siècle, les individus cherchaient des solutions pour analyser les données et identifier des caractéristiques communes.

Chapter 2

Le preprocessing

2.1 Le preprocessing et ses techniques

2.1.1 Définition de preprocessing

Le preprocessing des données dans le machine learning (ML) est une étape cruciale qui permet d'améliorer la qualité des données afin de promouvoir l'extraction d'informations significatives à partir des données. Le preprocessing des données dans Machine Learning fait référence à la technique de préparation (nettoyage et organisation) des données brutes pour les rendre adaptées à la construction et à la formation de modèles Machine Learning.

2.1.2 Les techniques du preprocessing

L'ouverture du dataset (open dataset)

Il faut ouvrir la dataset voulu pour pouvoir appliquer le preprocessing.

Le nettoyage du dataset (Data Cleaning)

- Convertir les attributs catégoriels en numérique (Encodage des données catégorielles)

  Les données catégorielles font référence aux informations qui ont des catégories spécifiques dans l'ensemble de données.

Les modèles d'apprentissage automatique sont principalement basés sur des équations mathématiques. Ainsi, vous pouvez intuitivement comprendre que le fait de conserver les données catégorielles dans l'équation causera certains problèmes puisque vous n'auriez besoin que de nombres dans les équations.

- Identifier et traiter les valeurs manquantes (missing values)
  Dans le preprocessing des données, il est essentiel d'identifier et de gérer correctement les valeurs manquantes, fondamentalement, il existe deux façons de gérer les données manquantes :
  Suppression d'une ligne particulière : Dans cette méthode, vous supprimez une ligne spécifique qui a une valeur nulle pour une caractéristique ou une colonne particulière où plus de 75% des valeurs sont manquantes. Cependant, cette méthode n'est pas efficace à 100%.
  Calcul de la moyenne : Cette méthode est utile pour les fonctionnalités contenant des données numériques telles que l'âge, le salaire, l'année, etc. Ici, vous pouvez calculer la moyenne, la médiane ou le mode d'une colonne.

La normalisation du dataset (Data Normalization)

La normalisation des données est une technique utilisée dans l'exploration de données pour transformer les valeurs d'un ensemble de données en une échelle commune. Ceci est important car de nombreux algorithmes d'apprentissage automatique sont sensibles à l'échelle des caractéristiques d'entrée et peuvent produire de meilleurs résultats lorsque les données sont normalisées.

2.2 Définition de l'apprentissage non supervisé

L'apprentissage non supervisé est une branche du machine learning, caractérisée par l'analyse et le regroupement de données non-étiquetées. Pour cela, ces algorithmes apprennent à trouver des schémas ou des groupes dans

les données, avec très peu d'intervention humaine. En termes mathématiques, l'apprentissage non supervisé implique l'observation de plusieures occurrences d'un vecteur X and l'apprentissage de la probabilité de distribution p(X) pour ces occurrences.

2.3 Définition de l'apprentissage supervisé

L'apprentissage supervisé utilise un jeu d'entraînement pour apprendre aux modèles à produire les résultats souhaités. Ce jeu de données d'apprentissage comprend des entrées et des sorties correctes, qui permettent au modèle d'apprendre au fil du temps.

Chapter 3

Les techniques et caractéristiques

3.1 Les algorithmes de clustering non supervisés

3.1.1 K-Means

K-Means est l'un des algorithmes de clustering les plus répandus. Il permet d'analyser une dataset caractérisées par un ensemble de descripteurs, afin de regrouper les données "similaires" en terme de "distance" en groupes (ou clusters).

3.1.2 K-Medoids

K-Medoids est un algorithme de clustering ressemblant à la technique de clustering K-Means, il diffère principalement de l'algorithme K-Means par la manière dont il sélectionne les centres des clusters.

3.1.3 AGNES

AGNES (Agglomerative Nesting) est l'un des algorithmes de clustering hiérarchique les plus populaires utilisés dans l'exploration de données.
L'algorithme AGNES utilise une approche "ascendante" pour le clustering hiérarchique. L'algorithme forme des clusters singleton de chacun des points de données. Il les regroupe ensuite de bas en haut dans la structure arborescente (appelée dendrogramme) jusqu'à ce que tous les points similaires

forment un seul cluster (représenté par la racine du dendrogramme).

3.1.4 DIANA

DIANA est également connu sous le nom d'algorithme de clustering DIvisie ANAlysis. Il s'agit de la forme d'approche descendante du clustering hiérarchique où tous les points de données sont initialement affectés à un seul cluster. De plus, les clusters sont divisés en deux clusters les moins similaires.

3.1.5 DBSCAN

DBSCAN signifie Density-Based Spatial Clustering of Applications with Noise, est un algorithme qui regroupe des points de données « densément groupés » dans un seul cluster. DBSCAN ne requiert que deux paramètres : epsilon et minPoints. Epsilon est le rayon du cercle à créer autour de chaque point de données pour vérifier la densité et minPoints est le nombre minimum de points de données requis à l'intérieur de ce cercle pour que ce point de données soit classé comme point central.

3.2 Les algorithmes de classification supervisés

3.2.1 K-Nearest Neighbors (KNN)

L'algorithme des k-voisins les plus proches, également connu sous le nom de KNN ou k-NN, est un classificateur d'apprentissage supervisé non paramétrique, qui utilise la proximité pour effectuer des classifications ou des prédictions sur le regroupement d'un point de données individuel.

3.2.2 Naive Bayes

Le classificateur Naïve Bayes est un algorithme d'apprentissage automatique supervisé populaire utilisé pour les tâches de classification telles que la classification de texte. Il appartient à la famille des algorithmes d'apprentissage génératif, ce qui signifie qu'il modélise la distribution des

intrants pour une classe ou une catégorie donnée. Cette approche repose sur l'hypothèse que les caractéristiques des données d'entrée sont conditionnellement indépendantes compte tenu de la classe, ce qui permet à l'algorithme de faire des prédictions rapides et précises.

3.2.3 Arbre de décision

Un arbre de décision est un schéma représentant les résultats possibles d'une série de choix interconnectés. Il permet à une personne ou une organisation d'évaluer différentes actions possibles en fonction de leur coût, leur probabilité et leurs bénéfices.

3.2.4 Réseau de neurones (NN)

Un réseau neuronal est l'association, en un graphe plus ou moins complexe, d'objets élémentaires, les neurones formels. Les principaux réseaux se distinguent par l'organisation du graphe (en couches,complets...), c'est-à-dire leur architecture, son niveau de complexité (le nombre de neurones, présence ou non de boucles de rétroaction dans le réseau), par le type des neurones (leurs fonctions de transition ou d'activation) et en fin par l'objectif visé:apprentissage supervisé ou non,optimisation,systèmes dynamiques...etc.

3.2.5 Machine à vecteurs de support (SVM)

SVM (Support Vector Machine ou Machine à vecteurs de support) est un algorithme d'apprentissage automatique supervisé qui peut être utilisé pour les problèmes de classification ou de régression. Toutefois, il est surtout utilisé dans les problèmes de classification.

Chapter 4

Interfaces

4.1 L'interface principale

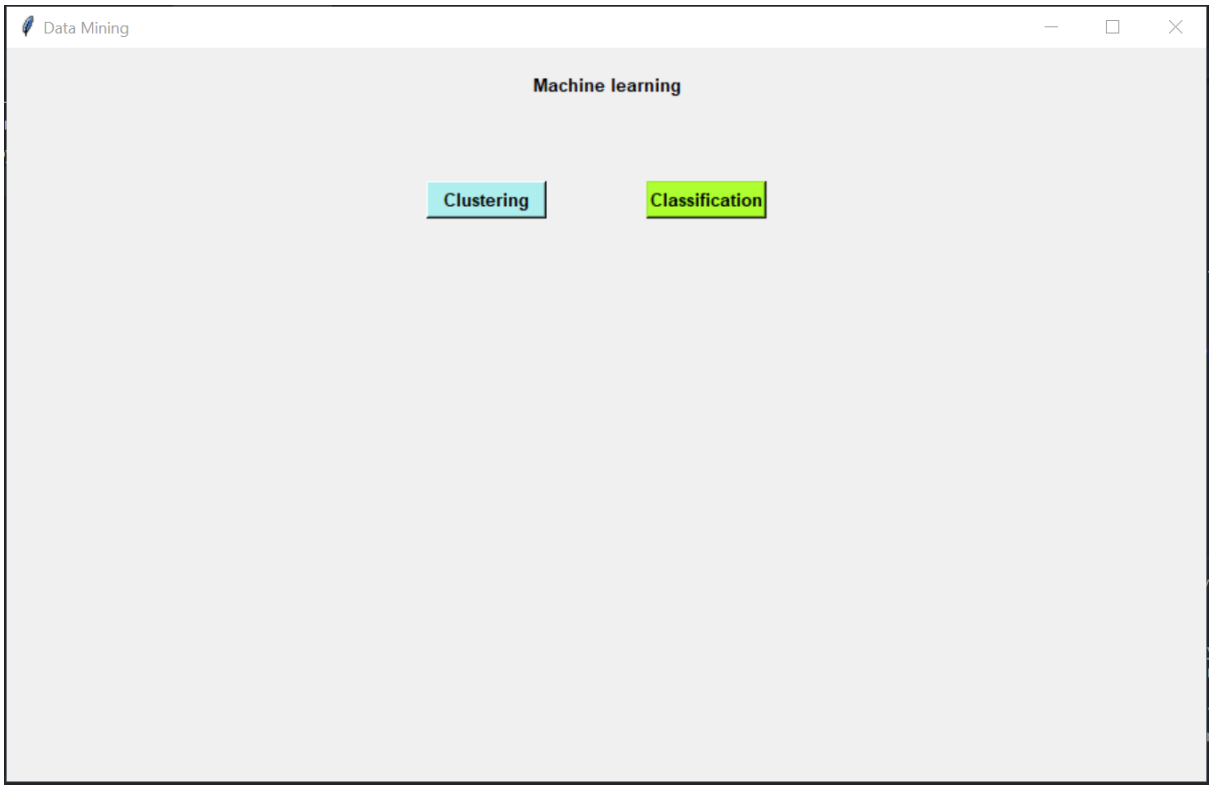


Figure 4.1: Interface principale

4.2 L'interface clustering

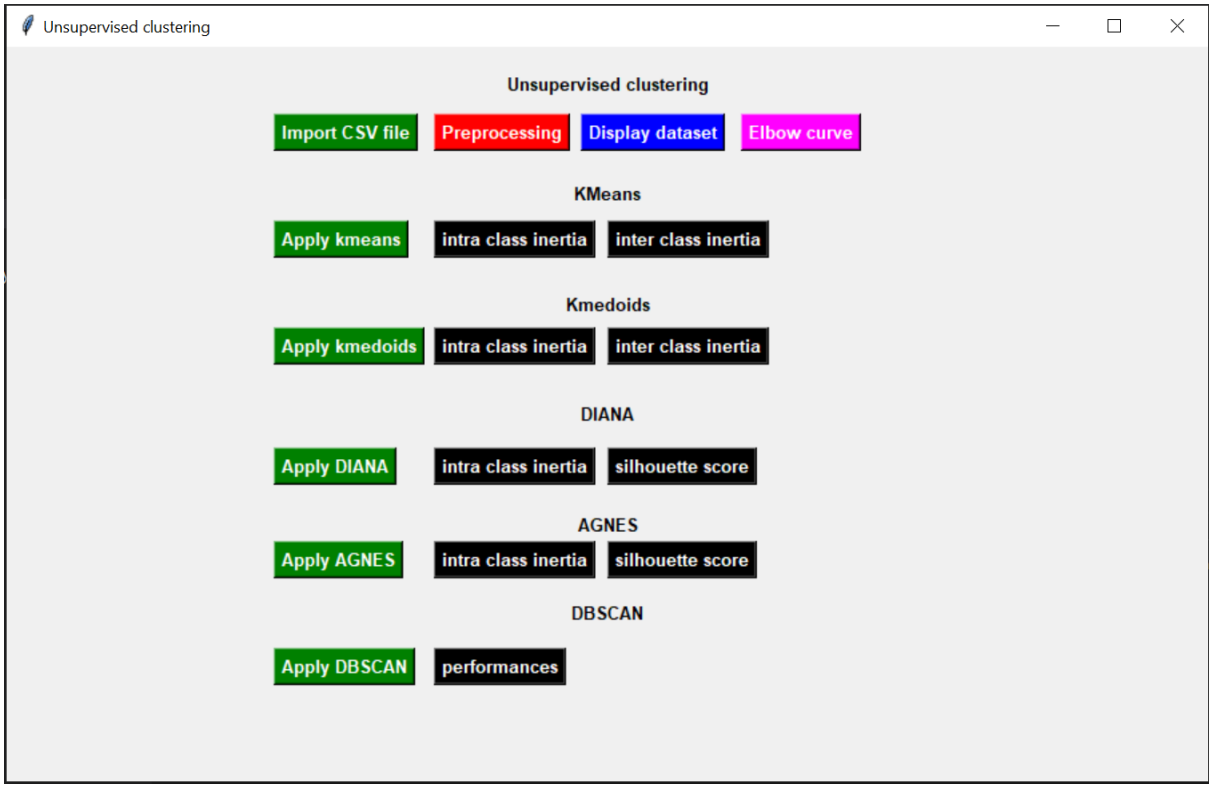


Figure 4.2: Interface clustering

4.3 L'interface classification

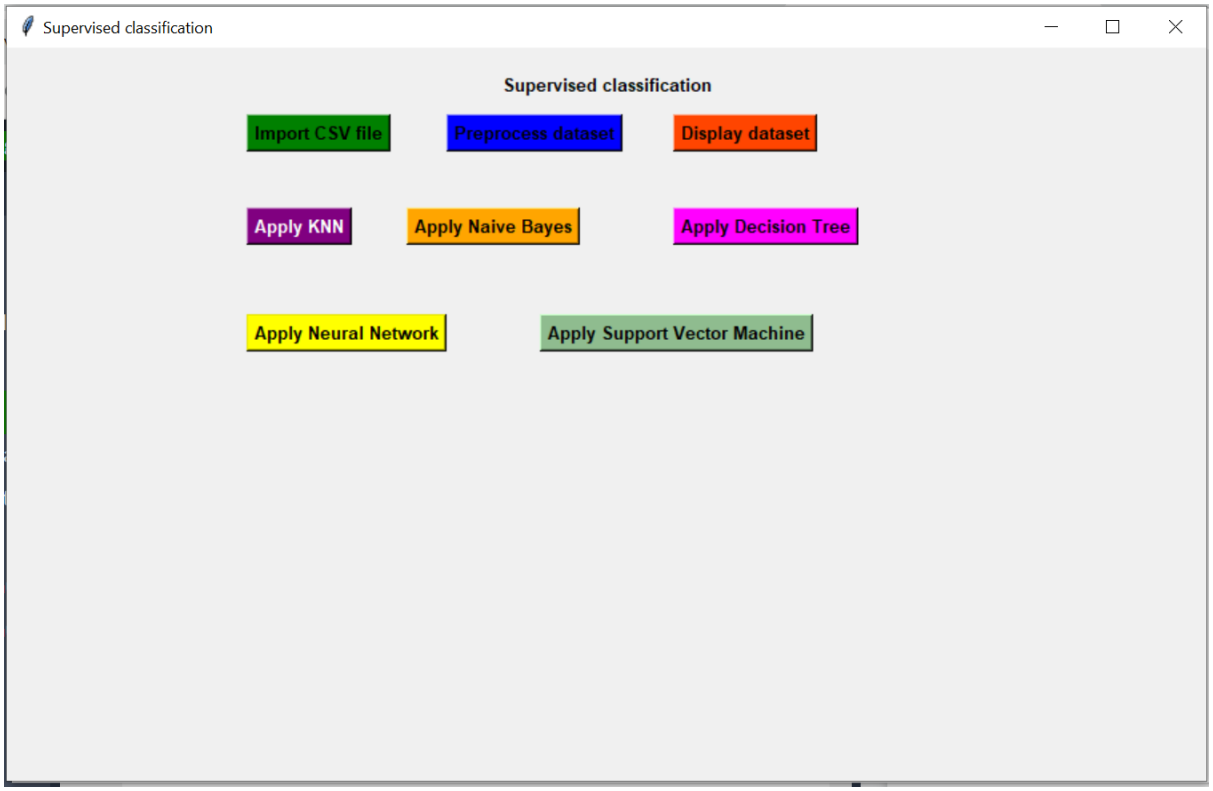


Figure 4.3: Interface classification

Chapter 5

Les resultats obtenus - clustering -

5.1 L'importation d'un dataset

Figure 5.1: Importer dataset

Figure 5.2: Le dataset diabetes

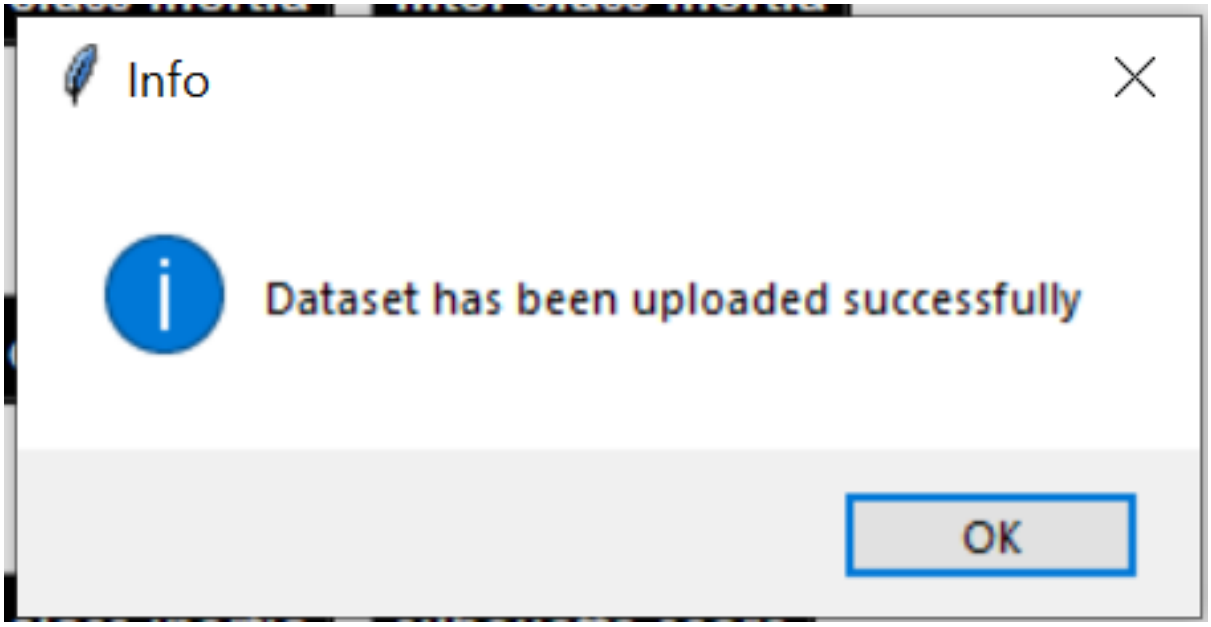


Figure 5.3: Output

5.2 Le preprocessing

Figure 5.4: Preprocessing

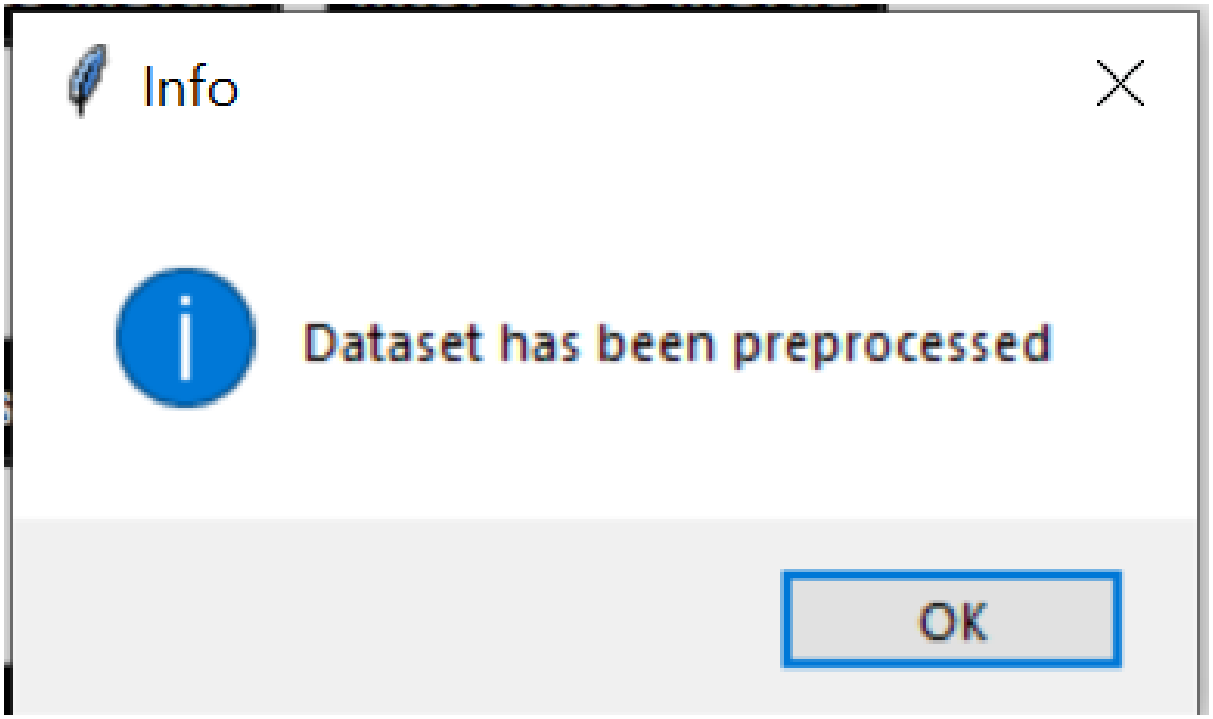


Figure 5.5: Output

5.3 La courbe d'elbow

Figure 5.6: Elbow

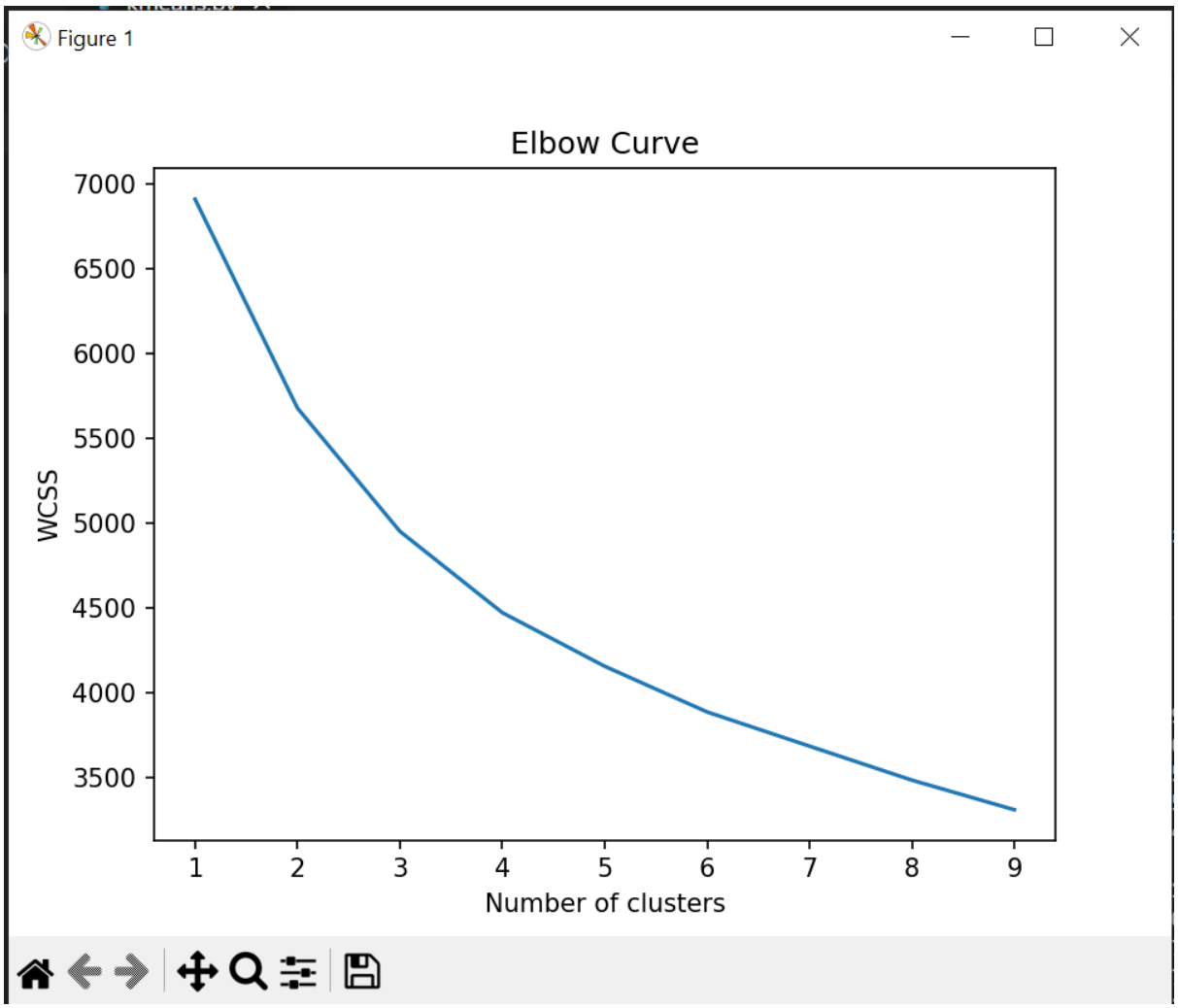


Figure 5.7: Output

Le nombre optimale des clusters est égale à : 2

5.4 Kmeans

Figure 5.8: Kmeans

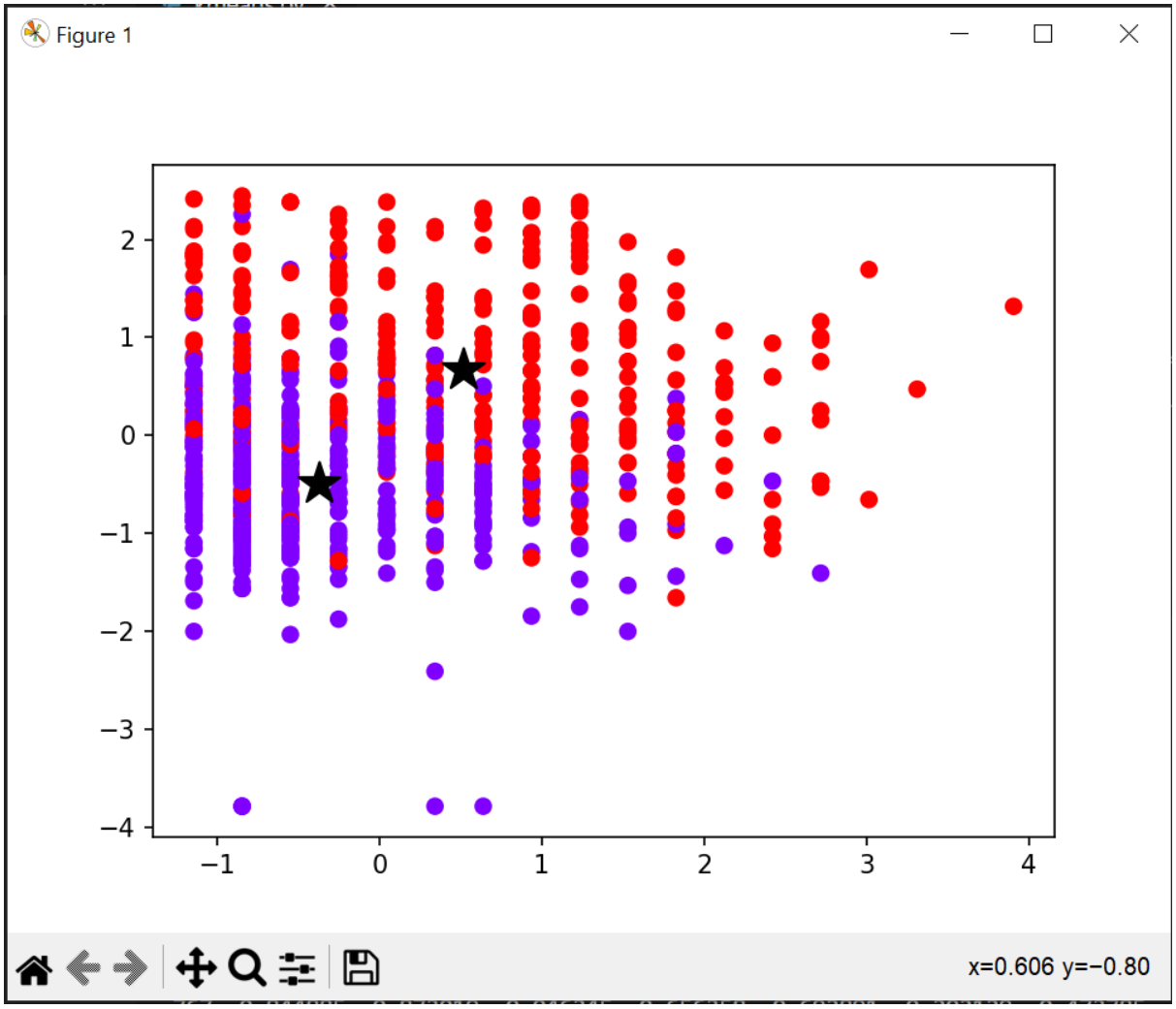


Figure 5.9: Output

5.4.1 L'inertie intra-classe

Figure 5.10: Inertie intra-classe

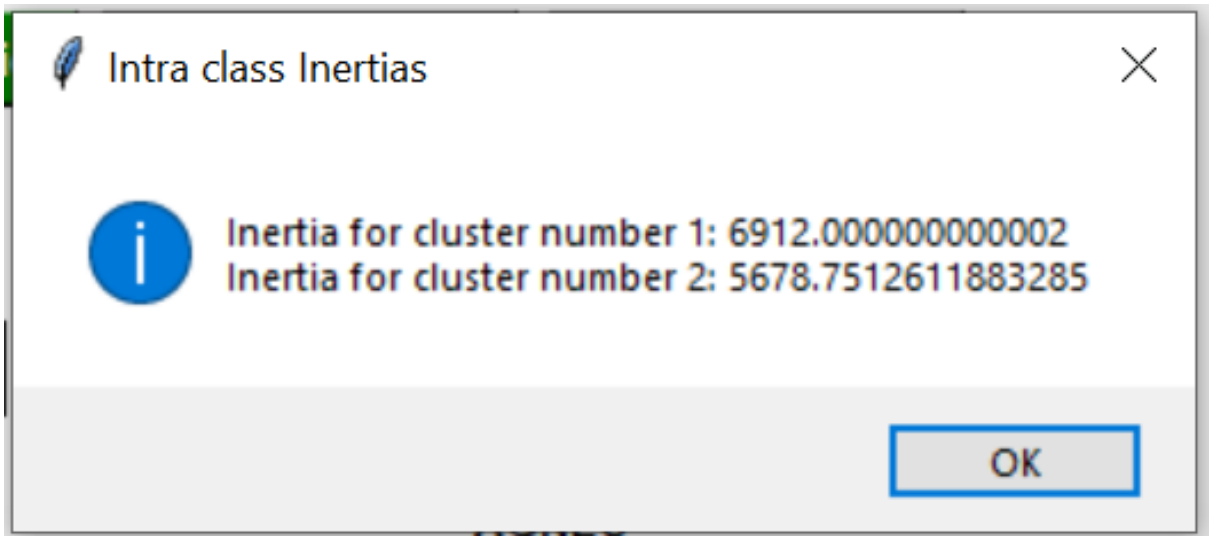


Figure 5.11: Output

5.4.2 L'inertie inter-classe

Figure 5.12: Inertie inter-classe

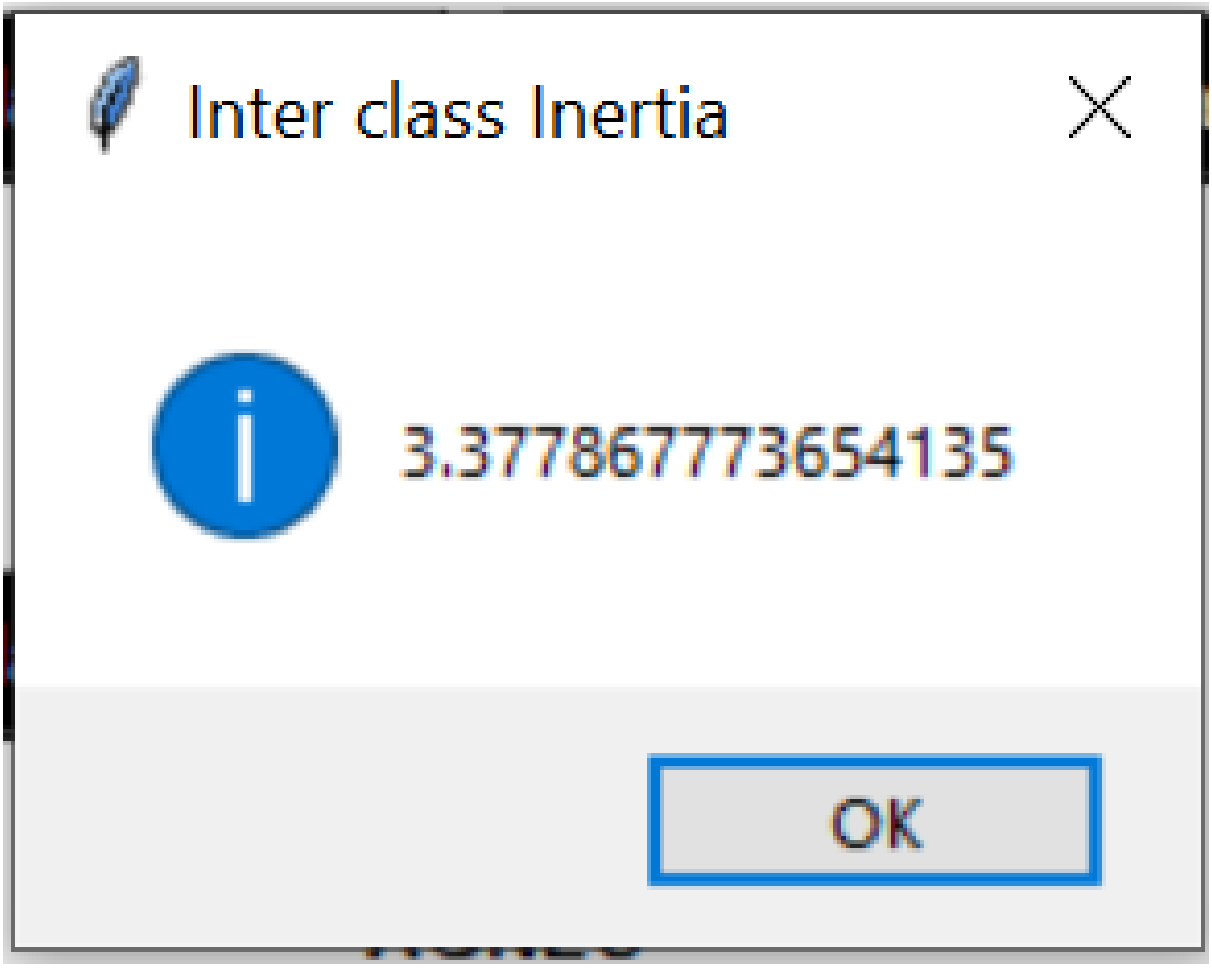


Figure 5.13: Output

5.5 Kmedoids

Figure 5.14: kmedoids

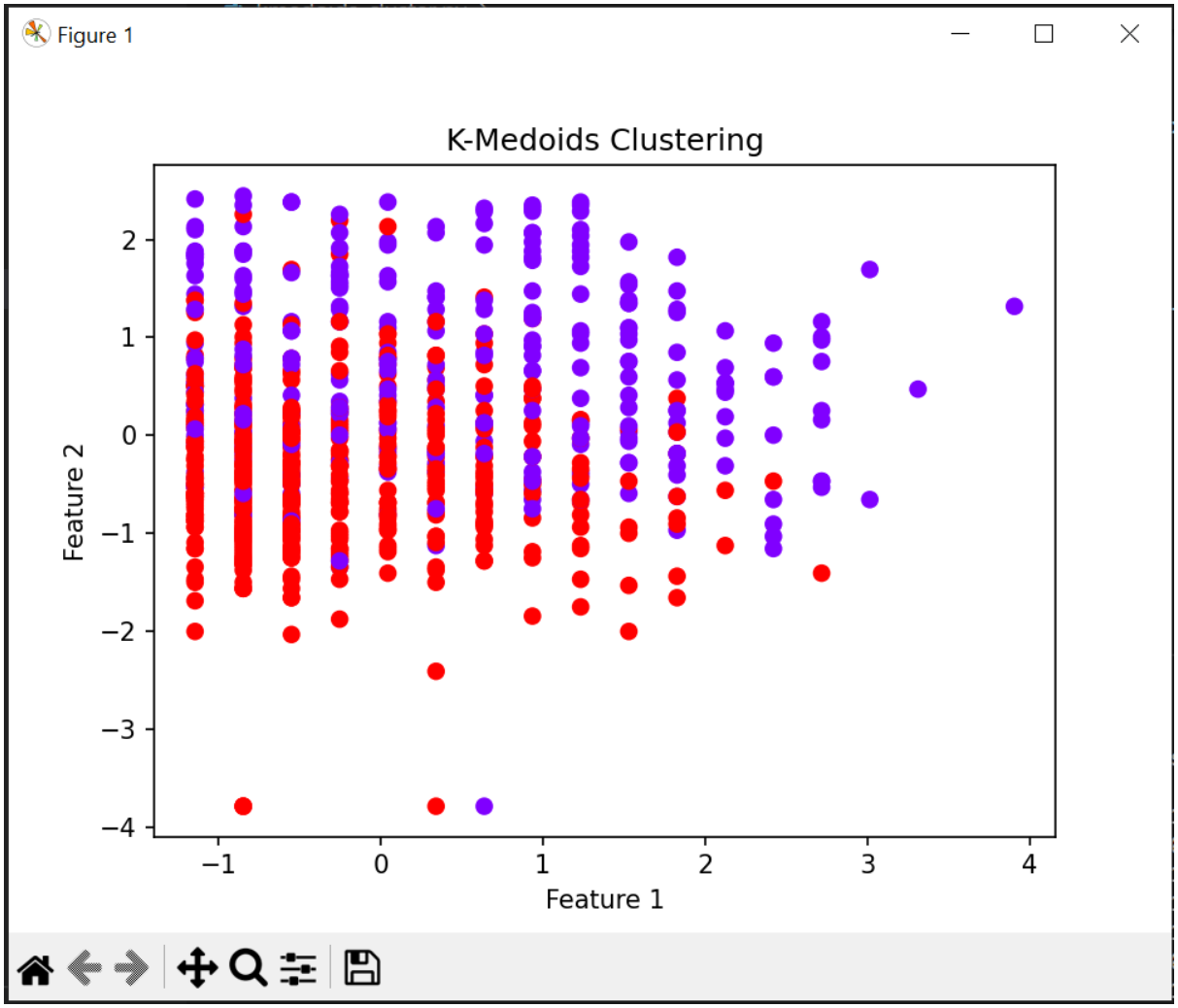


Figure 5.15: Output

5.5.1 L'inertie intra-classe

Figure 5.16: Inertie intra-classe

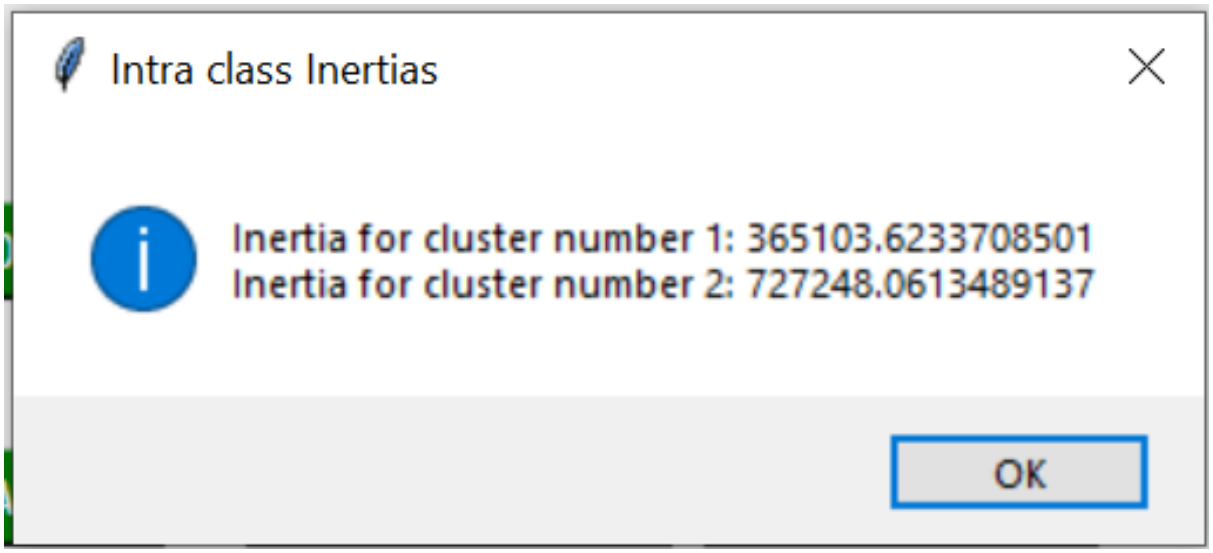


Figure 5.17: Output

5.5.2 L'inertie inter-classe

Figure 5.18: Inertie inter-classe

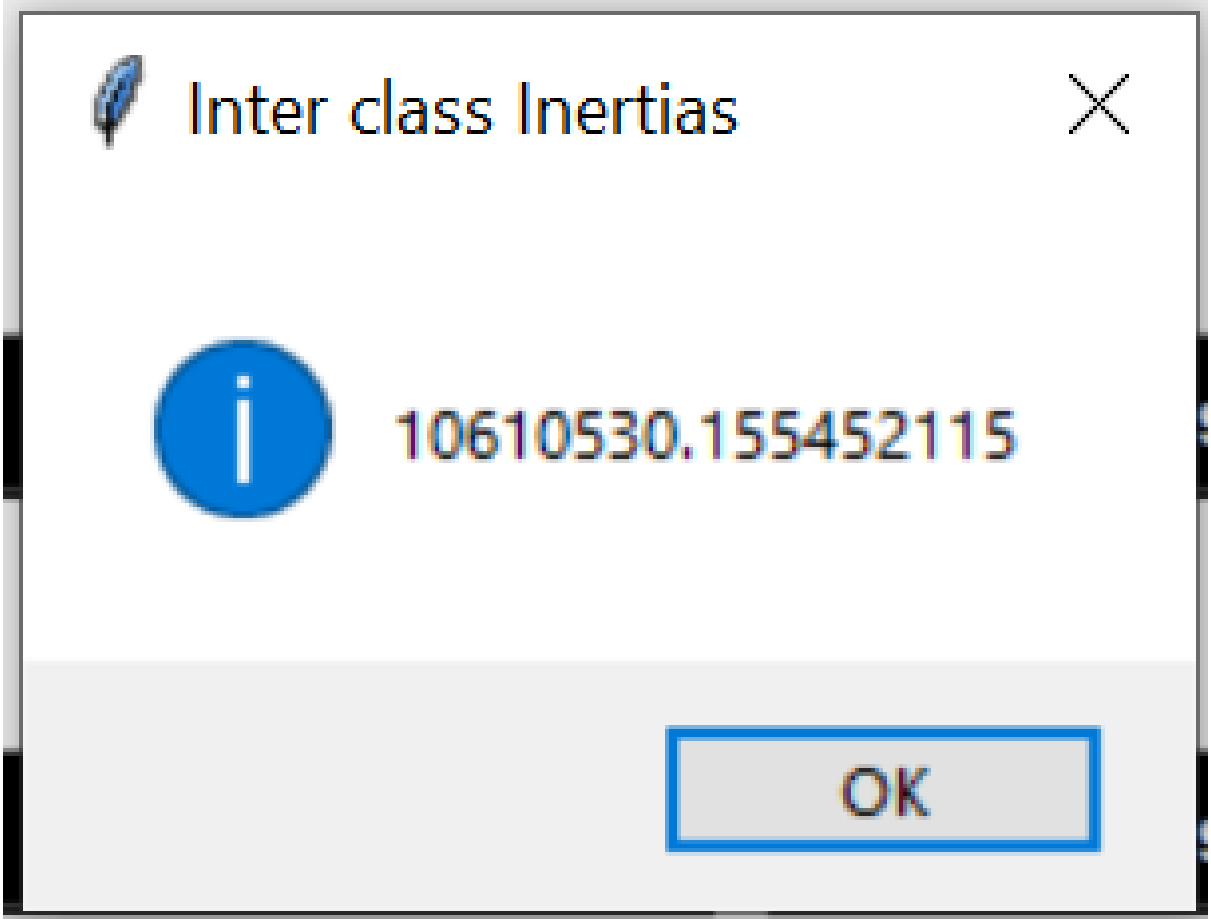


Figure 5.19: Output

5.6 AGNES

Figure 5.20: Agnes

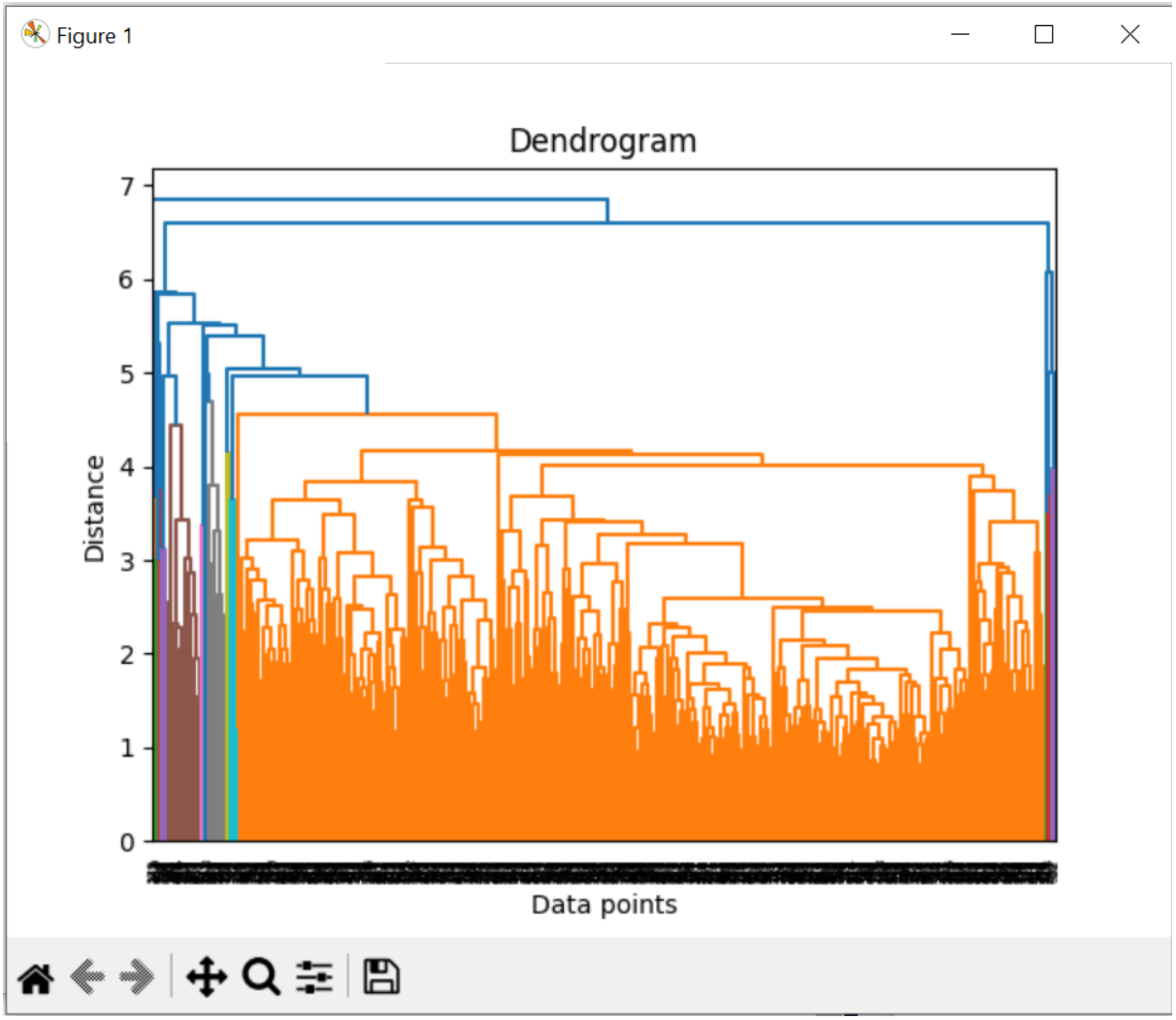


Figure 5.21: Output

5.6.1 L'inertie intra-classe

Figure 5.22: Inertie intra-classe

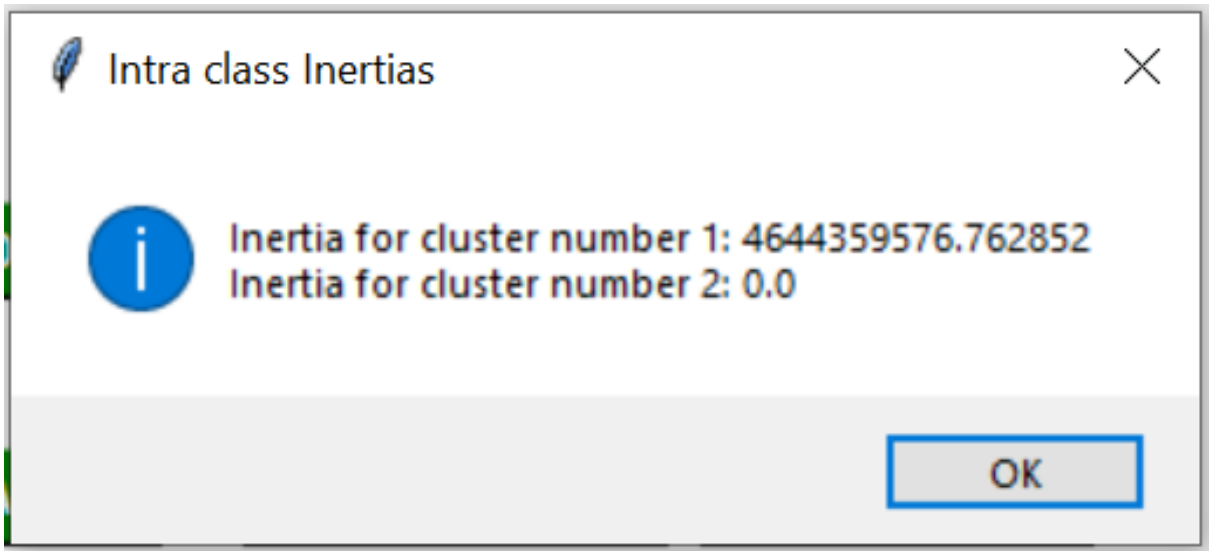


Figure 5.23: Output

5.6.2 Coefficient de silhouette

Figure 5.24: Silhouette

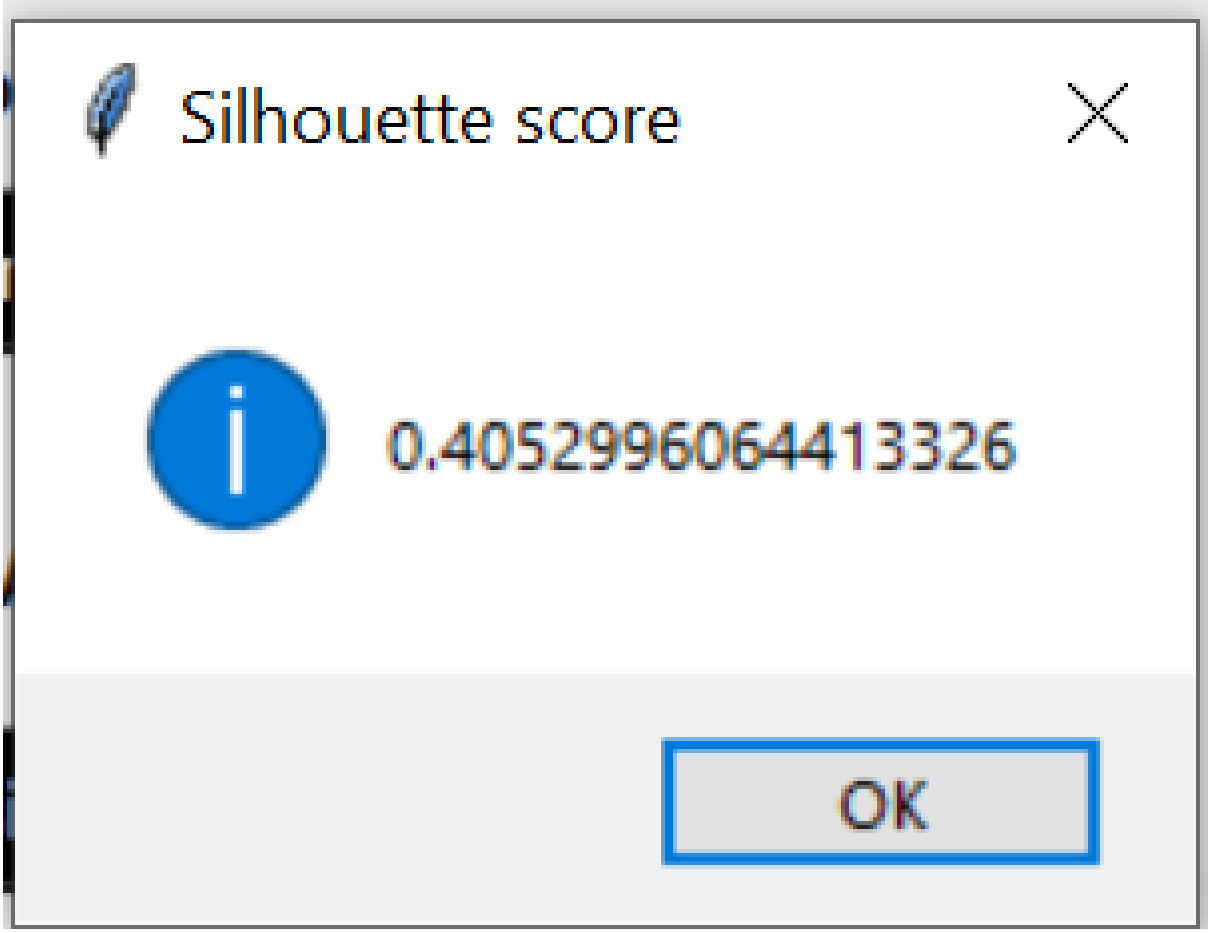


Figure 5.25: Output

5.7 DIANA

Figure 5.26: Diana

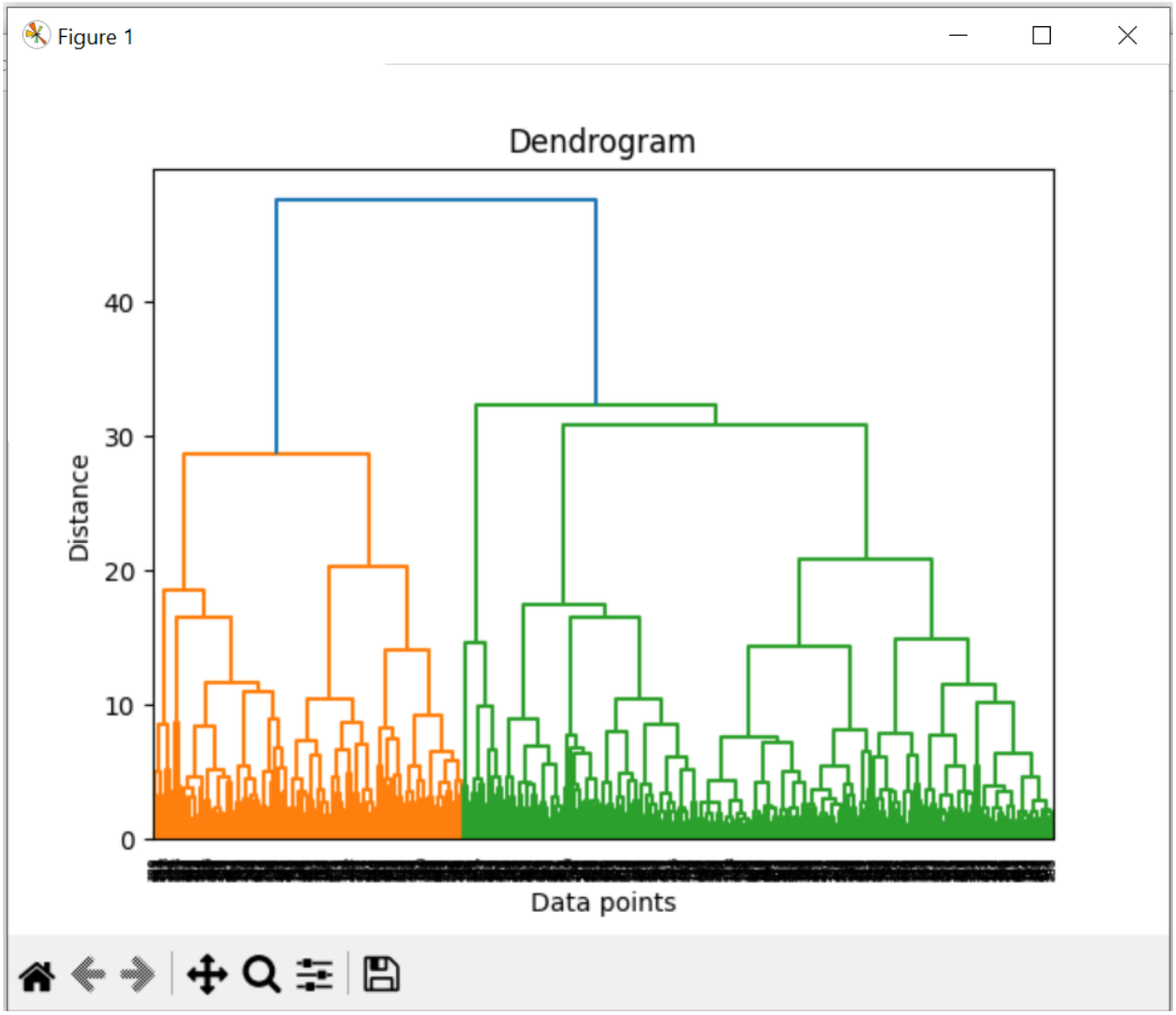


Figure 5.27: Output

5.7.1 L'inertie intra-classe

Figure 5.28: Inertie intra-classe

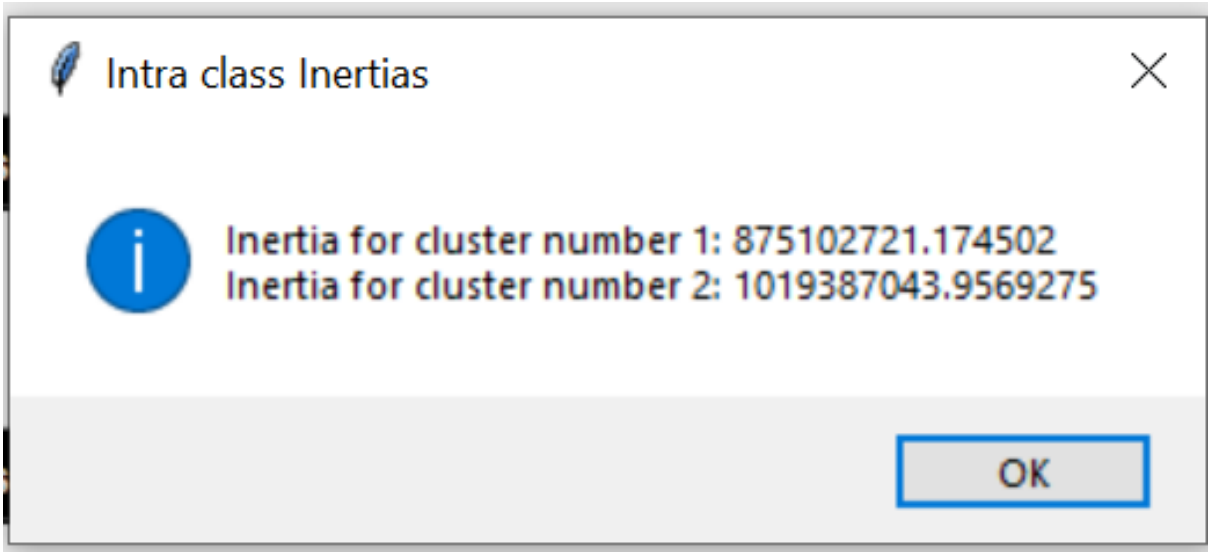


Figure 5.29: Output

5.7.2 Coefficient de silhouette

Figure 5.30: Silhouette

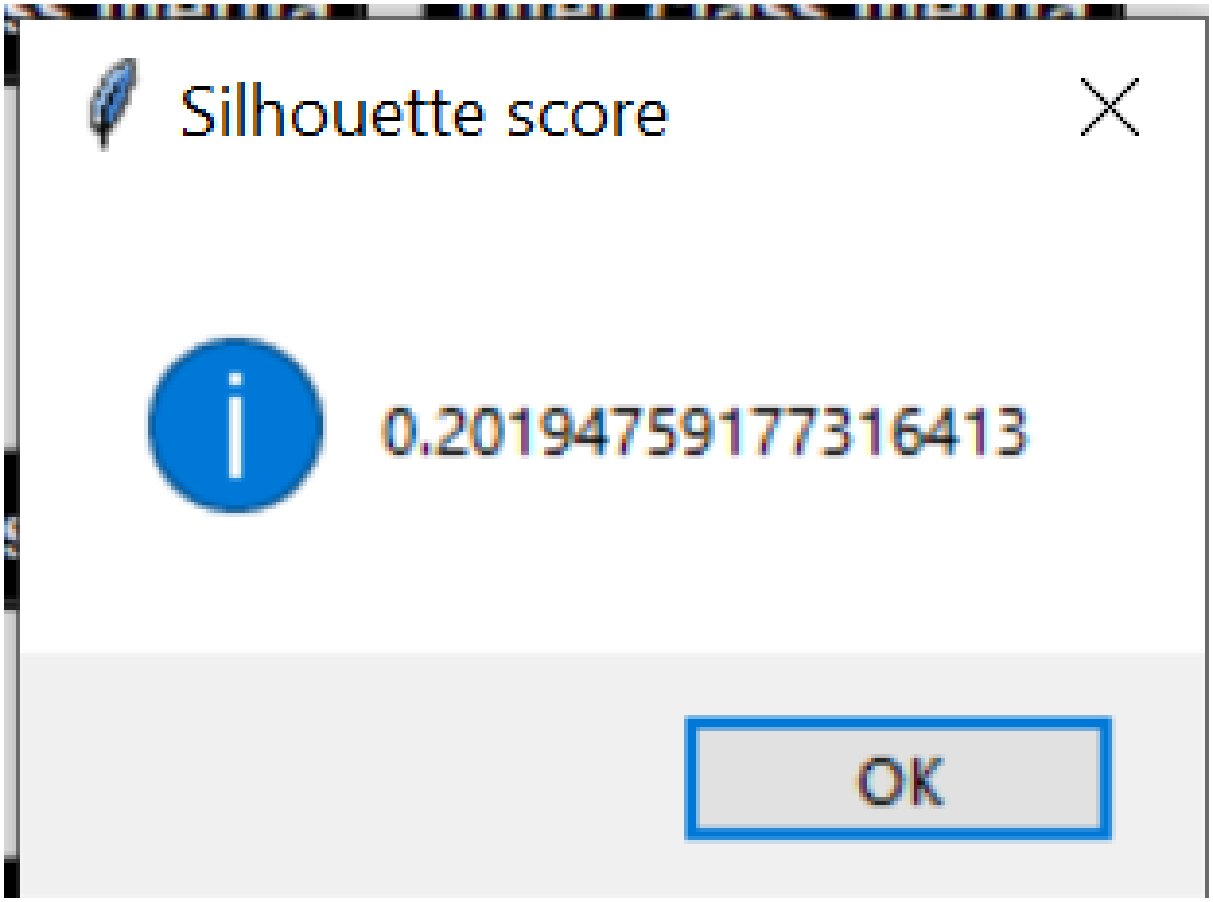


Figure 5.31: Output

5.8 DBSCAN

Figure 5.32: Dbscan

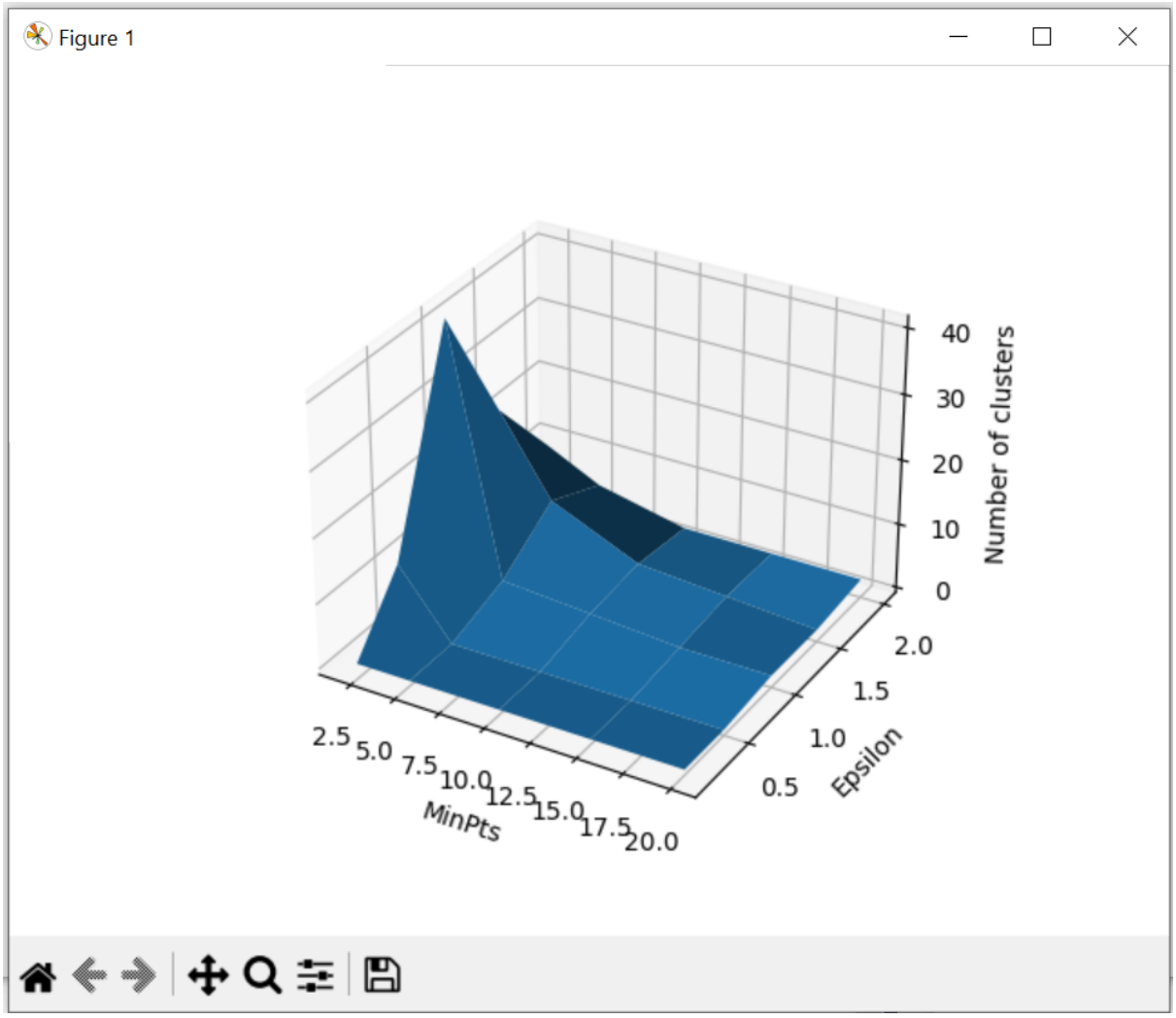


Figure 5.33: Output

5.8.1 Afficher les performanes

Figure 5.34: Performances

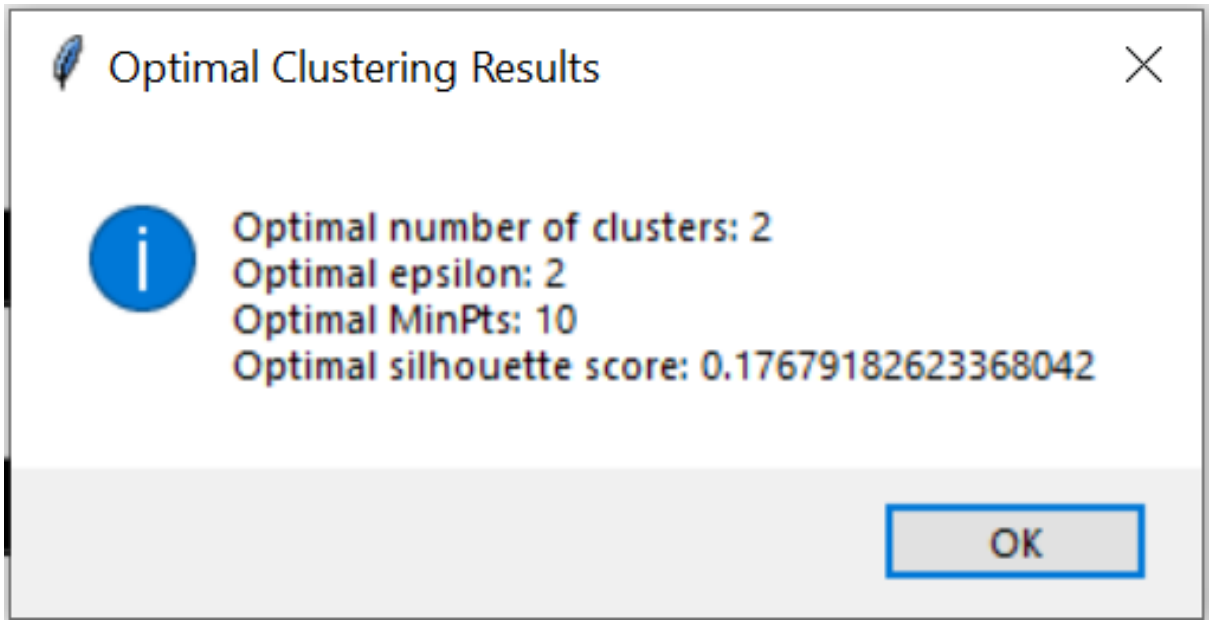


Figure 5.35: Output

Chapter 6

Les resultats obtenus - classification -

6.1 L'importation d'un dataset

Figure 6.1: Importer dataset

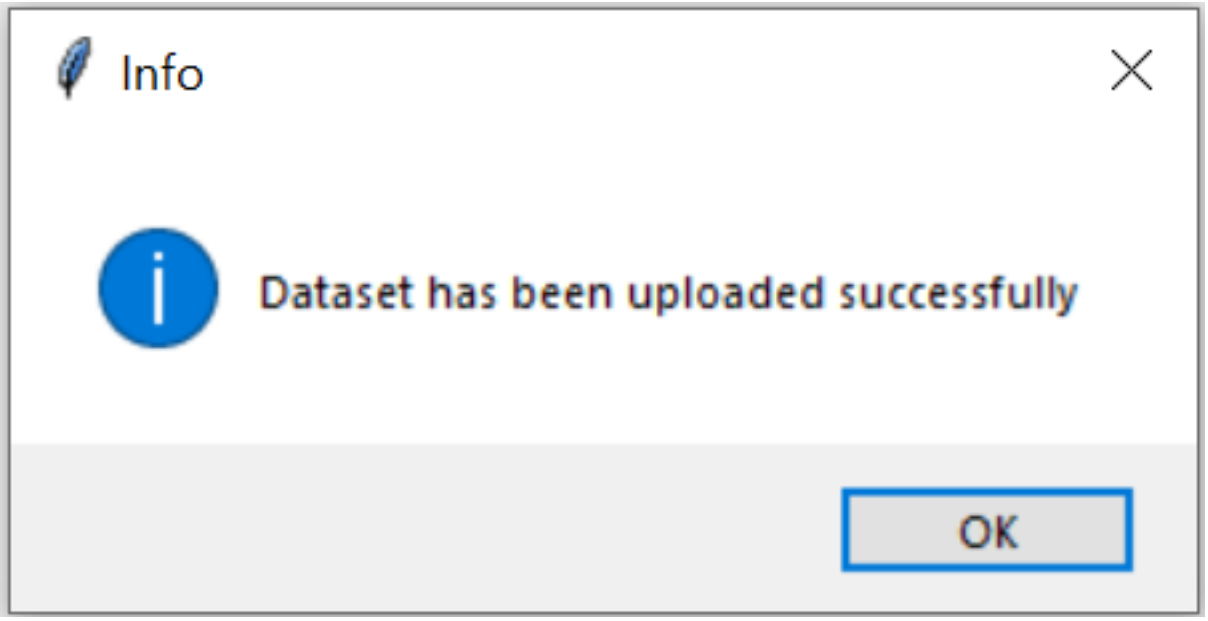


Figure 6.2: Output

6.2 Le preprocessing

Figure 6.3: Preprocessing

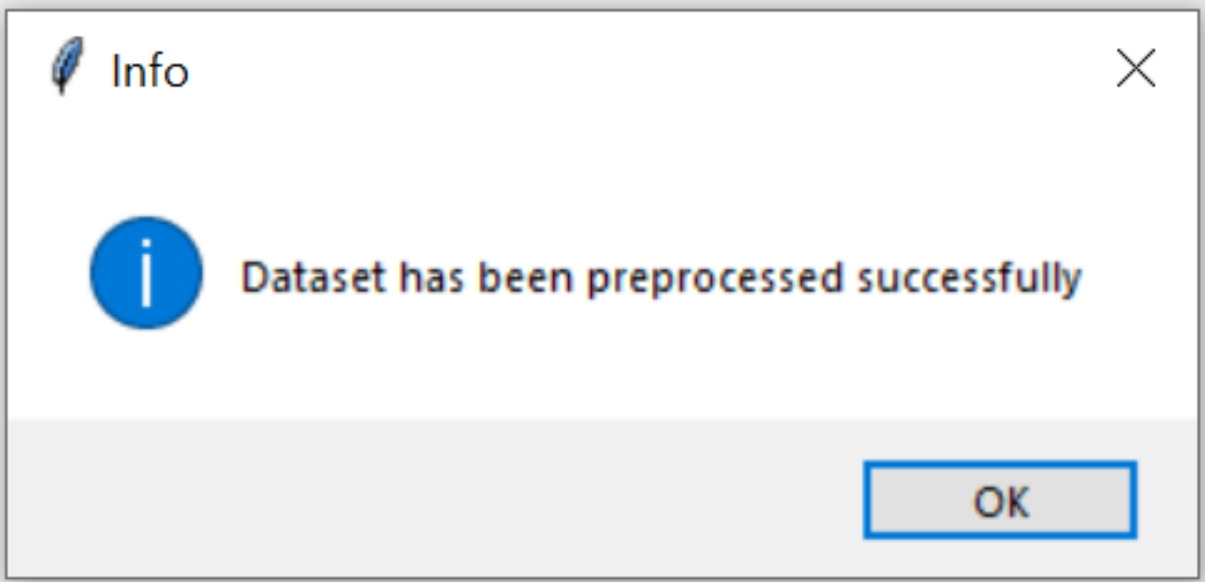
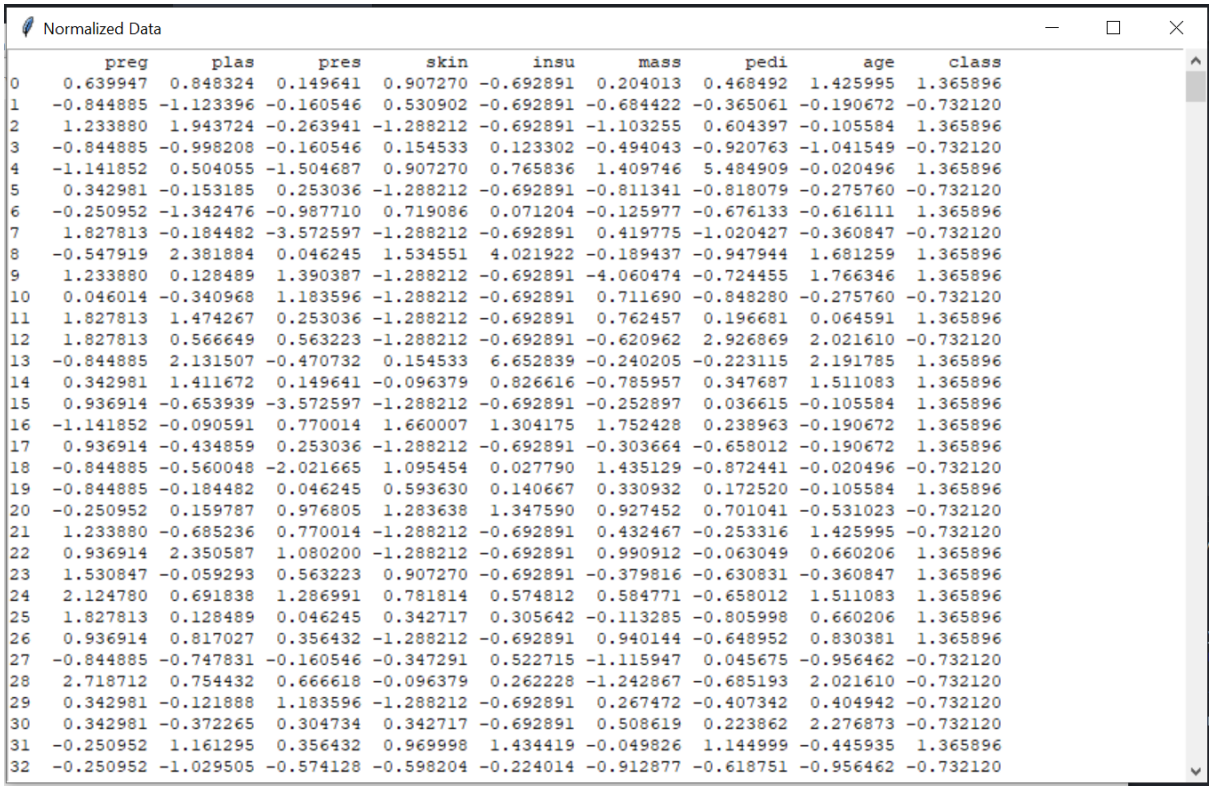


Figure 6.4: Output

6.3 L'affichage du dataset

Figure 6.5: Affichage



	preg	plas	pres	skin	insu	mass	pedi	age	class
0	0.639947	0.848324	0.149641	0.907270	-0.692891	0.204013	0.468492	1.425995	1.365896
1	-0.844885	-1.123396	-0.160546	0.530902	-0.692891	-0.684422	-0.365061	-0.190672	-0.732120
2	1.233880	1.943724	-0.263941	-1.288212	-0.692891	-1.103255	0.604397	-0.105584	1.365896
3	-0.844885	-0.998208	-0.160546	0.154533	0.123302	-0.494043	-0.920763	-1.041549	-0.732120
4	-1.141852	0.504055	-1.504687	0.907270	0.765836	1.409746	5.484909	-0.020496	1.365896
5	0.342981	-0.153185	0.253036	-1.288212	-0.692891	-0.811341	-0.818079	-0.275760	-0.732120
6	-0.250952	-1.342476	-0.987710	0.719086	0.071204	-0.125977	-0.676133	-0.616111	1.365896
7	1.827813	-0.184482	-3.572597	-1.288212	-0.692891	0.419775	-1.020427	-0.360847	-0.732120
8	-0.547919	2.381884	0.046245	1.534551	4.021922	-0.189437	-0.947944	1.681259	1.365896
9	1.233880	0.128489	1.390387	-1.288212	-0.692891	-4.060474	-0.724455	1.766346	1.365896
10	0.046014	-0.340968	1.183596	-1.288212	-0.692891	0.711690	-0.848280	-0.275760	-0.732120
11	1.827813	1.474267	0.253036	-1.288212	-0.692891	0.762457	0.196681	0.064591	1.365896
12	1.827813	0.566649	0.563223	-1.288212	-0.692891	-0.620962	2.926869	2.021610	-0.732120
13	-0.844885	2.131507	-0.470732	0.154533	6.652839	-0.240205	-0.223115	2.191785	1.365896
14	0.342981	1.411672	0.149641	-0.096379	0.826616	-0.785957	0.347687	1.511083	1.365896
15	0.936914	-0.653939	-3.572597	-1.288212	-0.692891	-0.252897	0.036615	-0.105584	1.365896
16	-1.141852	-0.090591	0.770014	1.660007	1.304175	1.752428	0.238963	-0.190672	1.365896
17	0.936914	-0.434859	0.253036	-1.288212	-0.692891	-0.303664	-0.658012	-0.190672	1.365896
18	-0.844885	-0.560048	-2.021665	1.095454	0.027790	1.435129	-0.872441	-0.020496	-0.732120
19	-0.844885	-0.184482	0.046245	0.593630	0.140667	0.330932	0.172520	-0.105584	1.365896
20	-0.250952	0.159787	0.976805	1.283638	1.347590	0.927452	0.701041	-0.531023	-0.732120
21	1.233880	-0.685236	0.770014	-1.288212	-0.692891	0.432467	-0.253316	1.425995	-0.732120
22	0.936914	2.350587	1.080200	-1.288212	-0.692891	0.990912	-0.063049	0.660206	1.365896
23	1.530847	-0.059293	0.563223	0.907270	-0.692891	-0.379816	-0.630831	-0.360847	1.365896
24	2.124780	0.691838	1.286991	0.781814	0.574812	0.584771	-0.658012	1.511083	1.365896
25	1.827813	0.128489	0.046245	0.342717	0.305642	-0.113285	-0.805998	0.660206	1.365896
26	0.936914	0.817027	0.356432	-1.288212	-0.692891	0.940144	-0.648952	0.830381	1.365896
27	-0.844885	-0.747831	-0.160546	-0.347291	0.522715	-1.115947	0.045675	-0.956462	-0.732120
28	2.718712	0.754432	0.666618	-0.096379	0.262228	-1.242867	-0.685193	2.021610	-0.732120
29	0.342981	-0.121888	1.183596	-1.288212	-0.692891	0.267472	-0.407342	0.404942	-0.732120
30	0.342981	-0.372265	0.304734	0.342717	-0.692891	0.508619	0.223862	2.276873	-0.732120
31	-0.250952	1.161295	0.356432	0.969998	1.434419	-0.049826	1.144999	-0.445935	1.365896
32	-0.250952	-1.029505	-0.574128	-0.598204	-0.224014	-0.912877	-0.618751	-0.956462	-0.732120

Figure 6.6: Output

6.4 K-Nearest Neighbors (KNN)

Figure 6.7: Apply KNN

6.4.1 Confusion matrix

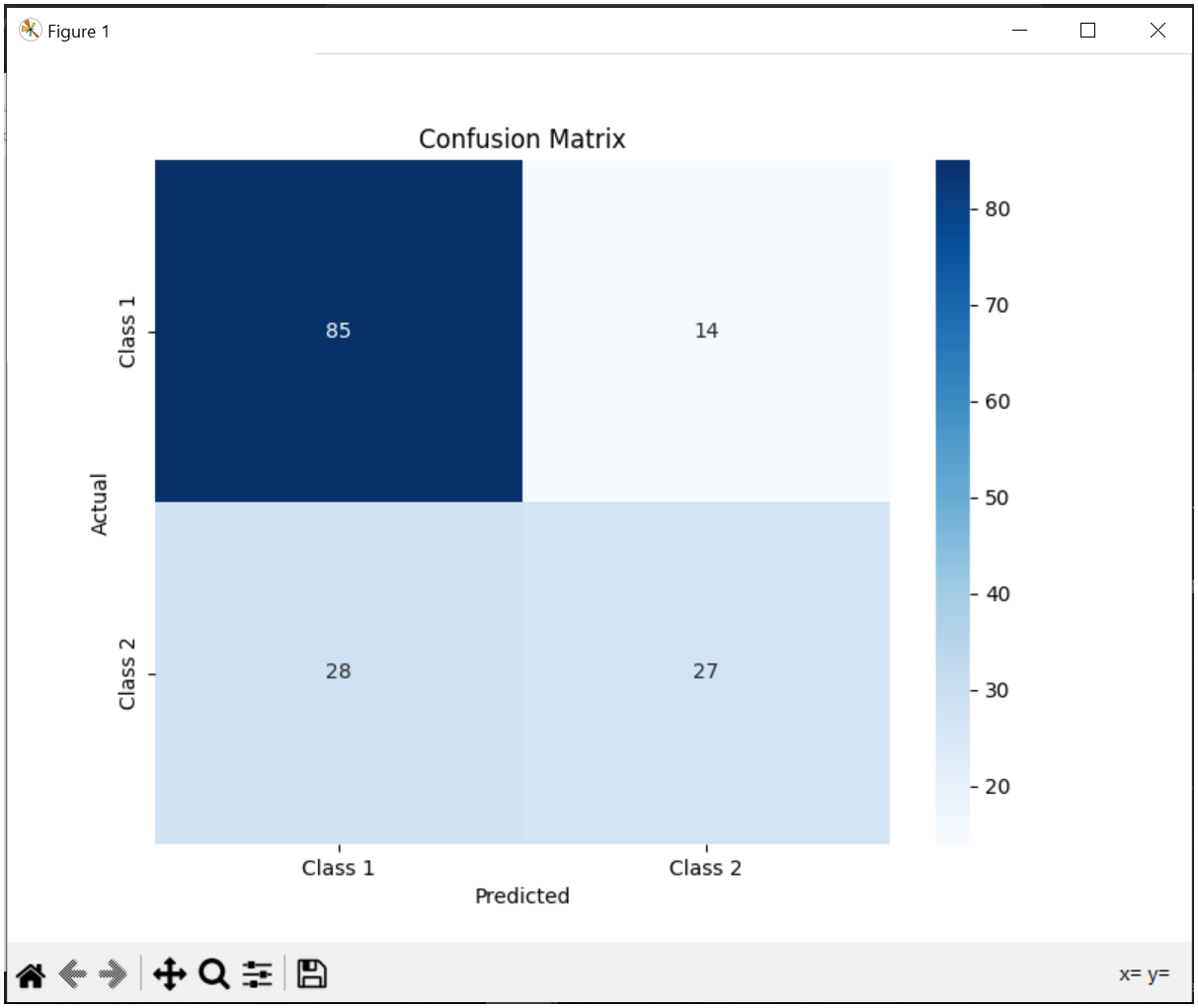


Figure 6.8: Confusion matrix KNN

6.4.2 Accuracy

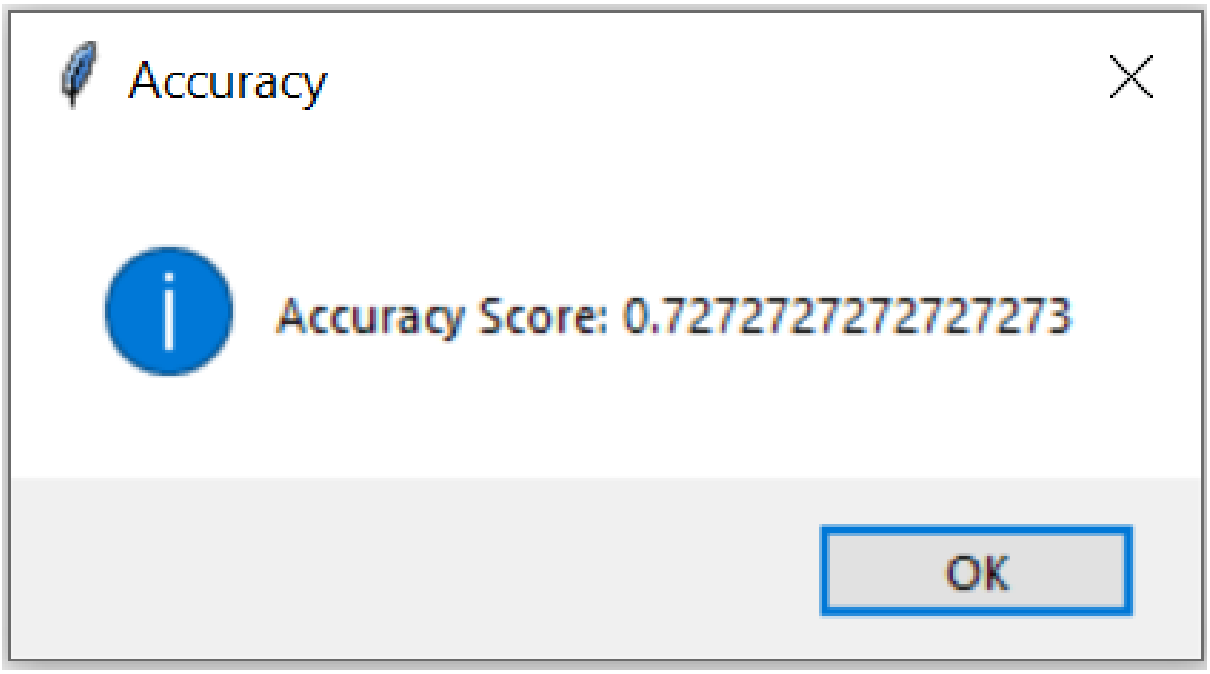


Figure 6.9: Accuracy KNN

6.4.3 F1 score

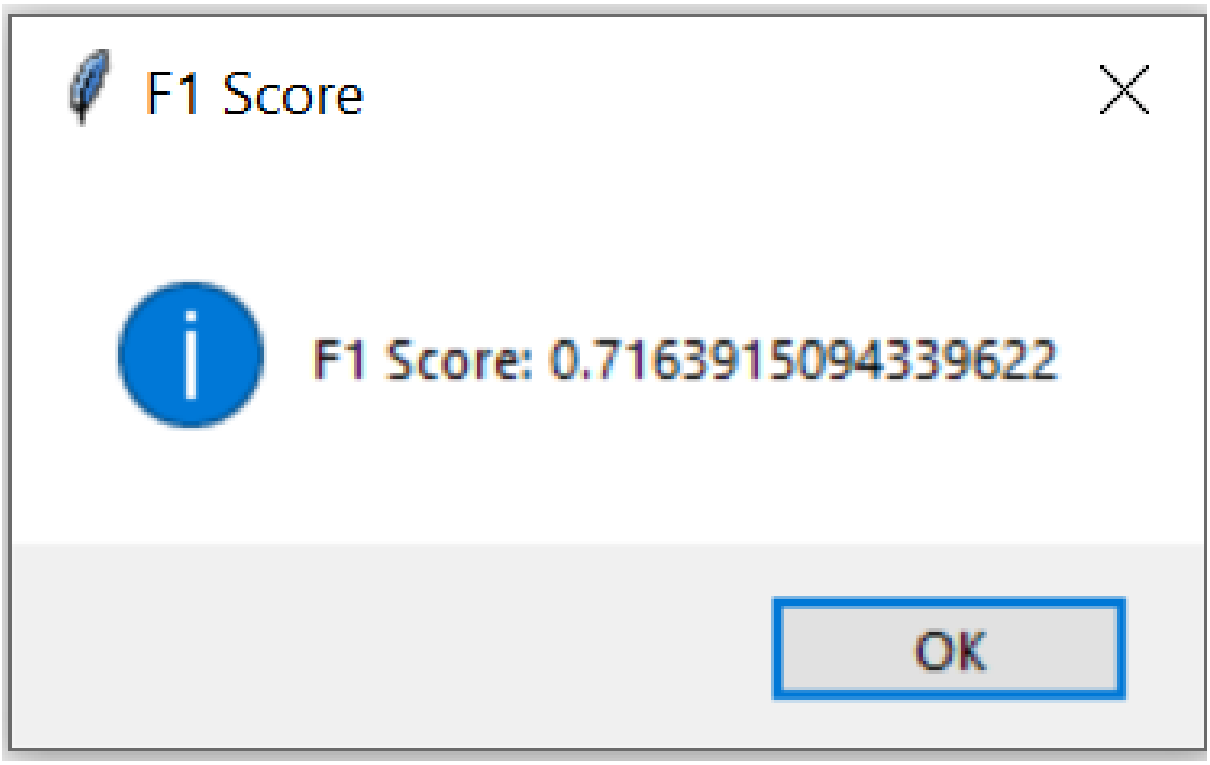


Figure 6.10: F1 score KNN

6.5 Naive Bayes

Figure 6.11: Apply Naive Bayes

6.5.1 Confusion matrix

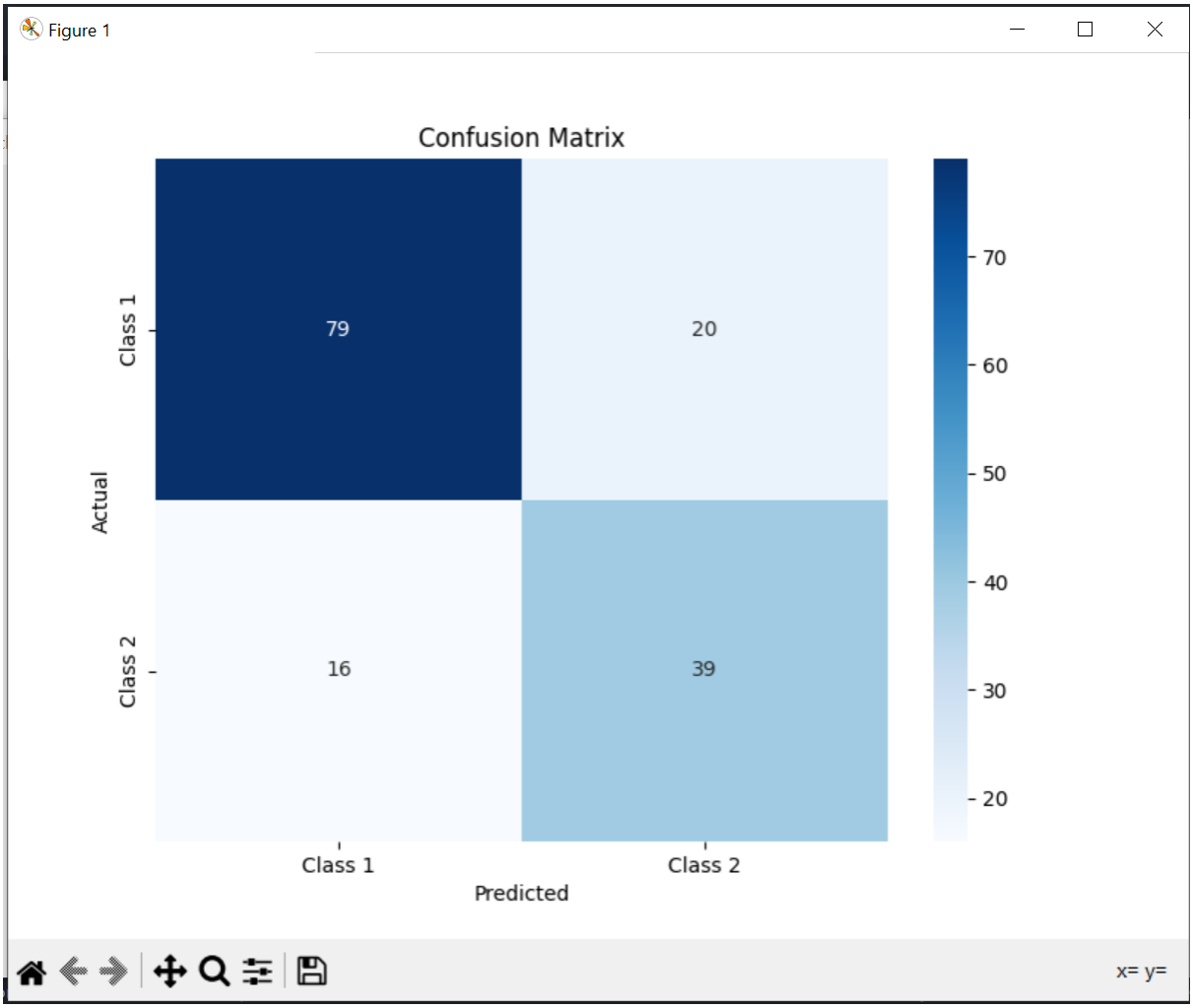


Figure 6.12: Confusion matrix Naive Bayes

6.5.2 Accuracy

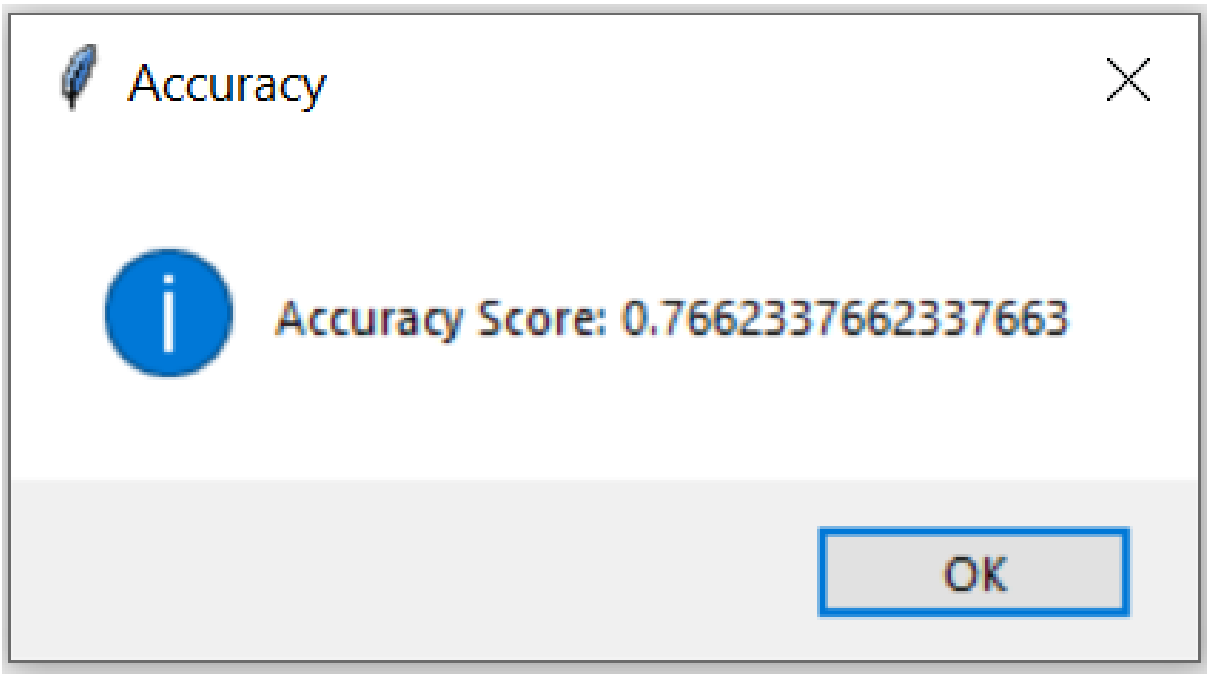


Figure 6.13: Accuracy Naive Bayes

6.5.3 F1 score

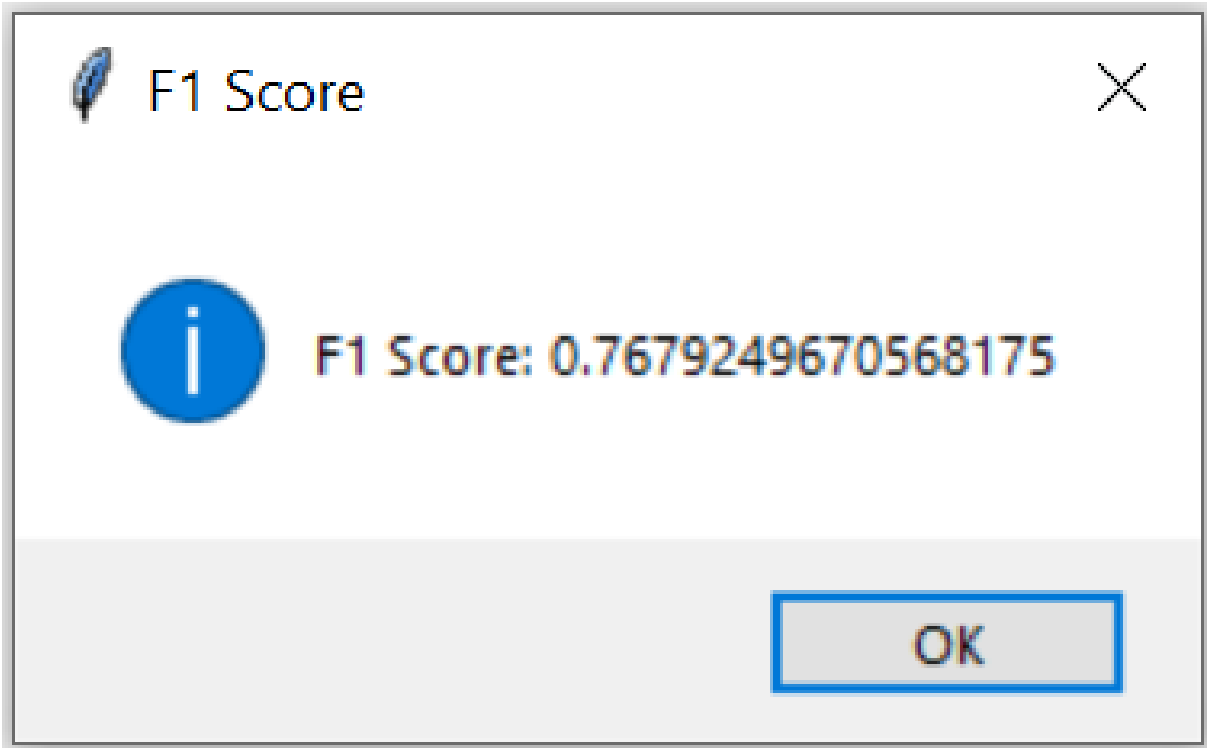


Figure 6.14: F1 score Naive Bayes

6.6 Arbre de décision

Figure 6.15: Apply Decision tree

6.6.1 Generated tree

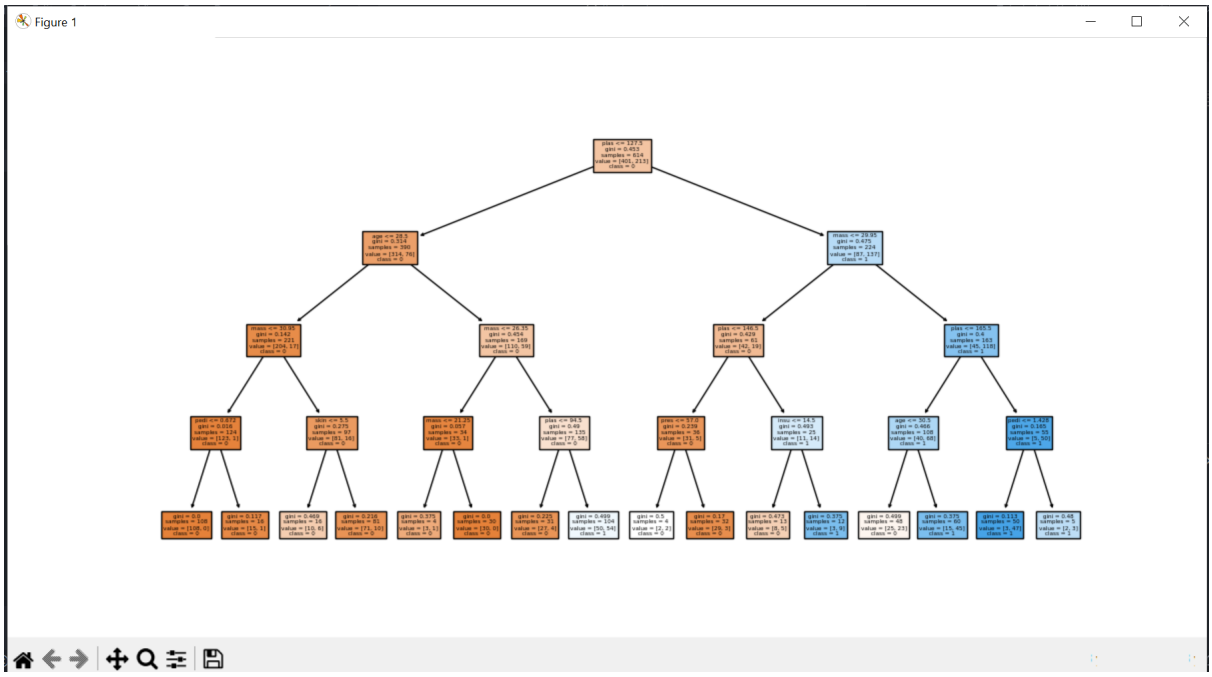


Figure 6.16: Generated tree

6.6.2 Confusion matrix

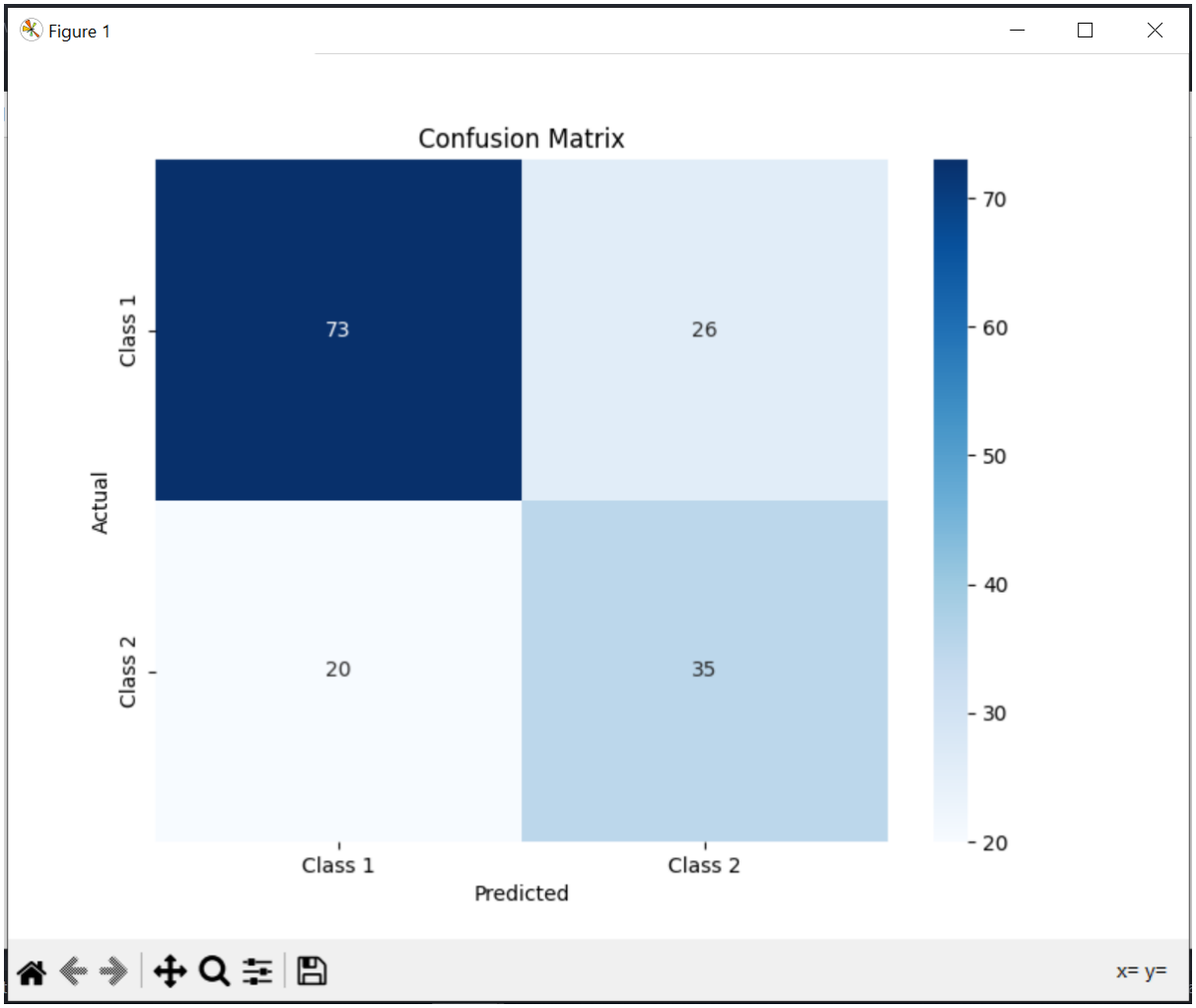


Figure 6.17: Confusion matrix Decision tree

6.6.3 Accuracy

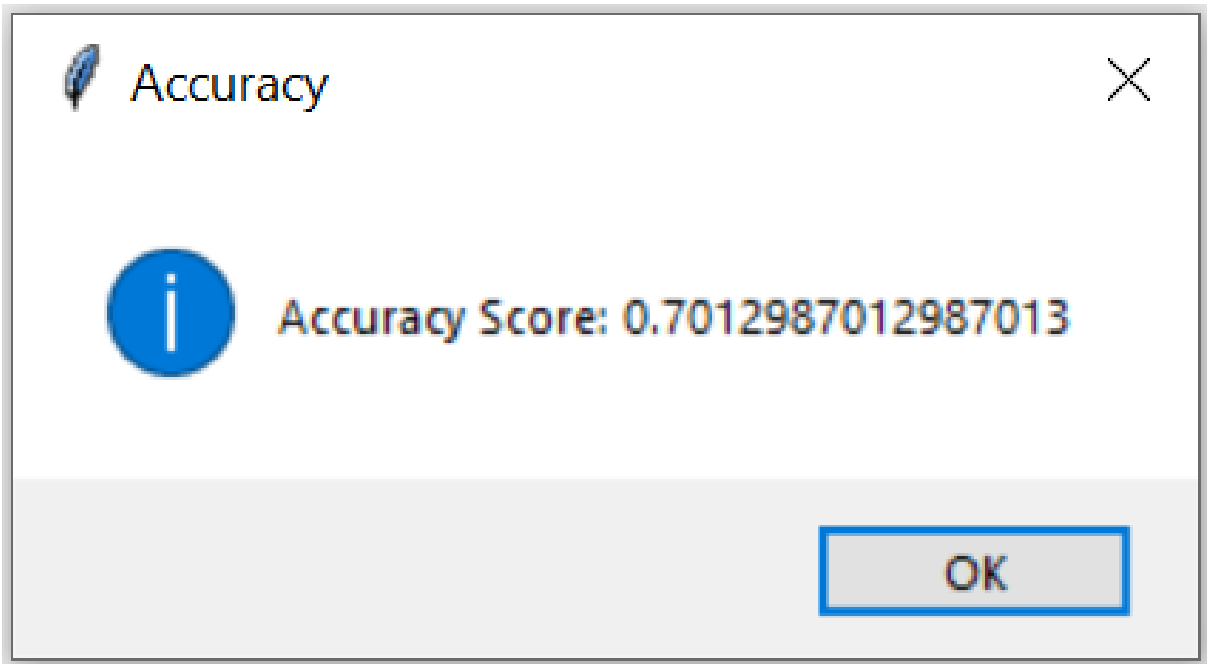


Figure 6.18: Accuracy Decision tree

6.6.4 F1 score

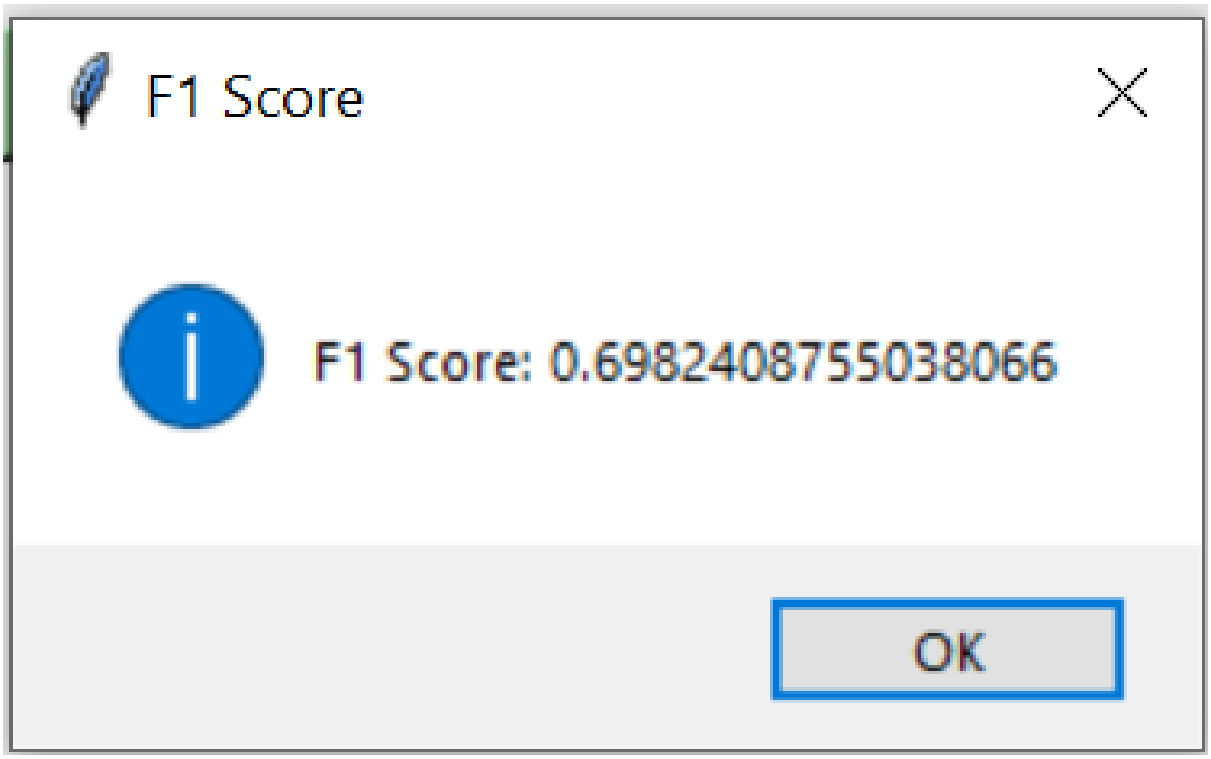


Figure 6.19: F1 score Decision tree

6.7 Machine à vecteurs de support (SVM)

Figure 6.20: Apply SVM

6.7.1 Report

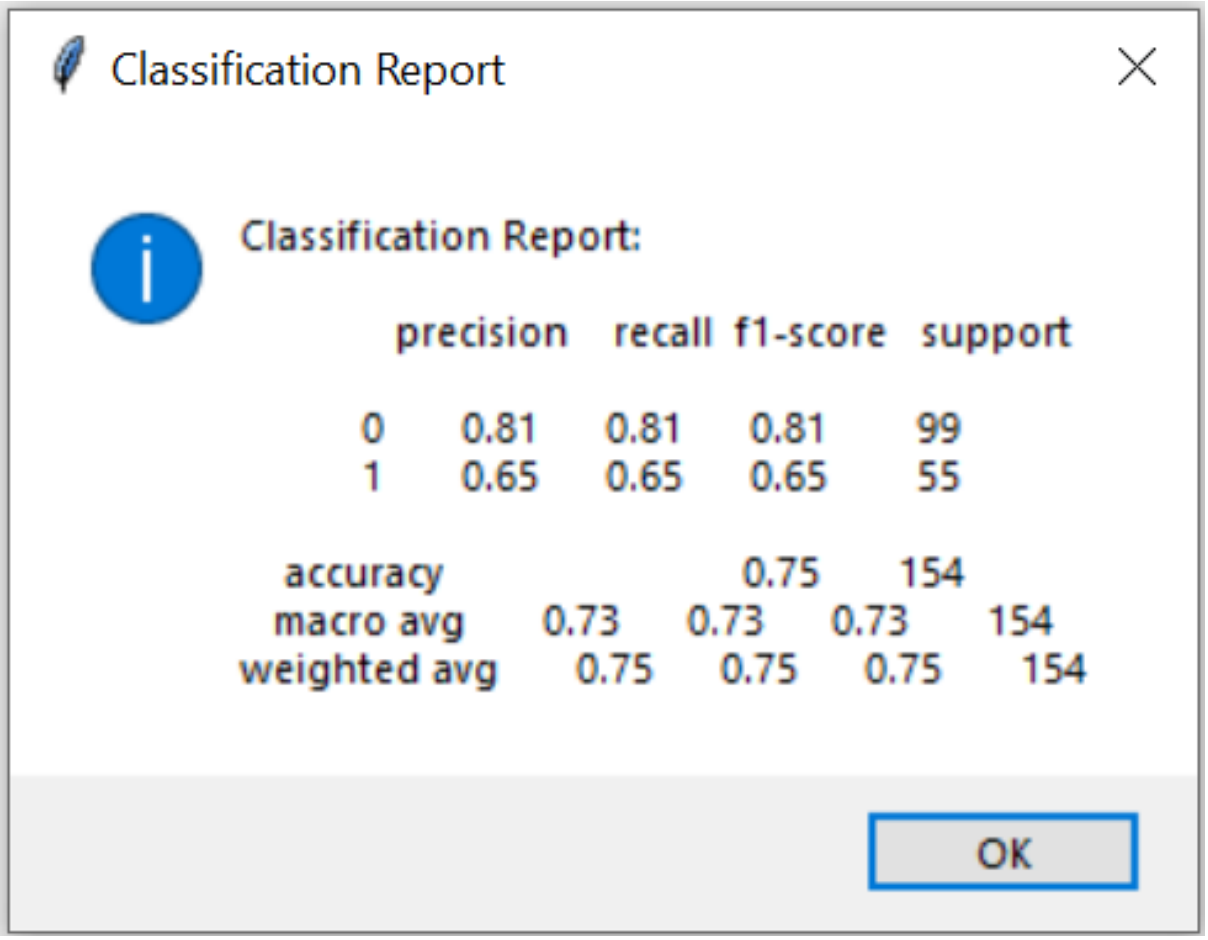


Figure 6.21: Report SVM

6.7.2 Confusion matrix

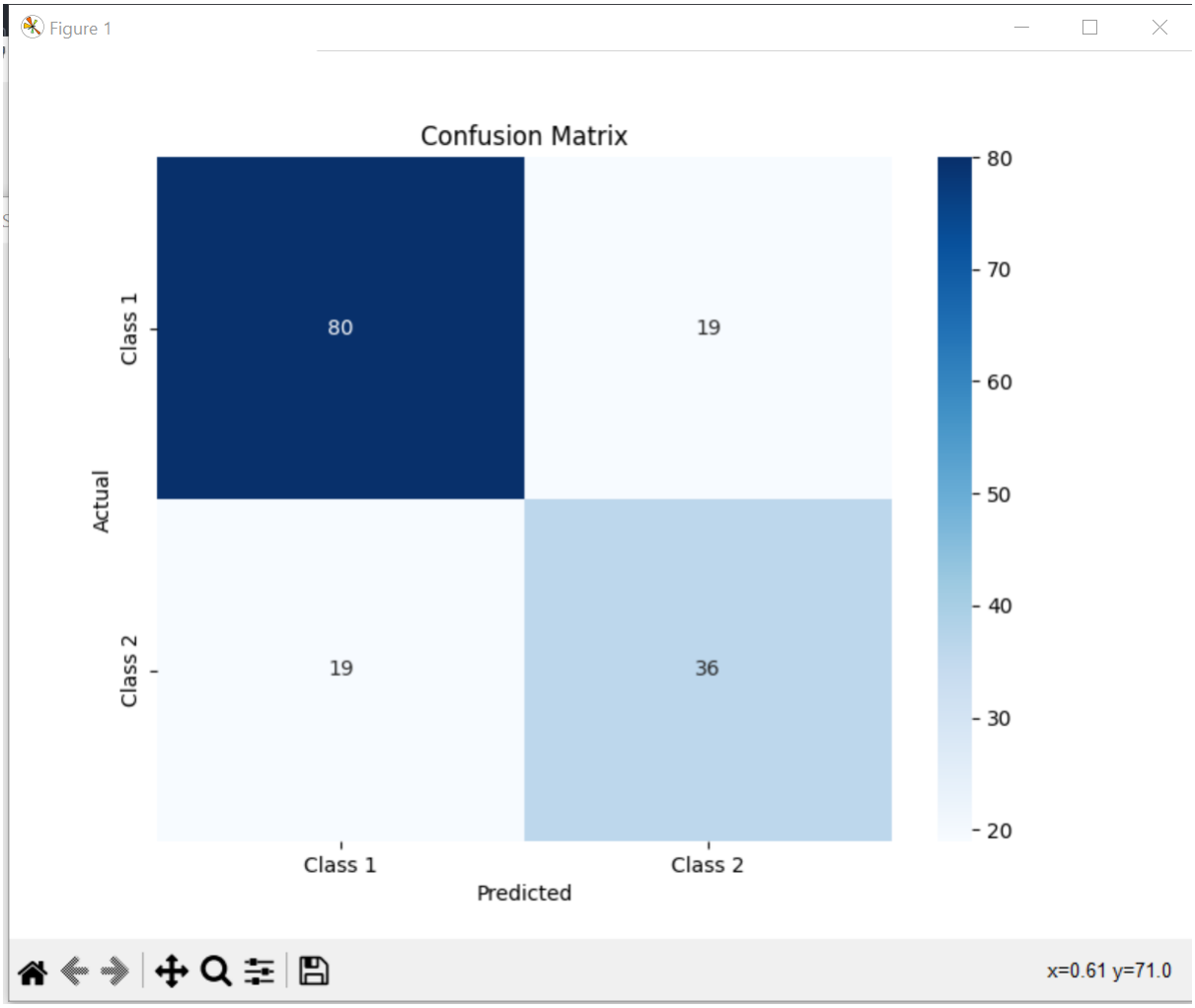


Figure 6.22: Confusion matrix SVM

6.7.3 Accuracy

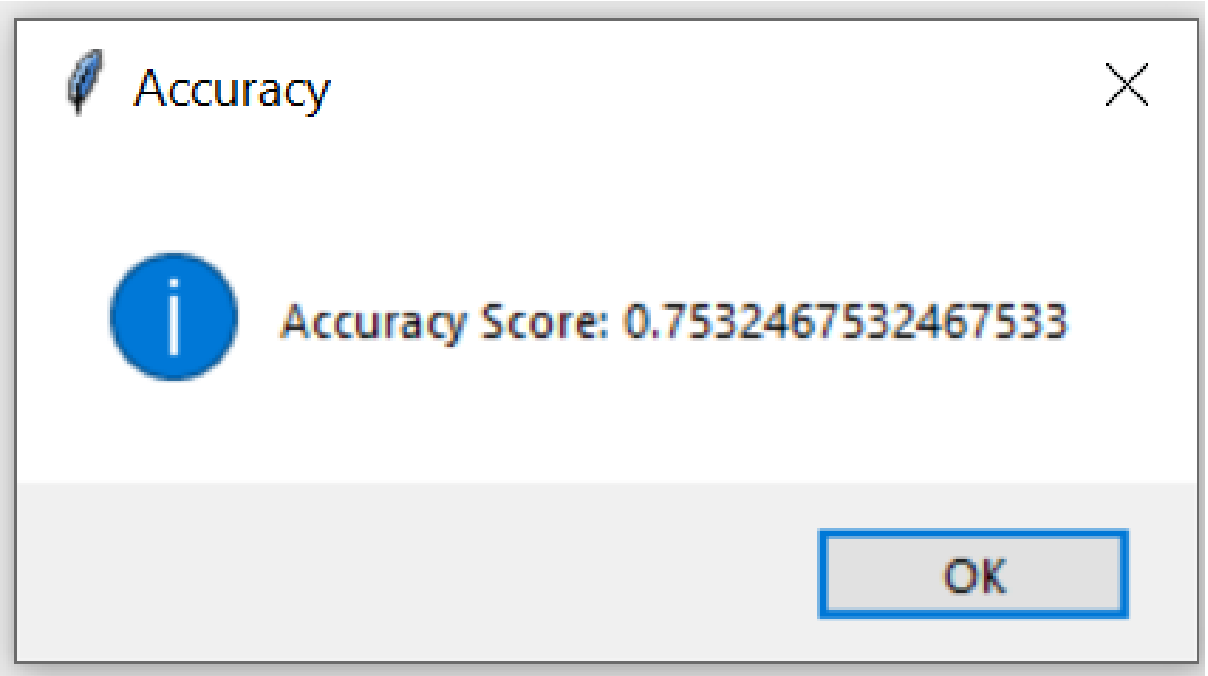


Figure 6.23: Accuracy SVM

6.7.4 F1 score

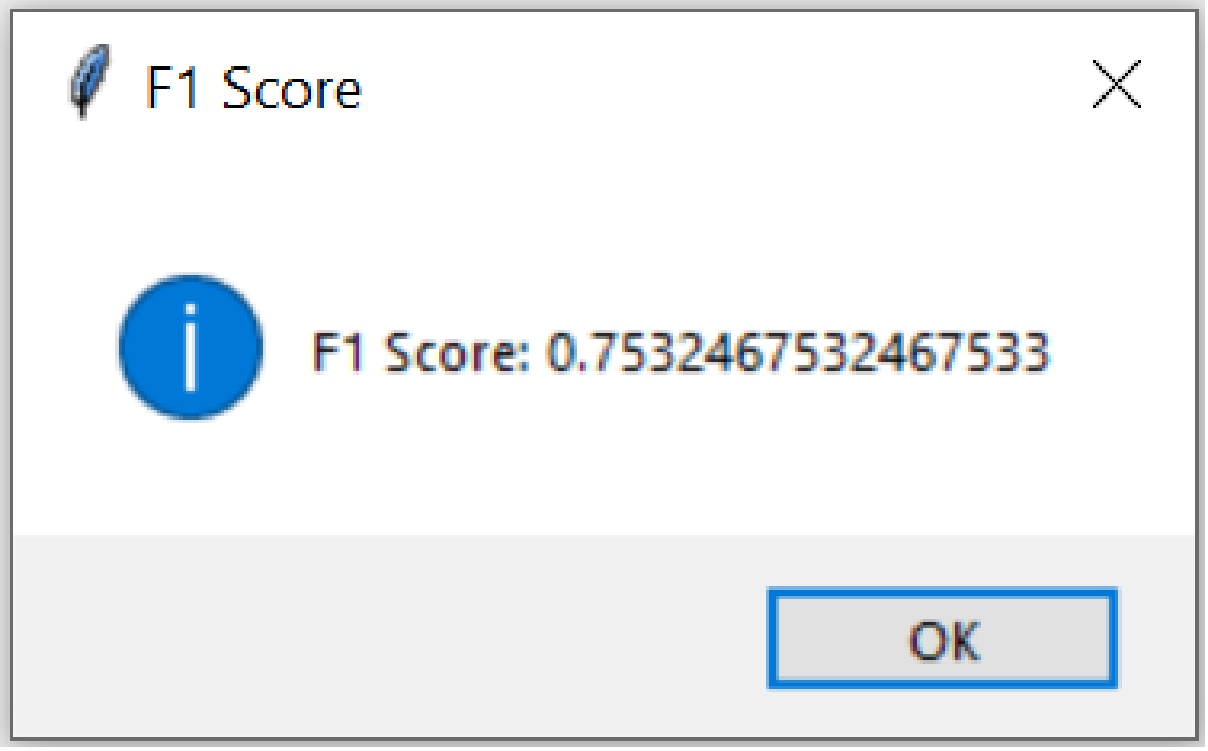


Figure 6.24: F1 score SVM

6.8 Réseau de neuronnes (NN)

Figure 6.25: Apply NN

6.8.1 Confusion matrix

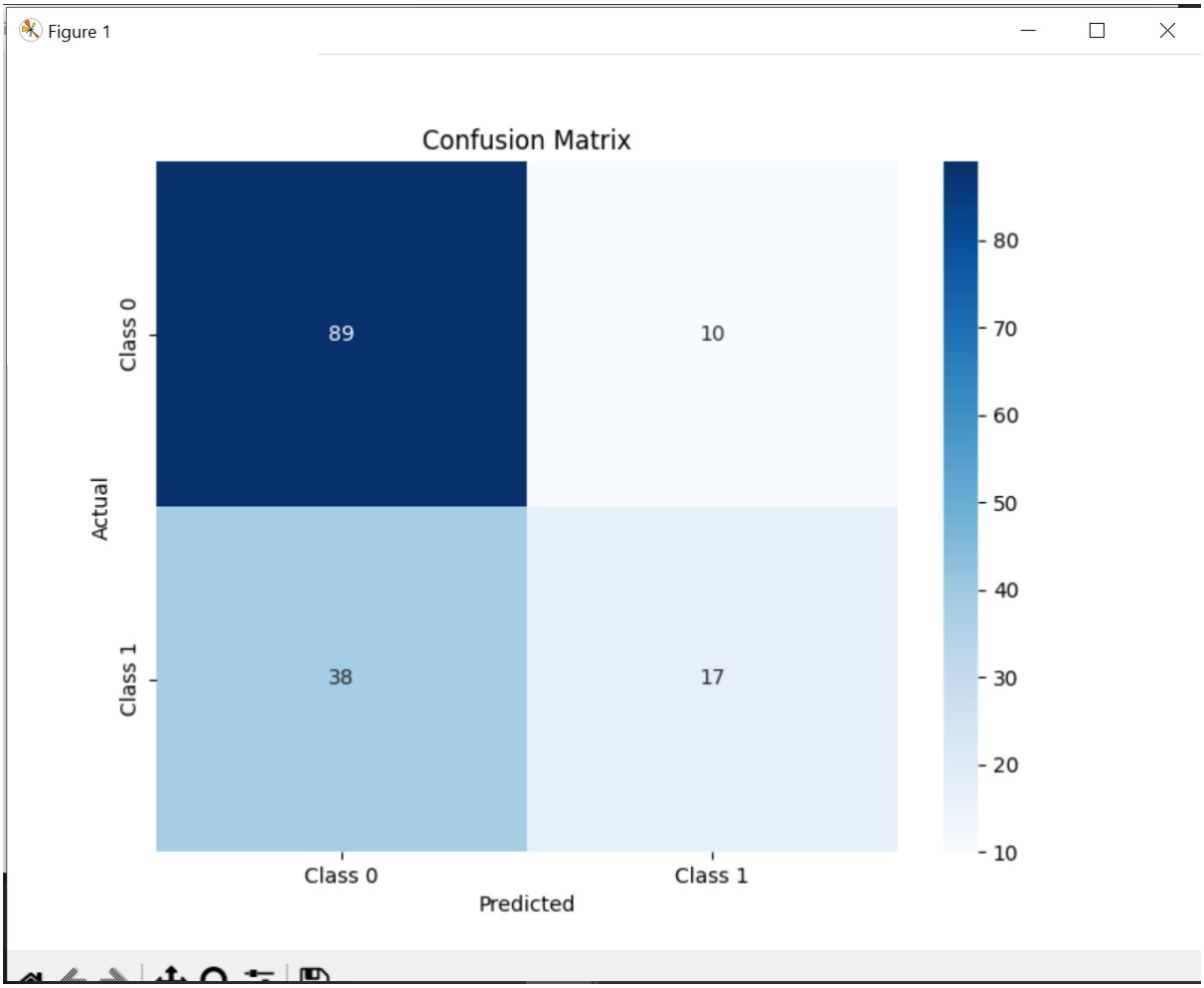


Figure 6.26: Confusion matrix NN

6.8.2 Accuracy

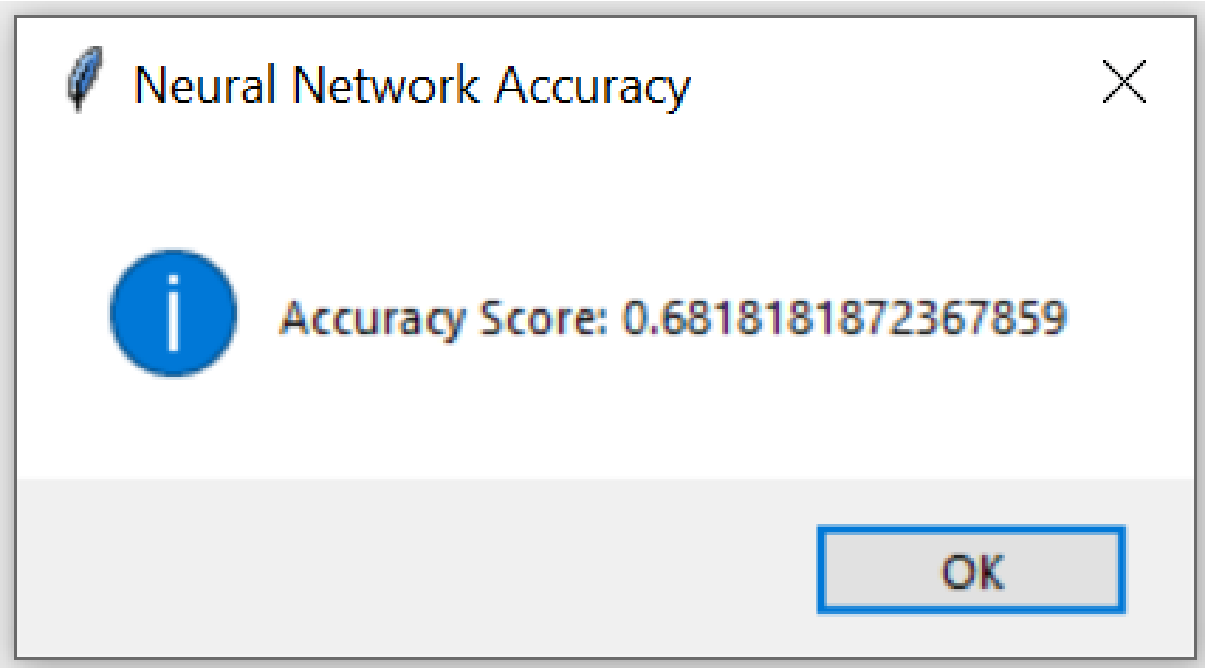


Figure 6.27: Accuracy NN

6.8.3 F1 score

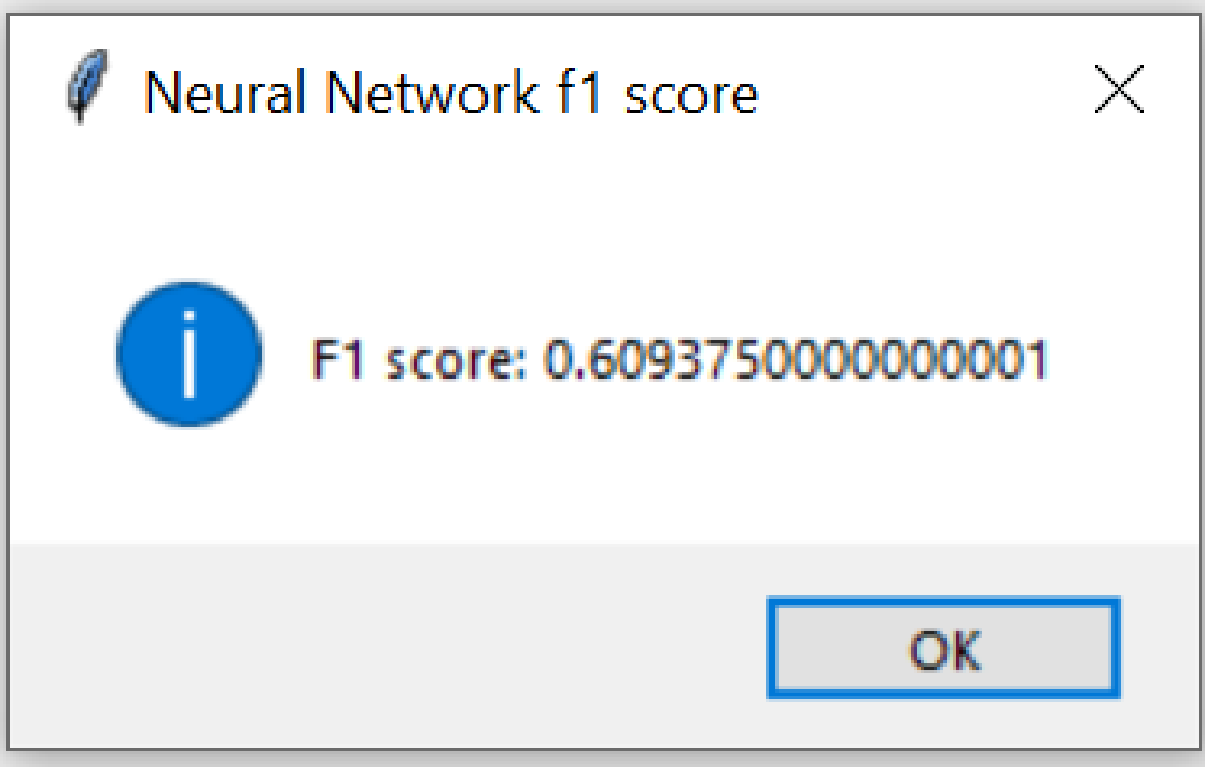


Figure 6.28: F1 score NN

Chapter 7

Conclusion générale

7.1 Apprentissage supervisé VS Apprentissage non supervisé

L'apprentissage non supervisé consiste à inférer les connaissances des classes sur la seule base des échantillons d'apprentissage, et sans savoir a priori à quelles classes ils appartiennent. Contrairement à l'apprentissage supervisé, on ne dispose que d'une base d'entrées et c'est le système qui doit déterminer ses sorties en fonction des similarités détectées entres les différentes entrées (règle d'auto organisation). On pourrait imaginer que l'algorithme d'apprentissage décide lui-même des classes qui existent et de la classification de chaque exemple.

Contrairement à l'apprentissage supervisé, dans l'apprentissage non-supervisé il n'y a pas d'oracle qui explicite les étiquettes. L'utilisation de ce type d'algorithme permet de trouver des structures, des dépendances entre descripteurs qui nous sont inconnues.