Noms : *Mohamed SELMI, Amir BEN MALLEM, Said MOHAMMED SEGHIR. GROUPE :14*

Rapport final HCV

I. Introduction

Notre sujet est de traiter un jeu de données sur l'Hépatite-C et ses complications. Nous avons à notre disposition des échantillons d'analyses sur différentes personnes et notre objectif est de réussir à définir si chaque personne est soit saine soit atteinte d'Hépatite-C, ou de ses deux complications : La Fibrose ou la cirrhose.
Ce sujet est donc un problème de classification multinomiale supervisée, et pour le résoudre nous allons utiliser l'apprentissage automatique.

II. Méthodes choisies

Nous avons sélectionné 3 méthodes parmi les différentes méthodes de classification supervisée, en faisant attention à sélectionner des méthodes qui conviennent le plus au problème à traiter et au jeu de données mis à disposition.
Méthodes à utiliser : **Knn**, **SVM, Régression logistique.**

III. Pre-processing

Après analyse des données nous avons constaté une distribution déséquilibrée entre les différentes catégories, ainsi que la présence de plusieurs valeurs null. Pour remédier à ça nous avons choisit de :

III.1 Remplacer les valeurs NULL :

Afin de remplacer les valeurs null, nous avons calculé la moyenne des valeurs de chaque attribut pour chaque catégorie. Puis on a remplacé les valeurs null par la moyenne de sa catégorie.

III.2 Rééquilibrage de données :

Pour rééquilibrer les données entre les classes (Catégories), nous avons complété le dataset original par des observations synthétiques des classes minoritaires à l'aide de la méthode SMOTE offerte par la librairie imbalanced-learn.

IV. *Protocole d'évaluation*

IV.1 Optimisation des modèles avec Grid Search :

Pour paramétrer nos modèles d'apprentissage afin de donner les meilleurs résultats, on a fait intervenir **Grid Search** qui est une méthode d'optimisation qui va nous permettre de tester une série de hyperparamètres, et de comparer les performances pour en déduire le meilleur paramétrage pour un modèle.

IV.2 Cross-Validation :

Pour optimiser les hyperparamètres une cross-validation est nécessaire pour vérifier la stabilité de l'apprentissage sur plusieurs découpages afin d'assurer que toutes les données servent à optimiser un paramètre. On peut également découper en training, test et validation mais cela réduit d'autant le nombre de données pour apprendre.

IV.3 Modéliser les résultats de prédiction à l'aide d'une matrice de confusion :

Pour évaluer la qualité de classification de chaque modèle. On a comparé les données classées avec des données de référence, puis évalué les prédictions à l'aide de la matrice de confusion qui nous donne différentes métriques telles que : La précision,le rappel, le score, l'exactitude.

V. Résultats et comparaisons

V.1 :Knn :

Accuracy score : 0.992

Classe	Précision	Rappel	f1 Score
0-Blood donor	0.99	0.98	0.99
1-Hépatite C	0.99	0.99	0.99
2-Fibrose	0.99	0.99	0.99
3-Cirrhose	1.00	1.00	1.00

Avantages KNN :

La modélisation KNN n'inclut pas de période d'entraînement car les données elles-mêmes sont un modèle qui servira de référence pour les prédictions et de ce fait, il stocke l'ensemble de données d'entraînement et n'en apprend qu'au moment de faire des prédictions en temps réel. Cela rend l'algorithme KNN beaucoup plus rapide que les autres algorithmes qui nécessitent un entraînement, comme : SVM, régression Logistique.

Inconvénient KNN:

L'algorithme Knn ne fonctionne pas très bien si on a un très grand nombre de classes à prédire, il est aussi très sensible aux données bruyantes et manquantes.

V.2 SVM :

Accuracy score : 0.992

Classe	Précision	Rappel	f1 Score
0-Blood donor	1	0.98	0.99
1-Hépatite C	0.98	0.99	0.99
2-Fibrose	0.99	1	0.99
3-Cirrhose	1	1	1

Avantages SVM :

- Le SVM fonctionne relativement bien lorsqu'il y a une marge de séparation claire entre les classes ce qui est le cas avec notre dataset.
- le risque de sur-entrainement est moindre avec le SVM et ce critère est important au vu du jeu de donnés restreint qu'on a (on ne peut prendre une validation set de taille suffisante pour vérifier s'il y a ou non un sur-entrainement).
- Le kernel trick est la vraie force du SVM. Avec une fonction de noyau appropriée, nous pouvons résoudre n'importe quel problème.
-

Inconvénient SVM:

- Le choix des hyperparamètres n'est pas facile plus particulièrement le choix du noyau idéal à notre problème.
- Bien que notre dataset soit relativement petit le temps d'entrainement est long car nous avons opté pour la grille search pour trouver les bons hyperparamètres donc larges combinaison d'hyperparamètres possible.
- Difficulté à visualiser l'impact des hyperparamètres C et gamma dans les cas ou la dimension est supérieure à 3.

V.3 Régression Logistique :

Accuracy score : 0.894

Classe	Précision	Rappel	f1 Score
0-Blood donor	0.92	0.97	0.94
1-Hépatite C	0.81	0.82	0.81
2-Fibrose	0.86	0.83	0.84
3-Cirrhose	0.99	0.98	0.98

Avantages Régression Logistique :

- La régression logistique est très efficace à entraîner et rapide dans la classification d'enregistrements inconnus. De plus, elle est peu sujette à l'over-fitting (tant que le dataset n'est pas de grande dimension (ex. Génomes)).
- La méthode fournit des probabilités bien calibrées pour chaque classe en plus de la classification finale, ce qui donne un meilleur aperçu sur la justesse du modèle sur le dataset.
- Elle peut utiliser un coefficient d'importance pour chaque variable si nécessaire, et elle est plus tolérante aux données aberrantes que Knn et SVM, la non standardisation des valeurs a donc moins d'effet sur elle.

Inconvénient Régression Logistique:

- L'exactitude est très bonne, mais l'algorithme marcherait mieux sur un dataset large et linéairement séparable.
- Cette méthode demande une non-colinéarité entre les variables explicatives et serait plus efficace si on ne gardait que les variables pertinentes, ce qui nécessite une analyse des données plus poussée.
-

V. Conclusion :

Le jeu de données présentées ne contient pas un grand volume d'information, nous avons donc sélectionné des algorithmes qui permettent d'assurer un bon fonctionnement sur notre dataset, en faisant un pré-traitement de nos données afin de rééquilibrer la distribution des données entre les classes à prédire, parmi les trois algorithmes utilisés nous avons pu obtenir une exactitude de 0.992 sur les deux algorithmes SVM et Knn, contrairement à l'algorithme de régression logistique qui donne une exactitude de 0.894, on peut dire que l'algorithme Knn et SVM est les plus adaptés au problème à résoudre et au jeu de données mis à disposition, mais au niveau du temps d'exécution l'algorithme la régression logistique nous offre un temps optimal, on peut donc utiliser cette dernière comme benchmark avant d'utiliser des algorithmes plus complexes.