Compte rendu

Manale EL GAAMOUSS
Melissa BAFDEL
Abdelatif SENHAJI
Groupe 6

Objectif :

Notre objectif est de modéliser la probabilité que le patient n'ait aucune maladie ou souffre d'une hépatite, d'une fibrose ou d'une cirrhose, en fonction de différents tests tels que des analyses de sang ou d'urine.

Protocole :

Pour notre étude on a choisi les modèles KNN, SVM, Random Forest pour prédire, auxquels on a appliqué la validation croisée et le grid search pour l'évaluation des performances, et la méthode SMOTE pour équilibrer les données, et la matrice de confusion pour l'évaluation des erreurs.

Avant d'appliquer ces méthodes on a fait du data cleaning pour notre dataset afin d'obtenir que des valeurs numériques et pour ceux qui sont nuls on les a remplis par la moyenne de la colonne pour chaque catégorie.

Analyse des résultats :

➢ Avec la validation croisée (K-Fold avec K=9):

Dans le cas du SVM, la moyenne de la précision est de (92 %), pour KNN elle est de (91%) et pour le Random Forest elle est de (99%), mais on a le recall des 3 modèles très faible dans la plupart des cas 0% et très élevé 100% dans la 1ere catégorie qui est blood donors.

➢ Avec le Grid Search

SVM: Les meilleurs paramètres trouvés pour cet algorithme sont: C=100, gamma=0.01.
On a une précision de 91% ,et le recall de la classe 1 est de 0%.
Knn: Le meilleur paramètre trouvé pour cet algorithme est: K=5.
On a une précision de 95%, mais en ce qui concerne le recall on trouve des résultats, avec le recall de la classe,0 est 100% et la classe 1 qui est 0% .
Random Forest : Les meilleurs paramètres trouvés pour cet algorithme sont: criterion=gini, max_depth=30, n_estimators=20.
Ce modèle nous donne une précision de 93% mais le recall qui est nul 0% pour les catégories 1 et 100% pour la classe 0.
On remarque d'après la matrice de confusion des 3 modèles que le nombre des vrais positifs pour la classe 0 est très élevé et le nombre des faux négatifs est élevé, alors que pour les autres classes généralement le nombre des vrais positifs est faible et cela est dû aux données déséquilibrées.

➢ Avec K-fold et Grid search

Après avoir trouvé les meilleurs paramètres on a ré-appliqué le 9-fold, on trouve que les résultats des 3 modèles ne sont toujours pas bons, ça est dû au fait qu'on a des données déséquilibrées, donc on a toujours la majorité des classes dont leurs recall est 0%, et la classe 0 est toujours très élevé.

➢ Avec SMOTE

On a utilisé la méthode SMOTE pour augmenter les données. Après l'avoir appliquer on a trouvé les résultats suivants :

SVM

	precision	recall	f1-score	support
0	0.96	0.92	0.94	119
1	0.98	1.00	0.99	128
2	0.91	0.89	0.90	154
3	0.86	0.92	0.89	131
4	0.99	0.96	0.98	135
accuracy			0.94	667
macro avg	0.94	0.94	0.94	667
weighted avg	0.94	0.94	0.94	667

```
array([[110,   2,   2,   4,   1],
       [  0, 128,   0,   0,   0],
       [  5,   0, 137,  12,   0],
       [  0,   0,  11, 120,   0],
       [  0,   1,   1,   3, 130]])
```

on remarque d'après la matrice de confusion que le nombre des vrais positifs sont élevés, et un nombre de 9, 17,11, 5 de faux négatifs pour les classes 0,2,3 et 4 et le recall est de 100% pour la classe 1, 92% pour la classe 0 et 3, 89% pour la classe 2 et pour la classe 4 est de 96% et avec une **précision de 94%**.

Knn

	precision	recall	f1-score	support
0	0.99	0.92	0.95	119
1	0.95	0.99	0.97	128
2	0.97	0.98	0.98	154
3	0.97	1.00	0.98	131
4	0.97	0.96	0.97	135
accuracy			0.97	667
macro avg	0.97	0.97	0.97	667
weighted avg	0.97	0.97	0.97	667

```
[[109   3   3   1   3]
 [  0 127   0   0   1]
 [  0   0 151   3   0]
 [  0   0   0 131   0]
 [  1   3   1   0 130]]
```

On remarque d'après la matrice de confusion que le nombre de vrais positifs est élevé et un nombre faible de faux négatifs par rapport au SVM de, et le recall qui presque le même dans toutes les classes, ce modèle a une **précision de 97%.**

Random Forest

	precision	recall	f1-score	support
0	1.00	0.97	0.98	119
1	0.99	1.00	1.00	128
2	0.99	1.00	0.99	154
3	0.99	1.00	1.00	131
4	1.00	1.00	1.00	135
accuracy			0.99	667
macro avg	0.99	0.99	0.99	667
weighted avg	0.99	0.99	0.99	667

```
[[115   1   2   1   0]
 [  0 128   0   0   0]
 [  0   0 154   0   0]
 [  0   0   0 131   0]
 [  0   0   0   0 135]]
```

On remarque d'après la matrice de confusion que le nombre de vrais positifs est très élevés et le nombre de faux négatifs très faible par rapport au 2 autres modèle, on a le recall presque le même dans toutes les classes, avec une **précision de 99%**

Avantages et limites :

SVM :

Avantages:

- Fonctionne bien sur de plus petits data sets.
- Sa grande précision de prédiction.

Inconvénients:

- Ne convient pas à des jeux de données plus volumineux, car le temps d'entraînement avec les SVM peut être long.
- Moins efficace sur les jeux de données contenant du bruit.

Knn :

Avantages:

- L'algorithme est polyvalent, il peut être utilisé pour la classification et la régression.
- Knn est un algorithme robuste envers les données bruitées.

Inconvénients:

- L'algorithme devient beaucoup plus lent a mesuré que le nombres d'exemples d'apprentissage augmente.
- L'étape de prédiction peut être lente, la complexité est de l'ordre de O(n)avec(k<<n).

Random Forest :

Avantages:

- Gestion des données manquantes.
- Adaptation à des problématiques de classification pour prédire une cible catégorielle.

Inconvénients:

- Parfois les arbres générés ne sont pas équilibrés.
- Random Forest est instable: des changements légers dans les données produisent des arbres très différents.

Conclusion:

On remarque que les résultats que Knn et Random Forest sont à peu près très proches, avec un taux d'erreur très bas, dont le meilleur algorithme de prédiction est le Random Forest avec une précision de 99%.