Séparateurs à vaste marge

I. Introduction :

L'algorithme de SVM a pour objectif de trouver la séparation entre deux classes d'objets avec l'idée que plus la séparation est large, plus la classification est robuste. SVM est capable de réaliser des classifications binaires et multiclasse.
Il prend en charge la classification binaire et la séparation des points de données en deux classes. Pour la classification multiclasse, le même principe est utilisé après avoir décomposé le problème de multiclassification en plusieurs problèmes de classification binaire. Nous allons utiliser la classification multiclasse.

II. Stratégie adoptée :

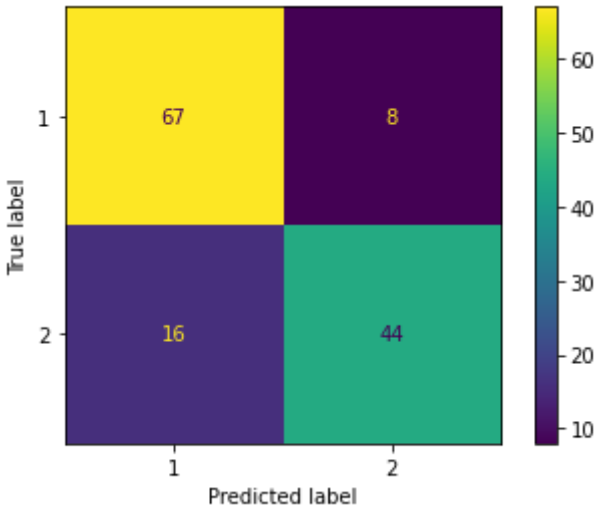
Pour commencer séparer les données entre données d'entraînement et de test en spécifiant le pourcentage de jeu de test (50 %) et en effectuant une stratification pour obtenir des données distribuées de manière uniforme.
Fixer la méthode à utiliser ainsi que leurs hyperparamètres correspondant à tester.
Créer une grille d'hyperparamètres contenante plusieurs valeurs possibles et tester toutes les combinaisons possibles.
Définir trois paramètres : Kernel: [linear, Rbf, polynomial] avec [gamma, C, degree]
Class weight pénalise les erreurs dans les échantillons.
Un poids de classe plus élevé signifie, mettre davantage l'accent sur une classe.
Nous utiliserons les poids suivant {1: 5, 2: 1}.

La méthode d'optimisation GridSearch nous permet d'entraîner le modèle en utilisant une recherche exhaustive sur tous les paramètres.
Les paramètres à fournir sont :
estimator = SVC() en précisant le poids avec la fonction Class weigh,
param_grid = les trois paramètres definie precedement,
cv = 5,
scoring = 'accuracy'

Pour finir la stratégie, nous appliquons avec la fonction d'ajustement fit qui ajuste les poids en fonction des valeurs de données afin d'obtenir une meilleure précision.

III. Résultats obtenus :

L'analyse du modèle SVM sur les données des maladies cardiaques nous a permis d'extraire les résultats suivants :




```
Meilleurs estimator:  SVC(class_weight={1: 5, 2: 1}, gamma=0.1)


Meilleurs paramètres:  {'C': 1.0, 'gamma': 0.1, 'kernel': 'rbf'}

Classification Report:
              precision    recall  f1-score   support

           1       0.81      0.89      0.85        75
           2       0.85      0.73      0.79        60

    accuracy                           0.82       135
   macro avg       0.83      0.81      0.82       135
weighted avg       0.82      0.82      0.82       135

Accuracy score:
0.8222222222222222
```

Nous pouvons constater que le meilleur hyperparamètre est {'C' : 1.0, 'gamma' : 0.1, 'kernel' : 'rbf'} avec un score de précision de 0,82 pour le jeu d'entrainement.

Nous utiliserons la matrice de confusion pour extraire les différentes métriques :
Précision : TP/(TP+FP)
Rappel = TP/(TP+FN)
Taux d'erreur : FP/(FP+TP)

Le label 1 possède une précision inférieure à celui du label 2.
Cependant, le Label 1 englobe un plus grand nombre de données que le 2, de ce fait le label 1 est meilleur que le label 2.

Pour finir le modèle a une précision de 0,82.

IV. Discussion :

Parmi les avantages de la stratégie adoptée, on retient la capacité à traiter de grandes dimensionnalités, nous avons utilisé 13 dimensions différentes pour nos analyses.

```
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(135, 13)
(135, 13)
(135, 1)
(135, 1)
```

De plus une bonne indication de la complexité du problème traité et une souplesse des paramètres qui est résistance au sur-apprentissage.
Cependant, on retrouve en inconvénients une difficulté à traiter de grande base avec des observations très élevées et le problème lorsque qu'on applique la multiplication des points supports sur les classes.

V. Conclusion :

L'algorithme de SVM a permis de trouver la séparation entre les différents classes.
SVM est capable de réaliser des classifications binaires et multiclasse.
La classification multiclasse réalisé avec la fonction GridSearch a permis de séparer les points de données avec une comparaison deux à deux entre les classes.
Nous avons donc obtenu une précision de notre modèle d'un taux de 82 %.
Notre modèle de test prédit sur un ensemble de 89 % pour une précision de 82 %, ce qui est convenable.

Il est possible de comparé notre stratégie avec celle réalisé dans le TP1 avec KNN.
L'algorithme des k plus proches voisins possède un meilleur de précision des maladies cardiaques.