Compte Rendu :

Support Vector Machine

Evaluation et sélection de modèle sur Heart data statlog.

Encadré par :

Marie SZAFRANSKI

Réalisé par :

Toufik GUENANE

1-Introduction :

Les maladies cardiovasculaires regroupent les pathologies qui touchent le cœur et l'ensemble des vaisseaux sanguins, comme l'athérosclérose, les troubles du rythme cardiaque, l'hypertension artérielle, l'insuffisance cardiaque ou encore les accidents vasculaires cérébraux, et tous ces problèmes peuvent conduire à la mort des personnes atteintes de cela. Afin de prévenir les risques de ces maladies et mieux se protéger, on doit comprendre et apprendre comment les détecter d'une manière efficace et sécurisé. Cette recherche vise à identifier les facteurs de risque de maladie cardiaque les plus pertinents ainsi qu'à prédire le risque global à l'aide de Machine à vecteurs de support (SVM) qui a pour but de la classifier et prédire si le patient présente un risque de développer une future maladie cardiaque.

2-Analyse de Données : (à regarder l'annexeSVM)

Figure 8 – La fréquence d'être malade ou non par rapport au type de douleur thoracique (Chest pain) : La figure suivante représente le un diagramme en bâtons qui décrit la fréquence d'être malade ou pas en fonction de type de douleur thoracique. On constate sur cette figure que les personnes qui ont une douleur thoracique de type Asymptote sont les plus susceptibles de développer des maladies cardiovasculaires contrairement aux autres types de douleur qui sont moins susceptibles.

Figure 1 – La fréquence d'être malade ou non par rapport à l'âge (générale) : La figure suivante représente des courbes qui décrivent la fréquence d'être malade ou pas en fonction de l'âge d'une manière générale. On constate sur la courbe bleue que la plupart des personnes malades appartiennent à la tranche d'âge entre 55 ans et 65 ans.

Figure 2 – La fréquence d'être malade ou non par rapport à l'âge (Détaillé) : La figure suivante représente un digramme en bâtons qui décrit la fréquence d'être malade ou pas en fonction de l'âge d'une manière détaillé. On constate sur cette figure que la plupart des personnes malades ont un âge de 58 ans en premier, après en deuxième on a les personnes qui ont 60 ans puis en troisième les personnes âgées de 59 ans et 62 ans.

Figure 3 – La fréquence d'être malade ou non en fonction de Glycémie à jeun (FBS Fasting Blood Sugar) : La figure suivante représente un digramme en bâtons qui décrit la fréquence d'être malade ou pas en fonction de Glycémie à jeun. On constate à partir de ce diagramme que la plupart des malades ont une Glycémie à jeun inférieur à 120 mg/dl et qui sont les plus susceptibles de développer des formes graves des maladies cardiovasculaires.

Figure 4 – La fréquence d'être malade ou non en fonction de Sex : La figure suivante représente un digramme en bâtons qui décrit la fréquence d'être malade ou pas en fonction de sexe. On constate sur cette figure que la plupart des malades sont de sexe masculins, on peut aussi dire que les males sont les plus susceptibles de développer des maladies cardiovasculaires contrairement aux femelles.

Figure 5 – La fréquence d'être malade ou non en fonction de la Thalassémie : La figure suivante représente un digramme en bâtons qui décrit la fréquence d'être malade ou pas en fonction de Type de Thalassémie. On constate sur cette figure que les personnes qui ont une thalassémie de type=7 sont les plus susceptibles de développer des maladies cardiovasculaires contrairement aux autres types de thalassémie qui sont moins susceptibles.

3-Machine à vecteurs de support (SVM algorithmes) :

Les machines à vecteurs de support ou séparateurs à vaste marge (Support Vector Machine, SVM) sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de discrimination(classification) et de régression. Les SVM sont une généralisation des classifieurs linéaires, ce sont des algorithmes d'apprentissage initialement construits pour la classification binaire avec l'idée de rechercher une règle de décision basée sur une séparation par hyperplan de marge optimale en trouvant de bonnes frontières de décision. Le passage à la recherche de surfaces séparatrices non linéaires est introduit en utilisant un noyau kernel (POLY(polynomiale) ou RBF(Gaussien)) qui code une transformation non linéaire des données, de suite on peut dire qu'on a deux types de SVM linéaires et non linéaires comme suit :

La fonction $k(x, x') = \langle x; x' \rangle_{\mathcal{R}^p}$ correspond à un hyperplan linéair dans ce cas : $\mathcal{H} = \mathbb{R}^p$.
$k(x, x') = (c + \langle x; x' \rangle)^d$ cherche une séparation par courbe polynômiale de degré au plus d
Le noyau Gaussien $k(x, x') = e^{\frac{-\|x-x'\|^2}{2\sigma^2}}$ cherche une séparation par frontière radiale (*Radial Basis Function*)
On peut choisir le « bon »noyau en utilisant une stratégie de validation croisée. En général, ce choix permet de gagner quelques pourcentages d'erreur.

Les SVM sont plus efficace dans les espaces de grande dimension.

4-Analyse des résultats :

4-1 GridSearchCV :

GridSearchCV (la recherche de grille) est le processus consistant à effectuer un réglage d'hyperparamètres comme « C » et « Gamma », afin de déterminer les valeurs optimales pour un modèle donné. Ceci est important car les performances de l'ensemble du modèle sont

basées sur les valeurs des hyperparamètres spécifiées dans une grille. GridSearchCV essaie toutes les combinaisons des valeurs passées dans le dictionnaire et évalue le modèle pour chaque combinaison à l'aide de la méthode de validation croisée. Par conséquent, après avoir utilisé cette fonction, nous obtenons une précision/perte pour chaque combinaison d'hyperparamètres et nous pouvons choisir celui qui offre les meilleures performances, et donnera aussi la meilleure combinaison basée sur le meilleur score de cross validation obtenu.

4.1.1 Kernel Linéaire :

Suite au découpage, split et normalisation des données 'Heart' et l'entrainement de ces dernières sur le model SVM de kernel linéaire, on commence à analyser nos résultats provenant de GridSearchCV :

On a un seul paramètre dans ce cas qui est 'C':

'C' est un paramètre de régularisation, et indique à l'optimisation SVM combien vous voulez éviter de mal classer chaque exemple d'entrainement des données d'apprentissage.

Les valeurs de **C** prises pour l'étude sont : **0.1, 1, 10, 100** avec des résultats obtenus comme suit : meilleur paramètre est **: Le meilleur paramètre est : {'C' : 0.1, 'kernel': 'linear'}**

Avec **précision** qui est à : **0.8564356435643564**

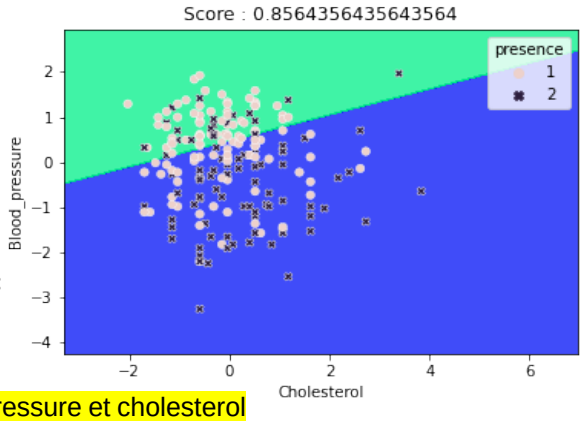

Figure 1 – Linear frontière de décision en 2D entre Blood_pressure et cholesterol

Matrice de confusion

A partir de la matrice de confusion on constate : true négative =34 ; faux positif = 4 ; faux négative = 6 ;true positif = 24 d'où le taux d'erreur est : 0.15 approximativement avec une précision de 85% sur l'ensemble de test.

```
tn : 34  fp : 4  fn:  6  tp : 24
              precision    recall  f1-score   support

           1       0.85      0.89      0.87        38
           2       0.86      0.80      0.83        30

    accuracy                           0.85        68
   macro avg       0.85      0.85      0.85        68
weighted avg       0.85      0.85      0.85        68

Le cout d'erreur est :  0.14705882352941177
```

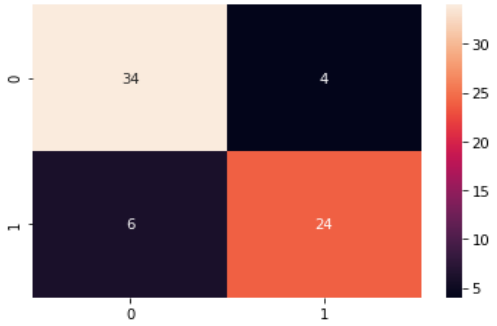

Figure 2 – Rapport de classification avec Linear SVM(GridSearchCV)

Figure 3 - Confusion Matrix LinearSVM(GridSerachCV)

4.1.2 Kernel RBF(Gaussien) :

Suite au découpage, split et normalisation des données 'Heart' et l'entrainement de ces dernières sur le model SVM de kernel RBF, on commence à analyser nos résultats provenant de GridSearchCV : On a un deux paramètres dans ce cas qui sont 'C' (veuillez regarder Kernel Linéaire) et 'Gamma':

'Gamma' : paramètre qui définit jusqu'où s'étend l'influence d'un seul exemple d'apprentissage.

Les valeurs prises pour l'étude sont **: 'Gamma'= « 0.01, 0.1 , 1 , 10 ».**
'C'= « 1, 10, 100, 1000, 10000» avec des résultats obtenu comme suit : Les **meilleurs paramètres** sont **: {'C': 1.0, 'gamma': 0.01, 'kernel': 'rbf'}**

Avec **précision** qui est à : **0.8564356435643564**

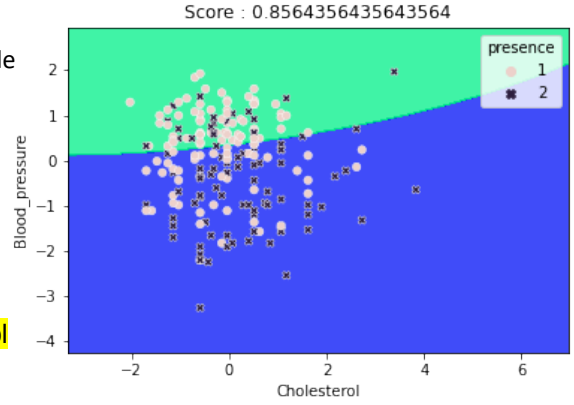

Figure 4 – RBF frontière de décision en 2D entre Blood_pressure et cholesterol

Matrice de confusion

A partir de la matrice de confusion on constate : true négative =33 ; faux positif = 5 ; faux négative = 6 ;true positif = 24 d'où le taux d'erreur est : 0.16 approximativement avec une précision de 85% sur l'ensemble de test.

```
tn : 33  fp : 5  fn:  6  tp : 24
              precision    recall  f1-score   support

           1       0.85      0.87      0.86        38
           2       0.83      0.80      0.81        30

    accuracy                           0.84        68
   macro avg       0.84      0.83      0.84        68
weighted avg       0.84      0.84      0.84        68

Le taux d'erreur est :  0.16176470588235295
```

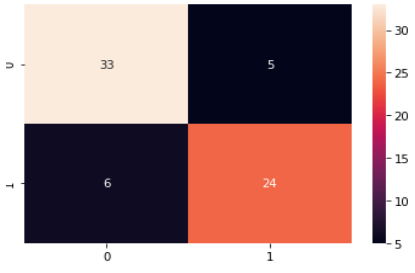

Figure 5 – Rapport de classification avec RBF SVM(GridSearchCV) Figure 6 - Confusion Matrix RBF_SVM(GridSerachCV)

4.1.3 Kernel POLY(Polynomial) :

Suite au découpage, split et normalisation des données 'Heart' et l'entrainement de ces dernières sur le model SVM de kernel POLY, on commence à analyser nos résultats provenant de GridSearchCV :

On a un deux paramètres dans ce cas qui sont 'C' (veuillez regarder Kernel Linéaire) et 'degree':

'degree' : paramètre qui définit jusqu'où s'étend l'influence d'un seul exemple d'apprentissage.

Les valeurs prises pour l'étude sont: **'degree'=«1,2,3,4,5,6,7».** **'C'= « 0.01, 0.1, 1, 10, 100»** avec des résultats obtenus comme suit : **Les meilleurs parametres** sont : **{'C': 0.1, 'degree': 1, 'kernel': 'poly'}**

Avec précision qui est à : 0.8514851485148515

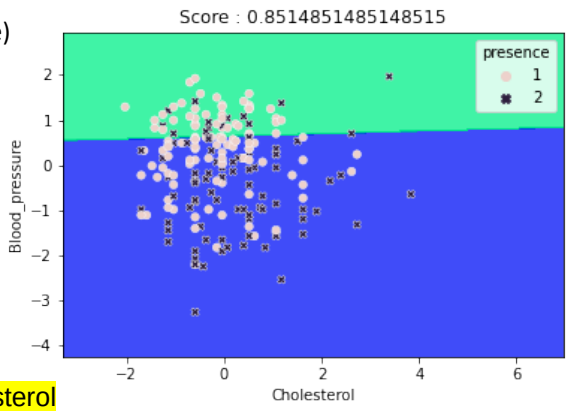

Figure 7 – POLY frontière de décision en 2D entre Blood_pressure et cholesterol

Matrice de confusion

A partir de la matrice de confusion on constate : true négative =33 ; faux positif = 5 ; faux négative = 6 ;true positif = 24 d'où le taux d'erreur est : 0.16 approximativement avec une précision de 85% sur l'ensemble de test.

```
tn : 33  fp : 5  fn:  6  tp : 24
              precision    recall  f1-score   support

           1       0.85      0.87      0.86        38
           2       0.83      0.80      0.81        30

    accuracy                           0.84        68
   macro avg       0.84      0.83      0.84        68
weighted avg       0.84      0.84      0.84        68

Le taux d'erreur est :  0.16176470588235295
```

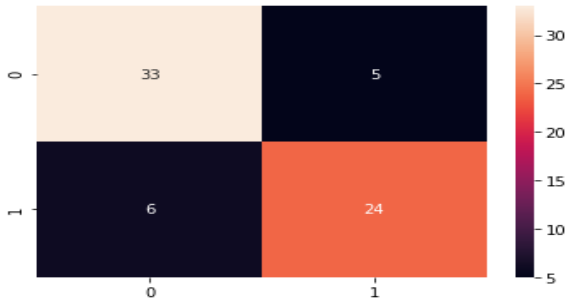

Figure 8 – Rapport de classification avec POLY SVM(GridSearchCV) Figure 9 - Confusion Matrix POLY_SVM(GridSerachCV)

4.2 Carte de performances en deux dimensions

Dans cette partie, pour les données non séparables linéairement (On travaillera sur POLY et RBF), on pourra visualiser la variation des performances du classifieur en fonction des différents hyper-paramètres, avec cette étude nous allons utiliser le modèle svm sans GridSearchCV pour voir la différence entre GridSearchCV et carte de performances en 2D.

4.2.1 kernel POLY

Suite au découpage, split et normalisation des données 'Heart' et l'entrainement de ces dernières sur le model SVM de kernel POLY, on commence à analyser nos résultats provenant de Carte de performances en 2D : On a un deux paramètres dans ce cas qui sont 'C'(veuillez regarder Kernel Linéaire) et 'degree': Les valeurs prises pour l'étude sont :

'degree'=«1,2,3,4,5,6,7». **'C'= « 1, 10, 100, 1000,10000»** avec des résultats de chaque combinaison que vous trouverez dans l'annexeSVM la figure suivante : « Figure 6 – Résultats de Carte de performances en deux dimensions POLY »

Matrice de confusion

A partir de la matrice de confusion on constate : true négative =34 ; faux positif = 4 ; faux négative = 7 ;true positif = 23 d'où le taux d'erreur est : 0.57 approximativement avec une précision de 84% sur l'ensemble de test.

```
tn : 34  fp : 4  fn:  7  tp : 23
              precision    recall  f1-score   support

           1       0.83      0.89      0.86        38
           2       0.85      0.77      0.81        30

    accuracy                           0.84        68
   macro avg       0.84      0.83      0.83        68
weighted avg       0.84      0.84      0.84        68

Le taux d'erreur est :  0.5735294117647058
```

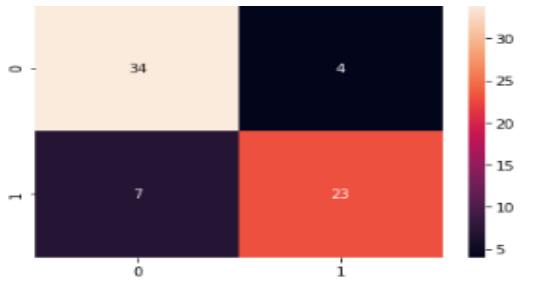

Figure 10–Rapport de classification avec POLY SVM(2D) Figure 11 - Confusion Matrix POLY_SVM(2D)

On constate que malgré une précision de 1 (regarder annexe figure 6 de AnnexeSVM) de l'entraiment, la précision de de test est de 84%.

4.2.2 kernel RBF

Suite au découpage, split et normalisation des données 'Heart' et l'entrainement de ces dernières sur le model SVM de kernel RBF, on commence à analyser nos résultats provenant de Carte de performances en 2D :

On a un deux paramètres dans ce cas qui sont 'C' (veuillez regarder Kernel Linéaire) et Gamma: Les valeurs prises pour l'étude sont : 'Gamma=« 0.01, 0.1 , 1 , 10». 'C'= « 1, 10, 100, 1000,10000»

Avec des résultats de chaque combinaison que vous trouverez dans l'annexeSVM la figure suivante :

« Figure 7 – Résultats de Carte de performances en deux dimensions POLY »

Matrice de confusion

A partir de la matrice de confusion on constate : true négative =37 ; faux positif = 1 ; faux négative = 23 ; true positif = 7 d'où le taux d'erreur est : 1.70 approximativement avec une précision de 65% sur l'ensemble de test.

tn : 37 fp : 1 fn: 23 tp : 7

	precision	recall	f1-score	support
1	0.62	0.97	0.76	38
2	0.88	0.23	0.37	30
accuracy			0.65	68
macro avg	0.75	0.60	0.56	68
weighted avg	0.73	0.65	0.58	68

Le taux d'erreur est : 1.7058823529411764

Figure 12–Rapport de classification avec RBF SVM(2D)

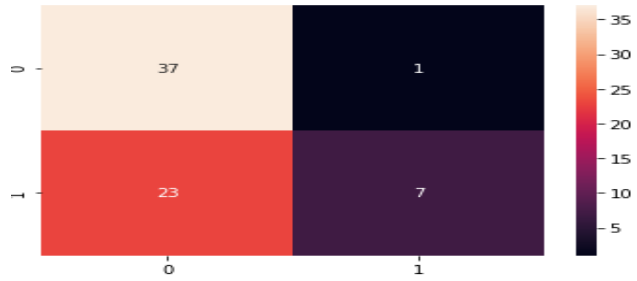

Figure 13 - Confusion Matrix RBF_SVM(2D)

On constate que malgré une précision de 1 (regarder annexe figure 7) de l'entraiment, la précision de de test est de 65%.

5. Avantages et les limites :

L'utilisation de GridSearchCV donne un meilleur résultat et un taux d'erreur bas, bien que GridSearchCV présente de nombreux avantages, il prend trop de temps et d'efforts à régler parfaitement le modèle. Une meilleure utilisation du temps peut être d'étudier plus avant les caractéristiques du SVM modèle qu'on fait avec les paramètres nécessaires. L'ingénierie de fonctionnalités et la sélection de sous-ensembles de fonctionnalités peuvent augmenter (ou diminuer) considérablement les performances de ce modèle. Cela demandera beaucoup plus d'efforts que de brancher des nombres dans une grille de paramètres, mais, en retour, on développera également davantage une compréhension de l'ensemble de données et on découvrira éventuellement de nouvelles relations entre les entités.

6. Conclusion

A partir des deux études effectuer avec GridSearchCV et Carte de performances en deux dimensions sur les différents Kernel linéaires et non linéaires (POLY et RBF) de SVM, on peut conclure que le meilleur modèle est le modèle linéaire en mode GridSearchCV, car il a donné le taux d'erreur le plus bas avec 15% par rapport au RBF avec 16% et au POLY avec 16%. Par rapport à algorithme de k plus proches voisins, SVM est plus efficace pour les problèmes qui peuvent pas être séparé linéairement et aussi dans les cas où il y'a moins de données contrairement à KNN qui est efficace sous un grand nombre de données, mais les deux types méthodes de classification supervisé prennent énormément de temps et d'efforts, ce que nous pousse à chercher une solution qui a les avantages des deux types de classifier, et on trouve RandomForest qui peut s'occuper des features catégoriques d'une excellente manière et peut gérer des espaces de grande dimension ainsi qu'un grand nombre d'exemples d'apprentissage.