Multimodal Processing, Recognition, and Interaction (MPRI)

Data preparation

Cross-validation

Confusion Matrix

Auteur : Spinelli Isaia

Prof : Stefano Carrino

Date : 24.11.2020

Salle : A2 – Lausanne

Classe : MPRI

Table des matières

[Introduction - 2 -](#_Toc55371612)

[OneClassSVM avec Scikit-learn - 2 -](#_Toc55371613)

[Novelty detection - 3 -](#_Toc55371614)

[Tâche 1 - 3 -](#_Toc55371615)

[Tâche 2 - 3 -](#_Toc55371616)

[Tâche 3 - 3 -](#_Toc55371617)

[Outliers detection - 4 -](#_Toc55371618)

[Conclusion - 5 -](#_Toc55371619)

[Difficultés rencontrées - 5 -](#_Toc55371620)

[Compétences acquises - 5 -](#_Toc55371621)

[Résultats obtenus - 5 -](#_Toc55371622)

[Annexe - 5 -](#_Toc55371623)

# Introduction

Ceci est un travail dirigé qui permet d’apprendre le processus complet pour la gestion d’un problème de machine learning.

…

Loading the dataset

Explore and visualize the dataset

1. Number of samples: **150**
2. Number of classes: **3**
3. Number of features: **4**
4. Are there probable outliers? (yes/no & why)

**(Oui) pour les largeur des sépales il y a 4 outliers**

Unbalanced dataset

1. Is the Iris dataset balanced? (yes/no & why)

**Oui car il y a 50 samples pour chaque classe**

Data preparation

1. In your opinion, why do we fix the seed?

**Pour avoir une sortie reproductible sur plusieurs appels de fonction.**

Rescaling/standardization

**Les feature avec des petites valeurs peuvent être négligée !**

1. Are these methods (QuantileTransformer, MinMaxScaler) a good alternative? Why? In your opinion, why in a regression problem should you rescale also the Y?

**Ca peut être utile et plus simple pour comparer ! Il ne faut pas oublier de reconvertir les valeurs**

**QuantileTransformer : Oui elle est robuste**

**MinMaxScaler : Base !!!! outliers ..**

k-fold cross-validation to search the “best” hyperparameters (using grid search)

1. max\_dept:
2. n\_estimators:
3. min\_samples\_split:
4. min\_samples\_leaf:

Detect under/overfitting using scikit-learn function “learning\_curve”

1. In your opinion, is the system overfitting, underfitting or doing well? What do you think and why?

Compute the confusion matrix using the test set

1. explain the line y\_pred = classifier.predict(X\_test\_scaled); Why we did not use y\_pred = classifier.predict(X\_train\_scaled)?

Compute accuracy, precision, recall, f1\_score

1. Go back to Step 7 and substitute the parameters used by the classifiers with the best parameters you found in Step 6. Do not forget to re-train the classifier (using classifier.fit()) on the training set. Is the performance of the Classifier improved?

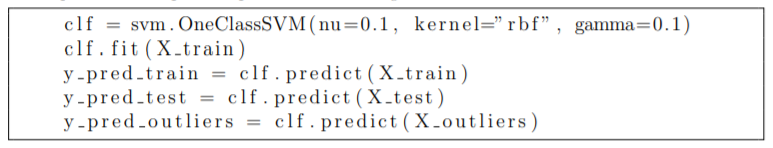
Optional exercises – Going deeper

1. Comment les échantillons sont générés ? Pourquoi y a-t-il deux pôles de concentration de points ?

Pour générer les échantillons, on génère des échantillons de type float à partir d'une distribution univariée « normale » (gaussienne) de moyenne 0 et de variance 1 en utilisant la fonction « numpy.random.rand » avec un sigma de 0.3. Ensuite, on ajoute ± 2 à toutes les valeurs des échantillons, ce qui double le nombre de donnés. C’est aussi ceci qui va provoquer les deux différents pôles de concentration de points.

Pour générer les échantillons outliers, il génère 40 échantillons de manière uniforme entre [-4 ; 4 [. (4 non compris).

1. Dans le code ci-dessous, à quoi correspondent les variables X\_train, X\_test et X\_outliers. Pourquoi n’a-t-on pas uniquement le training set et le test set ?



La variables X\_tain correspond aux données d’entrainement (training obersvations). La variable X\_test correspond aux données de test (new regular observations). La variable X\_outliers correspond aux données « outliers », autrement dit, les données distante des autres observations (new abnormal observation).

Nous avons maintenant un outliers set afin de vérifier la bonne détection des anomalies.

1. Pourquoi ne donne-t-on pas les étiquettes (ou labels) à la fonction fit() d’apprentissage ?

Car toutes les données dans le set d’entrainement sont de la même classe, des cancers bénins.

1. Tester d’autres noyaux (kernel) pour le classifieur et garder celui qui donne les meilleurs scores. Modifier ensuite les différents paramètres afin d’améliorer les prédictions (gamma, nu, et d’autres en fonction du noyau choisi - voir documentation).

Dans la documentation, voici les différents kernel possibles : linear, poly, rbf, sigmoid, precomputed. Dans mon cas, precomputed ne compile pas. Sinon, les meilleurs scores ont été produit par le kernel par défaut rbf.

Après plusieurs tentatives, je me suis rendu compte que ce n’était pas évident de rechercher les meilleurs paramètres car les données sont générées aléatoirement et cela peut faire varier notre système facilement. Personnellement, je trouvé que ces paramètres apportaient les meilleurs résultats de manière générale : nu=0.001 et gamma=0.2.

# Conclusion

## Difficultés rencontrées

* Bonne compréhension des principes et conséquences

## Compétences acquises

* Familiarisation avec le classifieur « OneClassSVM »
* Amélioration de la maîtrise du langage python
* Les différents paramètre de l’algorithme SVM

## Résultats obtenus

Toutes les étapes demandées du laboratoire ont été réalisées et les questions répondues. J’ai trouvé ce laboratoire particulièrement instructif car il était bien dirigé, ce qui m’a aidé à le réaliser pas à pas. De plus, le fait de travailler avec des données réelles est particulièrement intéressant.

# Annexe

* Code Pyhton de l’exercice 3 (ex3-breast-cancer\_outliers.py)

Date : 24.11.20

Nom de l’étudiant : Spinelli Isaia