meritis

Semaines 6 et 7: Diagnostique ML et SVM

Exercice 1 : Validation croisée et Diagnostique sur un ensemble de données sur le diabète.

L'ensemble des données sur le diabète consiste de 10 variables physiologiques, en particulier : l'âge, le sexe, le poids et la pression artérielle. Cette étude se base sur 442 patients avec un indice de progression de la maladie après un an.

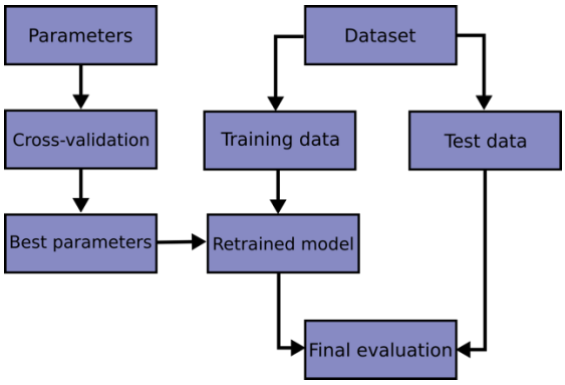
Plus d'information sur l'étude :
http://web.stanford.edu/~hastie/Papers/LARS/LeastAngle_2002.pdf

I. Apprentissage et Analyse des résultats

On s'intéresse dans un premier temps à apprendre

1. Préparer votre environnement, importer la base depuis Scitkit-learn.
2. Définir un objet *DataFrame* avec ces données et les champs suivants : « Âge, sexe, indice de masse corporelle, pression artérielle, et 6 mesures de sérum (TC, LDL, HDL, TCH, LTG et GLU) » et répartir vos données en Train et Test (avec un taux de 20 ou 30% pour les tests)
3. Utiliser la Régression Linéaire pour apprendre votre modèle.
4. Tracer sur un graphe (en utilisant matplotlib) les valeurs obtenus (sur l'axe des y) en fonction des valeurs observées (sur l'axe des x) et prédire le taux de réussite en utilisant la fonction *score* de votre modèle.
5. Que déduire de ces résultats ?

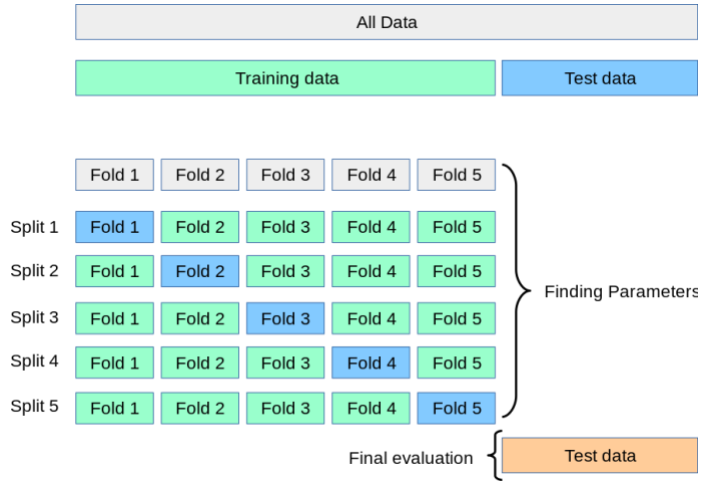
II. Diagnostique et Validation Croisée



Une des techniques diagnostique est la validation croisée. Celle-ci est basée sur la répartition de l'ensemble des données en Train/Test, mais est appliquée à plusieurs sous-ensembles. Pour

clarifier, on divise notre ensemble en *n* sous-ensembles et apprend notre modèle avec *n-1* sous-ensembles. Le dernier sous-ensemble est réservé aux tests.
Il existe plusieurs algorithmes qui permettent d'effectuer la validation croisée. Dans notre exemple, on se basera sur K-Folds.

K-Folds



Cet algorithme propose de diviser l'ensemble de données en *k* sous-ensembles où chaque sous-ensemble est utilisé *k-1* fois pour apprendre le modèle et une fois pour tester.
La procédure générale est la suivante :

1. Diviser votre ensemble de données en train et test.
2. Diviser l'ensemble train en *k* groupes.
3. Pour chacun des k groupes :
    a. Prendre un groupe pour validation (test) et les *k-1* autres pour l'apprentissage.
    b. Apprendre un modèle sur les ensembles *k-1* et le évaluer avec le *kème* groupe.
    c. Conserver le score d'évaluation et rejeter le modèle appris.
4. Votre modèle final utilisera les meilleurs paramètres calculés à l'étape 3.
5. Déterminer la précision de votre modèle en utilisant la base test.

N.B : N'oubliez pas de mettre dans une liste tous les modèles appris pendant l'étape 2.b.

Exercice 2 : Diagnostique pour choisir son modèle de classification

Dans l'exercice précédent, on a utilisé la validation croisée pour partitionner au mieux notre ensemble de données (entre Train et Test) et augmenter la précision de notre modèle. Maintenant, on s'intéresse à utiliser cette même technique mais pour choisir le meilleur modèle de classification et prédire s'il va pleuvoir demain ou pas en Australie, en se basant sur les observations des stations météorologiques australiennes.

Lien pour télécharger la base de données : https://www.kaggle.com/jsphyg/weather-dataset-rattle-package/discussion/78316

I. Étape de Prétraitement

1. Préparation des données

   a. Importer l'ensemble des données depuis le fichier csv et afficher les 5 premières lignes de vos données en utilisant la fonction *head()*
   b. Dropper les lignes avec des données manquantes.
   c. Dropper les colonnes inutiles de votre ensemble de données comme Date', 'Location', 'Evaporation', 'Sunshine', 'Cloud3pm', 'Cloud9am', 'RISK_MM'
   d. Remplacer vos observations par des chiffres pour RainToday et RainTomorrow (0 pour No et 1 pour Yes).
   e. Transformer les colonnes 'WindGustDir', 'WindDir9am', et 'WindDir3pm' en catégories binaires pour couvrir les différents cas possibles.
      i. Vous pouvez utiliser la fonction de pandas *getdummies(data, columns=list_de_categories, drop_ first=True)*
   f. Séparer les observations des features (X et y)

2. Norme Scalaire

   De nombreux algorithmes Machine Learning fonctionnent mieux lorsque les caractéristiques sont normalement distribuées et à une échelle relativement similaire. Cette échelle est d'habitude entre 0 et 1.

   « Standard Scaler » normalise une caractéristique en soustrayant la moyenne et en mettant à l'échelle de la variance. Pour rappel, la variance a pour objectif de diviser toutes les valeurs par l'écart-type. La variance et l'écart-type sont égaux à 1 et les valeurs tombent entre -1 et 1 pour une moyenne de distribution égale à 0.
   Z = (X – U) / S ; U étant la moyenne et S la variance

   a. Normaliser vos caractéristiques en utilisant la fonction fit_transform de la classe StandardScaler de Scikit-Learn et afficher les résultats.
   b. Diviser votre ensemble de données en train (80%) et test (20%)

II. Régression Logistique v/s SVM

Dans cette partie, on va faire appel à l'algorithme de K-Fold de la validation croisée déterminer le meilleure modèle de classification pour cet ensemble de données (SVM et régression logistique) à partir de la précision moyenne.

1. Diviser votre ensemble train en 10-Folds.
2. Les étapes à suivre sont les suivantes **pour chaque modèle :**
   a. Pour chacun des 10 groupes

i. Prendre un groupe pour validation (test) et les *k-1* autres pour l'apprentissage.
ii. Apprendre un modèle sur les ensembles *k-1* et le évaluer avec le *k*ème groupe.
iii. Conserver le score d'évaluation et rejeter le modèle appris.

b. Résumer les compétences du modèle en calculant la moyenne des scores conservés

3. Le meilleur modèle est choisi grâce au score calculer en 2.b.
4. Déterminer la performance du modèle choisi sur la base test

Sources

Observations were drawn from numerous weather stations. The daily observations are available from http://www.bom.gov.au/climate/data. Copyright Commonwealth of Australia 2010, Bureau of Meteorology.
Definitions adapted from http://www.bom.gov.au/climate/dwo/IDCJDW0000.shtml
This dataset is also available via the R package rattle.data and at https://rattle.togaware.com/weatherAUS.csv. Package home page: http://rattle.togaware.com. Data source: http://www.bom.gov.au/climate/dwo/ and http://www.bom.gov.au/climate/data.
And to see some nice examples of how to use this data: https://togaware.com/onepager/