Fonctions Classification

Version 2023

Benali Nafissa et Fuentes Vicente Laura

nov. 09, 2023

Contents:

CHAPITRE 1

general module

general.**CV_rep**(*Xtr*, *ytr*, *nfolds*)

La fonction crée des nouveaux jeux de données en sous-divisant les jeux de données en

Paramètres

- **Xtr** (*pd.DataFrame*) – Jeu de données à diviser, contenant les co-variables
- **ytr** (*pd.DataFrame*) – Vecteur à diviser, contenant la variable à prédire
- **nfolds** (*int*) – Nombre representant en combien de sous dataframes on souhaite diviser Xtr et ytr

Renvoie

liste contenant (n-folds) jeux de données y_new (list) : liste contenant (les n-folds) nouvelles version de ytr

Type renvoyé

X_new (list)

general.**Label_Encode**(*y_tr*, *new_pred*)

Fonction qui normalise les etiquettes de plusieurs jeux de données

Paramètres

- **y_tr** (*pd.DataFrame*) – vecteur qui contient les etiquettes que l'on utilisera pour entrainer le modèle
- **new_pred** (*np.array*) – vecteur auquel on appliquera la normalisation des etiquettes

Renvoie

même vecteur qu'avant avec les etiquettes normalisées

Type renvoyé

new_pred (np.array)

general.**submission**(*new_pred*, *name*, *date*, *X_test*)

Fonction qui genere un csv avec les valeurs à prédire associées à ses respectifs identificateurs Attention : il faudrait aussi verifier que obj_ID se retrouve dans le repertoire nommé

Paramètres

- **new_pred** (*np.array*) – vecteur contenant les predictions
- **name** (*string*) – méthode utilisée pour generer la prédiction (ex : "LDA")
- **date** (*string*) – date de la prédiction (ex : "04/10")
- **X_test** (*pd.DataFrame*) – Jeu de données test utilisé pour calculer la prédiction

CHAPITRE 2

KNN_sel_var module

KNN_sel_var.**choix_var_knn**(*X_tr*, *y_tr*, *vars*, *columns*)

Fonction qui calcule pour un vecteur de variables, le f1_score (par cross-validation) associé à l'ajout d'une nouvelle variable. Cette dernière choisit une nouvelle variable parmi cols et la rajoute sur vars.

Paramètres

- **X_tr** (*pd.DataFrame*) – Jeu de données d'entraînement avec les co-variables
- **y_tr** (*pd.DataFrame ou np.array*) – Jeu de données d'entraînement avec la variable à prédire
- **vars** (*pd.Index*) – vecteur avec les variables déjà selectionnées auparavant
- **columns** (*pd.Index*) – vecteur avec des nouvelles variables non inclues dans vars

Renvoie

retourne le vecteurs vars avec la nouvelle variable maxiimisant le f1_score cols (pd.Index) : retourne le vecteur de nouvelles co-variables sans la variable finalement choisie

Type renvoyé

vars (pd.Index)

KNN_sel_var.**main**(*X_tr*, *y_tr*, *X_te*, *y_te*)

Fonction qui effectue la selection de variables

Paramètres

- **X_tr** (*pd.DataFrame*) – Jeu de données d'entraînement avec les co-variables
- **y_tr** (*pd.DataFrame ou np.array*) – Jeu de données d'entraînement avec la variable à prédire
- **X_te** (*pd.DataFrame*) – Jeu de données test avec les co-variables
- **y_te** (*pd.DataFrame ou np.array*) – Jeu de données tets avec la variable à prédire

Renvoie

vecteur contenant le choix des variables aménant au meilleur f1_score par cross validation

Type renvoyé

choix_vars (pd.Index)

KNN_sel_var.**plot_vars**(*res_plot*, *vars*)

Fonction qui renvoit un plot avec les f1_scores associés aux prédicteurs issus de l'ajout itératif de chaque variable

Paramètres

— **res_plot** (*list*) – liste contenant les scores f1 associés à l'ajout de chaque variable
— **vars** (*pd.Index*) – vecteur contenant les variables ajoutées dans l'ordre

KNN_sel_var.**pred_acc_var**(*vars*, *X_tr*, *y_tr*, *X_te*, *y_te*)

Fonction qui pour des variables données, enmtraîne une knn et calcule l'erreur test associé.

Paramètres
— **vars** (*pd.Index*) – variables selectionnées
— **X_tr** (*pd.DataFrame*) – Jeu de données d'entraînement avec les co-variables
— **y_tr** (*pd.DataFrame ou np.array*) – Jeu de données d'entraînement avec la variable à prédire
— **X_te** (*pd.DataFrame*) – Jeu de données test avec les co-variables
— **y_te** (*pd.DataFrame ou np.array*) – Jeu de données test avec la variable à prédire

Renvoie
score f1 sur le jeu de données test

Type renvoyé
f1_score(y_te, pred, average= »weighted »)

KNN_sel_var.**train**(*X*, *y*, *X_test*, *vars*)

Fonction qui entraîne le modèle pour les variables choisies et crée une prédiction

Paramètres
— **X** (*pd.DataFrame*) – Jeu de données d'entraînement avec les co-variables
— **y** (*pd.DataFrame ou np.array*) – Jeu de données d'entraînement avec la variable à prédire
— **X_test** (*pdDataFrame*) – Jeu de données test avec les co-variables
— **vars** (*pd.Index*) – vecteur contenant les variables choisies au préalable

Renvoie
description

Type renvoyé
type

CHAPITRE 3

methode_fait_maison module

methode_fait_maison.**choix_seuils**(*X*, *y*, *folds*, *nb_seuils*, *n_var*)

Fonction qui choisi les seuils en fonction des résultats de cross validation

Paramètres

- **X** (*pd.DataFrame*) – Jeu de données avec les co-variables sur lesquel on veut entrainer les valeurs seuils
- **y** (*pd.DataFrame*) – Jeu de données avec la variable à prédire sur lequel on veut entrainer les valeurs seuils
- **folds** (*int*) – nombre de folds (sous-divisions) pour faire la cross-validation
- **nb_seuils** (*int*) – nombre de seuils à choisir
- **n_var** (*string*) – nom de la variable sur laquelle on travaille

Renvoie

seuil choisi pour distinguer les classes 1 et 0 seuil_2 (float) : seuil choisi pour distinguer les classes 0 et 2

Type renvoyé

seuil_0 (float)

methode_fait_maison.**f1**(*ytr*, *pred*, *lab*)

Fonction qui calcule le score f1 associé à une prédiction (pour un problème de classification binaire, C=[3,lab] avec lab={0,2})

Paramètres

- **ytr** (*pd.DataFrame*) – vecteur contenant les labels du jeu de données train
- **pred** (*np.array*) – vecteur contenant des prédictions (2 classes)
- **lab** (*_type_*) – _description_

Renvoie

valeur du score f1 associé à cette prédiction

Type renvoyé

(2*TP)/(2*TP + FP + FN) (float)

methode_fait_maison.**frontiere**(*Xtr*, *ytr*, *lab*, *folds*, *nb_seuils*, *n_var*)

Cette fonction crée un vecteur contenant, nb_seuils, seuils différents et teste leur performance (f1 score) pour classfifier le label lab à partir de la variable n_var. On utilisera la méthode de cross validation pour tester les performances moyennes de chaque seuil.

Paramètres
- **Xtr** (*pd.DataFrame*) – Jeu de données train contenant les co-variables
- **ytr** (*pd.DataFrame*) – Jeu de données train contenant la variable à prédire
- **(int** (*lab*) – 0 ou 2) : label pour lequel calculer la valeur frontière
- **folds** (*int*) – nombre de folds a créer pour performer la cross-validation
- **nb_seuils** (*int*) – nombre de seuils à tester entre la valeur min,max de n_var
- **n_var** (*string*) – nom de la variable sur laquelle on veut calculer le seuil

Renvoie
vecteur contenant les f1-scores moyennes de chaque seuil par cross-validation

Type renvoyé
results.mean(axis=0) (np.array)

methode_fait_maison.**melange**(*folds*, *Xtr*, *ytr*, *var1*, *seuil1*, *model2*, *vars*)

Fonction qui choisi la probabilité p (avec cross-validation) que l'on tirera sur une bernouilli pour mélanger deux prédictions

Paramètres
- **folds** (*int*) – nombre de
- **Xtr** (*np.array ou pd.DataFrame*) – Vecteur d'entrainement contenant les co-variables
- **ytr** (*np.array ou pd.DataFrame*) – Vecteur d'entrainement contenant la variable a predire
- **var1** (*string*) – variable sur laquelle se basent les seuil1
- **seuil1** (*list*) – liste avec les deux seuils choisis avec la méthode basée sur var1 (de la forme [seuil_0,seuil_2])
- **model2** – modèle 2 entraîné

Renvoie
probabilité que l'on utilisera pour mélanger deux predictions

Type renvoyé
proba

methode_fait_maison.**pred_mel**(*pred1*, *pred2*, *p*)

Fonction qui mélange deux prédictions avec probabilité p sur une Bernouilli

Paramètres
- **pred1** (*np.array*) – predicteur 1 (à mélanger)
- **pred2** (*np.array*) – prédicteur 2 (à mélanger)
- **p** (*np.float*) – probabilité pour la Bernouilli

Renvoie
nouvelle prédiction contenant le mélange des deux prédicteurs

Type renvoyé
tirage(p,p1,p2) (np.array)

methode_fait_maison.**predict**(*val0*, *val2*, *X_te*, *n_var*)

Crée notre prédiction en fonction des valeurs val0 et val2 calculées précédamment. Il crée un vecteur prédiction avec des valeurs : - 1 : n_var appartenant à]-inf,val0[- 0 : n_var appartenant à [val0,val2[- 2 : n_var appartenant à [val2,+inf[

Paramètres
- **val0** (*float*) – valeur seuil calculée pour distinguer la classe 1 et 0
- **val2** (*float*) – valeur seuil calculée pour distinguer la classe 0 et 2
- **X_te** (*pd.DataFrame*) – vecteur des covariables test sur lesquels on va regarder la valeur de n_var
- **n_var** (*string*) – nom de la variable sur laquelle on base la prédiction

Renvoie
vecteur contenant les prédictions de notre méthode_seuil

Type renvoyé
pred

methode_fait_maison.**tirage**(*p*, *pred1*, *pred2*)

Fonction qui tire melange deux vecteurs de probabilités. Elle repere les indices dans lequels les deux predicitions sont differentes et choisi une des deux valeurs en tirant aleatoirement une bernouilli avec proba p.

Paramètres
- **p** (*np.float*) – probabilité associé au tirage aléatoire de la loi de Bernouilli
- **pred1** (*np.array*) – Vecteur contenant les prédictions de la méthode 1
- **pred2** (*np.array*) – Vecteur contenant les prédictions de la méthode 1

Renvoie
nouveau vecteur de prédictions calculé à partir des deux autres prédictions

Type renvoyé
res

CHAPITRE 4

Indices and tables

— genindex
— modindex
— search

Index des modules Python

Index