Formation

“Projet évaluation en Data Science”

06 mars 2023

Sommaire

1. Techniques de prétraitement des données
2. Techniques d’optimisation des hyperparamètres
3. Modèles empilés

1 Pre-processing phase

1 – Data preprocessing : traiter les valeurs nulles

USE CASE 1

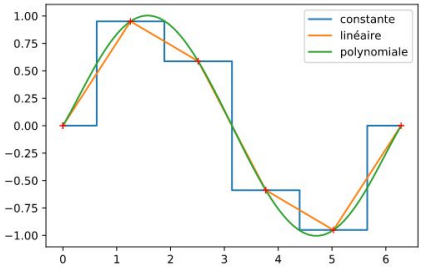
CONSOLIDATION ET NETTOYAGE DE LA SERIE TEMPORELLE

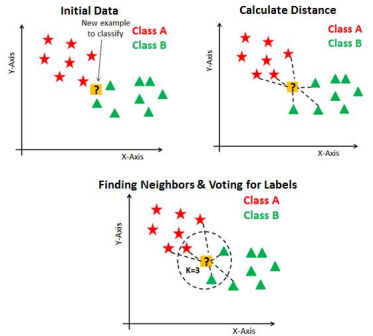
Quelques méthodes :

- Imputation :
  - par une constante,
  - par la moyenne,
  - par la médiane,
  - par la donnée la plus représentée, ...

- Interpolation : à partir d'un nombre fini de points reconstruire une fonction
  - linéaire,
  - polynomiale,
  - spline (une fonction polynomiale par morceaux),
  - cubic (cas particulier de polynôme d'ordre 3), ...

- K-nearest neighbors reconstruction





2 Modeling phase

2 - Optimisation des hyperparamètres

Hyperparameter Tuning with Grid Search and Random Search

	Grid search	Random search
Principe	Grid Search commence par définir une **grille** d'espace de recherche . La grille se compose de noms et de valeurs d'hyperparamètres sélectionnés, et **la recherche de grille** recherche de manière exhaustive la meilleure combinaison de ces valeurs données.	En recherche aléatoire, on définit **des distributions** pour chaque hyperparamètre qui peuvent être définies *uniformément* ou avec une *méthode d'échantillonnage* . La principale différence avec la recherche par grille réside dans la recherche aléatoire, toutes les valeurs ne sont pas testées et les valeurs testées sont sélectionnées au hasard.

Exemple	`# Définir la grille`
	`param_grid = {`
	`'n_estimators' : [50, 100, 200, 300],`
	`'min_samples_leaf' : [1, 5, 10],`
	`'max_depth' : [2, 4, 6, 8, 10],`
	`'max_features' : [ 'auto', 'sqrt'],`
	`'bootstrap' : [Vrai, Faux]}`
	
	`# Instancier GridSearchCV`
	`model_gridsearch = GridSearchCV(`
	`estimateur=rf_model,`
	`param_grid=param_grid,`
	`scoring='accuracy',`
	`n_jobs=4,`
	`cv=5,`
	`refit=True,`
	`return_train_score=True)`
		`# spécifiez les distributions à partir desquelles échantillonner`
		`param_dist = {`
		`'n_estimators': list(range(50, 300, 10)),`
		`'min_samples_leaf': list(range(1, 50)),`
		`'max_depth': list(range(2, 20)),`
		`'max_features' : ['auto', 'sqrt'],`
		`'bootstrap' : [Vrai, Faux]}`
		
		`# spécifier le nombre d'itérations de recherche`
		`n_iter = 50`
		
		`# Instancier RandomSearchCV`
		`model_random_search = RandomizedSearchCV(`
		`estimateur=rf_model,`
		`param_distributions=param_dist,`
		`n_iter=n_iter)`

Méthode en Python	GridSearchCV	RandomizedSearchCV

3 - Modèles empilés (principe)

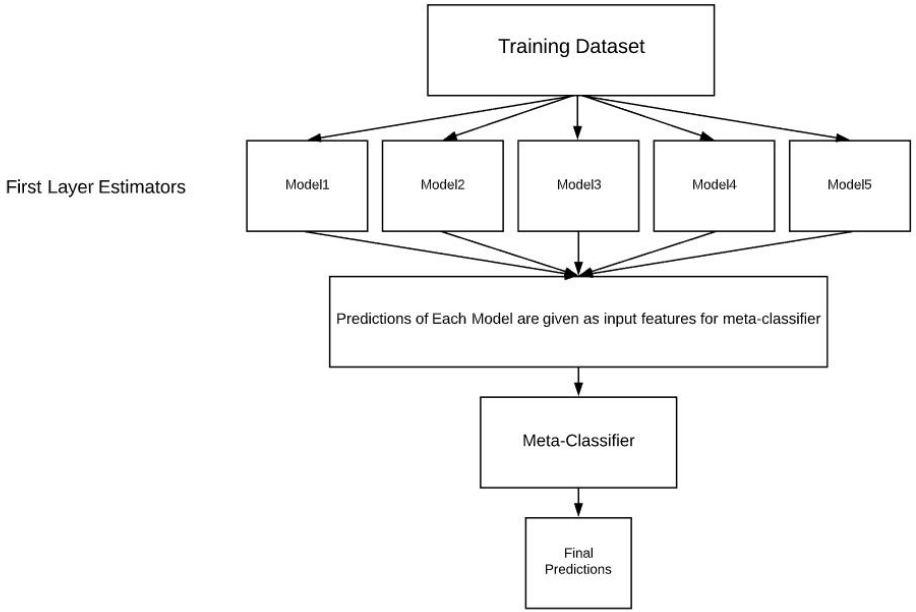
Empilement

Objectif : l'empilement est un moyen **d'assembler des modèles** de classification ou de régression,

Principe : il se compose d'estimateurs à deux couches.

1. La première couche comprend tous les modèles de référence utilisés pour prédire les sorties sur les jeux de données de test.

2. La deuxième couche consiste en un méta-classificateur ou un régresseur qui prend toutes les prédictions des modèles de base en entrée et génère de nouvelles prédictions.



3 - Modèles empilés (exemple)

Tutorial : https://www.geeksforgeeks.org/stacking-in-machine-learning-2/

Exemple : sklearn

Examples

```
>>> from sklearn.datasets import load_iris
>>> from sklearn.ensemble import RandomForestClassifier
>>> from sklearn.svm import LinearSVC
>>> from sklearn.linear_model import LogisticRegression
>>> from sklearn.preprocessing import StandardScaler
>>> from sklearn.pipeline import make_pipeline
>>> from sklearn.ensemble import StackingClassifier
>>> X, y = load_iris(return_X_y=True)
>>> estimators = [
...     ('rf', RandomForestClassifier(n_estimators=10, random_state=42)),
...     ('svr', make_pipeline(StandardScaler(),
...                           LinearSVC(random_state=42)))
... ]
>>> clf = StackingClassifier(
...     estimators=estimators, final_estimator=LogisticRegression()
... )
>>> from sklearn.model_selection import train_test_split
>>> X_train, X_test, y_train, y_test = train_test_split(
...     X, y, stratify=y, random_state=42
... )
>>> clf.fit(X_train, y_train).score(X_test, y_test)
0.9...
```


MILESKER
–
MERCI

“Garapen ekonomikoa xedea baino gehiago baliabide bat dela uste du HUPIk”

“Le Développement Économique est un Moyen et pas une Finalité”

HUPI SAS
Technopole Izarbel
2 Terrasse Claude Shannon
64210 Bidart

HUPI IBERICA SLU
Paseo Miramon N°170
20009 Donostia
/ San Sebastian

contact@hupi.fr