

Régression Logistique Binomiale - OVO

Régression Logistique Binomiale - One-Versus-One (OVO)

Théorie

La **régression logistique binomiale** est utilisée pour la classification binaire, mais elle peut être adaptée aux problèmes **multiclasse** via l'approche **One-Versus-One (OVO)**. Ici, un modèle est entraîné pour chaque paire de classes, ce qui peut améliorer la précision lorsque les classes sont bien séparées.

Hyperparamètres

Nous allons tester un seul hyperparamètre pour réduire le temps d'entraînement : - **Paramètre de régularisation (C)** : contrôle la pénalisation de la complexité du modèle (valeurs entre 0.1 et 1).

Exemple en Python

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.multiclass import OneVsOneClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Chargement des ensembles de données
train_data = pd.read_csv('covertypes_train.csv')
val_data = pd.read_csv('covertypes_val.csv')
```

```

test_data = pd.read_csv('coverttype_test.csv')

# Préparation des données
X_train = train_data.drop('Cover_Type', axis=1)
y_train = train_data['Cover_Type']

X_val = val_data.drop('Cover_Type', axis=1)
y_val = val_data['Cover_Type']

X_test = test_data.drop('Cover_Type', axis=1)
y_test = test_data['Cover_Type']

# Standardisation des données
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_val = scaler.transform(X_val)
X_test = scaler.transform(X_test)

# Recherche du meilleur hyperparamètre (C seulement)
C_values = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1] # Entre 0.1 et 1
train_accuracies = []
val_accuracies = []

for C in C_values:
    model = OneVsOneClassifier(LogisticRegression(solver='saga', C=C, penalty='l2', max_iter=1000))
    model.fit(X_train, y_train)

    y_train_pred = model.predict(X_train)
    y_val_pred = model.predict(X_val)

    train_accuracies.append(accuracy_score(y_train, y_train_pred))
    val_accuracies.append(accuracy_score(y_val, y_val_pred))

# Sélection du meilleur hyperparamètre
best_C = C_values[val_accuracies.index(max(val_accuracies))]
print(f"Meilleur hyperparamètre : C={best_C}")

# Affichage du graphique
plt.figure(figsize=(8, 6))
plt.plot(C_values, train_accuracies, marker='o', linestyle='dashed', label='Train Accuracy')
plt.plot(C_values, val_accuracies, marker='s', linestyle='dashed', label='Validation Accuracy')
plt.xlabel("Paramètre de régularisation (C)")

```

```

plt.ylabel("Précision")
plt.title("Impact de la régularisation sur la performance de la régression logistique (OVO)")
plt.legend()
plt.show()

# Modèle final avec le meilleur hyperparamètre
final_model = OneVsOneClassifier(LogisticRegression(solver='saga', C=best_C, penalty='l2', max_iter=1000))
final_model.fit(X_train, y_train)
y_test_pred = final_model.predict(X_test)

# Affichage de la matrice de confusion
conf_matrix = confusion_matrix(y_test, y_test_pred)
print("\nMatrice de confusion :")
print(conf_matrix)

print("\nÉvaluation sur l'ensemble de test")
print(classification_report(y_test, y_test_pred))

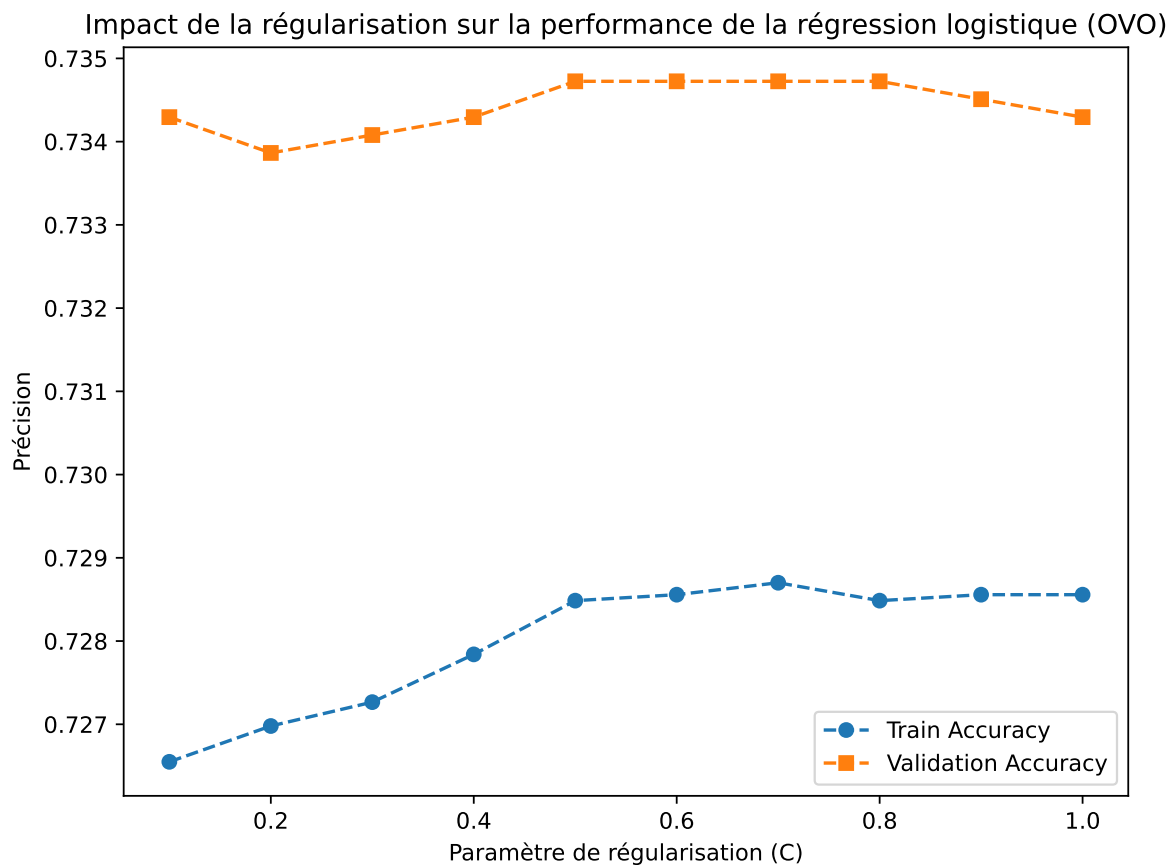
```

```

/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: ConvergenceWarning:
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: ConvergenceWarning:
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: ConvergenceWarning:
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: ConvergenceWarning:
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: ConvergenceWarning:
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: ConvergenceWarning:
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: ConvergenceWarning:
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: ConvergenceWarning:
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: ConvergenceWarning:
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: ConvergenceWarning:
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: ConvergenceWarning:
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: ConvergenceWarning:
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: ConvergenceWarning:
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: ConvergenceWarning:
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: ConvergenceWarning:
  warnings.warn(

```


[illegible]



```

/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergence
warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergence
warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergence
warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergence
warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergence
warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergence
warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergence
warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergence
warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergence
warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergence

```

```

warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergence
warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergence
warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergence
warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergence
warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergence
warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergence
warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergence
warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergence
warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergence
warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergence
warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergence

```

Matrice de confusion :

```

[[1210  471    0    0    0    2   25]
 [ 418 1789   29    0    1   23    1]
 [   0   30  221    1    0   29    0]
 [   0    0    6    9    0    6    0]
 [   1   68    1    0    0    4    0]
 [   0   35   81    0    0   28    0]
 [   70    0    0    0    0    0   89]]

```

Évaluation sur l'ensemble de test

	precision	recall	f1-score	support
1	0.71	0.71	0.71	1708
2	0.75	0.79	0.77	2261
3	0.65	0.79	0.71	281
4	0.90	0.43	0.58	21
5	0.00	0.00	0.00	74

6	0.30	0.19	0.24	144
7	0.77	0.56	0.65	159
accuracy			0.72	4648
macro avg	0.58	0.50	0.52	4648
weighted avg	0.70	0.72	0.71	4648