Régression Multinomiale - Classification Multiclasse

Régression Multinomiale (Softmax Regression)

Théorie

La **régression multinomiale**, aussi appelée **régression logistique multinomiale**, est une extension de la régression logistique qui permet de gérer plusieurs classes. Elle utilise une fonction **Softmax** en sortie pour assigner une probabilité à chaque classe.

Hyperparamètres

Nous allons optimiser 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.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('covertype_train.csv')
val_data = pd.read_csv('covertype_val.csv')
```

```
test_data = pd.read_csv('covertype_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
val_accuracies = []

for C in C_values:
    model = LogisticRegression(multi_class='multinomial', solver='saga', C=C, penalty='l2', r
    model.fit(X_train, y_train)
    y_val_pred = model.predict(X_val)
    acc = accuracy_score(y_val, y_val_pred)
    val_accuracies.append((C, acc))

# Sélection du meilleur hyperparamètre
best_C, best_acc = max(val_accuracies, key=lambda x: x[1])
print(f"Meilleur hyperparamètre : C={best_C}")

# Affichage du graphique
plt.figure(figsize=(8, 6))
plt.plot(C_values, [acc for C, acc in val_accuracies], marker='o', linestyle='dashed')
plt.xlabel("Paramètre de régularisation (C)")
plt.ylabel("Précision sur validation")
plt.title("Impact de la régularisation sur la performance de la régression multinomiale")
plt.show()

# Modèle final avec le meilleur hyperparamètre
```

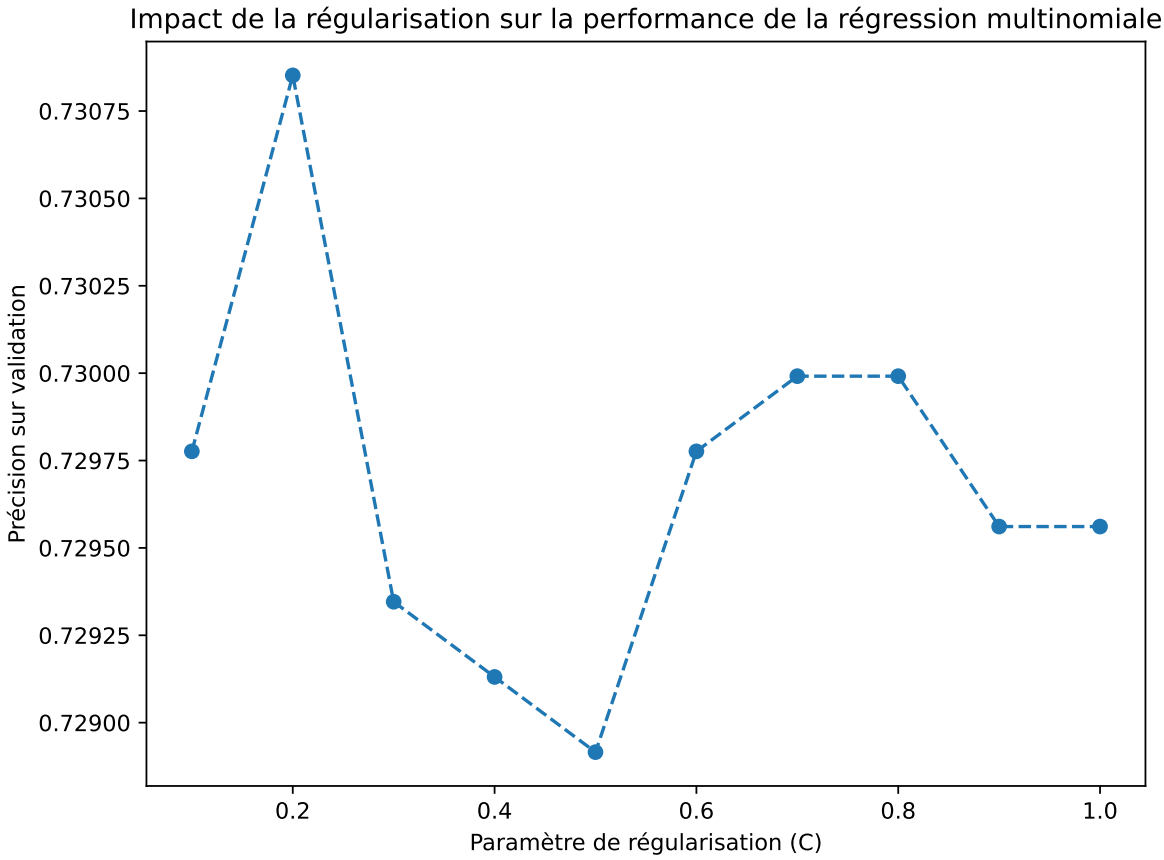
```
final_model = LogisticRegression(multi_class='multinomial', solver='saga', C=best_C, penalty=
final_model.fit(X_train, y_train)
y_test_pred = final_model.predict(X_test)

# 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/_logistic.py:1247: Futu
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergenc
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_logistic.py:1247: Futu
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergenc
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_logistic.py:1247: Futu
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergenc
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_logistic.py:1247: Futu
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergenc
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_logistic.py:1247: Futu
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergenc
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_logistic.py:1247: Futu
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergenc
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_logistic.py:1247: Futu
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_sag.py:349: Convergenc
  warnings.warn(
/home/ensai/.local/lib/python3.10/site-packages/sklearn/linear_model/_logistic.py:1247: Futu
  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/_logistic.py:1247: Futur
  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/_logistic.py:1247: Futur
  warnings.warn(
```

```
Meilleur hyperparamètre : C=0.2
```

```
/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/_logistic.py:1247: Futur
  warnings.warn(
```

```
Matrice de confusion :
[[1206  471    0    0    0    1   30]
 [ 415 1790   34    0    1   20    1]
 [   0   27  226    7    0   21    0]
 [   0    0    7   10    0    4    0]
 [   1   68    4    0    0    1    0]
 [   0   35   84    1    0   24    0]
 [  71    0    0    0    0    0   88]]

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       0.64      0.80      0.71       281
           4       0.56      0.48      0.51        21
           5       0.00      0.00      0.00        74
           6       0.34      0.17      0.22       144
           7       0.74      0.55      0.63       159

    accuracy                           0.72      4648
   macro avg       0.53      0.50      0.51      4648
weighted avg       0.70      0.72      0.71      4648

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