

# TP : Comparaison des Solveurs en Régression Logistique

INF4127 : Optimisation II

Membres du Groupe 4

Faculté des Sciences  
Département d'Informatique

17 janvier 2026

# Membres du Groupe

Matricule	Nom	Prénom
22W2163	DJATCHÉ-NKAMGANG	SYLVANO
24F2456	ESSUTHI MBANGUE	ANGE ARMEL
19M2351	TAGNE TALLA	IDRISS CHANEL
21T2899	GOUJOU GUIMATSA	ZIDANE

## Encadrement

- **Enseignant** : pr. Paulin Melatagia
- **Cours** : INF4127 - Optimisation II
- **Date** : 17 janvier 2026

# Plan de la Présentation

- 1 Introduction
- 2 Méthodologie
- 3 Solveurs Comparés
- 4 Résultats Principaux
- 5 Analyse de Convergence
- 6 Comparaison L1 vs L2
- 7 Recommandations
- 8 Conclusion

# Contexte et Objectifs

## Contexte

La régression logistique est un algorithme fondamental en Machine Learning. Son optimisation peut utiliser différents solveurs, chacun ayant ses spécificités.

## Objectifs du TP

- Comparer 5 solveurs de scikit-learn
- Analyser leurs performances
- Étudier l'impact des hyperparamètres
- Donner des recommandations pratiques

## Solveurs testés

`lbfgs, liblinear, newton-cg, sag, saga`

# Approche Expérimentale

## Datasets utilisés

- **Synthétique** :  $1000 \times 20$
- **Breast Cancer** :  $569 \times 30$
- **Iris** :  $100 \times 4$

## Métriques

- Accuracy
- Temps d'entraînement
- Nombre d'itérations
- Log Loss

## Étapes

- ① Préparation des données
- ② Standardisation
- ③ Entraînement des modèles
- ④ Évaluation
- ⑤ Comparaison

## Important

Toutes les données sont standardisées avant entraînement.

# Caractéristiques des Solveurs

Solveur	Type	Spécificités
lbfgs	Quasi-Newton	Généraliste
liblinear	Coordinate Descent	Rapide, L1/L2
newton-cg	Newton + CG	Précis, coûteux
sag	Gradient Stochastique	Grands datasets
saga	SAG amélioré	L1/ElasticNet

## Support des fonctionnalités

- **Multiclasse** : tous sauf liblinear
- **Régularisation L1** : liblinear et saga
- **Données standardisées** : requis pour lbfgs et newton-cg

# Performance sur Dataset Synthétique

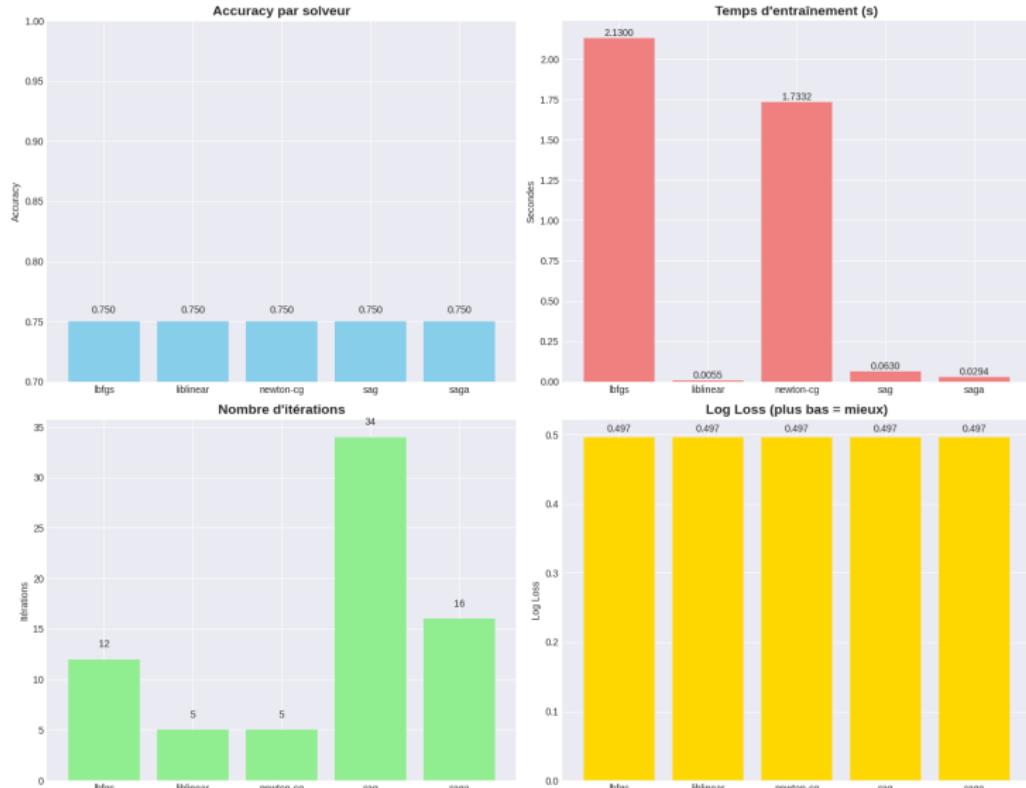
Solveur	Accuracy	Temps (s)	Itérations
LBFGS	0.7500	2.1300	12
LIBLINEAR	0.7500	0.0055	5
NEWTON-CG	0.7500	1.7332	5
SAG	0.7500	0.0630	34
SAGA	0.7500	0.0294	16

## Observations clés

- Même accuracy pour tous (0.75)
- liblinear est le plus rapide (0.0055s)
- lbfgs est le plus lent (2.13s)
- Tous ont convergé

# Visualisations

Comparaison des Solveurs - Dataset Synthétique



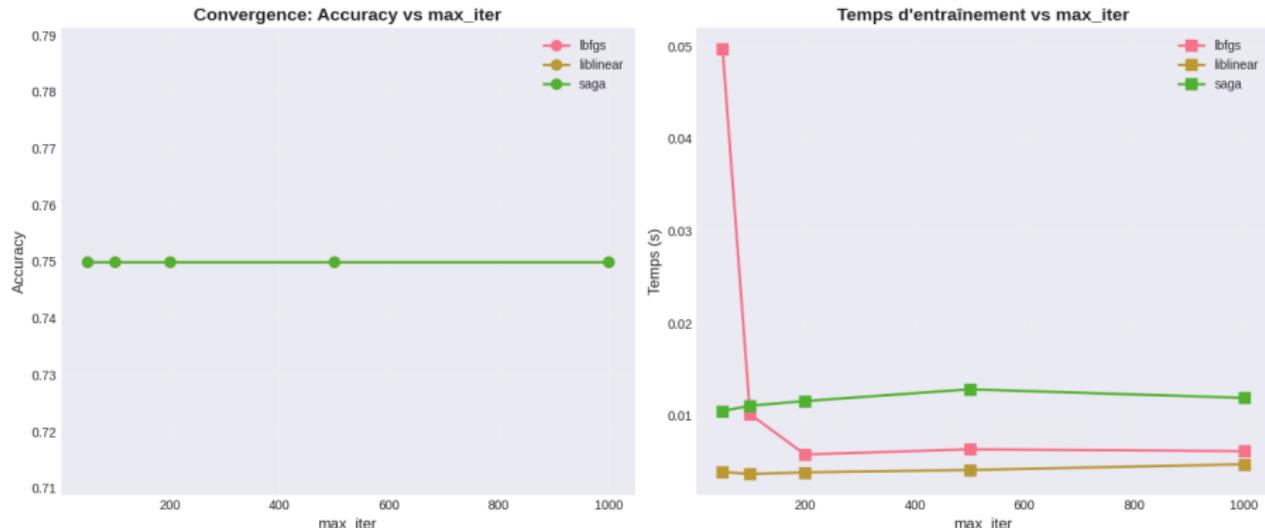
# Performance sur Données Réelles (Breast Cancer)

Solveur	Accuracy	Temps (s)
LBFGS	0.9883	1.4815
LIBLINEAR	0.9883	0.0058
SAGA	0.9883	0.1456

## Observations

- Excellente accuracy (98.83%) pour tous
- liblinear reste le plus rapide
- Résultats cohérents avec le dataset synthétique

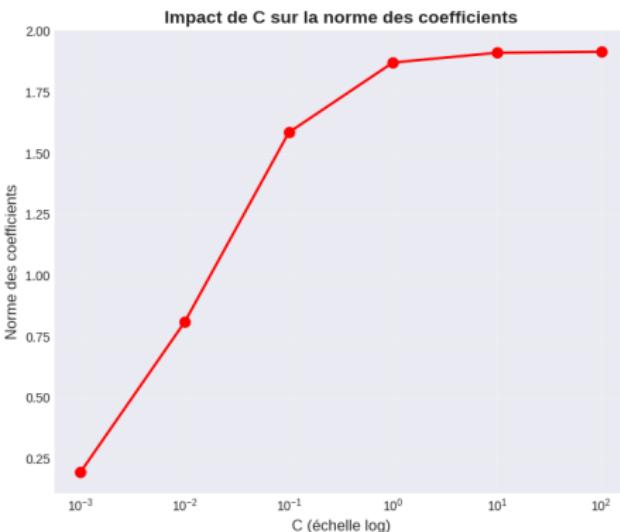
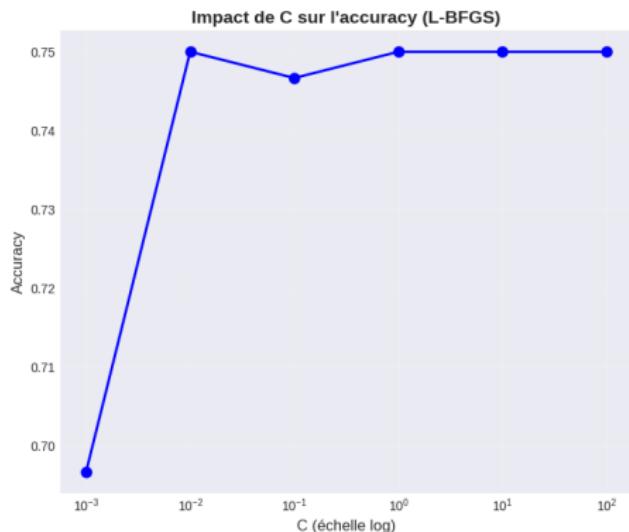
# Impact de max\_iter sur la Convergence



## Conclusions

- lbfgs converge dès 50 itérations
- liblinear stable mais moins précis
- saga montre des fluctuations

# Impact du Paramètre C



Plage optimale :  $10^{-2}$  à  $10^2$

- C trop petit ( $10^{-3}$ ) : sous-apprentissage
- C dans la plage optimale : accuracy stable
- Norme des coefficients augmente avec C

# Régularisation L1 vs L2

## L1 (Lasso)

- Sélection de features
- Accuracy : 0.7500
- Features sélectionnées : 15/20
- Supporté par : liblinear, saga

## L2 (Ridge)

- Régularisation douce
- Accuracy : 0.7500
- Tous coefficients non-nuls
- Supporté par : tous les solveurs

## Conclusion

Mêmes performances, choix dépend du besoin :

- **L1** pour sélection de variables
- **L2** pour stabilité numérique

# Guide de Sélection des Solveurs

Scénario	Solveur recommandé
Problème binaire, petit dataset	<b>liblinear</b> (rapide, supporte L1)
Problème multiclasse	<b>lbfgs</b> (robuste, bon compromis)
Grand dataset ( $n > 10^5$ )	<b>saga</b> (évolutif, supporte L1)
Sélection de variables (L1)	<b>saga</b> ou <b>liblinear</b>
Précision maximale requise	<b>newton-cg</b> (précis mais lent)

# Recommandations d'Hyperparamètres

## Paramètre C

- Valeur par défaut : 1.0
- Plage à tester :  $[10^{-3}, 10^3]$
- Grid search recommandé
- Validation croisée

## max\_iter

- Valeur par défaut : 100
- Pour `lbfgs` : 50-200
- Pour `saga` : 100-1000
- Toujours vérifier la convergence

## Bonnes pratiques

- ① Standardiser les données
- ② Tester plusieurs solveurs
- ③ Ajuster C par validation croisée
- ④ Monitorer la convergence

# Conclusions Principales

## Résultats clés

- ① Tous les solveurs obtiennent la même accuracy (0.75)
- ② liblinear est le plus rapide (0.0055s vs 2.13s pour lbfgs)
- ③ Tous convergent rapidement (5-34 itérations)
- ④ Bonne généralisation sur données réelles (98.83%)

## Recommandation finale

- **Par défaut** : utiliser lbfgs
- **Pour la rapidité** : liblinear (si problème binaire)
- **Pour grands datasets** : saga
- **Toujours** : standardiser et ajuster C

## Annexe : Code Principal

```
# Fonction d' valuation des solveurs
def evaluate_solver(X_train, X_test, y_train, y_test, solver):
    model = LogisticRegression(
        solver=solver,
        penalty='l2',
        C=1.0,
        max_iter=1000,
        random_state=42
    )

    start_time = time.time()
    model.fit(X_train, y_train)
    training_time = time.time() - start_time

    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)

    return {
        'accuracy': accuracy,
```

# Annexe : Résultats Complets

Solveur	Acc	Temps (s)	Itér	Conv	Log Loss	L1
LBFGS	0.7500	2.1300	12	oui	0.4967	non
LIBLINEAR	0.7500	0.0055	5	oui	0.4967	oui
NEWTON-CG	0.7500	1.7332	5	oui	0.4966	non
SAG	0.7500	0.0630	34	oui	0.4967	non
SAGA	0.7500	0.0294	16	oui	0.4967	oui

Table – Résultats détaillés sur dataset synthétique

- **Acc** : Accuracy (précision)
- **Temps (s)** : Temps d'entraînement en secondes
- **Itér** : Nombre d'itérations
- **Conv** : Convergence ( oui = convergence, non = pas de convergence)
- **Log Loss** : Perte logarithmique (plus bas = mieux)
- **L1** : Support de la régularisation L1 ( oui = prend en compte, non = ne prend pas en compte)

# Merci pour votre attention !

## Questions ?

Groupe 4 - INF4127

Université de Yaoundé I