

## **Rapport de comparaison des algorithmes FregSigCom et penFoFLPS**

Nous avons comparé deux algorithmes de régression sur le jeu de données de cyclisme.

Nous avons mis en place des modèles de régression multiples pour prédire la variable WATTS en fonction des autres variables. Pour chacun des algorithmes, nous avons utilisé les paramètres par défaut.

Après la mise en place des modèles, les erreurs quadratique moyennes (MSE) des prédictions obtenues avec les deux algorithmes sont les suivantes :

FregSigCom : 14143.1917335382

penFoFLPS : 459.498

Les résultats montrent que l'algorithme penFoFLPS a une MSE significativement plus faible que l'algorithme FregSigCom. Cela signifie que penFoFLPS est plus performant et plus précis dans ses prédictions.

Cette différence de performance est probablement due à la pénalité de somme des carrés de la pente utilisée par penFoFLPS. Cette pénalité permet de réduire le nombre de variables non significatives incluses dans le modèle. En conséquence, penFoFLPS est moins susceptible de surajuster les données et de faire des prédictions plus précises.

Sur la base des résultats obtenus, nous pouvons conclure que l'algorithme penFoFLPS est préférable à l'algorithme FregSigCom pour la prédiction de la puissance en cyclisme.

## Difficulté rencontré :

Nous n'arrivons pas à installer les library sur nos machines particulièrement la library FregSigCom qui ne marche que sur un seul ordinateur parmi les 4. Impossible de visualiser le jeux de données, la fonction n'acceptant que des objets, toutes transformations du jeux de données dénatures la structure des données.

## Structure avant transformations :

Data	
📄 cycling	Large list (9 elements, 28 MB) 🔍
\$ SECS :	num [1:216, 1:1800] 0 0 0 0 0 0 0 0 0 0 0 ...
\$ KM :	num [1:216, 1:1800] 0 0 0 0 0 0 0 0 0 0 0 ...
\$ WATTS:	num [1:216, 1:1800] 0 0 508 106 37 0 248 0 0 72 ...
\$ CAD :	num [1:216, 1:1800] 0 0 101 74 72 0 81 0 0 54 ...
\$ KPH :	num [1:216, 1:1800] 0 11.3 24.4 16.7 17.3 ...
\$ HR :	num [1:216, 1:1800] 112 77 122 82 0 77 148 79 87 76 ...
\$ ALT :	num [1:216, 1:1800] 305 201 199 221 163 ...
\$ SLOPE:	num [1:216, 1:1800] 0 0 0 0 0 0 0 0 0 0 0 ...
\$ TEMP :	num [1:216, 1:1800] 18 15 17 15 15 17 37 22 15 20 ...

## Structure après transformations :

\$ TEMP :	num [1:216, 1:1800] 18 15 17 15 15 17 37 22 15 20 ...
📄 data	216 obs. of 16200 variables 📊
\$ SECS.1 :	num 0 0 0 0 0 0 0 0 0 0 0 ...
\$ SECS.2 :	num 1 1 1 1 1 1 1 1 1 1 1 ...
\$ SECS.3 :	num 2 2 2 2 2 2 2 2 2 2 2 ...
\$ SECS.4 :	num 3 3 3 3 3 3 3 3 3 3 3 ...
\$ SECS.5 :	num 4 4 4 4 4 4 4 4 4 4 4 ...
\$ SECS.6 :	num 5 5 5 5 5 5 5 5 5 5 5 ...
\$ SECS.7 :	num 6 6 6 6 6 6 6 6 6 6 6 ...
\$ SECS.8 :	num 7 7 7 7 7 7 7 7 7 7 7 ...
\$ SECS.9 :	num 8 8 8 8 8 8 8 8 8 8 8 ...
\$ SECS.10 :	num 9 9 9 9 9 9 9 9 9 9 9 ...
\$ SECS.11 :	num 10 10 10 10 10 10 10 10 10 10 10 ...
\$ SECS.12 :	num 11 11 11 11 11 11 11 11 11 11 11 ...
\$ SECS.13 :	num 12 12 12 12 12 12 12 12 12 12 12 ...
\$ SECS.14 :	num 13 13 13 13 13 13 13 13 13 13 13 ...
\$ SECS.15 :	num 14 14 14 14 14 14 14 14 14 14 14 ...
\$ SECS.16 :	num 15 15 15 15 15 15 15 15 15 15 15 ...
\$ SECS.17 :	num 16 16 16 16 16 16 16 16 16 16 16 ...
\$ SECS.18 :	num 17 17 17 17 17 17 17 17 17 17 17 ...
\$ SECS.19 :	num 18 18 18 18 18 18 18 18 18 18 18 ...
\$ SECS.20 :	num 19 19 19 19 19 19 19 19 19 19 19 ...

Nous avons également essayer chercher à trouver des ressources sur internet mais nous n'avons rien trouvé malheureusement.

