Prévision des coefficients dynamiques

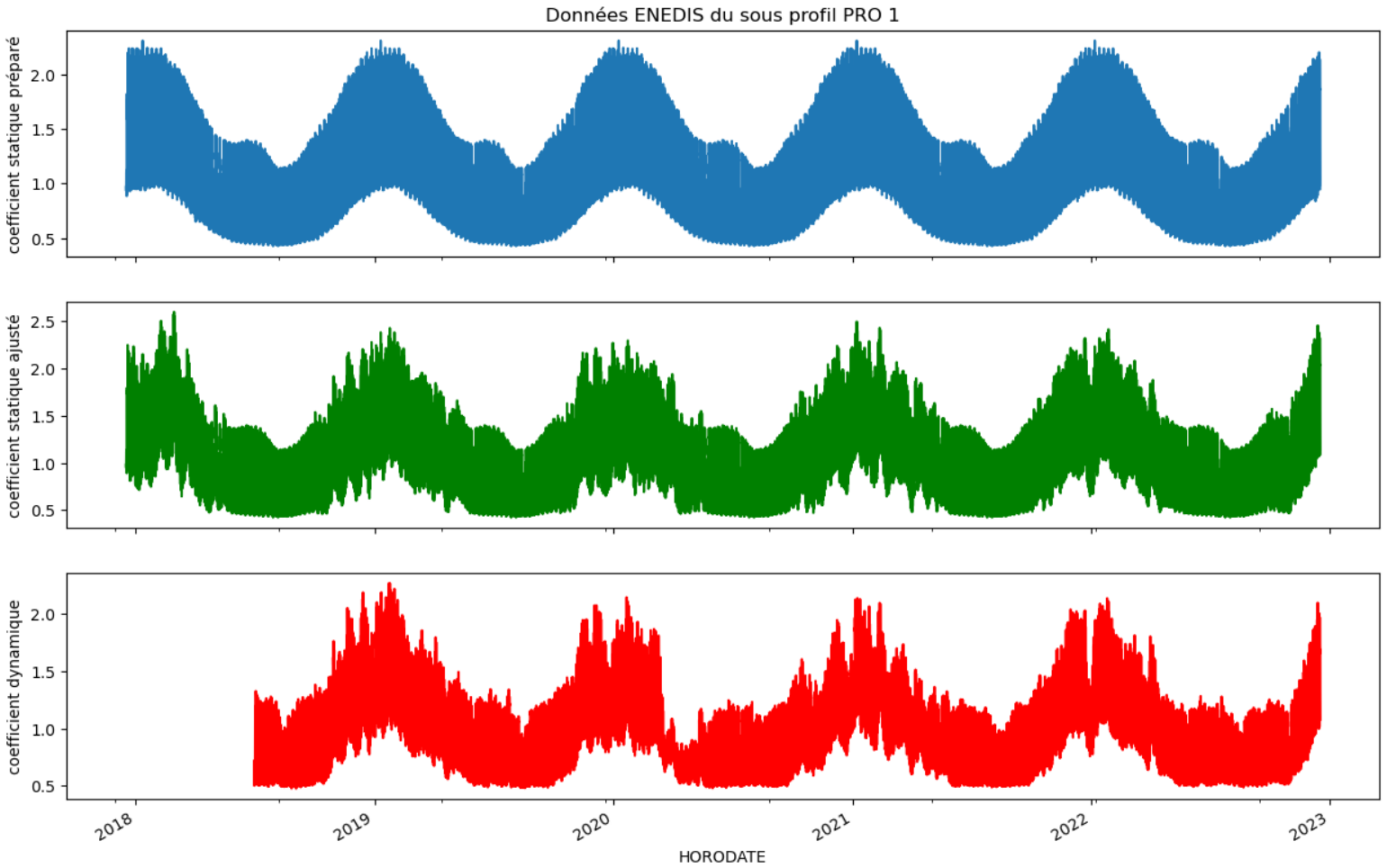
Machine learning – Alexandre Castanié

- Github code : https://github.com/CastanieAlexandre/Analyse-coefficients-dynamiques

ENEDIS met à disposition des données historiques de coefficients dynamiques. Est-il alors possible de prévoir en utilisant un algorithme de Machine Learning les coefficients dynamiques sur les années à venir ? Ces quelques pages résument rapidement les résultats que j'ai pu obtenir. On se concentrera uniquement sur le sous profil de consommation PRO1 P1. Je m'excuse pour les graphiques matplotlib suivants qui ne sont pas très beaux mais ils ont l'avantage de s'afficher sur github. Je compte créer un dashboard via streamlit si ce travail aboutit.

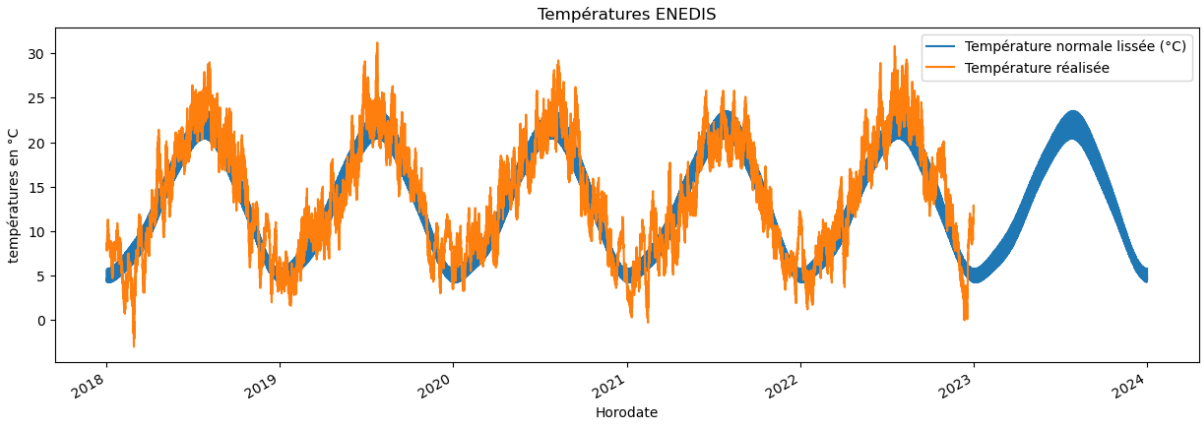
1. Les données

ENEDIS met à disposition des coefficients dynamiques mais également des coefficients préparés et ajustés.

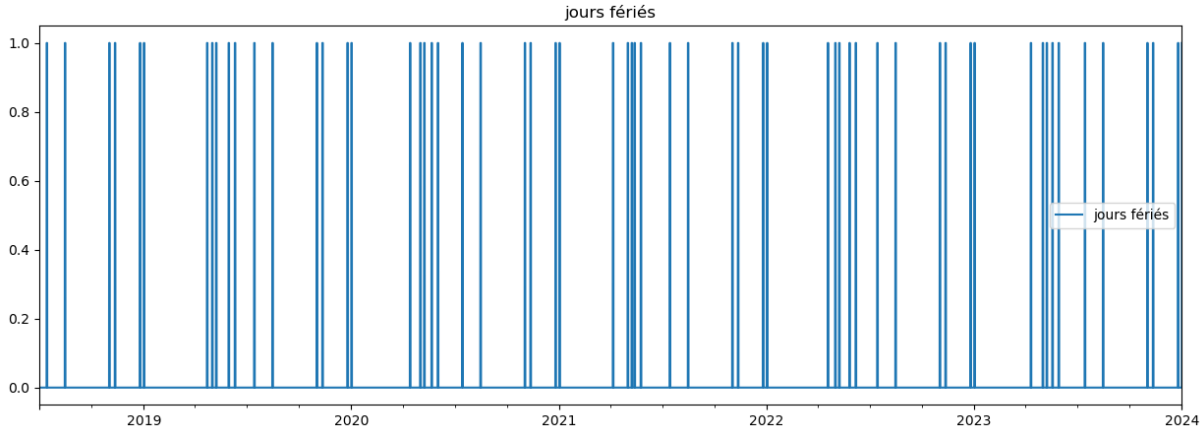


Je n'ai pas détecté d'outliers en particulier sur ces données. En revanche il manque certaines valeurs de coefficients dynamiques en 2020 d'où la forme singulière sur cette année.

ENEDIS fournit également les températures réalisées et lissées :



J'ai récupéré les jours fériés sur le site du gouvernement data.gouv.fr :



2. Entrainement

J'ai utilisé l'algorithme **XGBoost.**

J'ai d'abord cherché à optimiser les hyperparamètres. Pour cela j'ai utilisé un algorithme d'optimisation qui va tester au hasard des combinaisons d'hyperparamètres et renvoyer celle qui minimise les erreurs de prévision i.e. l'écart quadratique moyen (RMSE) :

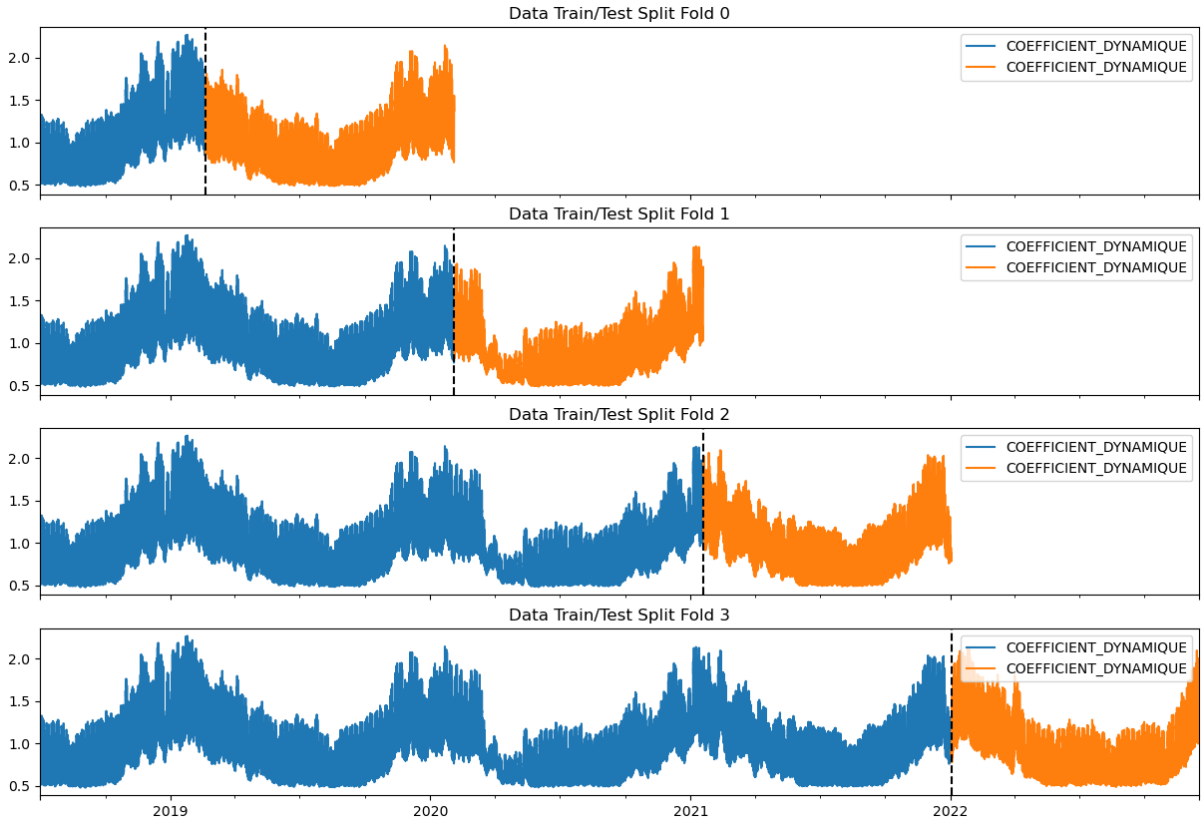
Best parameters: {'subsample': 0.8999999999999999, 'n_estimators': 400, 'max_depth': 10, 'eta': 0.3, 'colsample_bytree': 0.8999999999999999, 'base_score': 0.5}

Lowest RMSE: 0.08515488256853712

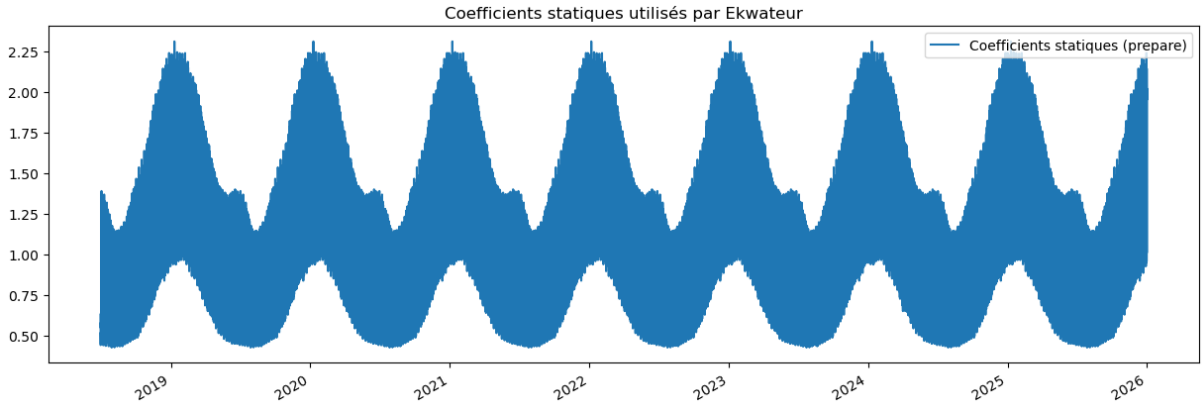
A partir de ces valeurs j'ai calibré le modèle en précisant les hyperparamètres par cross-validation en essayant de minimiser le RMSE. J'obtiens les valeurs suivantes :

```python
reg=xgb.XGBRegressor(objective="reg:squarederror",
                     eval_metric="rmse",
                     eta=0.5,
                     max_depth=14,
                     #min_child_weight=0,
                     #max_delta_step=0,
                     gamma=0.0,
                     subsample=1,
                     sampling_method='uniform',
                     colsample_bytree=1,
                     reg_alpha=0,
                     reg_lambda=1,
                     tree_method='hist',
                     booster='gbtree',
                     n_estimators=400,
                     base_score=1
                     )
```

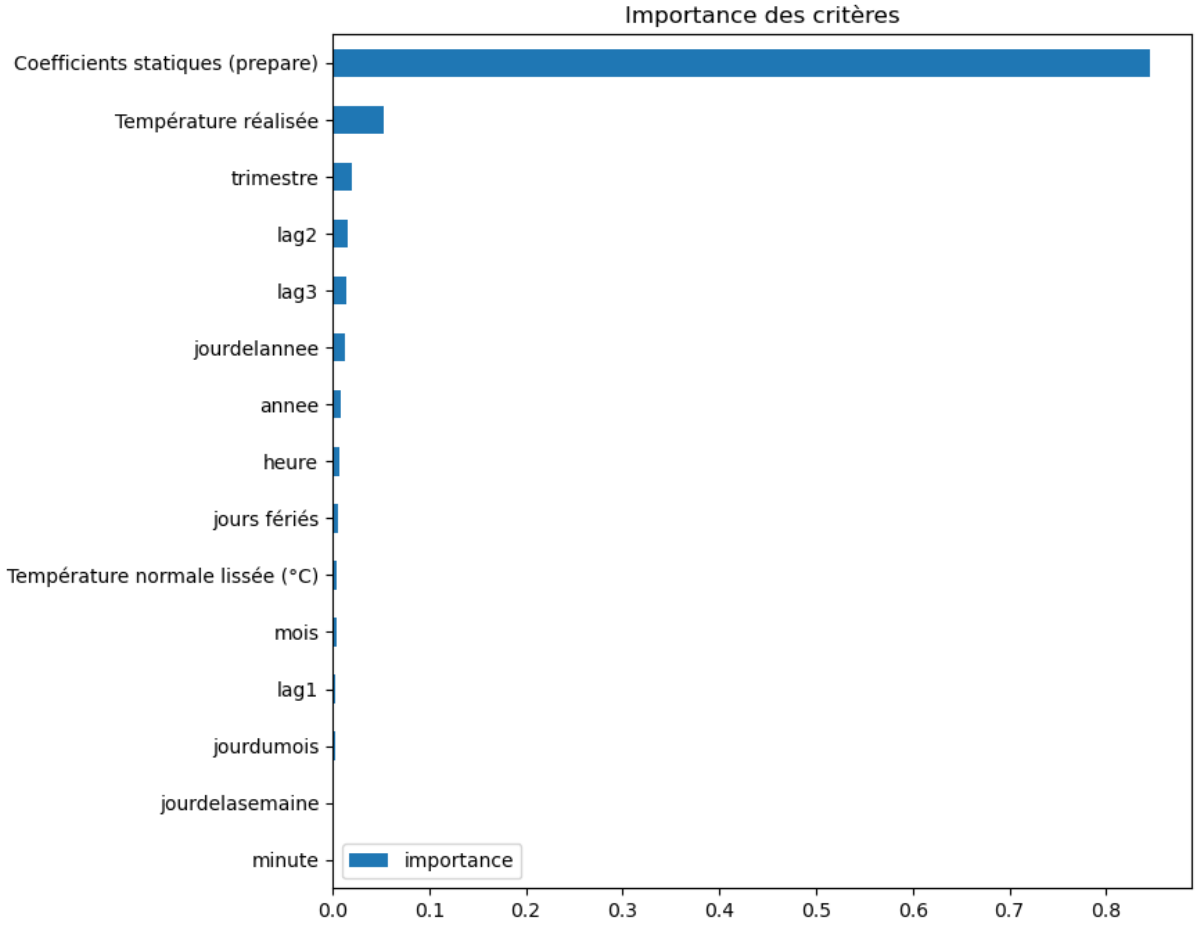
La cross validation départage les données de la manière suivante :



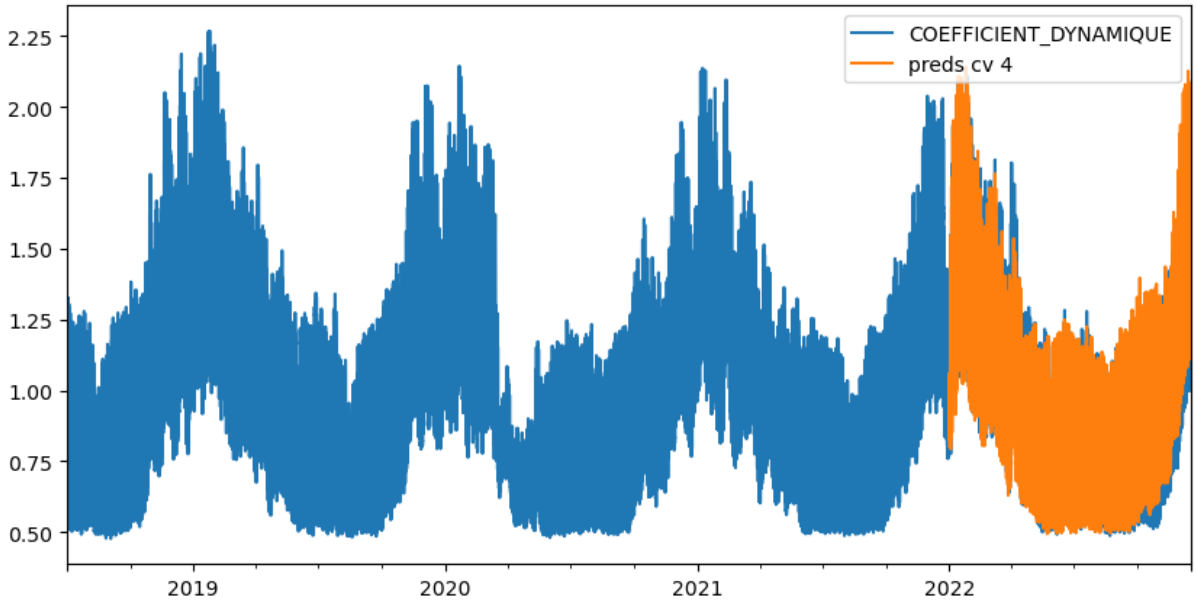
Les résultats n'étaient pas satisfaisants. Je me suis alors fait la réflexion suivante : pourquoi ne pas utiliser les coefficients statiques dont les prévisions dans le futur sont déjà faites pour entraîner mon modèle ?



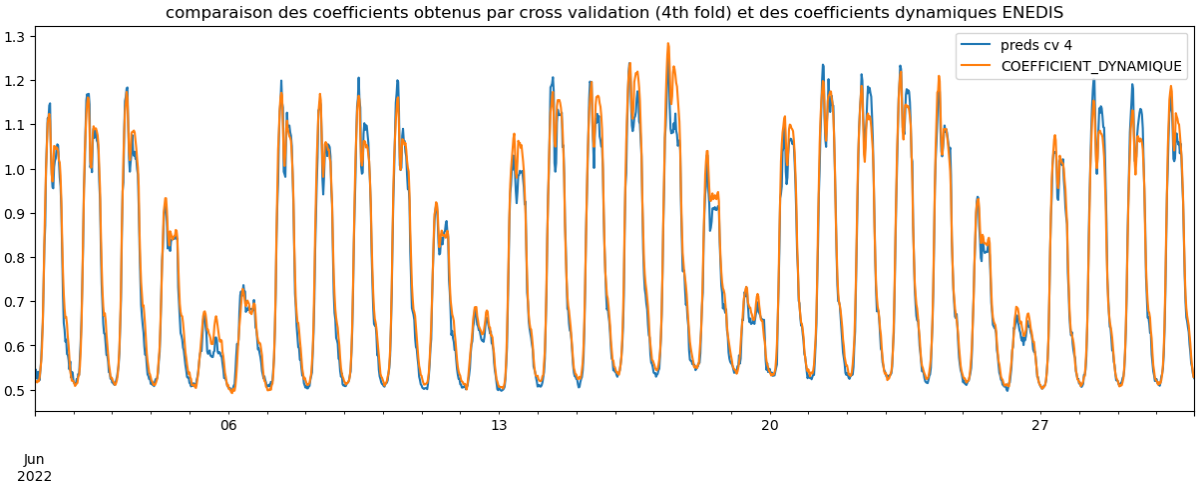
Le graphique ci-dessus est la combinaison des coefficients statiques prévisionnels d'Ekwateur et ceux des années précédentes d'ENEDIS. L'entrainement donne les résultats suivants :



Les coefficients statiques dominent entièrement les autres critères. Cela ne me surprend cependant pas car ils contiennent quasiment toutes les autres informations.



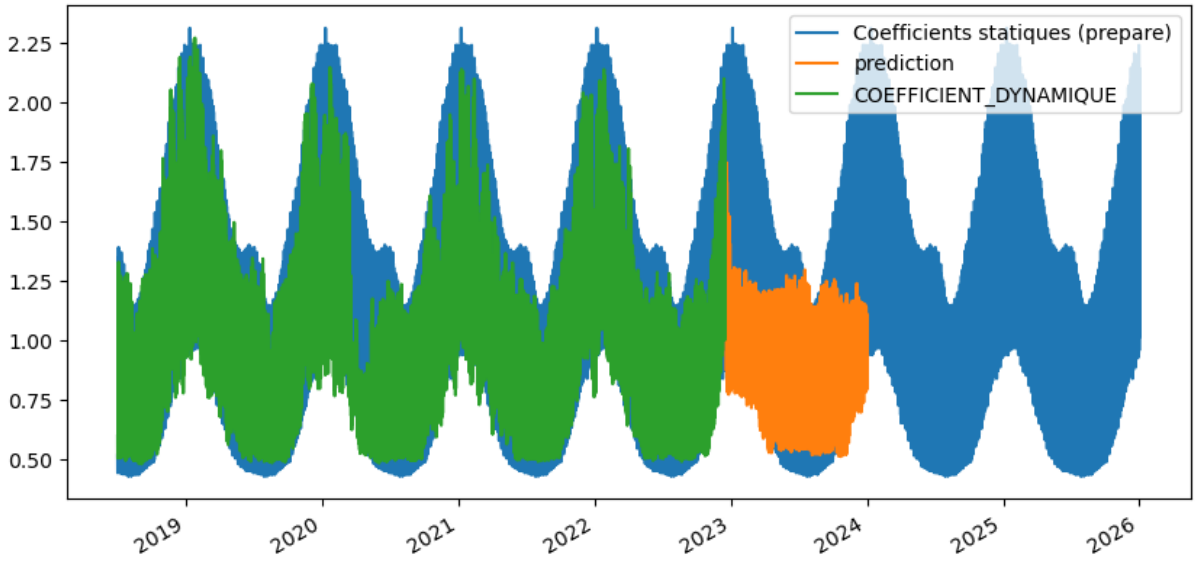
La prédiction sur 2022 est globalement bonne. Elle déraille cependant sur le pic du mois d'avril et en fin d'année. En revanche, sur certains mois, le modèle n'est vraiment pas loin comme par exemple en juin :



On a une erreur globale RMSE de 0.071 ce qui est assez élevé quand même. Je n'ai pas réussi à avoir mieux.

3. Prédiction

C'est là que le bât baisse : La prédiction sur l'année 2023 donne des résultats très mauvais comme en témoigne l'image suivante :



N’étant pas un expert dans l’analyse de donnée, je suis persuadé qu’il est possible d’améliorer mon résultat pour avoir quelque chose d’exploitable. Je pense par exemple que je me suis trompé dans le split pour la cross-validation.