Un UI et server par page

Les pages :

1. Présentation des données et du traitement pour les obtenir -> Shiny
2. Approche générale :

* Aggrégation des données par jour et heure -> Count, somme des poids et des distances entre essieux, température moyenne et vitesse moyenne
* Missing values : dates et heures avec des trous, système peu fiable. Méthode d’inputation des données au niveau aggrégée : création des dates et heures manquantes, puis interpolation linéaire ou spline
* Plot resultats de l’interpolation
* Box plots par variables catégorielles du temps
* Modèles classiques : augmented dickey-fully pour stationnarité, ACF et PACF plot pour les ordres AR et MA. ARIMA. Holtwinters. Décomposition temporelle. TBATS avec double seasonalité.
* Modèles linéares de base -> cross validation via time slice
* Séléction de variables de lags par stepwise (original et full)

modèles classiques, ARIMA, Holtwinters, décompostion temporelle, Cross validation, missing values, features seletion -> Rmarkdown

1. Modèles sur le train set, comparaison des performances (acccuracy) -> shiny

Original et full, modèles linéaires simple et avec interactions, modèle linéaire avec sélection de lags, modèle pénalisée, modèle random forest, modèle gradient boosting.

Plot du réel vs chaque prédiction du modèle sur le dataset d’apprentissage

Importance des variables pour random forest, gradient boosting.

1. Prédictions sur le test set, comparaison des performances -> shiny

Plot du réel vs chaque prédiction du modèle sur le dataset de test (modèles original et full)

Dygraph + dashboard shiny

Web scraping : explication des étapes avec des captures d’écran