UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

MASTER 1 ÉCONOMÉTRIE STATISTIQUES

2024/2025

Prédiction PRMs
Aéroport d'Edimbourg

Réalisé par:

Anne-Elisabeth Makosso
Christana Adefolami
Hanaa Talbi

Supervisé par:

M. Armand L'Huillier
M. Aryan Razaghi

Langage de programmation :

Python

Mai 2025

SYNTHÈSE

Afin de réaliser ce projet de prédiction du nombre de passagers à mobilité réduite (PRMs) par vol. On a utilisé la base de données **'PRMAggregated'** qui contient 112 860 enregistrements de données de vols avec 6 variables principales (MaxPax_MostConfident, Passengers, Avg_Seat_Load_Factor, count of FlightPerformance, Day_of_ScheduledDateTime_Local, TIME) nous avons décomposé notre travail en 4 parties :

1 Analyse exporatoire des données

1.1 Traitement des valeurs manquantes

En analysant notre base de données, nous avons constaté que les variables numériques **(PRMs, Avg. Seat Load Factor, Passengers, MaxPax MostConfident et Count of FlightPerformance)**.

Notre stratégie consistait d'abord à supprimer les enregistrements où toutes les variables clés sont manquantes.

La variable PRMs, qui représente notre variable cible, contenait le taux le plus élevé de valeurs manquantes (33,16%), soit environ 1/3 des données absentes, et il nous a paru judicieux de remplacer ces valeurs manquantes simplement par des 0, en considérant que l'absence de données indiquait en réalité l'absence de passagers à mobilité réduite sur ces vols. Grâce à l'histogramme post-imputation, on peut voir une distribution fortement concentrée sur les faibles valeurs avec une décroissance exponentielle (pic à 0-2 PRMs) avec une moyenne de 4 passagers à mobilité réduite par vol.

Quant aux trois autres variables numériques, qui interagissent entre elles **(Avg. Seat Load Factor, Passengers, MaxPax MostConfident)**, leur taux de valeurs manquantes était de 22%. Nous avons donc pu traiter les NaN en utilisant les relations mathématiques entre ces variables (imputation croisée), permettant ainsi le maintien de la logique opérationnelle des données. On a conclu par une élimination des dernières valeurs manquantes résiduelles.

Résultats:

- Une base de données complète et cohérente.
- Préservation de 78% des données originales.
- Relations causales maintenues entre Passengers, MaxPax et Load Factor.

1.2 Traitement des outliers

Pour le traitement et l'analyse des outliers qui portent sur les quatre variables numériques:

- MaxPax_MostConfident : Capacité maximale de passagers
- Passengers : Nombre réel de passagers
- Avg_Seat_Load_Factor : Taux de remplissage moyen
- PRMs : Passagers à mobilité réduite

Nous avons mené une analyse en deux étapes principales : la détection univariée avec la méthode IQR et les boxplots, puis la détection multivariée avec Isolation Forest.

Dans un premier temps, en appliquant **la méthode IQR** à chaque variable numérique, nous avons constaté que MaxPax_MostConfident présentait près de 1,91% de valeurs aberrantes liées à des capacités très élevées (jusqu'à 2319 passagers), ce qui révèle l'existence d'aéronefs de très grande capacité.

Passengers, de son côté, présente 1.74% d'outliers (jusqu'à 2194 passagers). Ces vols à très forte affluence peuvent être interprétés comme des vols associés aux périodes de pointe (vacances, événements spéciaux) ou aux destinations touristiques. Le fait que le maximum observé ici(2194) soit inférieur à la capacité maximale (2319) indique une utilisation optimale mais non saturée des gros porteurs.

Pour Seat_Load_Factor, environ 3.65% du dataset, notamment des valeurs supérieures à 1 (jusqu'à 1,14), ce qui est impossible logiquement car cette variable représente le rapport entre Passengers et MaxPax_MostConfident.

Enfin, la variable cible PRMs présentait environ 4,91% de valeurs extrêmes, mais cela semble refléter la réalité et peut-être s'expliquer par la nature spécifique des vols transportant des groupes de passagers à mobilité réduite (vols médicaux, transferts spécialisés...).

Ainsi, un traitement adapté a été appliqué à chaque variable. Nous avons employé la méthode de la transformation logarithmique afin d'atténuer l'influence des valeurs aberrantes sans forcément les supprimer. De plus, nous avons limité le taux de remplissage (Seat_Load_Factor) à 100%.

La méthode multivariée Isolation Forest nous a permis de confirmer les résultats précédents en détectant environ 3% d'observations atypiques correspondant aux réalités opérationnelles.

2 Feature Engineering

Nous avons créé plusieurs variables explicatives pour enrichir notre modèle :

- **Season_sector** : combine la saison et le type de vol (domestique/international), car les vols internationaux en été enregistrent souvent un nombre plus élevé de PRMs.
- **MomentDansLaJournée** : identifie le créneau horaire (matin, après-midi, soir), utile pour détecter des pics ou des préférences liées à l'affluence ou à l'organisation aéroportuaire.
- **PRMs_lag_1** : moyenne du nombre de PRMs la veille, permettant de repérer une continuité d'un jour à l'autre et de révéler des effets cachés (planification groupée, inertie, etc.).
- **Regions** : regroupe plusieurs variables géographiques afin de réduire la complexité du modèle tout en conservant l'information pertinente.
- **PRMs_cible** : recodage des valeurs de PRMs en classes par intervalles, ce qui permet de transformer le problème en classification. Ce choix est justifié par la forte dominance des cas où PRMs = 0 et la rareté des événements avec PRMs élevés.

3 Sélection de variables

Pour sélectionner les variables pertinentes, nous avons d'abord mené **une analyse univariée :** chaque variable a été testée individuellement pour évaluer son lien avec la cible. Ensuite, **une sélection multivariée** via un modèle **RandomForestClassifier** a permis d'identifier celles qui restent utiles en tenant compte des interactions entre variables. Cette double approche permet de retenir des variables à la fois informatives et complémentaires.

4 Modèle

- **XGBOOST:** est un algorithme de machine Learning supervisé basé sur les arbres de décision. Une fois qu'on a sélectionner les variables qui permettent le mieux d'expliquer notre modèle, on les incorpore dans cette algorithme pour obtenir les meilleurs résultats. On utilise également des hyper-paramètres pour optimiser au mieux les résultats notamment au travers des "gried search". De plus, en raison du déséquilibre qu'il existe entre les classes, nous avons décider d'ajouter un autre paramètre "sample weight".
- **CatBoost:** est aussi basé sur les arbres, mais il est particulièrement adapté aux variables catégorielles. Comme pour XGBoost, on ajuste les hyperparamètres. On a notamment utilisé "l2_leaf_reg" pour éviter le surapprentissage et mieux distinguer les classes proches, surtout entre 0 et 1.
- **XGBoost + SMOTE:** Avec ce modèle, on a ajouté SMOTE, une méthode qui permet de rééquilibrer les classes en générant des données pour la classe minoritaire. Le but est de mieux détecter les classes rares tout en profitant de la puissance de XGBoost.

En parallèle, on a utilisé **le filtre de Kalman** comme méthode complémentaire.On a d'abord séparé les données en deux parties : 80 pourcent pour l'entraînement, 20 pour le test.

On a créé une matrice X avec les variables utiles (passagers, saison, etc.) et une variable cible y qui correspond au log du nombre de PRMs, pour limiter l'impact des valeurs extrêmes.

Le filtre a été initialisé avec des coefficients à zéro et une incertitude forte. On suppose que les coefficients bougent lentement, donc on a mis un petit bruit dans la dynamique. Ensuite, à chaque pas de temps, le filtre met à jour les coefficients avec la donnée du jour. C'est une forme d'apprentissage en continu.Dans notre version, on n'a pas inclus l'étape de smoothing (qui permet d'utiliser aussi les infos futures pour lisser les estimations). On s'est contenté du filtrage pas à pas.Sur le test, on a utilisé le dernier état estimé pour faire les prédictions.

On a ensuite comparé avec les vraies valeurs à l'aide du MSE et du MAE. On a aussi visualisé les coefficients dans le temps ; On a aussi évalué les performances du modèle à l'aide d'une matrice de confusion, en comparant les classes prédites et les classes réelles. Mais cette méthode repose sur deux hypothèses fortes :une relation linéaire entre les variables et la cible,des erreurs qui suivent une distribution normale.Dans notre cas, ces hypothèses tiennent mal : le comportement des PRMs est souvent non linéaire, et les pics de demande ne sont pas normaux.

Le filtre détecte bien les tendances douces, mais pas les cas extrêmes. On a tenté (sans le finir) une version avec un filtre de Kalman étendu (EKF), qui permet de gérer la non-linéarité, mais faute de temps, on n'a pas pu aller plus loin. Les résultats sont dispos dans le notebook.