**Présentation Oral AIF**

**Slide 1 : Intro**

Bonjour, nous sommes Léa, Flavie et Lila et nous allons vous présenter notre projet sur le défi IA.

**Slide 2 : Plan**

Dans cette présentation, nous verrons :

* Quelle est stratégie de requêtage que nous avons employée et comment nous avons organisé nos codes.
* Nous parlerons ensuite du prétraitement des données afin de pouvoir appliquer les modèles de ML.
* Enfin, nous analyserons les résultats avec l’interprétabilité puis nous ferons une démonstration du Gradio et du Docker.

**Slide 3-6 : Stratégie de requêtage**

***Création de la matrice***

Pour explorer toutes les requêtes en évitant au maximum de requêter des doublons, nous créons une matrice contenant toutes les combinaisons possibles des paramètres *mobile*, *language* et *city*.

On parcourt 2 fois cette matrice en tirant au hasard une date parmi les 40 dates possibles.

***Exploration du jeu de test Kaggle***

On explore le jeu de test Kaggle pour identifier quels sont les paramètres à requêter de préférence. Avec cette analyse, on observe que :

* 1/3 des utilisateurs ont effectué des requêtes dans 2 langues différentes et seul 1 utilisateur a utilisé 3 langues

→ on va donc d’abord requêter 1 langue/utilisateur puis augmenter ce nb ensuite.

* La moitié des utilisateurs ont utilisé un ordinateur et l’autre moitié un mobile.

→ on requête ces 2 modalités en proportions égales.

Avec les graphiques, on voit que :

* Les modalités de city sont présentes de façon homogène et après un test d’indépendance du chi2 on observe que language a peu d’influence sur le prix.

→ Donc on requête uniformément les modalités de ces variables

* La grande majorité utilisateurs n’ont fait qu’1 requête.

→ On commence par faire 1 requête/utilisateur puis on augmentera ce nb par la suite.

Enfin,

* On voit que les dates des requêtes sont groupées en 3 périodes.

→ On requête donc préférentiellement selon ces dates puis on diversifie ensuite.

*Test Chi2:* *Le test d'hypothèse statistique du chi2 de Pearson est un exemple de test d'indépendance entre des variables catégorielles. Les résultats de ce test peuvent être utilisés pour la sélection des variables, où les variables qui sont indépendantes de la variable target peuvent être retirées de l'ensemble de données. Plus le Chi2-score est élevé et plus la feature est importante*

***Affinage de la stratégie de requêtage***

Pour affiner davantage notre stratégie de requêtage, nous effectuons des premières prédictions avec les modèles que l’on présentera par la suite puis nous observons où ces modèles sont le plus trompés.

Par exemple, on voit ici que la date 0, ainsi que les groupes de dates 1 à 8 jours et 14 à 22 jours sont celles pour lesquelles le modèle se trompe le plus et donc on requête à nouveau pour ces valeurs.

De même, on voit que le modèle n’arrive pas bien à prédire la langue Tchèque et donc on requête + cette langue.

**Slide 7 : Organisation des fichiers**

Pour l’organisation du travail de grupe, nous avons créé un fichier *set\_path.py* contenant toutes les variables globales et les chemins des dossiers.

→ *set\_path.py* est appelé par tous les autres fichiers. En le modifiant, on peut adapter le code à une autre machine.

On voit ici l’architecture de nos fichiers.

**Slide 8 : Prétraitement des données**

***Récupération et mise en forme des données***

* Nous commençons par concaténer : l’ensemble des requêtes réalisées, les features des hôtels (hotel\_id, group, brand…) données dans le défi, et le nb de requêtes/utilisateur que nous avons calculé.
* Nous assignons le bon type aux variables (qualitatives ou quantitatives)
* Nous supprimons les requêtes en double
* Puis nous transformons les variables price et stock pour les rendre + gaussiennes.

***Ajout de features extérieures***

Par la suite, pour tenter d’améliorer nos prédictions, nous ajoutons des features extérieures que nous pensons influentes sur le prix d’une chambre d’hôtel :

Le PIB du pays, le prix moyen / m2 dans les villes, le nb de touristes moyen / an par ville et le nb d’habitants par km2 par ville.

***Création du jeu d’entraînement***

Nous créons à présent le jeu d’entraînement. Comme nous avons beaucoup de données, nous utilisons 60% pour le train set, et 20% pour le set de validation et test.

***Encoding des variables qualitatives***

*One-hot encoding* : dans un premier, nous avons fait du one-hot encoding sur les variables qualitatives en mettant drop\_first=True pour éviter les redondances.

Cependant, cela crée de nombreuses colonnes notamment pour la variable *hotel\_id* qui a beaucoup de modalités et pose est déconseillé pour les méthodes basées sur les arbres.

*Target & Frequency* *encoding* : On fait ensuite du target encoding pour les variables qui ont + de 3 modalités et du frequency encoding pour les variables qui 3 modalités ou moins.

Le target encoding consiste à remplacer la modalité d’une variable qualitative par la valeur moyenne que prend la target pour cette modalité. Le frequency encoding consiste à remplacer la modalité d’une variable qualitative par sa fréquence.

Ce type d’encoding est calculé sur le set de train et appliqué au set de train, de validation et de test. Il est approprié aux modèles basés sur les arbres mais peut entraîner de l’Overfitting.

*Rq : Si on utilise le one-hot encoding pour les méthodes basées sur les arbres, il est conseillé de ne pas mettre l’option drop first=True, de manière `a ne pas perdre d’information pour les splits. De plus les splits sont très restrictifs pour les variables codées en one-encoding (on va à gauche si c’est d’une modalité et à droite pour toutes les autres modalités).*