Rapport de projet

Cloud Computing - M1 IoT

H3 Hitema - 01/02/2021

Table des matières

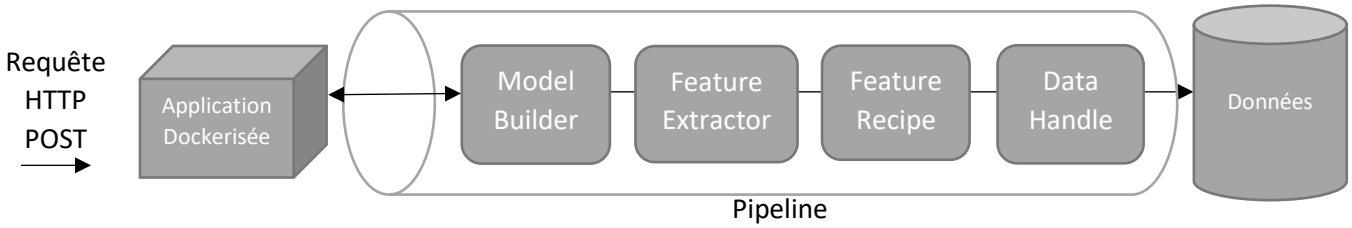
Contexte

Le projet consiste à présenter en tant qu'**Ingénieur junior en Machine** Learning pour un client une mini **pipeline** allant de la collecte des données au développement d'une **API** en passant par le paquetage **Docker** ainsi que les tests et la rédaction d'une documentation.

Celui-ci est rendu disponible sur un répertoire **Git** public avec un document **README.md** rédigé et sera déployé et hébergé sur une solution en **Cloud**, sur un serveur distant.

Le pipeline

Le pipeline de machine learning se présente sous la forme suivante :



Celle-ci est composée des éléments suivants :

- Data Handler :
  - Récupération des données (ici dans un Google Cloud Storage)
  - Mise en commun des datasets
- Feature Recipe :
  - Identification des types de données
  - Suppression des informations inutiles
  - Gestion des duplicatas
  - Suppression des éléments vides selon un seuil
- Feature Extractor :
  - Suppression des informations non utilisées dans le cadre de l'étude
  - Séparation des données en modèles d'entrainement et de test
- Model Builder :
  - Prédiction
  - Sauvegarde du model
  - Affichage de la précision
  - Chargement du model
  - Entrainement des données

Collecte des données

Les données utilisées dans ce projet proviennent du **Google Cloud Storage** (GCS) et concernent les caractéristiques des logements proposées par l'application **Airbnb** ainsi que leurs informations de réservation passées :

- **listing_final.csv**, qui répertorie un ensemble de logements avec leurs informations : *https://storage.googleapis.com/h3-data/listings_final.csv*
- **price_availability.csv**, qui retrace les informations de locations de ces logements : *https://storage.googleapis.com/h3-data/price_availability.csv*

La réception de ces données dans le pipeline se fait via le **Data Handler**.

L'API

L'API se base sur **l'algorithme de régression linéaire** afin d'effectuer une **prédiction** sur le **prix d'un logement** selon les critères suivants :

- *La latitude*
- *La longitude*
- *Le facteur de prix hebdomadaire*
- *Le facteur de prix mensuel*
- *La capacité (nombre de personnes)*
- *Le nombre de salles de bain*
- *Le nombre de chambres*
- *Le nombre de lits*

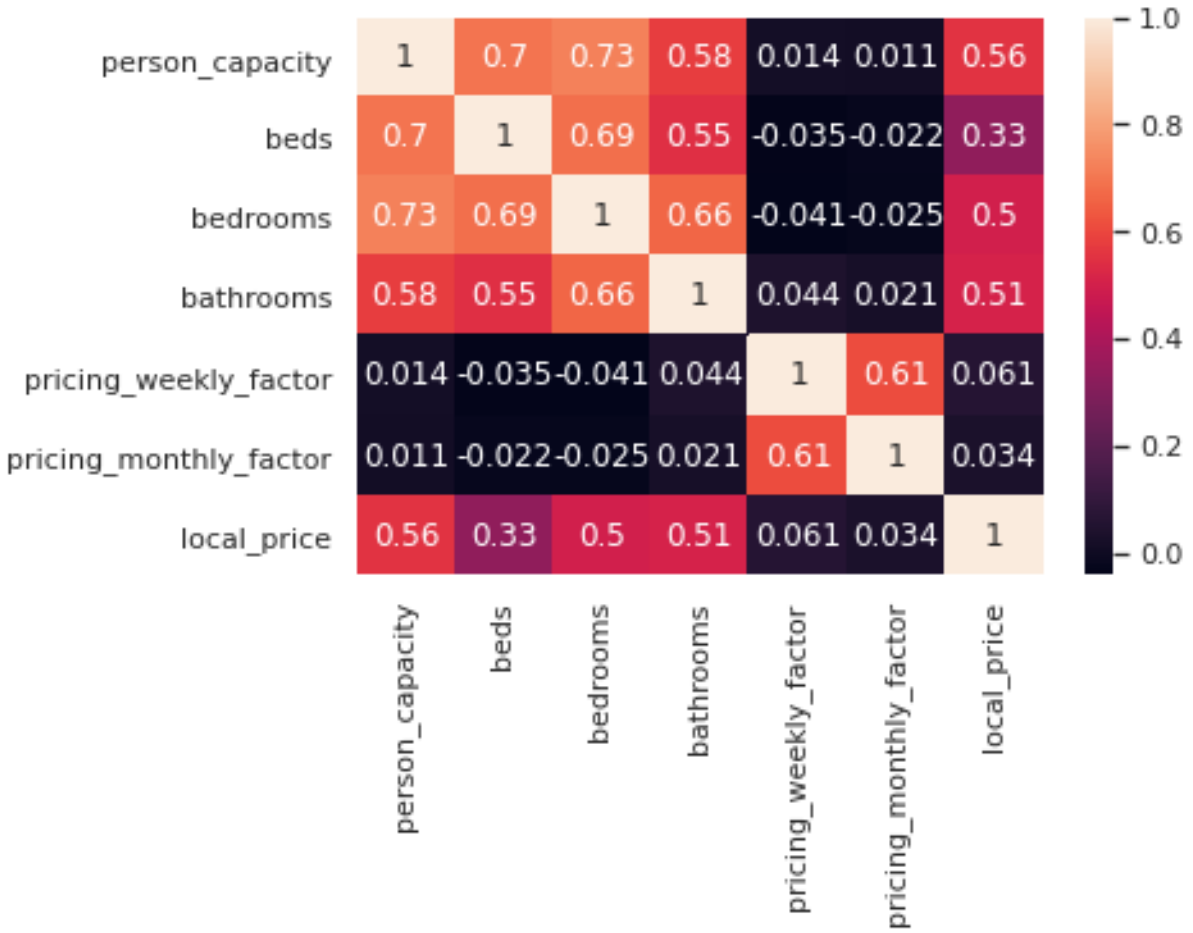
Cette API dispose d'un endpoint « ***/predict*** » permettant d'envoyer à l'application via une requête **HTTP POST** les différents critères accompagnés de leurs valeurs respectives dans le but de **prédire le prix** estimé du logement concerné par les informations saisies.

Enfin, grâce à **FastAPI**, L'API propose un endpoint « ***/docs*** » permettant d'accéder à une interface d'administration de l'ensemble des points de terminaison (ici, notre unique endpoint « /predict ») afin d'y faire appel, notamment ceux nécessitant un **body** comme les requêtes HTTP POST dans lesquelles il est possible d'**envoyer des informations** à notre application directement via le navigateur web sans avoir besoin d'effectuer la manipulation en ligne de commande ou en passant par un outils tiers tel que Postman.

Lien vers le endpoint « /docs » : *https://ml-template-api.herokuapp.com/docs*

Analyse des données

La **Heatmap** ci-dessous permet d'analyser le coefficient de **corrélation** entre les différents attributs des données. Plus le coefficient s'**approche de 1** et plus les attributs à l'intersection de ce coefficient sont corrélés :



Nous pouvons prendre les corrélations les plus pertinentes et constater que le coefficient est :

- De **0.73** entre la **capacité de personnes** et le **nombre de chambres**
- De **0.70** entre la **capacité de personnes** et le **nombre de lits**
- De **0.66** entre le **nombre de lits** et le **nombre de chambres**

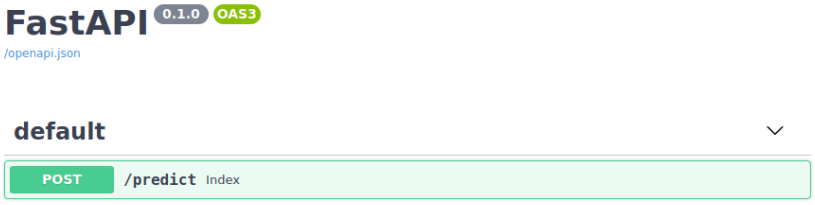
Paquetage Docker

L'**image** Docker utilisée par notre conteneur se base sur l'image **Uvicon-Gunicorn-FastAPI Python 3.7**. Le conteneur Docker contient les diverses librairies utiles telles que Numpy et Pandas. Il a été déployé sur un serveur distant **Heroku** à l'adresse suivante :

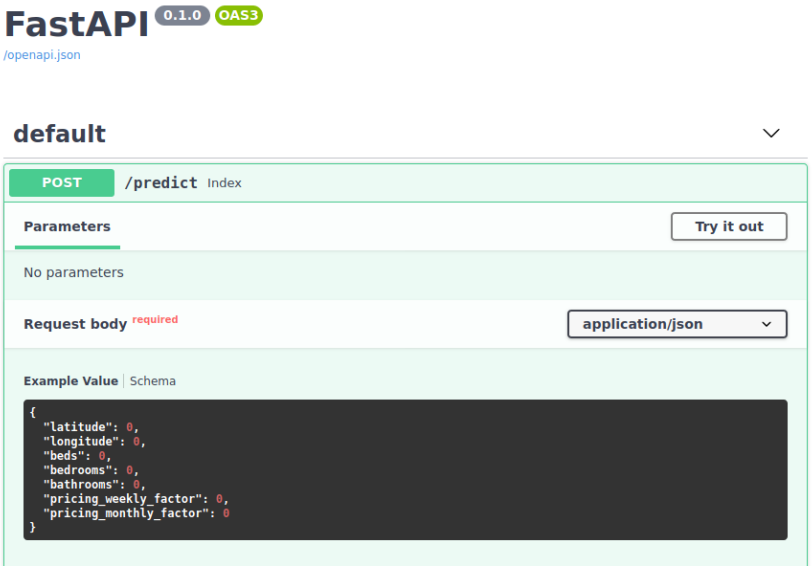
https://ml-template-api.herokuapp.com

Test de l'application finale

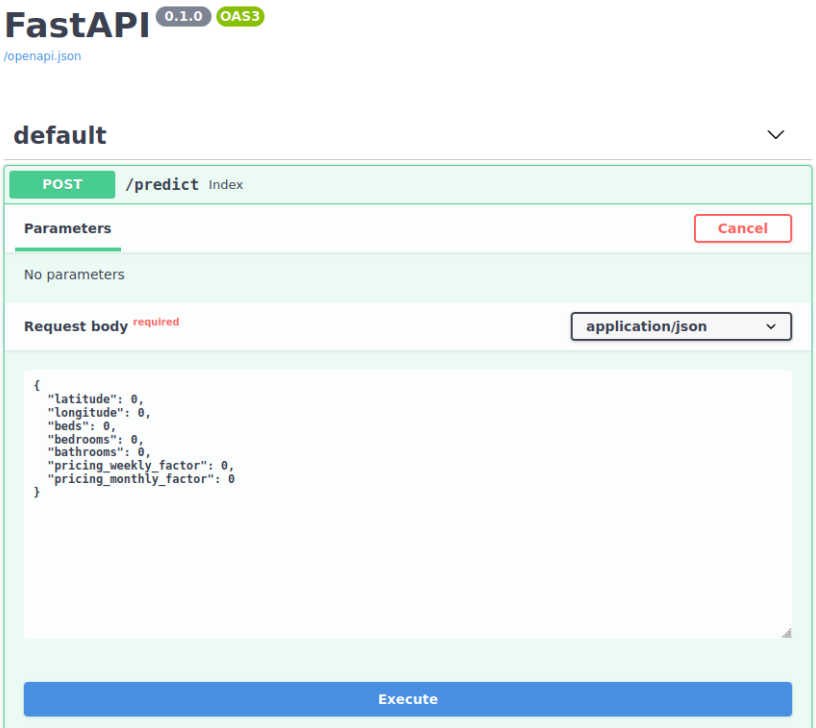
Pour tester l'application, rendez-vous sur l'interface de gestion de l'API via le endpoint « /docs ». Un seul endpoint sera disponible : le « /predict » :



En cliquant sur ce endpoint, un menu sera déroulé avec des information concernant ce dernier :



Cliquez sur "Try it out" pour essayer ce endpoint, saisissez les valeurs des paramètres demandés et cliquez sur "Execute" :



L'API retourne l'estimation du prix du logement en fonction des critères saisis.