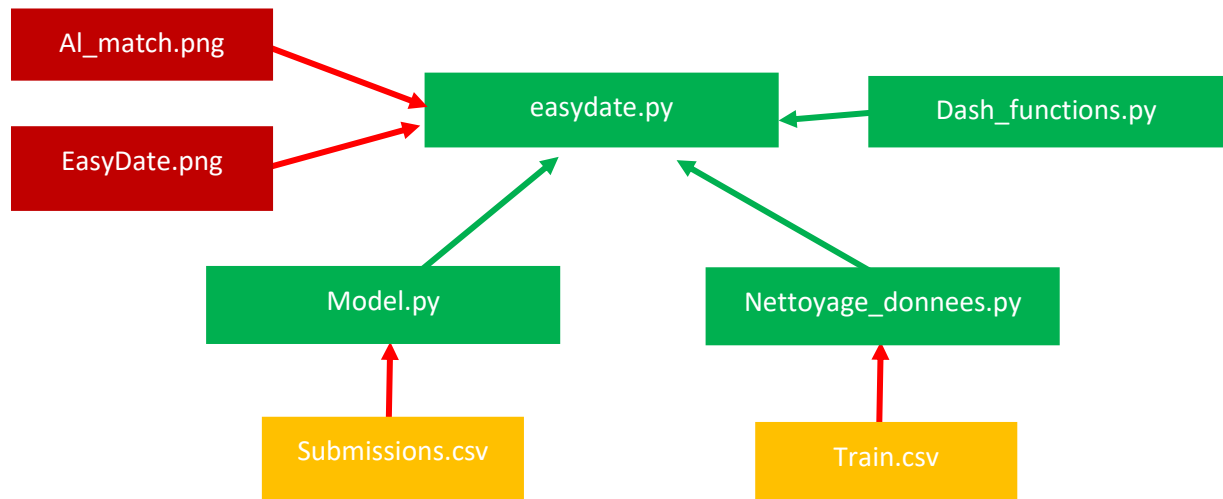


Documentation technique

Dans le cadre de ce projet nous avons décidé de créer plusieurs fichiers. Nous avons ensuite appelé certains fichiers dans d'autres afin d'utiliser les fonctions. Cela nous a permis de mieux nous organiser et d'avoir des codes plus compréhensibles et plus condensés.

Les relations entre les différents fichiers sont les suivantes :



- ➔ Appel grâce à un chemin
- ➡ Appel d'une variable de fichier parent grâce à import
- Document python
- Document csv
- Document png

Les fichiers png

Les fichiers png contiennent les logos et un graphique qui apparaissent dans notre application.

Les fichiers csv

Les fichiers csv contiennent nos données d'apprentissage (`train.csv`) et nos données à prédire (`submissions.csv`)

Les fichiers python

1. Nettoyage

Le premier fichier python que nous avons conçu lors de la conception de notre modèle a été le fichier `Nettoyage_donnees.py`.

Ce fichier nous a servi à nettoyer les données d'apprentissage. Nous avons tout d'abord importé notre base d'apprentissage puis créé une fonction qui permet de nettoyer la base d'apprentissage mais aussi de créer de nouvelles variables ou d'enlever celles qui ne sont pas présentes dans la base de données à prédire.

2. Modèle

Nous nous sommes ensuite intéressés à la création du modèle, nous l'avons conçu dans le fichier `model.py`.

Pour concevoir le meilleur modèle nous avons récupéré la base de données propre que nous venions de créer. Nous avons :

- Séparer nos données en deux parties, les données train (80% des données) et les données test (20%).
- Créer des fonctions pour supprimer les données manquantes, mais aussi séparer la variable à expliquer et les variables explicatives.
- Défini l'instance de Smote (technique de suréchantillonnage des observations minoritaires), puis nous l'avons appliqué sur l'ensemble des variables de nos données train.
- Conçu notre modèle de KNN Classifier (`knn`), que nous avons ensuite testé et qui nous a aidé à prédire la variable à expliquer de notre base de test.
- Créer une fonction qui évalue le f1 score de notre modèle.
- Charger les données à prédire (submissions) et créer une fonction qui permet gérer certains outliers mais aussi créer de nouvelles variables (à l'aide de celle de notre base de données) qui sont essentielles pour utiliser notre modèle.
- Nous avons ensuite prédit les matchs de notre base submissions.

3. Application Dash

Nous avons ensuite décidé de nous occuper de l'application Dash. Pour ce faire nous avons dans un premier temps créé le fichier `Dash_function` puis le fichier `easydate.py`.

Le fichier `Dash_function` regroupe l'ensemble des fonctions qui nous ont permis de créer nos différents graphiques et tableaux sur l'application mais aussi la fonction permettant l'affichage des logos sur notre application.

Le fichier `easydate` est le corps de notre application. C'est ici que l'on a appelé les différentes fonctions de `Dash_function` pour afficher les logos mais aussi les graphiques et tableaux. De plus, c'est dans ce fichier que l'on a créé les deux pages de notre application (grâce à `dcc.Tabs`). Nous avons aussi créé le filtre à appliquer sur nos visuels.