Documentation technique

Nettoyage des données pour l'affichage

Nous avons dans un premier temps procéder au nettoyage et recodage pour certaines variables de nos fichiers « train.csv » et « submissions.csv ». Nous avons dans un premier temps procéder aux différents traitements sur les valeurs manquantes en utilisant la technique des KNN. Nous avons également réalisé un recodage sur l'ensemble de nos variables pour qu'elles soient de type quantitatif pour l'homogénéité et la simplification de nos traitements. Certaines variables plus spécifiques comme l'âge nous ont demandé un recodage en différentes classes pour définir les différentes tranches d'âges de nos individus.

Modèle prédictif

Une fois nos fichiers « train » et « submissions » nettoyés, nous avons élaboré un modèle prédictif directement accessible via le fichier « Machinelearning.py ». Afin de percevoir le modèle le plus performant pour nos données, nous avons dans un premier temps travaillé sur le jeu de donnée train qui a été nettoyé comme évoqué précédemment. Pour ce faire :

- Procéder au découpage du Dataset en deux parties distinctes avec 50% des données pour notre apprentissage et les 50% restants pour la partie test.
- On réalise l'instanciation de Smote pour l'ensemble de nos données apprentissage (Smote est une technique de suréchantillonnage des observations minoritaires).
- Mise en place du modèle prédictif sur nos données apprentissage par la méthode de AdaBoostClassifier avec l'utilisation de la fonction fit.
- Elaboration de notre prédiction des matchs sur nos données tests grâce à la fonction predict.
- On procède à la création d'un DataFrame pour reconstituer la bonne structure avec d'une part la colonne « iid_pid » ainsi que notre colonne prédiction.
- Analyse des performances du modèle grâce aux différents indicateurs dont le f1_score.

On obtient en sortie notre fichier « prediction.csv »

Application Dash

Après obtention de notre modèle de prédiction, nous avons réalisé une application Dash sous Visual Studio Code directement accessible via le fichier python « app.py ».

Afin de rationaliser et de structurer notre code, nous avons utilisé la fonction callback pour créer les interactions interface utilisateurs. Nous l'avons ainsi intégré dans nos différents onglets de notre application pour faire le passage entre l'accueil du site et les onglets Dashboard et prédiction.

Pour notre page d'accueil du site, nous avons intégré les différents logos : « Easy Date » et « AI Match » grâce à la librairie plotly en utilisant la méthode add_layout_image.

Notre deuxième onglet intitulé Dashboard est constitué d'une liste déroulante permettant la sélection entre le fichier train d'une part et le fichier submissions de l'autre. Pour ce faire, nous avons utilisé la méthode ddc.dropdown de Plotly. La visualisation de nos graphiques en diagramme circulaire sont réalisées à l'aide de la méthode ddc.RadioItems qui permet grâce à nos callbacks de venir récupérer nos données correspondantes en sélectionnant sur nos variables descriptives tel que le genre, l'âge ou encore le domaine d'étude et l'origine des individus.

Enfin, sur notre même onglet Dashboard, nous avons ajouté un nuage des activités les plus populaires auprès des individus dans un autre fichier python intitulé «Activite.py». Pour créer les graphiques souhaités, nous avons utilisé la méthode wordcloud à la fois pour obtenir un nuage des mots sur nos données train et submissions. Nous avons par la suite exporter nos graphique sous format image «png» pour les intégrer par la suite dans notre fichier de base « app.py » pour les faire afficher sur l'application.

Dans le dernier onglet de notre application Speed Dating on retrouve notre prédiction réalisée par notre modèle sur le jeu de donnée. Nous avons ainsi décidé de représenter cette prédiction de matchs en élaborant un graphique qui utilise le module Pyplot de la librairie Matplotlib. Nous avons ainsi confronté notre prédiction avec la fréquence de sorties et de rendez-vous des individus. Nos graphiques ont été exporter au format image «png» comme vu précédemment pour les intégrer dans notre module initial pour l'affichage.