Documentation technique

AI match

Informations

Nom du projet	AI match
Type de document	Documentation technique
Date	17/10/2022
Version	1.8
Mots-clés	Modèle Modélisation Prédiction Evaluation Dash
Auteurs	Christelle KIEMDE Fatimetou HAIDARA Pierre DUBRULLE

Tables des matières

1. Résumé du document

Ce document est la documentation technique officielle du projet AI match.

C'est un projet qui consiste à réfléchir à un modèle permettant de prédire si deux personnes vont matcher selon le formulaire complété préalablement de la rencontre et à produire un tableau de bord interactif.

Dans ce document nous expliquons les différentes démarches pour y arriver.

Le document comprend la partie nettoyage des données, les modèles appliqués et la partie visualisation de l'application.

2. Librairies

Dans cette partie, nous expliquons les librairies principales que nous avons utilisé pour la prédiction et le tableau de bord.

Le tableau suivant nous donne le récapitulatif des librairies utilisées :

Librairies	**Rôles**
Pandas	Lecture et traitement des données
sklearn	-Découpage des données en train/test - l'arbre de classification -f1_score -sélection des variables -Traitement des données manquantes par les k plus proches voisins -K plus proches voisins -Random Forest
mode	Traitement des données manquantes par le mode
numpy	Pour les calculs
imblearn	Traitement des données déséquilibrées
pickle	Sauvegarder le modèle
Dash	Pour l'application

3. Nettoyage des données

Après l'importation des deux bases de données, nous avons d'abord testé si les deux bases ont les mêmes variables puis nous avons supprimer les variables différentes.

Le nettoyage consiste d'abord à changer les virgules par des points et à changer le type de certaines variables en float puis en int.

Ensuite, le nettoyage a consisté à traiter les données manquantes. Pour le faire, nous avons scinder la base de données **train** en deux : les qualitatives à part et les quantitatives d'autre part. Pour le traitement des données manquantes dans la base quantitative, nous avons utilisé les K plus proches voisins et pour le traitement qualitatif par le mode.

La dernière étape du nettoyage de données était la partie labélisation. Il s'agissait de recoder les modalités.

4. Modèle

Avant de commencer nous avons scinder la matrice des explications en listant les variables à mettre dans le modèle, puis un codage disjonctif complète a été faite sur les variables qualitatives. Ensuite nous avons scinder les données en échantillons d'apprentissage et test.

Nous avons d'abord fait un arbre de décision simple qui nous a permis de sélectionner les variables les plus importantes à mettre dans le modèle.

On reprend le modèle arbre de décision avec les variables les plus importantes sur les données d'apprentissage.

Ensuite, nous prédisons sur les données de test, puis nous évaluons le modèle avec le F1_score.

Pour chaque modèle, nous l'appliquons sur les données d'apprentissage, nous prédisons sur les données de test, puis nous évaluons le modèle.

Nous avons appliqué par la suite l'arbre de décision avec la méthode **GridSearchCV.**

La modélisation 2, nous avons utilisé d'abord **SMOTE** qui est une technique très utile pour rééquilibrer les données numériques en entrée d'un modèle de Machine Learning.

Ensuite, nous avons appliquons l'arbre de décision avec GridSearchCV sur les données équilibrées.

Nous avons également utilisé **SBordeline SMOTE** qui est une variante de l'algorithme SMOTE original et nous avons fait tourner le modèle.

La modélisation 3 a consisté à utiliser le random forest sur les données équilibrées avec SMOTE.

Enfin nous avons utilisé le K plus proches voisins. Nous avons cherché la valeur K pour laquelle nous avons une meilleure précision, puis nous avons prédit et évalué le modèle.

Le meilleur modèle est l'arbre de décision avec application de SMOTE.

Ce modèle a été sauvegardé en format pickle pour une utilisation ultérieure.

Nous avons fait ressortir l'importance des variables sur le meilleur modèle.

Il sera utilisé également pour la compétition Kaggle.

5. Compétition Kaggle

Nous avons traité les données de la base submission en corrigeant les données manquantes.

Ensuite nous avons utilisé le modèle retenu pour prédire sur les données submission.

Nous avons enregistré la prédiction sous format CSV pour le soumettre à la compétition kaggle.

6. Application Dash

AI MATCH est un tableau de bord pour pouvoir jouer avec les données et des modèles à travers un tableaux de bord interactif.

Construit avec :

Dash - Serveur principal et composants interactifs

Plotly Python - Utilisé pour créer les tracés interactifs

Dash DAQ - composants techniques stylisés

Exigences :

Nous avons créé un environnement virtuel distinct exécutant Python 3 pour cette application et installer toutes les dépendances requises.

Comment utiliser cette application

Exécutez cette application localement en :

Python index.py

Ouvrez http://127.0.0.1:8050/ dans votre navigateur, vous verrez un tableau de bord de mise à jour en direct.

Qu’est-ce que cette application montre

Nous avons 8 fichiers sont nécessaires pour le fonctionnement de l’application.

`app.py` : L'application dash est déployé.
index.py : Qui contient le Sidebar, qui permet de naviguer entre nos différentes pages, et tous les graphiques plotly .
`data.py` : contient des fonctions de préparation du Dataframe pour les graphiques
`train` : Le fichiers CSV pour les graphes et les prédictions.
`/documentation` : Les fichiers de documentations technique et fonctionnelle de l'application.
`requirements.txt` : contenant les modules python qui seront installés lors de la construction de l'application.
`runtime.txt` Nécessaire pour Heroku car il indique (le serveur HTTP Gunicorn) la version de Python utiliser.
`Procfile` : Nécessaire pour Heroku , donne comme information le type de processus que va exécuter (processus Web Gunicorn) et le point d'entrée de l'application Python (index.py).

Les fonctions

Cette fonction callback permet de créer une page Accueil et nos différentes pages

```
@app.callback(Output("page-content", "children"), [Input("url", "pathname")])

def render_page_content(pathname):

    if pathname == '/':

        return [html.Div([dcc.Markdown('''

            ###  TROUVER UNE PERSONNE PARFAITEMENT COMPATIBLE

            Cette application est un projet de Easy Date réalisé par [Haidara Fatimetou, KIEMDE Christelle, DUBRULLE Pierre]

            à l'aide de Dash de Plotly,



            C’est la raison pour laquelle le projet AI match a vu le jour !

        ''')],className='home',style={'text-align':'center'}),
```

html.Div(html.Img(src=logo), style={'text-align':'center'}),]

elif pathname == '/Analysis':

Nous avons une Sidebar, qui nous permet de naviguer entre nos différentes pages(Home ,Analysis et Le profiles.

```
sidebar = html.Div(
                                                    [
                                                        #html.H2("AI MATCH",
                                                    className="display-5"),
                                                        html.Img(src=logo1,
                                                    height="200",width="230"),
                                                        html.H2("Data
                                                    Explorer", className="display-
                                                    5"),
                                                        html.Hr(),
                                                        dbc.Nav(

                                                    [dbc.NavLink("Home", href="/",
                                                    active="exact")],
                                                            vertical=True,
                                                            pills=True,
                                                        ),
                                                        html.Hr(),
                                                        html.H2("Historical
                                                    Analysis", className="lead"),
                                                        dbc.Nav(
                                                            [

                                                    dbc.NavLink("Analysis",
                                                    href="/Analysis",
                                                    active="exact"),

                                                    dbc.NavLink("Profiles of
                                                    participants",
                                                    href="/Profiles",
                                                    active="exact")
                                                            ],
                                                            vertical=True,
                                                            pills=True,
                                                        ),
                                                        html.Hr(),
                                                        html.H2("Machine
                                                    Learning Analysis",
                                                    className="lead"),
                                                        dbc.Nav(
```

```
                [

dbc.NavLink("Projections and
Regression",
href="/projection",
active="exact"),
                #
dbc.NavLink("Batting
Analysis", href="/player",
active="exact"),
                #
dbc.NavLink("Pitching/Feilding
Analysis", href="/field",
active="exact"),
            ],
            vertical=True,
            pills=True,
        ),
    ],
    style=SIDEBAR_STYLE,
)
```

Création de l'application Dash et définir le serveur d'applications comme variable pour le déploiement.

```
app = dash.Dash(__name__,
external_stylesheets=[dbc.themes.SANDSTONE],
        meta_tags=[{'name': 'viewport',
'content': 'width=device-width, initial-
scale=1'},],)
srv = app.server

# set app callback exceptions to true
app.config.suppress_callback_exceptions =
True

# set applicaiton title
app.title = 'AI MATCH Data Explorer'
```