CHATENET Nathan
FAILLAFAUX Thomas

4201C - DATA ENGINEERING DEVELOPPER GUIDE

Table des matières :

1. Installation des packages nécessaires

- Installer « Sélénium » pour le scrapping

Tout le code de ce projet a été développé en local, et non via « Docker », pour la raison que nous avons utilisé Sélénium et un driver pour le scrapping de nos données. Il faudra ainsi s'assurer d'avoir installé :
« Google Chrome » sur l'ordinateur.
« WebDriver » : qui s'installe normalement dans notre code en local, ou utilisé un chromedriver associé.
« Sélénium » : via la commande « !pip install Selenium », qui se situe normalement dans les notebooks associés.

- Installer PyMongo pour les bases de données

La partie insertion dans une base de Données nécessite PyMongo et une liaison de Mongo avec Python. Toutes les informations utiles pour télécharger PyMongo et la mise en relation avec Python se situe dans le lien ci-dessous.

https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/

- Installer les autres package nécessaires

D'autres packages sont importés et utilisés, notamment :
« Flask » via la commande « !pip install Flask ».
« Plotly » via la commande « !pip install plotly », et pourra nécessiter l'option plotly-orca
« Pandas » via la commande « !pip install pandas ».
« Numpy » via la commande « !pip install numpy ».

2. Le Scraping

A. Emplacement des fichiers

La partie Scraping se situe dans le dossier « Data_Traitement ». Cette partie est divisée en deux, car nous avons récupérer des données sur deux sites différents. Les deux fichiers notebook particulier portent l'appellation « <nom_du_site>_scraping ».

B. Description de la contenance des fichiers

La manière de procédé est la suivante, on définit d'abord un web Driver sur Chrome qui nous permettra de prendre les données. On récupère les éléments sous l'ID qui nous intéresse.
Afin de récupérer les données qui nous intéresse, on récupère le texte associé aux éléments pris, qui sera ainsi sous la forme d'une grande liste. Ces données sont ainsi nettoyées en fonction de nos besoins futures. Le nettoyage est exclusivement destiné à cette récupération de données et il est ainsi non modifiable afin de ne pas entraîner d'erreurs pour la suite du projet. Après le projet, on met les données sous forme de dataframe, que l'on exportera en csv afin d'effectuer la mise sous Mongo.

3. Vers Mongo

A. Emplacement des fichiers

La mise sous Mongo s'effectue dans le même dossier que le scraping, dans le dossier « Data_Traitement ». Les deux fichier notebook portent l'appellation « <nom_du_site>_vers_Mongo ».

B. Description de la contenance des fichiers

Ayant auparavant déjà nettoyé nos données et mis tout correctement dans un dataframe propre, on définit une nouvelle collection, puis on définit nos données que l'on veut mettre dans la base en important juste le dataframe. Ce dataframe est ensuite mis sous forme de dictionnaire. On veillera d'avoir correctement nettoyé la base de données avant d'importer les nôtres.

4. Fichier Python pour l'application Flask

A. Description et emplacement des fichiers

Ce fichier sert principalement de liens entre l'application Flask et le code HTML derrière le site internet. Le fichier est dénommé « Application.ipynb » et crée un fichier Python, nommé « contenu_app.py » et permet ensuite son exécution via la commande « ! python contenu_app.py ». L'application sera mise en place à l'exécution de cette commande.

B. Importation des modules

On importe d'abord tous les modules dont on a besoin afin d'effectuer l'application. On a ensuite l'importation des différentes bases de données et documents « csv » crée auparavant.

C. Graphiques

La partie suivante contient la création des graphiques. Ceux-ci sont des bar graphes particulier montrant la réussite des équipes sous différents aspects. Ils sont mis à jour à chaque fois que le scraping est effectué étant donné que les graphiques sont effectués à partir des données récupéré par celui-ci. Chacune des fonctions définit un graphique. On a d'abord les différents tracés, un ou deux selon les représentations, puis les à côté comme le titre des graphiques, ou des axes. Puis on effectue le tracé avec la fonction « figure » et on l'importera sous forme d'images dans le dossier « static », par l'appellation des fonctions suivants leur définition.

D. Les « App Route »

Les « App Route » décrivent l'accès au lien de l'appli par rapport aux templates html décrit plus bas. Chaque route appelle, lorsque c'est nécessaire, sa base de données. On lui définira un nom qui servira de référence lors de l'appel de la page. Une modification des noms au niveau des définitions doit aussi entraîner une modification des liens « href » définit sur chaque document html correspondant.

5. Fichier Static pour l'application Flask

A. « Static Description »

Les deux images « description1 » et « description2 » sont des images statiques qui ne se modifient pas et sont présentes pour apporter une explication supplémentaire au rôle du projet. Elles sont directement liées au fichier « template_description.html », leur modification ou leur enlèvement doit donc entraîner aussi une modification de ce dernier fichier.

B. « Static Plot »

Les images « Plot » dans le dossier « Static » sont définis à partir des graphiques précédemment crées lors de la création de l'application via le fichier « Python ». Ces photos changent régulièrement en fonction des matchs effectués, puisqu'elles s'inspirent des statistiques récupérées. Elles sont référencées dans le fichier « graph.html », une modification de leur nom lors de leur importation doit donc entraîner une modification de leur nom associé dans la source du fichier « graph.html ».

6. Fichiers template.html pour l'application Flask

Le dossier « template » contient quatre fichiers html permettant la création de la forme de l'application. Ces quatre fichiers sont liés et sont appelés lors d'un changement de route correspondant par la forme à un clique sur le lien.

A. « template_description.html »

Cette page est principalement composée de texte, encadré par les balises <p>. Ce texte est justifié et encadré par une bordure. Les titres et sous-titres sont eux tous centrés. Ensuite viennent deux images définies par une classe dans le « head » permettant de les encadrés et de les centrés. Pour modifier la taille des images, on modifiera le paramètre « height » pour la hauteur et « width » pour la largeur si cela est nécessaire. Enfin, la dernière partie de cette page se compose, comme pour toutes les autres, d'un menu d'accès aux autres pages. On veillera ainsi à éviter de ne pas modifier le texte derrière « href », représentant l'accès à la route particulière et donc à une nouvelle page.

B. « template_graph.html »

Cette page représente la page de graphique. Comme pour la page précédente, les graphiques étant représenté par des photos, on veillera à ne pas modifier la source. Comme pour la page précédente, on à la même classe afin de centrer ces images, et celle-ci sont ici encadrés par la caractéristique « border ». De même que précédemment, le texte est encadré par différentes couleurs et est justifié. On retrouvera sur la fin un menu de sélection d'autres pages.

C. « template_classement.html »

Cette page se présente différemment des deux précédentes car elle contient une table permettant d'examiner les données. C'est là qu'intervient notre base de données crée précédemment. On définit comme précédemment une classe afin de centrer le tableau. L'argument « colgroup » qui suit nous permet ici d'attribuer à chaque cases la même largeur. Ensuite, entre les balises <tr> puis <th>, on définit un titre pour chaque colonne. Enfin, à l'aide d'une boucle et de notre base de données, on remplit notre table.
De plus, de même que précédemment, on définira du texte encadré expliquant les graphiques. On aura de même un menu d'accès aux autres liens de l'application web.

D. « template_calendrier.html »

Cette dernière page se structure de la même façon que la page précédente, utilisant une base de données et une boucle afin de remplir l'autre table des meilleures côtes récupérés. De même, un menu avec les liens pour passer de pages en page sera affiché en bas de cette page-ci.