



Projet Stimul x CentraleSupélec

Notice d'utilisation



https://gitlab.com/latitudes-exploring-tech-for-good/stimul/1819_stimul/

README

Name	Last commit	Last update
notebooks	commit notebooks by gao	1 week ago
rapport	commit rapport by gao	1 week ago
README.md	Update README.md	1 week ago

Les Notebooks :

Name	Last commit	Last update
..		
Notebook_KPI_com.ipynb	commit notebooks by gao	1 week ago
Traitement des fichiers Json.ipynb	commit notebooks by gao	1 week ago
Vrai_Notebook_Temporalité.ipynb		

Ils nécessitent des données en .csv séparés par des « ; » ou les fichiers .json extraits de GoogleAnalytics, placés dans le même dossier que le fichier Jupyter Notebook.

Fichiers nécessaires avec le format utilisé :

Notebook_KPI_com.ipynb :

1 Activity.csv

1_Activity.csv
1 id;category_id;customer_id;duration;date;state;deletedAt;createdAt;updatedAt;createdBy_id;updatedBy_id;deletedBy_id;feedback_rate;feedback_text;;
2 1;1;5;09/05/2016 15:40;failed;NULL;09/05/2016 15:24;13/05/2016 09:32;6;4;NULL;NULL;NULL;;
3 2;1;1;12/05/2016 14:12;done;NULL;12/05/2016 13:59;13/05/2016 09:32;4;4;NULL;NULL;NULL;;
4 3;1;1;13/05/2016 09:50;new;13/05/2016 16:48;13/05/2016 09:33;13/05/2016 09:33;4;4;4;NULL;NULL;;

2 Activity Category.csv

1_Activity.csv	2_Activity_Category.csv
1 id;image_id;name;deletedAt;createdAt;updatedAt;createdBy_id;updatedBy_id;deletedBy_id;MET;activity_category_type_id	
2 1;8;Autre;NULL;09/05/2016 15:22;14/12/2017 15:17;3;1;NULL;0;1	
3 2;9;Aerobique;NULL;09/05/2016 16:43;14/12/2017 15:05;3;3;NULL;7.3;1	

4 Customer_Lesson.csv

```
4_Customer_Lesson.csv
1 id;lesson_id;customer_id;state;date
2 1;1;1;done;08/05/2016 10:00
3 2;2;1;done;15/05/2016 10:00
4 3;3;1;done;22/05/2016 10:00
```

6_Metric.csv

```
6_Metric.csv
1 id;user_id;category_id;value;date;deletedAt;createdAt;updatedAt
2 1;6;5;5237;17/05/2016;NULL;28/11/2017 11:51;28/11/2017 11:51
3 2;6;5;4512;16/05/2016;NULL;28/11/2017 11:51;28/11/2017 11:51
4 3;6;5;3589;15/05/2016;NULL;28/11/2017 11:51;28/11/2017 11:51
5 4;6;5;6396;14/05/2016;NULL;28/11/2017 11:51;28/11/2017 11:51
```

7_Metric_Category.csv

```
7_Metric_Category.csv
1 id;name
2 1;sleep
3 2;wakeup_sleep
4 3;light_sleep
```

11_Customer.csv

```
11_Customer.csv
1 id;user_id;gender;height;weight;deletedAt;createdAt;updatedAt
2 1;6;2;180;71.0;NULL;09/05/2016 14:47;04/02/2019 15:21
3 4;9;2;186;82.0;NULL;10/05/2016 18:01;26/04/2018 10:10
4 5;12;1;165;62.0;NULL;20/05/2016 20:16;26/06/2018 09:26
```

11_User.csv

```
11_User.csv
1 id;birthday;deletedAt;createdAt;updatedAt
2 6;09/02/2018;NULL;09/05/2016 14:47;13/02/2019 15:00
3 9;05/03/1992;NULL;10/05/2016 18:01;13/02/2019 15:23
4 12;15/08/1979;NULL;20/05/2016 20:16;25/01/2019 11:26
```

cohortes.csv

```
cohortes.csv
1 id;user_id;gender;height;weight;deletedAt;createdAt;updatedAt;disease_id;
2 1;6;2;180;71.0;NULL;09/05/2016 14:47;04/02/2019 15:21;1;
3 4;9;2;186;82.0;NULL;10/05/2016 18:01;26/04/2018 10:10;1;
4 5;12;1;165;62.0;NULL;20/05/2016 20:16;26/06/2018 09:26;1;
5 6;13;2;175;29.6;NULL;26/05/2016 10:21;03/04/2018 14:35;1;
```

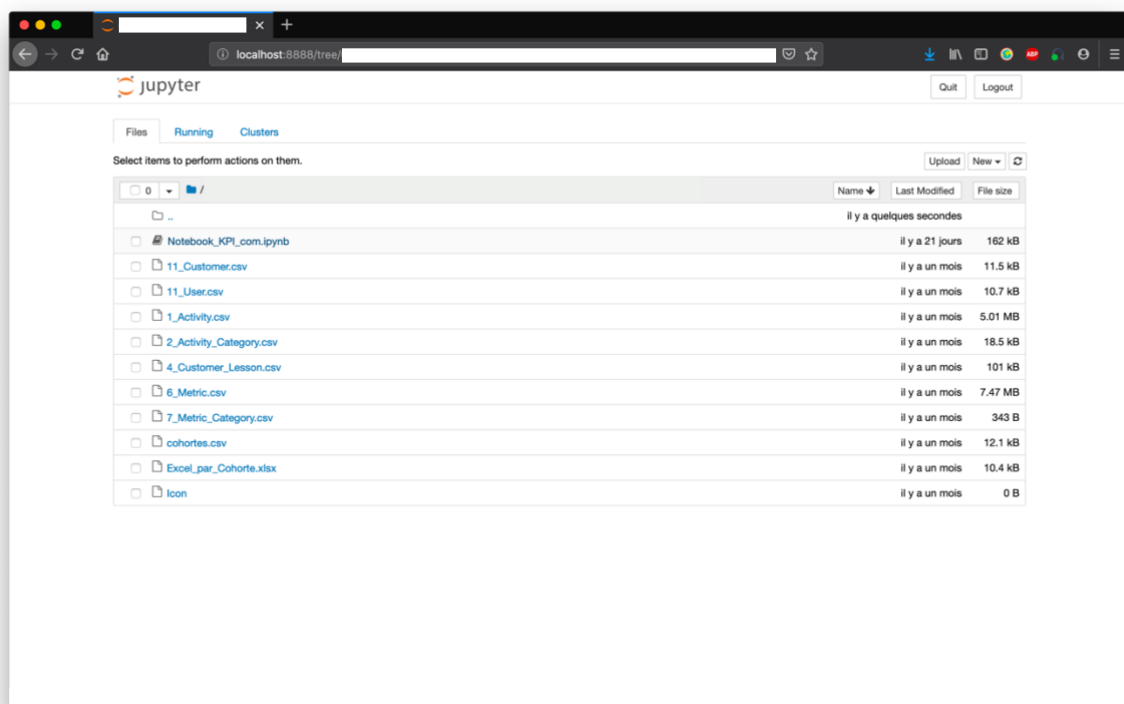
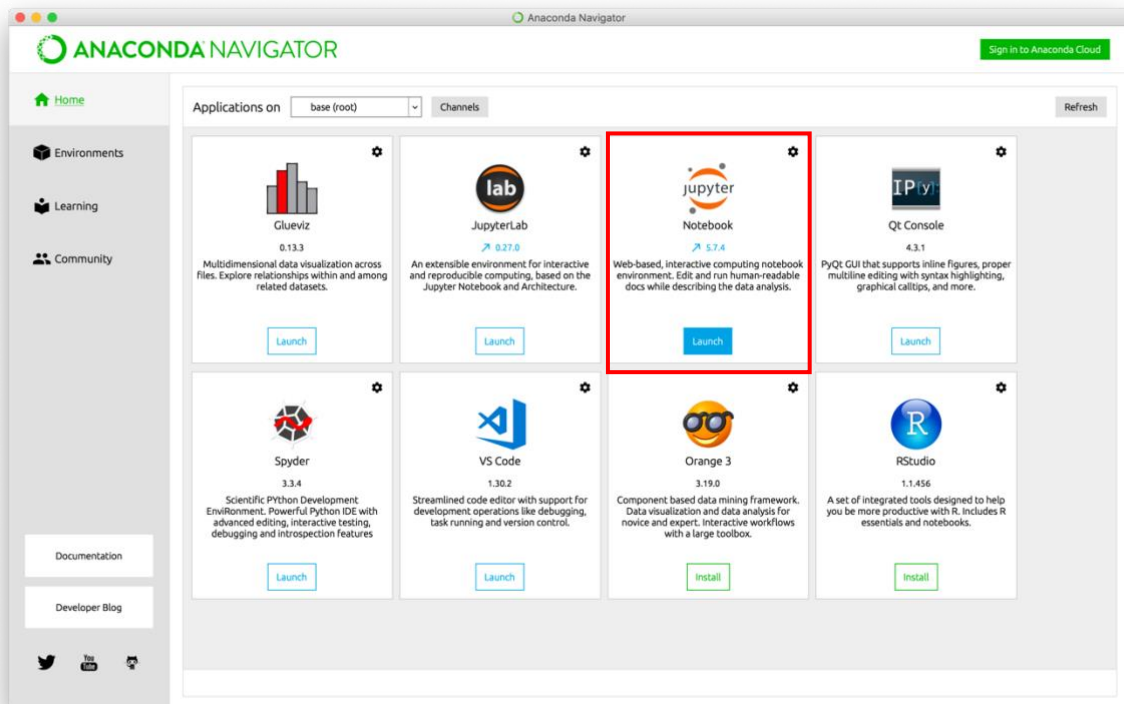
Traitement des fichiers Json.ipynb :

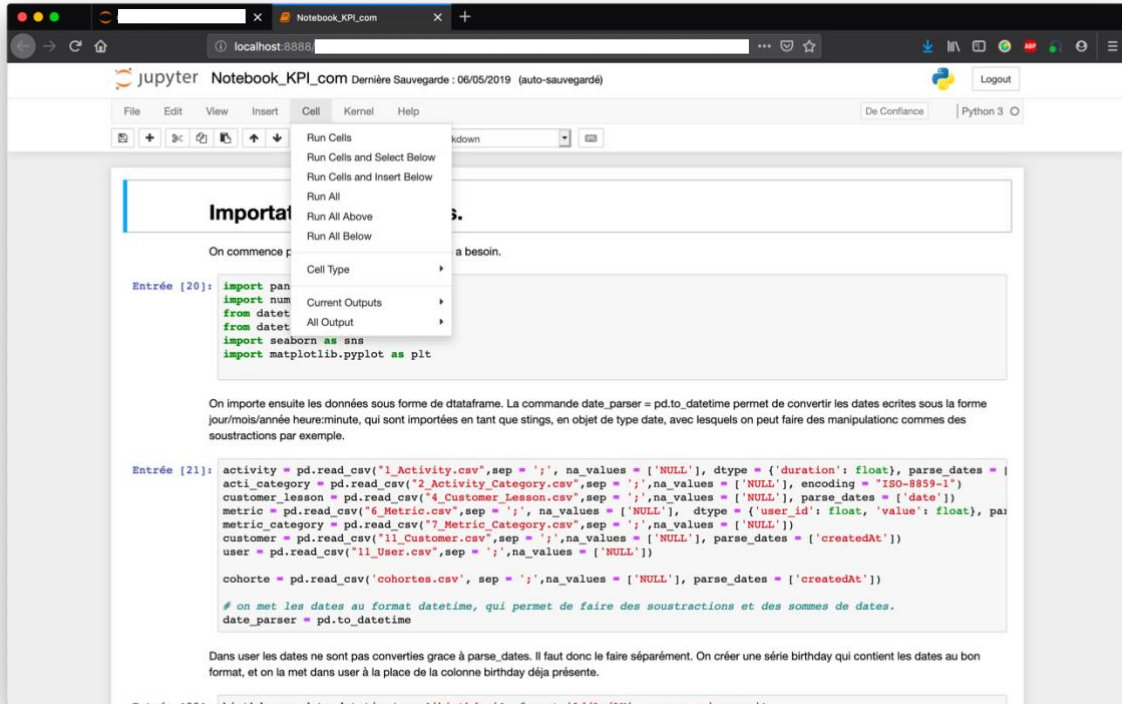
```
578_1_user-report-export.json
578_2_user-report-export.json
579_1_user-report-export.json
....
```

Vrai_Notebook_Temporalité.ipynb :

```
1_Activity.csv
2_Activity_Category.csv
6_Metric.csv
cohortes.csv
```

Les Notebooks s'ouvrent avec Jupyter (compris dans le navigateur Anaconda) :





The screenshot shows a Jupyter Notebook titled "Notebook_KPI.com" with a menu open over the "Cell" tab. The code in the notebook includes:

```

Entrée [20]: import pandas as pd
import numpy as np
from datetime import datetime
import seaborn as sns
import matplotlib.pyplot as plt

On importe ensuite les données sous forme de dataframe. La commande date_parser = pd.to_datetime permet de convertir les dates écrites sous la forme
jour/mois/année heure:minute, qui sont importées en tant que strings, en objet de type date, avec lesquels on peut faire des manipulations comme des
soustractions par exemple.

Entrée [21]: activity = pd.read_csv("1_Activity.csv", sep = ';', na_values = ['NULL'], dtype = {'duration': float, parse_dates = ['date']})
acti_category = pd.read_csv("2_Activity_Category.csv", sep = ';', na_values = ['NULL'], encoding = "ISO-8859-1")
customer_lesson = pd.read_csv("4_Customer_Lesson.csv", sep = ';', na_values = ['NULL'], parse_dates = ['date'])
metric = pd.read_csv("6_Metric.csv", sep = ';', na_values = ['NULL'], dtype = {'user_id': float, 'value': float, parse_dates = ['date']})
metric_category = pd.read_csv("7_Metric_Category.csv", sep = ';', na_values = ['NULL'])
customer = pd.read_csv("11_Customer.csv", sep = ';', na_values = ['NULL'], parse_dates = ['createdAt'])
user = pd.read_csv("11_User.csv", sep = ';', na_values = ['NULL'])

cohort = pd.read_csv("cohortes.csv", sep = ';', na_values = ['NULL'], parse_dates = ['createdAt'])

# on met les dates au format datetime, qui permet de faire des soustractions et des sommes de dates.
date_parser = pd.to_datetime

```

Below the code, there is a paragraph explaining the use of `date_parser = pd.to_datetime` to convert date strings into datetime objects for easier manipulation.

Un rapport écrit qui détaille nos résultats et les analyses :

