

TEXT MINING PROJECT

En commande line

- Pour lancer le projet il faut cloner le projet

```
https://github.com/Aida73/text-mining.git
```

- Activer l'environnement virtuel:
 - Ouvrir le projet cloné dans un terminal et se déplacer dans le dossier **methode_2**
 - Tapez **source env/bin/activate**
- Installer les packages nécessaires
 - Verifier si vous avez un python 3.6+
 - Verifier si java 8+ est installée
 - **pip install -r requirements.txt**
- Corriger un dataset
 - Tapez dans le terminal ***python utils.py correct_target /path/of/csv/dataset target***
 - exemple : ***python utils.py correct_target /Users/user/Desktop/text-mining/VariableCibles.csv profession***

```

user@MacBook-Pro-de-User projet_stage_text_mining % cd methode_2
user@MacBook-Pro-de-User methode_2 % source env/bin/activate
(env) user@MacBook-Pro-de-User methode_2 % python utils.py correct_target /Users/user/Desktop/text-mining/VariableCibles.csv profession
-----the correction starts-----
Pandas Apply: 100%| 2000/2000 [00:00<00:00, 154117.36it/s]
Pandas Apply: 100%| 2000/2000 [00:09<00:00, 208.04it/s]
      profession      Corrected
1 responsable societe seunte globe Responsable société s'ente globe
2      menagere      ménagère
3      menagere      ménagère
4 directrice commerciale Directrice commerciale
5      eleve      élevé
Tapez 1 pour enregistrer la base et 0 pour quitter: 1
Donner le nom du dataset : corrected
Enregistrer !
(env) user@MacBook-Pro-de-User methode_2 %

```

- On peut des fois rencontrer quelques corrections mal effectuées. Cependant on pourra toujours les corriger.

corrected.csv			Ouvrir
	profession	Corrected	
	responsable societe seunte globe	Responsable société s'ente globe	
	menagere	ménagère	
	menagere	ménagère	
	directrice commerciale	Directrice commerciale	
	eleve	élevé	
	eleve	élevé	
	chauffeur	chauffeur	
	ouvrier agricole	Ouvrier agricole	
0	agent du cadastre impots et domaine	Agent du cadastre impôts et domaine	
1	autres	autres	
2	pilote engen	Pilote engen	
3	commerçant	commerçant	

- Leur correction a été prise en compte. Pour ce faire:
 - Tapez sur le terminal ***python utils.py correct_outliers /path/to/corrected/dataset target***
 - Exemple: ***python utils.py correct_outliers /Users/user/Downloads/corrected.csv Corrected***

```
(env) user@MacBook-Pro-de-User methode_2 % python utils.py correct_outliers /Users/user/Downloads/corrected.csv Corrected
Entrer l'élément à corriger: élevé
Élément de remplacement: élève
Donner le nom du dataset pour l'enregistrer: corrected_2
Enregistrer !
(env) user@MacBook-Pro-de-User methode_2 %
```

corrected_2.csv				Ouvrir
	Unnamed: 0	profession	Corrected	
0	1	responsable societe seunte globe	Responsable société s'ente globe	
1	2	menagere	ménagère	
2	3	menagere	ménagère	
3	4	directrice commerciale	Directrice commerciale	
4	5	eleve	élève	
5	6	eleve	élève	
6	7	chauffeur	chauffeur	
7	9	ouvrier agricole	Ouvrier agricole	
8	10	agent du cadastre impots et domaine	Agent du cadastre impôts et domaine	
9	11	autres	autres	
10	12	pilote engin	Pilote engin	
11	14	commercant	commerçant	
12	15	commercante femme au foyer	commerçante femme au foyer	
13	16	etudiant droits des affaires	étudiant droits des affaires	

- Trouver les categories: on va essayer de catégoriser la colonne corrigé. Ceci, en utilisant les catégories données par l'utilisateur. Il pourra donner autant de catégories qu'il voudra. Pour ce faire:
 - Tapez **`python utils.py find_categories /path/of/corrected/dataset target`**
 - Exemple : **`python utils.py find_categories /Users/user/Downloads/corrected_2.csv profession`**

```

(env) user@MacBook-Pro-de-User methode_2 % python utils.py find_categories /Users/user/Downloads/corrected_2.csv profession
Tapeze 1 ajouter une categorie: 1
categorie : ménagère
['ménagère']
Tapeze 1 ajouter une autre catégorie categorie ou 0 pour la classification: 1
categorie : enseignement
['ménagère', 'enseignement']
Tapeze 1 ajouter une autre catégorie categorie ou 0 pour la classification: médical
Vous ne pouvez choisir qu'entre 0 et 1 !
Tapeze 1 ajouter une autre catégorie categorie ou 0 pour la classification: 1
categorie : transport
['ménagère', 'enseignement', 'transport']
Tapeze 1 ajouter une autre catégorie categorie ou 0 pour la classification: 1
categorie : administration
['ménagère', 'enseignement', 'transport', 'administration']
Tapeze 1 ajouter une autre catégorie categorie ou 0 pour la classification: 0
['ménagère', 'enseignement', 'transport', 'administration']

```

```

-----finding categories-----
Pandas Apply: 100%| 2000/2000 [00:00<00:00, 10619.96it/s]
Pandas Apply: 100%| 2000/2000 [00:00<00:00, 74802.78it/s]
Pandas Apply: 100%| 2000/2000 [00:00<00:00, 7507.40it/s]
Pandas Apply: 100%| 2000/2000 [00:00<00:00, 24723.42it/s]
Pandas Apply: 100%| 2000/2000 [00:00<00:00, 10071.20it/s]
Pandas Apply: 100%| 2000/2000 [00:00<00:00, 21324.66it/s]
Pandas Apply: 100%| 2000/2000 [00:00<00:00, 7035.96it/s]
Pandas Apply: 100%| 2000/2000 [00:00<00:00, 24168.81it/s]
Pandas Apply: 100%| 2000/2000 [00:00<00:00, 580727.45it/s]
Tapeze 1 pour enregistrer la base et 0 pour quitter: 1
Donner le nom du dataset : categorized
Enregistrer !
None                1099
enseignement         372
administration       246
ménagère            189
transport            61
transport,administration  13
enseignement,administration  12
transport,enseignement   8
Name: Categorie, dtype: int64
(env) user@MacBook-Pro-de-User methode_2 %

```

profession	Corrected	Categorie
responsable societe seunte globe	Responsable société s'ente globe	administration
menagere	ménagère	ménagère
menagere	ménagère	ménagère
directrice commerciale	Directrice commerciale	None
eleve	élève	enseignement
eleve	élève	enseignement
chauffeur	chauffeur	transport
ouvrier agricole	Ouvrier agricole	enseignement
agent du cadastre impots et domaine	Agent du cadastre impôts et domaine	enseignement,administration
autres	autres	None
pilote engin	Pilote engin	None
commerçant	commerçant	None
commercante femme au foyer	commerçante femme au foyer	ménagère
etudiant droits des affaires	étudiant droits des affaires	enseignement
auxilliaire de vie	auxiliaire de vie	None
commercante	commerçante	None
aide soignante	aide-soignante	None

rhinorrhée	rhinorrhV@e	rhinorrhV@e
rhinorrhée	rhinorrhV@e	rhinorrhV@e
polyuries essoufflement	Polyuries essoufflement	None
dysneés	dyspnV@es	None
rhinorrhée	rhinorrhV@e	rhinorrhV@e
rhinorrhée	rhinorrhV@e	rhinorrhV@e
contact a risque	Contact V† risque	None
rhinorrhée	rhinorrhV@e	rhinorrhV@e
anosmie	anosmie	anosmie
rhinorrhée	rhinorrhV@e	rhinorrhV@e
douleur poitrine essoufflement manque d air	Douleur poitrine essoufflement manque d'air	None
fatigue	fatigue	Douleurs musculaires
rhinorrhée difficulté respiratoire	rhinorrhV@e difficultV@ respiratoire	Douleurs musculaires,rhinorrhV@e
rhinorrhée	rhinorrhV@e	rhinorrhV@e
rhinorrhée courbatures dyspnee	rhinorrhV@e courbatures dyspnV@e	rhinorrhV@e
rhinorrhée	rhinorrhV@e	rhinorrhV@e
rhinorrhée	rhinorrhV@e	rhinorrhV@e
dyspnee	dyspnV@e	None
difficultés a respirer douleurs thoraciques	difficultV@s v† respirer douleurs thoraciques	None
rhume perte odorat	Rhume perte odorat	anosmie
douleurs thoraciques	Douleurs thoraciques	None

Sur RStudio

On a mise en place une API avec Django REST framework pour pouvoir utiliser les méthodes développées pour la correction et la catégorisation.

- Comme pour la première approche, il faut cloner le projet :

```
https://github.com/Aida73/text-mining.git
```

- Cette fois-ci, on va s'intéresser au dossier **data_quality_project**
 - `cd path/to/the/project/data_quality_project`
- Activation de l'environnement virtuel: ***source env/bin/activate***
- Si l'environnement virtuel n'existe pas, on peut le créer et on l'active: ***python3 -m createvirtualenv nom_de_l'environnement***
- Installation de packages nécessaires:
 - Sur le même répertoire où se trouve le fichier **requirements.txt**, on exécute la commande suivante: ***pip install -r requirements.txt***
- Lancement du serveur: on se déplace dans le dossier **dq_project** (*qui est le projet en tant que telle*) et on exécute: ***python manage.py runserver***

```
user@MacBook-Pro-de-User projet_stage_text_mining % cd data_quality_project
user@MacBook-Pro-de-User data_quality_project % ls
Data-Quality-Project.R  dq_project  env  requirements.txt
user@MacBook-Pro-de-User data_quality_project % source env/bin/activate
(env) user@MacBook-Pro-de-User data_quality_project % ls
Data-Quality-Project.R  dq_project  env  requirements.txt
(env) user@MacBook-Pro-de-User data_quality_project % pip install -r requirements.txt
Requirement already satisfied: appnope==0.1.3 in ./env/lib/python3.9/site-packages (from -r requirements.txt)
```

```
(env) user@MacBook-Pro-de-User data_quality_project % cd dq_project
(env) user@MacBook-Pro-de-User dq_project % python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
January 25, 2023 - 14:37:15
Django version 4.1.5, using settings 'dq_project.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Une fois que le serveur tourne, on peut aller sur RStudio et appeler les endpoints adéquates pour la correction et la catégorisation.

On aura besoin d'installer le package **httr** pour pouvoir effectuer des requêtes Http.

- **Correction :**

- On effectue un post à l'endpoint de correction en lui fournissant le dataset et la colonne à corriger: ***POST(url,body = list(file = name_of_dataset,target=target_column)***
- Transformer les résultats de l'API en dataframe: ***new_data <- as.data.frame(do.call(cbind, result_correction\$file))***
- On peut enregistrer les données dans un fichier csv: ***write.csv(new_data, paste0(path_to_save_data,"corrected.csv"), row.names=FALSE)***

```
library(httr)
dataset="/Users/user/Desktop/text-mining/VariableCibles.csv"
path_to_save_data="/Users/user/Desktop/"

url <- "http://127.0.0.1:8000/api/correction"
result_correction = content(POST(url,body = list(file = dataset,target="profession")))

#result to dataframe
new_data <- as.data.frame(do.call(cbind, result_correction$file))

#change list class to character in the dataframe
new_data$profession <- unlist(new_data$profession, use.names = FALSE)
new_data$Corrected <- unlist(new_data$Corrected, use.names = FALSE)

#save dataframe as csv
write.csv(new_data, paste0(path_to_save_data,"corrected.csv"), row.names=FALSE)
```

R	Global Environment	
Data		
new_data	20000 obs. of 2 variables	
\$ profession: chr	"responsable societe seunte globe" "menagere" "menagere" "di...	
\$ Corrected : chr	"Responsable société s'ente globe" "ménagère" "ménagère" "Dir...	
result_correction	Large list (5.1 MB)	
Values		
dataset	"/Users/user/Desktop/text-mining/VariableCibles.csv"	
path_to_save_data	"/Users/user/Desktop/"	
url	"http://127.0.0.1:8000/api/correction"	
Files	Plots	Packages
Help	Viewer	

	profession	Corrected
1	responsable societe seunte globe	Responsable société s'ente globe
2	menagere	ménagère
3	menagere	ménagère
4	directrice commerciale	Directrice commerciale
5	eleve	élève
6	eleve	élève
7	chauffeur	chauffeur
8	ouvrier agricole	Ouvrier agricole
9	agent du cadastre impots et domaine	Agent du cadastre impôts et domaine
10	autres	autres
11	pilote engin	Pilote engin
12	commercant	commerçant
13	commercante femme au foyer	commerçante femme au foyer
14	etudiant droits des affaires	étudiant droits des affaires
15	auxilliaire de vie	auxiliaire de vie
16	commercante	commerçante
17	aide soignante	aide-soignante
18	eleve	élève
19	program officier onu	programme officier ONU
20	aide a domicile	Aide à domicile
21	enseignant ministre agriculture	Enseignant ministre agriculture
22	femme au foyer	Femme au foyer
23	eleve	élève

- **Catégorisation:**

- On effectue un post à l'endpoint de categorization en lui fournissant le dataset corrigé (enregistré précédemment), la colonne cible qu'on devait corriger (pour effectuer une deuxième correction en se basant sur les catégories), et la liste de catégorie à utiliser pour la classification: ***POST(url2, body = list(file = corrected_datset,target=column_target,elements=list_categories))***

```
url2 <- "http://127.0.0.1:8000/api/categorization"
list_professions <- paste("ménagère","enseignement")
result_categorization = content(POST(url2, body = list(file = "/Users/user/Desktop/corrected.csv",target="profession",elements=list_professions)))

new_data2 <- as.data.frame(do.call(cbind, result_categorization$corrected_dataset))
new_data2$profession <- unlist(new_data2$profession, use.names = FALSE)
new_data2$Corrected <- unlist(new_data2$Corrected, use.names = FALSE)
new_data2$Categorie <- unlist(new_data2$Categorie, use.names = FALSE)
write.csv(new_data2, paste0(path_to_save_data,"categorized2.csv"), row.names=FALSE)
```

	profession	Corrected	Categorie
1	responsable societe seunte globe	Responsable société s'ente globe	None
2	menagere	ménagère	ménagère
3	menagere	ménagère	ménagère
4	directrice commerciale	Directrice commerciale	None
5	eleve	élève	None