Dans l'énoncé du TP, vous mentionnez que les zones sont du type "Vieux-Port", "China town", "Petite Italie". Mais les zones dans les data sont avec les zip code et code postal. Est-ce qu'il faut faire un transfert ou les garder comme ça?

Il faut les garder comme ça.

Pour le feature #6, est-ce qu'on doit différencier les villes qui ont le même nom, mais qui sont en réalité deux villes différentes? Il est possible de le faire avec les zip code, mais ce n'est pas si évident à priori, puisque certaines villes ont plusieurs zip code (ou code postal) différents.

Non pas besoin de les traiter (c'est hors du scope du travail)

Pour l'horaire (feat 26-30), est-ce qu'on peut considérer les restaurants qui n'ont aucune donnée comme donnée manquante, mais que les restaurants qui ont seulement quelques jours sans horaire comme étant lors de ces journées?

- Oui
  - On considère l'horaire manquant pour un restaurant si toutes les valeurs sont manquantes pour tous les jours de la semaine
  - Si les valeurs sont vides pour un restaurant pour seulement quelques jours de la semaine alors le restaurant est fermé pour ces jours-là.

Est-ce qu'il y a un avantage de travailler avec Google Colab?

- Quelques avantages:
  - Pas besoin de préparer l'environnement ou installer les librairies standards comme scikit-learn et pandas, ils sont déjà installés pour vous
  - Si votre environnement (ordinateur) est très lent ou avec des ressources moins que Colab (Colab offre 12GB de RAM)

Horaires.csv : Est-ce qu'un horaire de 00:00-00:00 signifie que le restaurant est ouvert 24h ?

Oui ça doit être considéré comme 24 h

Calcul #1 : Lorsque vous demandez un nombre, est-ce que c'est acceptable d'avoir un float, où faut-il toujours les convertir en Int. Pour les nombres à virgule (moyenne, écart-type, etc.) Est-ce qu'on doit arrondir les nombres ?

- C'est acceptable d'avoir un float, en fait pour quelques attributs comme la moyenne ou l'écart-type ça doit être un float (avec des chiffres après la virgule)
- Pas besoin d'arrondir les nombres

#1.24 : On demande l'écart-type par année. Est-ce qu'on s'attend à avoir un écart-type par année de check_in par restaurant ou un écart-type par restaurant. Ex : id 1 : (2019,0.3) (2018, 0.4) OU id1 ecart-type = 0.5.

Par restaurant -> id1 ecart-type = 0.5.

Pour chaque restaurant, on calcule le nombre de checkin par année (Count) et puis on prend l'ecart-type des valeurs

Ex : id1 : (2019,100) (2018,50) (2017,90) => id1 ecart-type = 21.6

Juste pour être sûre, quand vous dites dans énoncé (page 1) ''Le training dataset que vous avez préparé dans un fichier csv'', est-ce qu'il s'agit bien de fichier features.csv généré au cours de la tâche 1 et modifié au cours de la tâche 2?

Celui de la tache 2 : features_finaux.csv

Est-ce que je comprends bien que le choix d'approche pour la tâche 2 (prétraitement) est à notre discrétion? Si oui, cela signifie que les différentes équipes peuvent avoir les données d'apprentissage qui se diffèrent. Dans ce cas-ci, comment vous allez évaluer le travail? Est-ce que le résultat qu'on obtient doit correspondre exactement à ce que vous avez (calculé) pour être considéré comme correct?

Pour la tache de prétraitement, il existe plusieurs approches pour l'accomplir. Par exemple, parfois on peut remplacer les valeurs manquantes par zéro, d'autres fois c'est plus convenable de les remplacer par la moyenne ou le mode des autres valeurs du feature. Cela dit, il y a des méthodes qui ne s'appliquent pas sur certains features. Par exemple, si le feature prend des valeurs discrètes (exp : 1,2,3) et un enregistrement manque de valeur, on ne peut pas remplacer la valeur manquante par la moyenne des valeurs des autres enregistrements, car ça va nous donner des décimaux comme 1.5 ou 2.5, des valeurs qui ne s'appliquent pas pour ce feature.

Ce qui est important c'est de justifier la méthode utilisée pour chaque feature dans le rapport.

Est-ce qu'il y a une structure suggérée pour le rapport en pdf?

Oui. Un document est disponible dans moodle qui relate les directives de rédaction du rapport.

Je ne comprends clairement pas comment calculer feature 23 (moyenne_checkin). Est-ce que je comprends bien, que si on a par exemple un resto qui a reçu 10 checkins dans 2012, 40 dans 2013 et 100 dans 2019, la moyenne devrait être calculée à partir de nombre des années indiqués, donc 3; et dans ce cas-ci on obtient (10 + 40 +100) / 3 = 50 visites par année en moyenne. Ou bien, je devrais diviser par le nombre d'années dans intervalle, donc par 8, et cela me donnera 18,75 visites?

La première méthode: **(10 + 40 +100) / 3 = 50**

Pour feature 36 (prix), le jeu de données contient les valeurs 0, 1, 2, 3 et 4 pour décrire le prix. Quelle est la règle de transformation de ces valeurs dans l'échelle 1-2-3 demandée dans énoncé?

- 0 est considéré comme valeur manquante.
- Pour ce travail, et des fins de simplifications, 4 doit-être considéré comme 3.

Est-ce que c'est possible d'avoir un exemple de calcul d'attribut? Juste pour être sûr qu'on fait de bonnes choses et pour apprendre les bonnes pratiques et les astuces comment travailler avec Pandas. Je pose cette question aussi parce que l'exécution de code n'est pas vraiment rapide, alors je ne peux pas comprendre si c'est dû à ma mauvaise façon de travailler avec les jointures de tables et utilisation de fonctions lambda avec apply() qui nuisent à performance ou bien c'est normal avec la taille de données aussi grande.

- Il faut éviter l'usage de la fonction **apply()** sauf s'il y a une fonction complexe qui doit être appliquée sur chaque enregistrement.
- Autrement, il faut utiliser des dataframes avec la fonctionnalité de merge
  - Exemple : Si on veut calculer un feature qui s'appelle « moyenne_X » qui est la moyenne d'une colonne X dans la table «avis » :

```
dataframe_temporaire =
avis.groupby('restaurant_id')['X'].mean().reset_index(name='moyenne_X')
features = pd.merge(features,dataframe_temporaire,how="left", on="restaurant_id")
```

  - avis.groupby('restaurant_id') : regroupe les enregistrement de la table avis par « restaurant_id » et puis la fonction ['X'].mean() calcule la moyenne de la colonne ['X'] par restaurant_id.
  - reset_index(name='moyenne_X') : renomme la nouvelle colonne calculée à « moyenne_X »
  - Les résultats sont sauvegardés dans le dataframe : dataframe_temporaire
  - Puis on joigne les 2 dataframe pour transférer la colonne 'moyenne_X' du dataframe « dataframe_temporaire » au dataframe features

C'est quoi exactement un feature? Il y a aussi le concept du dataframe qui n'est pas clair pour moi. Est-ce qu'il faut faire un dataframe pour chaque resto et que pour chaque dataframe il faut rouler les 37 features ? De plus, le feature est seulement selon les données du resto ou il est selon tous les données de tous les restos?

- Le feature c'est un attribut ou une colonne dans un ensemble de données qui est utilisé pour l'entrainement des modèles d'apprentissage automatique. Quand vous lancer le premier fichier python « features.py », un fichier csv va être généré « feature.csv ». Dans ce fichier csv, les lignes sont des restaurants et les colonnes sont les features ou les attributs. Vous allez trouver que le fichier « features.csv » contient par défaut 3 features (moyenne_etoiles, ville et zone). Vous devrez ajouter les autres 34 features demandés dans l'énoncé à ce fichier.

- Le dataframe est un objet de la libraire Pandas de python. Le dataframe est simplement une table qui contient les données. Par exemple, au début du fichier code « feature.py », vous trouvez la commande :

- outilisateurs = pd.read_csv(data_path + "utilisateurs.csv")
  - Cette commande charge les données du fichier « utilisateurs.csv » dans un objet « utilisateurs» de type dataframe, qui est un simple conteneur des données du fichier « utilisateurs.csv »
  - En fait, vous pouvez vérifier le contenu de l'objet « utilisateurs» en insérant la commande suivante :
    - print(utilisateurs)

- Dans le fichier « features.py » vous devez ajouter les features au dataframe « features » initialisé au début du fichier par la ligne :
  - features = restaurants[['restaurant_id', 'moyenne_etoiles', 'ville', 'zone', 'ferme']].copy()

- Les features doivent être calculés par restaurant , alors à la fin de votre travail, le nombre d'enregistrements dans le dataframe « features » et le fichier « features.csv » et « features_finaux.csv » doit être égal au nombre total de restaurants dans le fichier « restaurants.csv » qui est 34268.

Dans le fichier features.py, pour les questions #15, #16, #20, #22, #23 et #24, on veut calculer ces attributs par restaurant?

Oui.

Exemple pour le #22, j'imagine qu'on veut le nombre total de visites par restaurant?

Oui.

Pour le calcul de nb_restaurants_zone
Ce qui est problématique pour moi, Dans la vidéo sur le TP, le démonstrateur a dit que la jointure doit être faite sur "restaurant_id". Alors que le dataframe temporaire n'a pas "restaurant_id".

Il y avait une petite erreur de prononciation. Dans la vidéo, il est dit qu'on joigne les tables par le id, alors qu'il fallait simplement dire : le id de la table temporaire, qui est dans ce cas "zone".

Pour le calcul de nb_avis_favorables_mention.
Définition: Le nombre total d'avis qui ont reçu au moins une mention "useful" ou "funny" ou "cool" ET le nombre d'étoiles de l'avis est >=3.

Est ce qu'on veut dire ici : Le nombre total d'avis qui ont reçu au moins une mention "useful" ou "funny" ou "cool" ET le nombre d'étoiles de l'avis est >=3 pour ce restaurant ? ou le nombre total de tous les restaurants?

On veut dire : Le nombre total d'avis qui ont reçu au moins une mention "useful" ou "funny" ou "cool" ET le nombre d'étoiles de l'avis est >=3 pour ce restaurant.

Pour le calcul de nb_avis_favorables_mention.
Définition : L'écart type de la moyenne des étoiles par année. Il faut estimer la moyenne des étoiles par années. Puis, calculer l'écart type sur ces valeurs.

Est-ce qu'il faut créer une colonne dans features avec la même valeur copiée pour toutes les rows soit 34268 fois?

Non, c'est calculé par restaurant.
Pour chaque restaurant on calcule la moyenne des étoiles par année et puis on prend l'écart type des moyennes des années.

Pour un restaurant X:

Restaurant X				
Année	2014	2015	2016	2017
Avis	3	5	4	4
	2	4	2	1
	4	2	4	4
	4	1	3	
	3	4		
Moyenne par année	3.2	3.2	3.25	3
ecart-type	**0.10**			

Pour le calcul de nb_heures_ouverture_semaine

Est-ce qu'on doit définir une fonction Python qui calcule la durée d'ouverture, c'est-à-dire, faire l'extraction des horaires d'ouverture et fermeture de la chaine de caractères (format H:M-H :M) pour les 7 attributs de la table « Horaires »), et puis les convertir en « datetime » et puis faire la différence ?
Ou bien
Il est préférable de faire la recherche dans la bibliothèque Pandas d'une fonction qui calcule automatiquement la durée d'ouverture à partir de la chaine de caractères (format H :M-H :M) et l'appliquer sur les 7 attributs de la table « Horaires » ?

Les deux méthodes seront acceptées. Vous pouvez rechercher s'il existe une telle librairie dans pandas. Si elle n'existe pas, vous pouvez l'implémenter vous-même.

Pour le calcul de nb_heures_ouverture_semaine

	11:0-2:30	11:0-2:30	11:0-2:30	11:0-2:30	11:0-2:30	11:0-2:30

Comment considère-t-on les horaires suivants ?
La plupart des données suivent le format 24h mais de rares cas, comme celui-ci, sont au format 12h. Doit-on les gérer comme des format 24h, ce qui produira du bruit ou bien les convertir en format 24h (11:30-2:30 deviendrait 11:30-14:30) ?
Je suis parti du principe qu'il fallait les convertir en format 24h, j'ai donc produit ce code, auriez-vous une piste pour une solution plus optimale ?

Votre proposition est acceptable comme solution « ajouter 12 heures à l'heure de fermeture ». Mais dans ce cas on remarque que ce restaurant sera ouvert seulement 3 ½ chaque jour.

Une autre solution c'est d'ajouter 24 heures à l'heure de fermeture. Dans ce cas nous considérons que l'heure de fermeture est 2:30 AM. Mais pour ce cas-là vous deviez traiter le format des heures comme « datetime », car nous sommes entrain de considérer que l'heure de fermeture est le lendemain.

Pour avoir une idée du niveau de détails et de développement attendu dans le rapport, à combien de pages environ est-ce que vous vous attendez? Combien de pages vous semblent trop, et combien vous semblent pas assez?

Il n'y a pas un maximum pour le nombre de pages. On apprécie toujours un rapport très bien écrit qui suit les directives avec une présentation analyse des résultats adéquate. Une analyse sommaire/superficielle avec des mots génériques n'est pas recommandée.