INF7370 Apprentissage automatique – Hiver 2023

Travail pratique # 1

Envoyé le 18 janvier 2023
Date de remise : le 27 février 2023 à 9h30

Important

1. Le TP est noté sur 25 points.
2. **Le travail peut être fait individuellement ou en équipe de deux étudiant-e-s maximum**.
3. Vous devez travailler seulement avec les trois fichiers : features.py ET processing.py ET learning.py
4. Ne changé pas les noms de ces trois fichiers Python.
5. Inclure votre code dans les endroits spécifiés dans ces trois fichiers Python.
6. Indiquer votre nom dans les trois fichiers.
7. Vous devez soumettre :
   - Le training dataset que vous avez préparé dans **un fichier csv**. Ne pas remettre les fichiers de données originaux.
   - Les trois fichiers de code : features.py, processing.py et learning.py
   - Votre rapport en format pdf identifié à votre nom. **Le rapport doit suivre les directives mentionnées dans l'énoncé ainsi que les directives mentionnées dans le fichier "Directives pour la rdaction du rapport du TP1 - disponible dans moodle.**
   - Remettre le training dataset (format csv), le rapport en pdf, les trois fichiers de code Python dans **un seul fichier zip identifié à votre nom**. N'utiliser pas un nom générique (exemple : TP1).
8. Soigner la présentation de votre rapport.
9. La date de remise est le 27 février 2023 à 9h30. Moodle ferme à 9h30. Aucun travail ne sera accepté à partir de cette heure.
10. **La remise se fait via Moodle uniquement. Aucune remise par courriel**.
11. **Les soumissions par courriel ne seront pas corrigées et le TP sera considéré comme non remis**.

Contexte

Le but de ce travail est d'analyser et de comparer la performance de quelques algorithmes d'apprentissage automatiques : Arbre de décision, Forêt d'arbres décisionnels (Random Forest) et Classification bayésienne naïve. Les algorithmes doivent être entraînés pour classifier des restaurants en deux situations : fermeture définitive / restaurant encore ouvert. À cette fin, vous devez utiliser le langage **Python** avec la librairie **scikit-learn**. Pour entraîner et tester les algorithmes, il faut travailler avec des données offertes par Yelp [1]. Nous avons déjà effectué un premier traitement sur les données brutes afin de sélectionner seulement les restaurants avec les informations qui sont nécessaires pour ce TP.

Yelp

Yelp est une application mobile qui sert comme un répertoire en ligne pour la recherche des entreprises et des commerces de différents types. Yelp permet aux utilisateurs de chercher les restaurants par de différents critères, comme : par région, par type de cuisine (p. ex. Chinoise, Italienne, Française), par catégorie (p. ex. Bar, Pub, Café). Yelp permet également de fournir d'autres informations comme les heures d'ouverture, gamme de prix, etc.

En plus, Yelp offre aux utilisateurs la possibilité de rédiger des avis sur le restaurant qu'ils ont visités. Cela sert comme une sorte d'évaluation des clients du service fourni au restaurant. L'avis d'un client consiste en un texte court et d'un score de 1 à 5 étoiles. Les utilisateurs peuvent aussi écrire des conseils "Tips" sur les restaurants et signaler la date de leur visite "Check-in". Yelp offre aussi un petit réseau social où les utilisateurs peuvent être amis avec d'autres utilisateurs. Les usagers de Yelp peuvent aussi étiqueter les avis postés sur le site comme "useful", "funny" ou "cool". Ces réactions associées aux avis relatent une certaine indication de la qualité et l'authenticité de l'avis lui-même. Aussi, Yelp désigne un certain nombre d'utilisateurs comme utilisateurs "Élites".

1. Yelp offre une partie de ses données pour des buts d'enseignement et de recherche. Données disponibles à partir de https ://www.yelp.com/dataset

Données

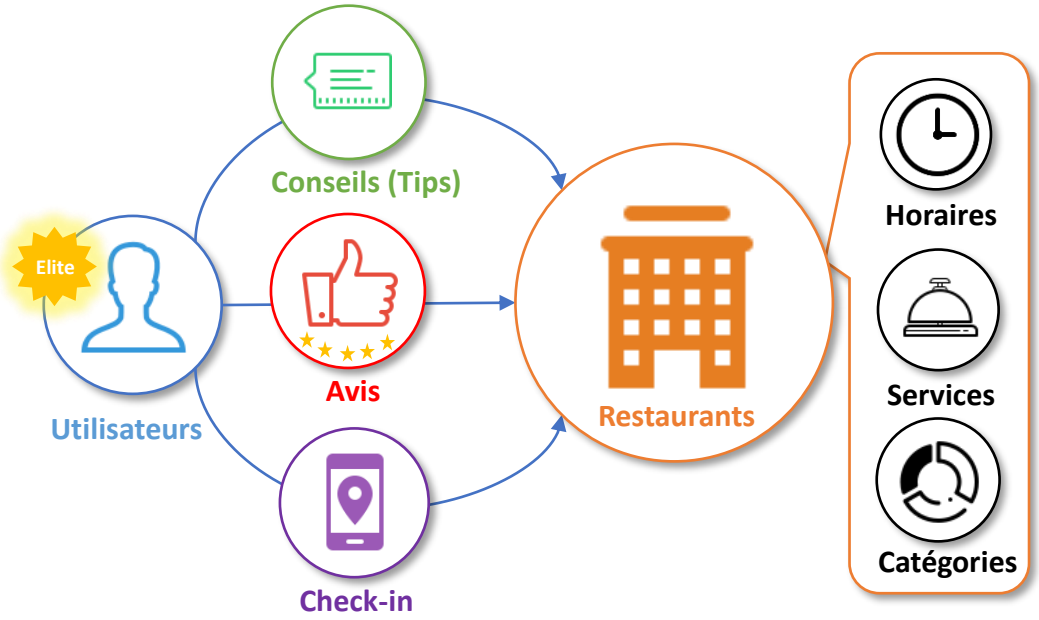

FIGURE 1 – Schéma décrivant les relations entre les composantes principales des données.

Les données associées à ce TP sont disponibles dans **8 tables** en format **csv** (fichiers délimités par des virgules) [2] dans le OneDrive du cours. Le schéma présenté dans la Figure 1 démontre les relations entres les différentes composantes des données :

- **Utilisateurs** : Contient la liste des utilisateurs avec quelques détails comme les noms d'usager et leurs statuts (Élite ou Régulier).
- **Avis** : Contient les informations qui se rattachent aux avis rédigés pour chaque restaurant avec le nombre d'étoiles accordé. On trouve également le nombre de réactions des autres usagers (p. ex. "useful", "funny", "cool") associé à chaque avis. Il est à noter que nous avons enlevé le texte des avis afin d'éviter de créer une base de données volumineuse.
- **Conseils (tips)** : Contient les informations qui se rattachent aux conseils (tips) rédigés par les utilisateurs sur les restaurants avec le nombre de "compliments" reçu pour chaque conseil. Il est à noter que les conseils sont un moyen de transmettre des informations clés sur un restaurant - comme le meilleur moment pour visiter ou plats du jours - sans rédiger un avis complet [3]. Il est à noter que nous avons enlevé le texte des conseils (tips) afin d'éviter de créer une base de données volumineuse.

2. Nous avons déjà extrait ces données et effectué une première préparation qui a résulté en des ensembles de données réparties sur 8 fichiers csv

3. `https://www.yelp-support.com/article/What-are-tips?l=en_US`

— **Check-in** : Contient les dates et heures de visite par restaurant.

— **Restaurants** : Contient la liste des restaurants avec les informations suivantes : le nom du restaurant, l'emplacement, nombre d'étoiles (moyenne générale) et le statuts : fermeture définitive ou restaurant encore ouvert (c'est la classe qui guide l'entraînement des algorithmes d'apprentissage).

— **Horaires** : Contiens les heures d'ouverture de chaque restaurant pour les sept jours de la semaine.

— **Services** : Contiens la liste des services offerts par chaque restaurant. Exemple : bon pour les groupes, avec une terrasse, bon pour les enfants, livraison possible, etc.

— **Catégories** : Contiens la liste des catégories des restaurants (Breakfast & Brunch, Italien, Seafood, etc.).

Le tableau 1 affiche le nombre d'enregistrements de chaque table de données (chaque fichier csv fourni). À la fin de l'énoncé, vous trouverez davantage de détails sur chaque table de données.

Table	Taille
Utilisateurs	748 247
Avis	207 3685
Conseils (tips)	416 803
Check-in	549 2282
Restaurants	34 268
Horaires	34 268
Services	34 268
Catégories	89 945

TABLE 1 – Le nombre d'enregistrements dans chaque ensemble de données.

Tâches

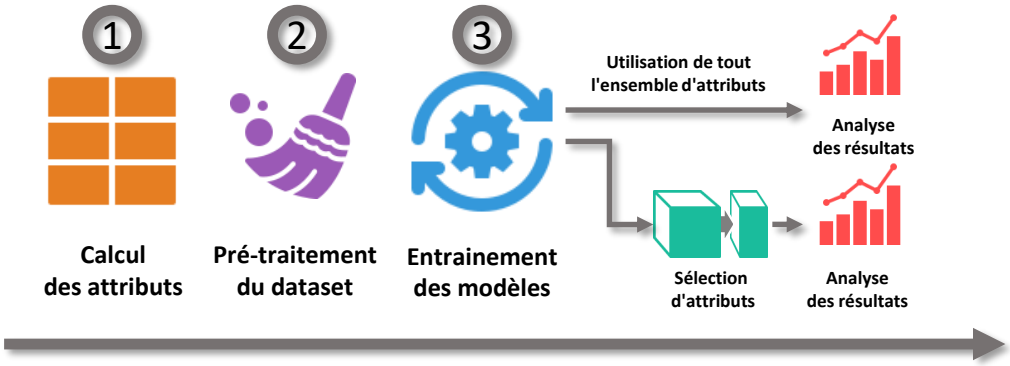

FIGURE 2 – Les tâches principales du travail.

Le travail est divisé en 3 tâches. Tâche 1 : calcul des attributs (features engineering) à partir des données fournies. Tâche 2 : pré-traitement du dataset issu de la première tache. Tâche 3 : entraînement des modèles et analyse des résultats.

Tache 1 : Calcul des attributs (features engineering)

Votre première tâche consiste à utiliser les données collectées (les 8 fichiers csv) pour bâtir un ensemble d'apprentissage exploitable par les algorithmes d'apprentissage automatique. Pour construire cet ensemble d'apprentissage, il faut calculer, pour chaque restaurant, un certain nombre d'attributs/caractéristiques (features) qui aident à distinguer les restaurants fermés (fermeture définitive) des restaurants encore ouverts. Pour la construction des features nous recommandons l'usage de la libraire **Pandas**. Vous devez implémenter les 37 features [4] suivants, **et ce pour chaque restaurant** :

1. **moyenne_etoiles** : La moyenne générale des scores en étoiles (entre 1 et 5).
2. **ville** : La ville où se trouve le restaurant.
3. **zone** : La partie de la ville où se trouve le restaurant (p. ex. Vieux-Port, China Town, Little Italy, etc.).
4. **nb_restaurants_zone** : Le nombre de restaurants dans la zone associée au restaurant en question.
5. **zone_categories_intersection** : Le nombre de restaurants dans la même zone qui partagent au moins une catégorie avec le restaurant en question.

4. Dans votre implémentation, il faut absolument utiliser les mêmes noms de features qui sont listés dans cet énoncé.

6. **ville_categories_intersection** : Le nombre de restaurants dans la même ville qui partagent au moins une catégorie avec le restaurant en question.
7. **nb_restaurant_meme_annee** : Le nombre de restaurants qui sont ouverts leurs portes dans la même année que le restaurant en question. Ici, on considère que la première année d'un restaurant correspond à l'année de la première publication d'un avis sur ce restaurant sur Yelp.
8. **ecart_type_etoiles** : 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.).
9. **tendance_etoiles** : La différence entre la moyenne des étoiles de la dernière année et la moyenne des étoiles de la première année d'un restaurant. Ici, on considère la première année d'un restaurant correspond à l'année de la première publication d'un avis sur ce restaurant sur Yelp.
10. **nb_avis** : Le nombre total d'avis pour ce restaurant.
11. **nb_avis_favorables** : Le nombre total d'avis favorables et positifs pour ce restaurant. On considère un avis "favorable" si son nombre d'étoiles est $>= 3$.
12. **nb_avis_defavorables** : Le nombre total d'avis défavorables pour ce restaurant. On considère un avis comme "défavorable" si son nombre d'étoiles est < 3.
13. **ratio_avis_favorables** : Le nombre d'avis favorables et positifs sur le nombre total d'avis pour ce restaurant.
14. **ratio_avis_defavorables** : Le nombre d'avis défavorables sur le nombre total d'avis pour ce restaurant.
15. **nb_avis_favorables_mention** : 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 $>= 3$.
16. **nb_avis_defavorables_mention** : 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.
17. **nb_avis_favorables_elites** : Le nombre total d'avis favorables pour un restaurant qui sont rédigés par des utilisateurs élites. Dans ce travail, on considère un utilisateur élite si son statut est "élite" (élite = 1 dans la table Utilisateurs) ET il a rédigé au moins 100 avis au total ET il a au moins 100 avis avec mention.
18. **nb_avis_defavorables_elites** : Le nombre total d'avis défavorables pour un restaurant qui sont rédigés par des utilisateurs élites. Dans ce travail, on considère un utilisateur élite si son statut est "élite" (élite = 1 dans la table Utilisateurs) ET il a rédigé au moins 100 avis au total ET il a au moins 100 avis avec mention.
19. **nb_conseils** : Le nombre total de conseils (tips) associés à un restaurant.
20. **nb_conseils_compliment** : Le nombre total de conseils qui ont reçu au moins un compliment (voir Table Conseils).

21. **nb_conseils_elites** : Le nombre total de conseils sur un restaurant qui sont rédigés par des utilisateurs élites. Dans ce travail, on considère un utilisateur élite si son statut est "élite" (élite = 1 dans la table Utilisateurs) ET il a rédigé au moins 100 avis au total ET il a au moins 100 avis avec mention.
22. **nb_checkin** : Le nombre total de visites.
23. **moyenne_checkin** : La moyenne de visites par année.
24. **ecart_type_checkin** : L'écart type de visites par année. Ici, on calcul l'écart type pour le total des visites par année.
25. **chaine** : Prend 0 ou 1. La valeur 1 indique que le restaurant fait parti d'une chaîne (p. ex. McDonald). On considère un restaurant comme il fait partie d'une chaîne, s'il existe un autre restaurant dans la base de données qui a le même nom.
26. **nb_heures_ouverture_semaine** : Le nombre total d'heures d'ouverture du restaurant par semaine.
27. **ouvert_samedi** : Si le restaurant est ouvert les samedis (valeur booléenne : 0 ou 1).
28. **ouvert_dimanche** : Si le restaurant est ouvert les dimanches (valeur booléenne : 0 ou 1).
29. **ouvert_lundi** : Si le restaurant est ouvert les lundis (valeur booléenne : 0 ou 1).
30. **ouvert_vendredi** : Si le restaurant est ouvert les vendredis (valeur booléenne : 0 ou 1).
31. **emporter** : Si le restaurant offre le service à emporter (valeur booléenne : 0 ou 1).
32. **livraison** : Si le restaurant offre le service de livraison (valeur booléenne : 0 ou 1).
33. **bon_pour_groupes** : Si le restaurant est approprié pour les groupes (valeur booléenne : 0 ou 1).
34. **bon_pour_enfants** : Si le restaurant est approprié pour les enfants (valeur booléenne : 0 ou 1).
35. **reservation** : Si on a besoin de faire une réservation au restaurant (valeur booléenne : 0 ou 1).
36. **prix** : Le niveau de prix du restaurant. Il existe trois niveaux, 1 (abordable), 2 (moyen) et 3 (coûteux).
37. **terrasse** : Si le restaurant a une terrasse (valeur booléenne : 0 ou 1).

Une fois les features sont calculés (et ce pour chaque restaurant), il faut regrouper tous les restaurants dans un même fichier et ajouter une colonne qui désigne la classe : 1 pour les restaurants qui ont fermé leurs portes définitivement et 0 pour les restaurants encore ouverts.

Tache 2 : Pré-traitement

Cette étape représente une phase de prétraitement afin de rendre les données cohérentes et prêtes pour l'analyse. Cette étape consiste en :

- Remplacement des valeurs manquantes. Les valeurs manquantes peuvent être remplacées par des 0, ou remplacées par la moyenne ou le mode. La méthode choisie doit dépendre de la nature de chaque feature.
- Pour les deux attributs "ville" et "zone" avec des valeurs symboliques, il faut effectuer une transformation de ces symboles en utilisant la fonction Categorical (de la librairie Pandas). Exemple utilisant la fonction *Categorical* de Pandas :

```
import pandas as pd

features.ville = pd.Categorical(features.ville)
features.ville = features.ville.cat.codes
```

Vous pouvez aussi appliquer d'autres mesures de traitement aux données. Indiquez chaque méthode de traitement utilisée et expliquer votre démarche.

Tache 3 : Entraînement des modèles et analyse comparative

3.1 Utilisation de tout l'ensemble d'attributs

- Appliquer les algorithmes : Arbre de décision, Random Forest, Bagging, AdaBoost et Classification bayésienne naïve.
- Comparer la performance de ces algorithmes en se basant sur :
  - Le taux des vrais positifs (TP Rate) – de la classe Restaurants fermés définitivement.
  - Le taux des faux positifs (FP Rate) – de la classe Restaurants fermés définitivement.
  - F-measure de la classe Restaurants fermés définitivement.
  - La surface sous la courbe ROC (AUC).
- Illustrer vos résultats dans des tables et/ou graphes.
- Afficher la matrice de confusion de chaque algorithme.
- Analyser vos résultats. Indiquer les points forts et points faibles de chaque algorithme et expliquer pourquoi. Vous devez bien élaborer cette partie. Fournir une analyse critique et non des commentaires génériques.

3.2 Sélection d'attributs :

Identifier les 10 meilleurs attributs en utilisant la mesure du Gain d'information (Information Gain). Selon les résultats obtenus (suite à l'application du gain d'information) faire :

— Monter les 10 meilleurs attributs dans un tableau (par ordre croissant selon le score obtenu par le Gain d'information).

— Appliquer les algorithmes : Arbre de décision, Random Forest, Bagging, AdaBoost, Classification bayésienne naïve sur l'ensemble de données, mais avec les 10 attributs sélectionnés seulement.

— Comparer la performance de ces algorithmes en se basant sur
  — Le taux des vrais positifs (TP Rate) – de la classe Restaurants fermés définitivement.
  — Le taux des faux positifs (FP Rate) – de la classe Restaurants fermés définitivement.
  — F-measure de la classe Restaurants fermés définitivement.
  — La surface sous la courbe ROC (AUC).

— Illustrer vos résultats dans des tables et/ou graphes.

— Afficher la matrice de confusion de chaque algorithme.

— Quelles sont vos conclusions par rapport aux tests effectués sur l'ensemble de données avec tous les attributs (tests effectués dans Tâche 3.1) ?

Voici les librairies qui peuvent être utilisées :

— **Scikit-Learn** : Bibliothèque libre et open source implémentée avec Python, dédiée a l'apprentissage automatique. Elle est conçue pour s'harmoniser avec d'autres bibliothèques Python (ou autres), notamment NumPy, SciPy, etc.

— **Scikit-plot** : Bibliothèque libre et open source implémentée avec Python, utilise pour l'affichage des graphiques principalement les courbes ROC et AUC.

— **Numpy** : "Numerical Python" fournit une interface pour stocker et effectuer des opérations sur les données. D'une certaine manière, les tableaux Numpy sont comme les listes en Python, mais Numpy permet de rendre les opérations beaucoup plus efficaces, surtout sur les tableaux de grande taille qui sont au cœur de l'écosystème de la Data Science.

— **Pandas** : Fournis deux structures de données fondamentales, la "Serie" et le "DataFrame". On peut voir ces structures comme une généralisation des tableaux et des matrices de Numpy. La différence entre les structures de Pandas et celles de Numpy c'est la définition explicite par l'utilisateur des indices et des index sur les objets (matrices).

— **Matplotlib** : "Mathematic Plot library" est une bibliothèque de traçage Python 2D qui produit des figures de qualité de publication dans une variété de formats papier et d'environnements interactifs entre plates-formes. Matplotlib peut être utilise dans les scripts Python, les Shells Python et IPython, les notebook Jupyter, etc.

Google Colab

Vous pouvez utiliser Google Colab pour réaliser ce travail. Pour utiliser Colab, vous avez besoin d'un compte de Google (Gmail). L'usage de Colab est gratuit et illimité.

Sachez toutefois que les fichiers que vous générer ou télécharger seront supprimés lorsque la session sera déconnectée. La session ne dure que 6 heures au maximum.

Pour télécharger les images sur Colab, il est recommandé de les télécharger en tant que fichier zip, puis de les décompresser sur Colab. Vous pouvez utiliser la commande suivante pour les décompresser :

```
!unzip donnees.zip
```

Si vous ne voulez pas télécharger vos fichiers à chaque fois, vous pouvez placer les données sur Google Drive et connecter Colab à MyDrive par la commande[5] :

```
from google.colab import drive
drive.mount('/content/drive')
```

Mais sachez qu'avec cette méthode, l'entraînement sera plus lent, car les fichiers devront être retirés de MyDrive.

5. Au lancement de cette commande, Colab va vous présenter un lien pour authentifier l'accès à MyDrive. Il faut copier le code généré par Google dans Colab afin de connecter le Drive.

Description des données

Utilisateurs

Colonne	Description
utilisateur_id	L'identifiant (ID) de l'utilisateur
nom	Le nom de l'utilisateur
elite	Utilisateur élite ? 1 : Oui 0 : Non
nb_avis	Le nombre total d'avis rédigés par cet utilisateur
nb_avis_mention	Le nombre total d'avis rédigés par cet utilisateur que d'autres utilisateurs ont étiquetés cool ou funny ou useful

Avis

Colonne	Description
avis_id	L'identifiant (ID) de l'avis
utilisateur_id	Référence de l'identifiant (ID) de l'utilisateur
restaurant_id	Référence de l'identifiant (ID) du restaurant
etoiles	Le nombre d'étoiles accordé
useful	Le nombre de personnes qui ont trouvé cet avis useful
funny	Le nombre de personnes qui ont trouvé cet avis funny
cool	Le nombre de personnes qui ont trouvé cet avis cool
date	La date de la rédaction de l'avis

Conseils (Tips)

Colonne	Description
utilisateur_id	Référence de l'identifiant (ID) de l'utilisateur
restaurant_id	Référence de l'identifiant (ID) du restaurant
nb_compliments	Le nombre de compliments que ce conseil a reçus
date	La date de la rédaction du conseil

Check-in

Colonne	Description
restaurant_id	Référence de l'identifiant (ID) du restaurant
date	La date et l'heure de la visite au restaurant

Restaurants

Colonne	Description
restaurant_id	L'identifiant (ID) du restaurant
nom	Le nom du restaurant
moyenne_etoiles	La moyenne générale d'étoiles de ce restaurant (entre 1 et 5)
ville	La ville où se situe le restaurant
zone	La zone de la ville où se situe le restaurant
ferme	Le statut du restaurant (c'est la classe à utiliser pour l'entraînement) 1 : Fermé 0 : Ouvert

Horaires

Colonne	Description
restaurant_id	Référence de l'identifiant (ID) du restaurant
lundi	L'heure d'ouverture et de fermeture du restaurant (format H :M-H :M)
mardi	L'heure d'ouverture et de fermeture du restaurant (format H :M-H :M)
mercredi	L'heure d'ouverture et de fermeture du restaurant (format H :M-H :M)
jeudi	L'heure d'ouverture et de fermeture du restaurant (format H :M-H :M)
vendredi	L'heure d'ouverture et de fermeture du restaurant (format H :M-H :M)
samedi	L'heure d'ouverture et de fermeture du restaurant (format H :M-H :M)
dimanche	L'heure d'ouverture et de fermeture du restaurant (format H :M-H :M)

Services

Colonne	Description
restaurant_id	Référence de l'identifiant (ID) du restaurant
emporter	Si le restaurant offre des repas à emporter 1 : Oui 0 : Non
livraison	Si le restaurant offre le service de livraison 1 : Oui 0 : Non
bon_pour_groupes	Si le restaurant est bon pour les groupes 1 : Oui 0 : Non
bon_pour_enfants	Si le restaurant est bon pour les enfants 1 : Oui 0 : Non
reservation	Si on a besoin de faire une réservation au restaurant 1 : Oui 0 : Non
prix	Le niveau de prix du restaurant 1 : Abordable 2 : Moyen 3 : Coûteux
terrasse	Si le restaurant a une terrasse 1 : Oui 0 : Non

Catégories

Colonne	Description
restaurant_id	Référence l'identifiant (ID) du restaurant
categorie	La catégorie du restaurant Un restaurant peut avoir plusieurs catégories (p. ex. Sandwiches, Gluten-Free, Vegetarian, Vegan)