Université du Québec à Montréal

INF7710 - Théeorie et applications de la fouille d'associations

Projet Final

Mining Yelp Data
Pour la prédiction de la fermeture &
La recommandation des restaurants

Par
Maroun Haddad - HADM15088902

Avril 2020

Table des matières

Chapitre 1

Introduction

Le secteur des restaurants est un marché diversifié et en plein essor, mais qui souffre d'une volatilité et d'une vulnérabilité de nombreux facteurs économiques, géographiques et culturels. Dans ce travail, nous utilisons les données offertes par Yelp pour s'attaquer à deux taches reliées à l'amélioration de la performance des restaurants. Premièrement, nous essayons de prédire la fermeture d'un restaurant en fouillant ses différentes caractéristiques et son historique et en trouvant des règles d'associations reliées aux succès des restaurants dans les différentes régions de Montréal. Deuxièmement, nous tentons recommander des restaurants à un utilisateur en construisant des modèles qui peuvent bien encoder la complexité des relations Client/Restaurants pour fouiller les tendances générales reliées aux goûts des utilisateurs et la qualité des restaurants afin de bien prédire l'évaluation qu'un utilisateur donnera à un restaurant.

1.1 L'application Yelp

Yelp est une application mobile qui sert comme un répertoire en ligne pour la recherche des entreprises et des commerces de différents types. En fait, les restaurants été toujours au coeur de cette application et c'est sur ce type d'entreprise que nous nous focalisons dans cette étude. Yelp permet aux utilisateurs de rechercher les restaurants par de différents critères, comme leurs régions, leurs catégories telles que la cuisine du restaurant (chinois, italien ...), leurs types (Bar, pub, café...) et les attributs et les services dont ils offrent, exemple leurs heures d'ouverture, leurs gammes de prix et leur atmosphère (figure1.1).

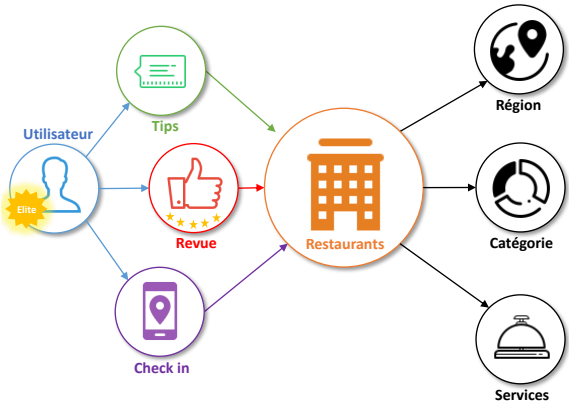

FIGURE 1.1 – Composantes de l'application Yelp

En plus, Yelp offre aux utilisateurs la possibilité de rédiger des revues sur le restaurant qu'ils ont visité qui servent comme une sorte d'évaluation de leur expérience durant leur visite. Leur revue consiste d'un texte court et d'un score de 1 à 5 étoiles. Cette revue sert comme référence aux autres utilisateurs sur la qualité de l'établissement évalué. Les utilisateurs peuvent aussi écrire des commentaires "Tips" sur les restaurants et signaler le temps de leur visite "checkin". Yelp offre aussi un petit réseau social où les utilisateurs peuvent être amis avec d'autres utilisateurs et interagir avec les revues des autres en signalant si le revue été utile, rigolo ou cool. Les réactions sur la revue sont une indication de la qualité et l'authenticité de la revue elle-même. Aussi, Yelp désigne un certain nombre d'utilisateurs comme utilisateurs "Élites". Même si Yelp ne divulgue pas le processus de choix des utilisateurs élites(wikipedia, 2020), il est spéculé que les revues de ces utilisateurs ont plus de poids que les utilisateurs réguliers.

1.2 Description du Dataset

Amérique du Nord		**Montréal**	
Restaurant	59371	**Restaurant**	5420
Utilisateur	1148098	**Utilisateur**	45022
Revue	4201684	**Revues**	137992
Commentaire	810342	**Commentaire**	9031
Checkin	57402	**Checkin**	5420

TABLE 1.1 – Les figures du dataset Yelp

Yelp offre un dataset pour les recherches(yelp, 2020) formé de 5 fichiers en format XML. Le tableau (1.1) décrit le dataset entier filtré par type "Restaurant", ces données couvrent l'Amérique du Nord. Pour l'analyse et la tâche de recommandation, nous focalisons sur les données de Montréal. Nous listons par la suite les attributs que nous allons utiliser dans notre étude, tableau(1.2).

- **Restaurant** : contient la liste des restaurants avec leurs coordonnées, leur nombre d'étoiles (moyenne générale), leur statut (ouvert ou fermé), leurs catégories (un restaurant peut avoir plusieurs catégories, exemple : Café / Italien), leurs services et leurs heures d'ouverture.
- **Utilisateur** : Contient la liste des usagers de l'application avec leur statut(Élite ou Régulier) et la liste de leurs amis.
- **Revue** : Contient les textes des revues rédigées par les utilisateurs avec leurs nombres d'étoiles, les réactions que cette revue a reçues et la date d'émission.
- **Commentaire** : La liste des commentaires écrits par les utilisateurs avec les réactions qu'ils ont reçus.
- **Checkin** : La liste des dates de visite par restaurant.

Restaurant	**Utlisateur**	**Revue**	**Commentaire**	**Checkin**
Nom	Nom	Utilisateur	Utilisateur	Restaurant
Code Postal	Elite	Restaurant	Restaurant	Date
Longitude	Amis	Etoiles	Réactions	
Latitude		Réactions	Texte	
Etoiles		Texte	Année	
Statut		Année		
Catégories				
Services				
Heurs ouverts				

TABLE 1.2 – Liste de tables du dataset Yelp

1.3 La démarche du travail

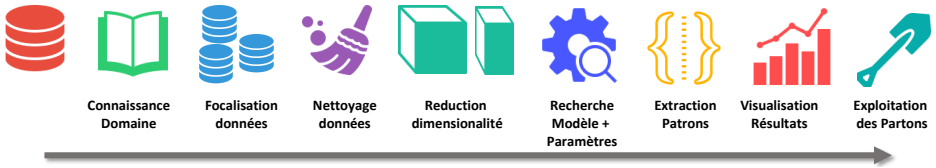

FIGURE 1.2 – Le processus KDD

Pour les deux taches que nous allons exécuter sur les données, nous suivons les étapes du processus KDD(Fayyad *et al.*, 1996). Premièrement nous focalisons sur un sous-ensemble des données, puis nous exécutons une suite de requêtes pour développer de la connaissance sur le domaine que nous étudions. Ensuite, nous nettoyons les données et nous réduisons leur dimensionnalité. Après, nous cherchons les meilleurs modèles et leurs meilleurs paramètres. Enfin, nous extrayons des patrons et nous visualisons les résultats.

Dans le chapitre suivant nous commençons par la première tâche, où nous visons prédire la fermeture d'un restaurant selon ses différents critères.

Chapitre 2

Prédiction de Fermeture

Dans ce chapitre nous détaillons la démarche de la première tache qui se concerne avec la prédiction de la fermeture des restaurants. Nous commençons par faire une suite de requêtes sur le dataset de Montréal pour comprendre les facteurs qui affectent le statut d'un restaurant. Nous construisons par la suite les attributs relatifs aux connaissances que nous avons acquis .

2.1 Analyse et Assemblage des Attributs

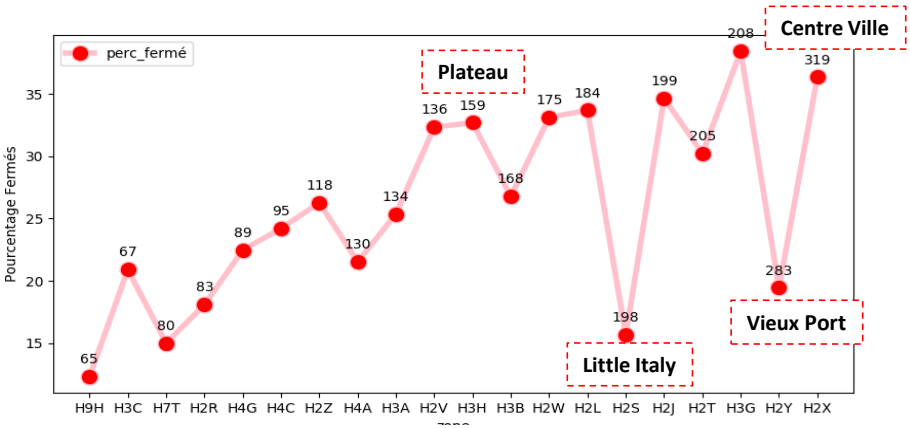

FIGURE 2.1 – Le processus KDD

Premièrement, nous regardons la stabilité des différentes zones à Montréal. Par stabilité on veut dire le pourcentage de restaurants fermés dans chaque zone. Nous divisons les zones selon la première section du code postal (exemple Vieux-Port : H2Y).

Quand nous traçons sur un graphe le nombre de restaurants par zone (figure 2.1), nous remarquons qu'il y a une corrélation positive entre le nombre de restaurants et le pourcentage de restaurants fermés dans cette zone. Ceci est normal, car plus qu'il y a de restaurants dans une certaine zone, plus il y a de compétition locale et plus le risque de fermeture va augmenter.

Alors, nous ajoutons les attributs :

- **zone** : la zone du restaurant, la première section du code postal.
- **zone_number_restaurants** : Nombre de restaurants dans la zone du restaurant.

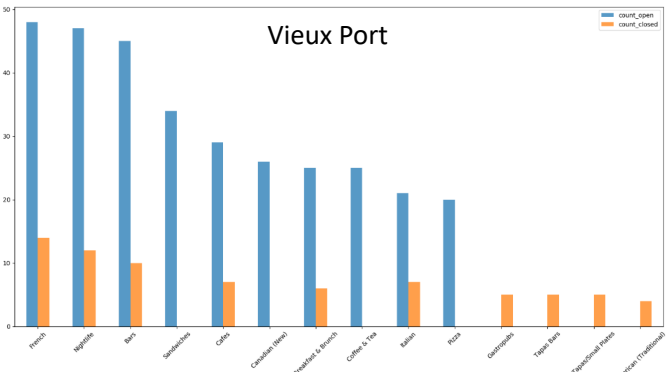

FIGURE 2.2 – Catégories des zones stables

Or, on remarque qu'il a 2 grandes exceptions à cette règle, le Vieux-Port et Little Italie, où le nombre de restaurants est élevé, mais les 2 régions sont plus ou moins stables. Alors pour discerner la différence entre ces zones et d'autres zones non stables, nous regardons les catégories par zones. Nous traçons les tops 10 catégories des restaurants ouverts contre les tops 10 catégories fermées (figure 2.2). Nous remarquons que pour le Vieux-Port et Little Italie, ils sont dominés par certaines catégories stables, exemple les restaurants français pour le vieux port et les restaurants italiens pour Little Italie.

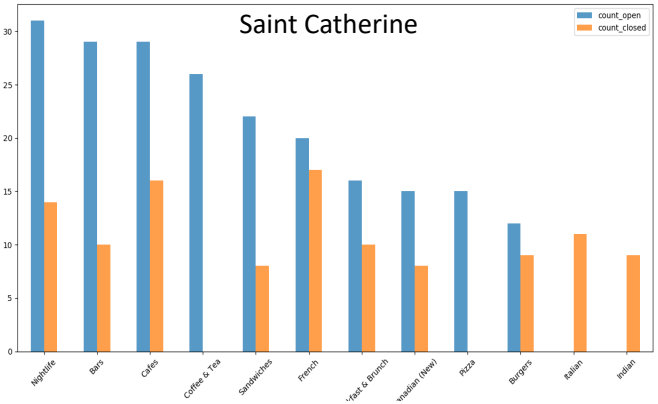

FIGURE 2.3 – Catégories des zones non-stables

Or, quand on regarde les catégories de Sainte-Catherine et le Golden Square, nous remarquons que cette dominance par certaines catégories est absente et toutes les catégories sont plus ou moins non-stables.

Alors nous ajoutons les attributs suivants :

- **catégories** : La liste des catégories auxquelles le restaurant appartient.
- **category_zone_inter** : Le nombre de restaurants dans la même zone qui partagent les mêmes catégories avec le restaurant.
- **category_city_inter** : Le nombre total de restaurants qui partagent les mêmes catégories avec le restaurant.

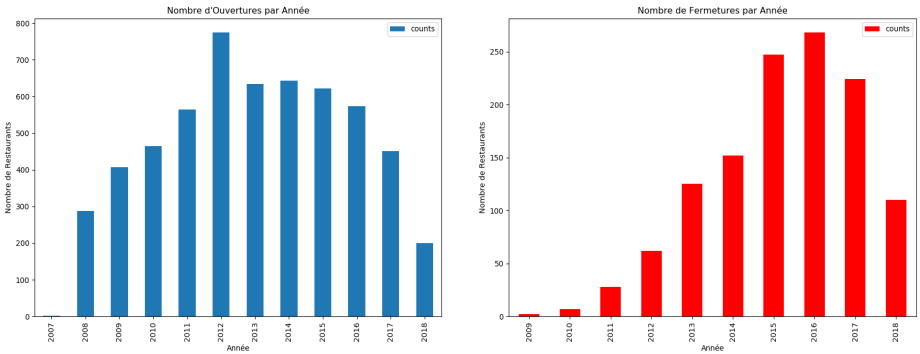

FIGURE 2.4 – Année d'ouverture vs de fermeture

Deuxièmement, nous regardons le nombre de restaurants qui sont ouverts par année, contre le nombre de restaurants qui sont fermés par année (figure 2.4). Nous remarquons qu'il y a l'apparence d'un cycle de "Boom and Bust" où pour

quelques années le nombre de restaurants qui ouvrent augmente progressivement jusqu'où on atteint un sommet, suivie par une série d'années où on a un déclin du nombre de restaurants qui ouvrent et un incrément dans le nombre de restaurants qui ferment leurs portes. Alors l'année du lancement du restaurant encode une certaine information sur l'état économique de la ville et peut emporter des informations sur la probabilité de fermeture de ce restaurant.

Alors nous ajoutons les attributs :

- **business_first_year** : L'année du lancement du restaurant
- **business_first_year_count** : Le nombre des autres restaurants qui sont lancés durant la même année du lancement du restaurant.

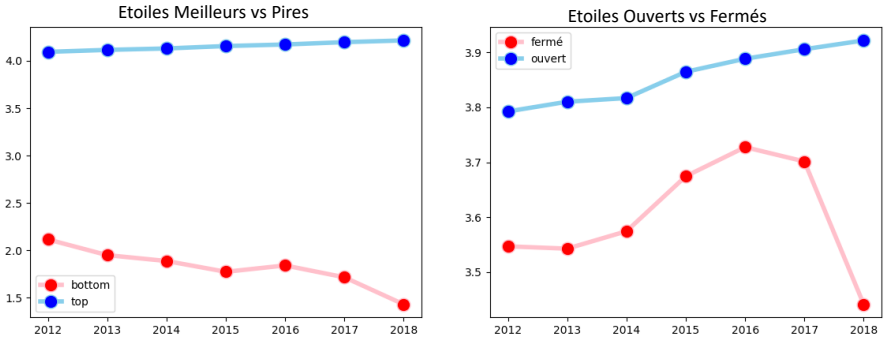

FIGURE 2.5 – Tendance de la qualité des restaurants

Ensuite, nous suivons la tendance de la qualité des restaurants au fil des années. Nous comparons les meilleurs restaurants ($>$4 étoiles) et les pires restaurants ($\leq$ 2 étoiles) aussi que les restaurants ouverts contre ceux qui sont fermés. Nous remarquons que la qualité du service des meilleurs restaurants et des restaurants ouverts est plus stable que celle des pires restaurants et des restaurants fermés, alors que la qualité de ces derniers a tendance à se détériorer avec les années (figure 2.5).

Alors, nous ajoutons les attributs :

- **std_stars** : L'écart type des étoiles des revues à travers les années.
- **trend_stars** : La moyenne des étoiles des revues de la dernière année moins la moyenne des étoiles des revues de l'année du lancement par restaurant.

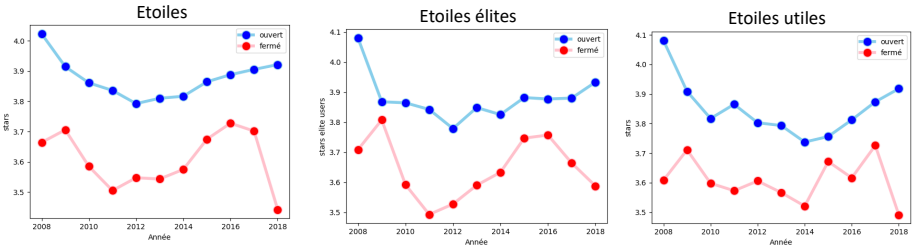

FIGURE 2.6 – Étoiles restaurants ouverts vs fermés

Puis, nous regardons la moyenne des étoiles à travers les années pour les restaurants ouverts contre les restaurants fermés. Nous évaluons les 3 types de revues, **Générales** (rédigées par tous les utilisateurs), **Élites** (rédigées par les utilisateurs élites) et **Utiles** (les revues qui ont reçu des réactions d'autres utilisateurs). Nous trouvons que pour les trois catégories, les évaluations des restaurants ouverts dépassent par une grande marge celles des restaurants fermés (figure 2.6). Alors la qualité du service et l'évaluation du restaurant par les clients jouent un grand rôle dans le statut du restaurant.

Alors, nous ajoutons les attributs suivants :

- **review_count** : Le nombre total de revues par restaurant.
- **good_reviews_count** : Le nombre de bonnes revues par restaurant (nombre d'étoiles >=3)
- **bad_reviews_count** : Le nombre de mauvaises revues par restaurant (nombre d'étoiles <3)
- **good_reviews_ratio** : La proportion des bonnes revues.
- **bad_reviews_ratio** : La proportion des mauvaises revues.
- **good_useful_review_count** : Le nombre de bonnes revues utiles que le restaurant a reçu.
- **bad_useful_review_count** : Le nombre de mauvaises revues utiles que le restaurant a reçu..
- **good_elite_review_count** : Le nombre de bonnes revues par les élites que le restaurant a reçu..
- **bad_elite_review_count** : Le nombre de mauvaises revues par les élites que le restaurant a reçu..

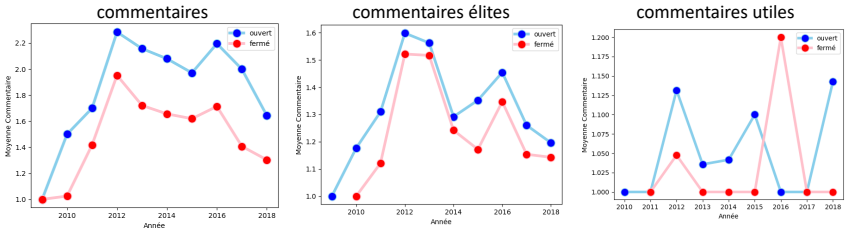

FIGURE 2.7 – Commentaire restaurants ouverts vs fermés

Nous remarquons la même tendance pour les commentaires, où les restaurants ouverts reçoivent plus de réactions du public que ceux qui sont fermés (2.7) :

- **tips_count** : Le nombre total de commentaires par restaurant.
- **tips_usefull_count** : Le nombre de commentaires utiles (qui ont reçu des réactions)
- **tips_elite_count** : Le nombre de commentaires par les utilisateurs élites.

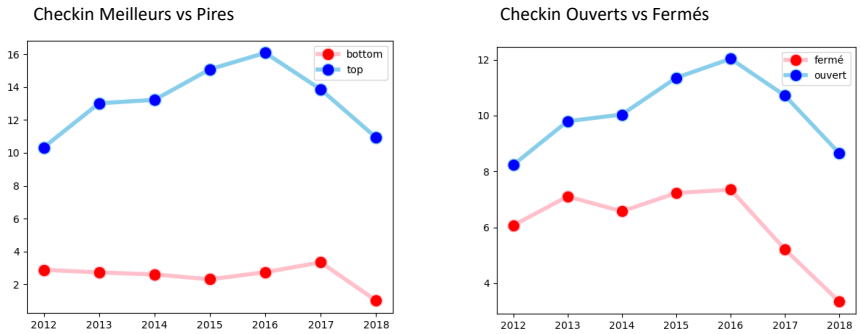

FIGURE 2.8 – Tendance des Checkins des restaurants

De même, nous analysons la tendance des visites des clients aux restaurants à travers les années, nous remarquons que la moyenne des clients qui signalent leur visite aux meilleurs restaurants et aux restaurants ouverts dépasse par une grande marge celles des pires restaurants et des restaurants fermés.

Alors nous ajoutons les attributs suivants :

- **checkin_count** : Le nombre de checkins par restaurant.
- **average_checkin** : La moyenne des checkins à travers les années.
- **std_checkin** : L'écart type des checkins à travers les années.

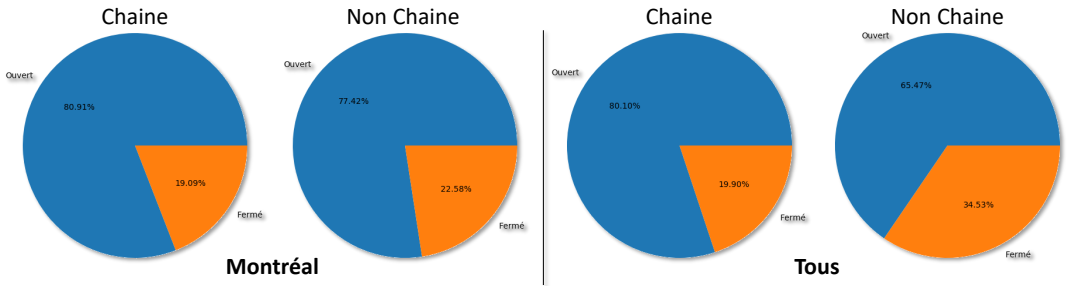

FIGURE 2.9 – Tendance de la qualité des restaurants

Selon (Cassel, 2018), les restaurants qui font partie d'une chaîne sont plus sécuritaire que ceux qui ne le sont pas. Alors, on a tracé le pourcentage de restaurants ouverts vs le pourcentage de restaurants fermés pour les restaurants chaîne et non-chaîne. Pour Montréal, on a remarqué qu'il n'y a pas vraiment une grande différence (figure 2.9|gauche). Alors que pour l'Amérique du Nord en total, on a remarqué que les restaurants qui font partie d'une chaîne sont 15% plus sécuritaires que ceux qui ne le sont pas (figure 2.9|droite).

Alors nous avons ajouté l'attribut :

- **is_chain** : Si le restaurant fait partie d'une chaîne de restaurants.

Nous avons aussi ajoutés les attributs suivants que nous avons évalués comme pertinents à l'apprentissage :

- **total_opening_hours** : Nombre total d'heures ouvert par semaine.
- **is_open_saturday** : Si le restaurant est ouvert les samedis.
- **is_open_sunady** : Si le restaurant est ouvert les dimanches.
- **is_open_monday** : Si le restaurant est ouvert les lundis.
- **RestaurantsTakeOut** : Si le restaurant offre le service à emporter.
- **RestaurantsGoodForGroups** : Si le restaurant est bon pour les groupes.
- **RestaurantsReservations** : Si le restaurant exige une réservation.
- **RestaurantsPriceRange2** : La marge de prix du restaurant (abordable,moyenne, chère).
- **OutdoorSeating** : Si le restaurant a une terrasse.
- **GoodForKids** : Si le restaurant est bien pour les enfants.
- **RestaurantsDelivery** : Si le restaurant offre le service de livraison.

2.2 Prétraitement des données

Pour le prétraitement des données, nous appliquons one-hot-encoding pour les catégories et les services et nous encodons les zones.
Pour le traitement des valeurs manquantes :

- Pour l'attribut **total_opening_hours**, nous remplaçons les enregistrements qui manquent cet attribut par la moyenne des autres restaurants dans la même zone qui partage les mêmes catégories du restaurant. Si aucun record n'est trouvé, nous remplaçons la valeur par la moyenne des restaurants dans la même zone. Sinon, nous la remplaçons par la moyenne de tous les restaurants.
- Pour **is_open_Saturday**, **is_open_Monday**, **is_open_sunady** nous utilisons la même méthode, mais avec le mode au lieu de la moyenne.

Dans la deuxième étape, nous éliminons l'attribut ID et les enregistrements doubles. Ensuite, nous normalisons les données. Enfin nous appliquons la technique d'Information Mutuelle pour la réduction de dimensionnalité et nous prenons les tops 150 attributs. Le tableau (2.1) montre les tops 10 attributs avec leurs scores.

Rank	Nom Colonne	Mutual Info Score
1	**good_reviews_ratio**	**0.055679**
2	**category_city_inter**	**0.030603**
3	**review_count**	**0.023946**
4	**bad_reviews_ratio**	**0.021523**
5	**checkin_count**	**0.021184**
6	**zone**	**0.020286**
7	**std_stars**	**0.018932**
8	**is_chain**	**0.018252**
9	**total_opening_hours**	**0.017653**
10	**business_first_year_count**	**0.017614**

TABLE 2.1 – Les tops 10 attributs avec leurs scores

2.3 Expérimentations et résultats

Pour tous nos expérimentations nous utilisons python avec les librairies pytorch, scikit-learn et mlxtend sur une machine avec un GPU NVIDIA GeForce GTX 1050 (12GB).

Nous balançons les données d'entraînement et de test de façon que le nombre d'enregistrements dans les 2 classes (ouvert et fermé) soit égal. Le tableau (2.2) décrit les détails des données utilisées pour l'apprentissage.

Données Total	33006
Données Entrainement	26404
Données Test (20%)	6602
Overts (Entrainement)	13191
Fermés (Entrainement)	13213
Overts (Test)	3312
Fermés (Test)	3290

TABLE 2.2 – Données d'apprentissage pour la prédiction de fermeture

Nous entraînons 7 types de modèles sur les données. Nous utilisons les modèles classiques suivants : Bagging, Boosting, Random Forest, Arbre de Décision, Régression Logistique et Naive Bayes.
Nous construisons aussi un réseau de neurones à 2 couches avec pytorch. Les couches sont de taille 64 avec une activation RELU et un Dropout de 20% sur toutes les couches. Nous entraînons pour 500 époques avec une patience de 10. Nous utilisons la fonction de perte Binary Cross Entropy avec l'optimisateur Adam et un taux d'apprentissage de 0.01.

Le tableau (2.3) montre les résultats de l'apprentissage. Nous évaluons les modèles sur l'accuracy et la F-mesure. Nous rapportons les meilleurs résultats.

Modèles	Accuracy	F-measure	Precision	Recall	AUC	FPR	TPR
Bagging	**0.839**	**0.841**	0.834	0.848	0.839	0.17	0.848
MLP	0.836	0.836	0.84	0.833	0.836	0.161	0.833
AdaBoost	0.834	0.832	0.842	0.823	0.834	0.156	0.823
Radom Forest	0.827	0.826	0.832	0.82	0.827	0.166	0.82
Decision Trees	0.763	0.762	0.766	0.758	0.763	0.233	0.758
Logistic Regression	0.733	0.74	0.724	0.756	0.733	0.29	0.756
Naive Bayes	0.526	0.671	0.515	0.961	0.525	0.912	0.961

TABLE 2.3 – Les résultats de la prédiction de fermeture

Les modèles qui ont performé le mieux sont surlignés en vert. Nous achevons la meilleure exactitude de 84% avec le modèle Bagging.

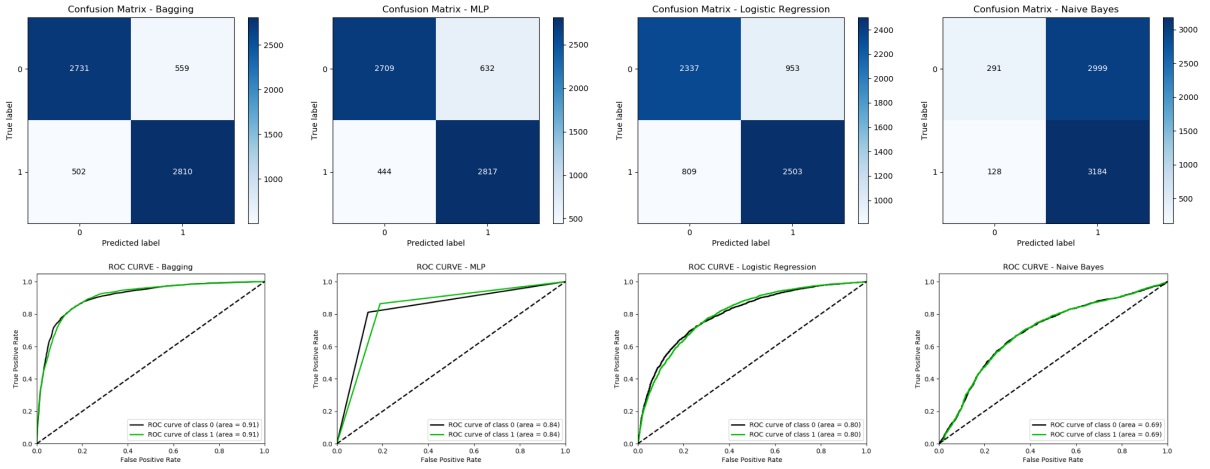

FIGURE 2.10 – Matrices de confusion et courbes ROC de 4 modèles

2.4 Extraction des Patrons

Pour l'extraction des patrons, nous traitons les attributs de chaque restaurant comme un panier. Nous utilisons seulement le dataset de Montréal. Pour les attributs nominaux et ordinaux, nous concaténons le nom de l'attribut avec la valeur (exemple is_open_saturday1). Pour les attributs continus, nous les discrétisons en utilisant K-binning avec un nombre de paniers égal à 3. De sorte que 1 :Petit, 2 :Moyen et 3 :Grand, (exemple review_count2 $\longrightarrow$ nombre de revues moyen). Nous appliquons FP-Growth (Han *et al.*, 2000) pour l'extraction des patrons. Notez que Apriori donnait une erreur de mémoire sur notre machine.

Pour bien pouvoir interpréter les patrons, nous focalisons sur les zones et nous filtrons les règles avec "is_closed0" i.e. "Restaurant Ouvert" comme conséquent pour extraire les règles qui affectent le succès d'un restaurant. Note que proprement, nous devrons filtrer "is_closed1" i.e. "Restaurant Fermé" comme conséquent. Mais, comme les restaurants fermés sont rares en comparaison avec les restaurants ouverts ($\sim$ 20%), il est très difficile qu'ils apparaissent dans les règles et nous devrons diminuer beaucoup le minimum support pour les retrouver ce qui rend la plupart des règles fouillées non représentatives.

Nous présentons par la suite l'analyse de quelques zones de Montréal selon les règles extraites par FP-Growth. Note : nous enlevons les attributs continus, comme par exemple le nombre d'étoiles, car par leur présence ils vont dominer les règles et nous voulons extraire des règles associées aux services offerts par les restaurants.

antecedents	consequents	support	confidence	lift
GoodForKids0, is_open_saturday1, is_open_monday1, **RestaurantsGoodForGroups1**, is_open_sunday1	is_closed0	0.111	0.810	1.301
GoodForKids0, RestaurantsDelivery0, is_open_saturday1, is_open_monday1, RestaurantsGoodForGroups1, is_open_sunday1	is_closed0	0.104	0.800	1.286
GoodForKids0, is_chain0, is_open_saturday1, is_open_monday1, RestaurantsGoodForGroups1	is_closed0	0.101	0.816	1.311
is_open_saturday1, **RestaurantsGoodForGroups1**, GoodForKids0, RestaurantsReservations1	is_closed0	0.121	0.804	1.293
is_open_saturday1, is_open_monday1, **GoodForKids0**, **RestaurantsReservations1**	is_closed0	0.107	0.846	1.360
is_open_saturday1, is_open_sunday1, **GoodForKids0**, **RestaurantsReservations1**	is_closed0	0.107	0.825	1.326
is_open_saturday1, **RestaurantsPriceRange22**, GoodForKids0	is_closed0	0.111	0.829	1.333
is_open_saturday1, RestaurantsGoodForGroups1, **RestaurantsPriceRange22**, GoodForKids0	is_closed0	0.101	0.838	1.347
OutdoorSeating0, RestaurantsDelivery0, **RestaurantsGoodForGroups1**, RestaurantsPriceRange22, RestaurantsTakeOut1	is_closed0	0.104	0.800	1.286
GoodForKids1, RestaurantsDelivery0, is_open_saturday1, RestaurantsGoodForGroups1, RestaurantsPriceRange22, RestaurantsTakeOut1	is_closed0	0.104	0.800	1.286
is_open_saturday1, RestaurantsDelivery0, stars4.0, GoodForKids1	is_closed0	0.101	0.816	1.311
is_open_saturday1, RestaurantsPriceRange22, stars4.0	is_closed0	0.130	0.816	1.312

FIGURE 2.11 – Règles d'association Sainte Catherine - minsup 0.1

Pour la zone Sainte-Catherine (centre-ville), nous remarquons que les règles avec GoodForKids0, RestaurantGoodForGroups1, RestaurantPriceRange22 et RestaurantReservation1 sont fréquentes. Cela signifie que les restaurants avec un certain succès dans cette zone ne sont pas adaptés aux enfants ou les familles, ils sont bons pour les groupes et sont plus ou moins coûteux. Ceci s'aligne avec la nature de la zone au centre-ville et aux types de restaurants que nous trouvons là-bas.

antecedents	consequents	support	confidence	lift
RestaurantsReservations0, is_open_saturday1	is_closed0	0.327	0.842	1.232
RestaurantsReservations0, **RestaurantsTakeOut1**	is_closed0	0.327	0.821	1.200
RestaurantsTakeOut1, is_open_saturday1	is_closed0	0.439	0.811	1.187
RestaurantsTakeOut1, is_chain0, is_open_saturday1	is_closed0	0.372	0.820	1.200
is_open_sunday1, **RestaurantsTakeOut1**, is_open_saturday1	is_closed0	0.372	0.839	1.227
is_open_sunday1, **RestaurantsTakeOut1**, is_chain0	is_closed0	0.347	0.800	1.170
is_open_sunday1, **RestaurantsTakeOut1**, is_chain0, is_open_saturday1	is_closed0	0.311	0.847	1.239
RestaurantsTakeOut1, is_open_monday1, is_open_saturday1	is_closed0	0.327	0.810	1.185
RestaurantsTakeOut1, RestaurantsDelivery0	is_closed0	0.367	0.800	1.170

FIGURE 2.12 – Règles d'association Plateau - minsup 0.3

Pour la zone Plateau, nous remarquons que les restaurants dans cette zone ne demandent pas une réservation et offrent le service à emporter. Cela s'aligne avec la nature de la zone qui est "Trendy" et populaires avec les jeunes.

antecedents	consequents	support	confidence	lift
GoodForKids1, **RestaurantsDelivery0**	is_closed0	0.468	0.813	1.113
GoodForKids1, is_open_monday1, **RestaurantsDelivery0**	is_closed0	0.459	0.810	1.109
GoodForKids1, **RestaurantsDelivery0**, **RestaurantsTakeOut1**	is_closed0	0.432	0.814	1.115
GoodForKids1, **RestaurantsDelivery0, OutdoorSeating0**	is_closed0	0.423	0.810	1.110
GoodForKids1, RestaurantsDelivery0, **RestaurantsGoodForGroups1**	is_closed0	0.396	0.800	1.096
GoodForKids1, is_open_monday1, **RestaurantsDelivery0**, **RestaurantsTakeOut1**	is_closed0	0.423	0.810	1.110
GoodForKids1, is_open_monday1, **RestaurantsDelivery0, OutdoorSeating0**	is_closed0	0.414	0.807	1.106
GoodForKids1, RestaurantsDelivery0, OutdoorSeating0, RestaurantsTakeOut1	is_closed0	0.396	0.815	1.117
GoodForKids1, OutdoorSeating0, RestaurantsDelivery0, RestaurantsTakeOut1, is_open_monday1	is_closed0	0.387	0.811	1.112
GoodForKids1, is_open_monday1, RestaurantsDelivery0, RestaurantsGoodForGroups1	is_closed0	0.396	0.800	1.096
GoodForKids1, RestaurantsGoodForGroups1, OutdoorSeating0, RestaurantsDelivery0, RestaurantsTakeOut1	is_closed0	0.333	0.804	1.102
GoodForKids1, RestaurantsGoodForGroups1, OutdoorSeating0, RestaurantsDelivery0, RestaurantsTakeOut1, is_open_monday1	is_closed0	0.333	0.804	1.102
GoodForKids1, is_chain0, RestaurantsTakeOut1	is_closed0	0.432	0.800	1.096
GoodForKids1, RestaurantsDelivery0, is_chain0	is_closed0	0.405	0.818	1.121

FIGURE 2.13 – Règles d'association China Town - minsup 0.3

Pour China Town, nous constatons que les règles dominantes sont GoodForkids1, RestaurantDelivery0, OutdoorSeeting0 et RestaurantTakeOut1. Cela s'aligne avec la nature touristique de la zone où les restaurants doivent être ciblés vers les familles pour qu'ils réussissent.

Dans le chapitre suivant, nous détaillons le processus de la deuxième tache concernant la recommandation des restaurants aux utilisateurs par préférence.

Chapitre 3

Recommandation

Dans ce chapitre nous détaillons la démarche de la deuxième tache qui se concerne avec la prédiction de la fermeture des restaurants. Nous commençons par modéliser le problème en forme de graphe hétérogène et en introduisant les types d'architectures que nous utilisons pour apprendre sur les graphes. Ensuite, nous décrivons la technique d'augmentation des données en utilisant les associations et enfin nous présentons les résultats de l'apprentissage.

3.1 Modélisation du problème

Nous modélisons la relation entre les utilisateurs et les restaurants sous forme d'un graphe hétérogène bipartie (figure3.1). Les noeuds sont de 2 types, utilisateurs et restaurants et les arêtes sont les revues rédigées par les utilisateurs aux restaurants. Chaque nombre d'étoiles par revue est représenté par un type d'arête. Nous traitons le problème de recommandation des restaurants comme un problème de classification semi-supervisé des arêtes sur un graphe hétérogène. Où le modèle s'entraîne à étiqueter un sous-ensemble des arêtes et prédit les étiquettes d'un autre sous-ensemble. Ici les étiquettes sont les types des arêtes qui sont le nombre d'étoiles par revue.

Par la suite, nous entraînons un modèle Deep-learning pour faire l'apprentissage end-to-end sur le graphe hétérogène. Mais avant de passer aux modèles, nous discutons dans la section suivante la technique d'augmentation des données en utilisant FP-Growth.

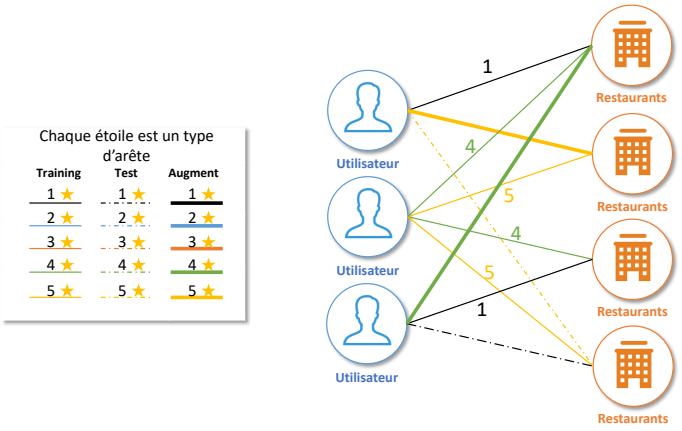

FIGURE 3.1 – Modèle du graphe hétérogène

3.2 Augmentation par FP-Growth

Un problème que nous rencontrons durant l'apprentissage sur les graphes "sparces" est la rareté des connexions pour chacun des noeuds. Alors, nous n'avons pas beaucoup d'exemples d'entraînement par utilisateur. En plus, ajouter de nouveaux noeuds (utilisateurs ou restaurants) va augmenter la complexité de l'apprentissage sans nécessairement ajouter de nouveaux exemples aux noeuds originaux. Une façon de combler cela est en augmentant le nombre des arêtes des noeuds originaux. Mais nous ne pouvons pas les ajouter d'une façon aléatoire, alors que les arêtes ajoutées doivent être consistantes avec les tendances sous-jacentes dans les données. Pour cela, nous extrayons des patrons des données que nous utilisons comme relations potentielles pour augmenter le nombre d'arêtes dans le graphe afin d'améliorer l'apprentissage.

Premièrement, nous filtrons les données d'entraînement pour chaque nombre d'étoiles. Nous considérons l'ensemble des restaurants évalués par chaque utilisateur comme notre panier. Nous appliquons FP-Growth sur le dataset filtré et nous prenons les ensembles fréquents de taille 2. Pour chaque utilisateur, nous vérifions, s'il est déjà connecté à un des 2 restaurants, nous le connectons à l'autre avec le même nombre d'étoiles filtré au début du processus. Nous répétons le processus pour les 5 types d'étoiles.

Note : Il aurait été beaucoup plus avantageux d'appliquer FP-Growth sur tout le dataset avec tous les nombres d'étoiles et de filtrer les règles par conséquent, où le conséquent est le nombre d'étoiles. Et pour chaque nombre d'étoiles, nous faisons les nouvelles connexions selon les restaurants dans les antécédents. Cela garantit une association entre les restaurants et le nombre d'étoiles et pas une simple fréquence d'apparition. Nous laissons cela pour un travail futur.

3.3 Les Modèles GCN

Pour faire l'apprentissage sur les graphes, nous utilisons les techniques de Deep Learning sur les graphes. Ces techniques utilisent le processus de passage et d'agrégation de message . En littérature, on réfère à ces modèles comme GNN (Graph Neural Networks). Une des variations les plus connues des GNN est le GCN (Graph Convolutional Networks) développée par (Kipf et Welling, 2017). C'est le type d'architecture que nous allons utiliser comme base pour notre modèle. Dans chacune des couches du GCN, chaque noeud collecte tous les messages (attributs) de son voisinage direct et les agrège selon une fonction d'agrégation comme : Max, Somme, Moyenne. Les messages sont accumulés d'une couche en couche et cela permet l'exploration de plusieurs niveaux de proximité (figure 3.2).

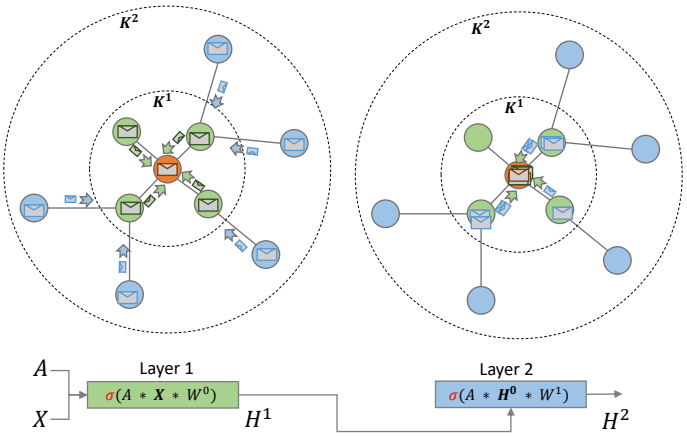

FIGURE 3.2 – Passage et agrégation des message par couche

Pour la première couche on fait passer la matrice d'adjacence et la matrice d'attribut comme entrée. Si le graphe est non attribué nous utilisons one-hot-encoding comme attributs. La règle la plus simple est l'agrégation par Somme. Pour agréger par Somme, il suffit de multiplier la matrice d'adjacence par la matrice d'attribut. Une matrice d'identité est additionnée à la matrice d'adjacence pour permettre l'accumulation des messages. Le tout est multiplié par une matrice de poids et passé par une fonction d'activation : ReLU, Tanh ou Sigmoïde. La sortie de chaque couche sert comme une entrée pour la couche suivante. La sortie de la dernière couche est passée par une Sigmoïde ou Softmax pour produire une probabilité d'appartenance à une classe et la propagation de l'erreur en arrière est utilisée pour ajuster les poids selon l'erreur calculée entre la prédiction et le groundtruth.

3.4 Architecture rGCN

Pour l'apprentissage sur le graphe hétérogène nous utilisons rGCN(Schlichtkrull *et al.*, 2018) une variation du GCN qui est adaptée pour les graphes hétérogènes et les knowledge graphes. La différence la plus importante entre les GCN et les rGCN est que ces derniers introduisent une matrice de poids pour chaque type d'arête. Alors le modèle est entrain d'apprendre à séparer entre les différents types d'entités et de relations dans le graphe. Chaque couche applique la fonction suivante :

$$H^l = \sigma(\sum_{r \in R} \sum_{j \in N_i 0} W_r h_j^{l-1}) \tag{3.1}$$

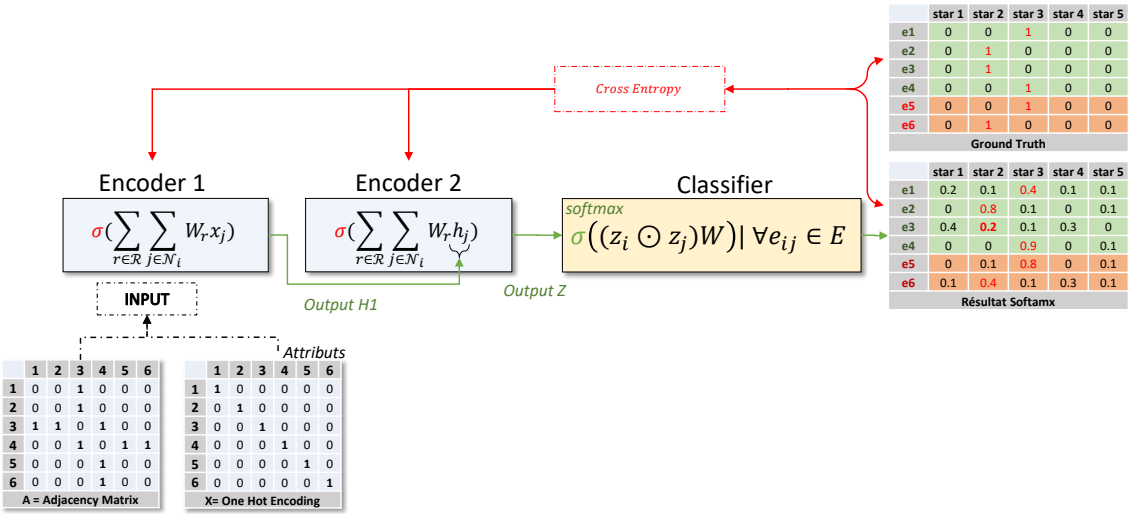

FIGURE 3.3 – Architecture rGCN

Pour classifier les arêtes (le nombre d'étoiles des revues), dans la dernière couche, nous échantillons toutes les arêtes marquées pour l'entraînement. Puis, nous multiplions le embedding des deux noeuds de l'arête élément par élément (produit de hadamard) pour générer un embedding pour toute l'arête. Les embeddings des arêtes sont multipliés par une matrice de poids à apprendre. Enfin, le tout est passé par une Softmax pour générer les probabilités d'appartenance aux différentes classes des arêtes. Le schéma (3.3) détaille les différentes composantes de l'architecture.

3.5 Expérimentations et résultats

Pour toutes nos expérimentations nous utilisons python avec les librairies pytorch, Networkx et DGL(Deep Graph Library) (Wang *et al.*, 2019) sur une machine avec un GPU NVIDIA GeForce GTX 1050 (12GB).

Nous utilisons les données de Montréal, nous échantillons 1000 utilisateurs et tous les restaurants. Nous divisons les données pour un apprentissage semi-supervisé. Nous prenons les revues avant 2017 pour l'entraînement et toutes les revues de 2017 et delà pour le test. Le tableau (3.1) liste les détails des données.

Dataset	Montréal
Utilisateurs	1000
Restaurants	5622
Revue Train (<2017)	8548
Revue Test (>=2017)	1923

TABLE 3.1 – Détails des données de la recommandation

Le tableau (3.2) détaille la distribution des classes dans les données d'entraînement et de test. On remarque que les classes ne sont pas balancées. L'augmentation des données va aider à mitiger ce problème. Pour l'augmentation des données, nous testons sur 3 minimums support. Le nombre d'arêtes ajoutées pour chaque minsup est détaillé dans le tableau (3.3).

Entraînement		**Test**	
Étoiles	**Arêtes**	**Étoiles**	**Arêtes**
1	235	**1**	46
2	622	**2**	119
3	1752	**3**	350
4	3646	**4**	753
5	2293	**5**	655

TABLE 3.2 – Distribution des classes

minsup 0.01		**minsup 0.008**		**minsup 0.005**	
Étoiles 1	996	Étoiles 1	989	Étoiles 1	2972
Étoiles 2	0	Étoiles 2	2716	Étoiles 2	2867
Étoiles 3	911	Étoiles 3	0	Étoiles 3	1983
Étoiles 4	1757	Étoiles 4	1823	Étoiles 4	2868
Étoiles 5	4242	Étoiles 5	954	Étoiles 4	954

TABLE 3.3 – Le nombre d'arêtes ajoutées par augmentation

Pour l'entraînement de nos modèles GCN et rGCN, nous utilisons 2 couches avec une taille de 16 neurones pour chacune. Nous utilisons l'activation RELU sur toutes les couches internes avec 0% dropout. Nous entraînons pour 300 époques avec une patience de 30. Nous utilisons la fonction de perte Cross Entropy avec l'optimisateur Adam et un taux d'apprentissage de 0.001. Nous évaluons la performance des modèles avec RMSE (Root Mean Squared Error) qui calcule l'erreur entre les étoiles originales des revues et les étoiles prédites par le modèle sur les arêtes tests..

Le tableau (3.4) détaille les résultats des modèles. Les modèles rGCN surperforment les modèles GCN vanilla. Alors, adaptés le modèle aux différents types d'arêtes améliore la performance. Aussi, on remarque que le modèle rGCN-Augmenté avec un minimum support de 0.01 donnait le meilleur résultat. Alors, on conclut qu'augmenter les arêtes du graphe avec des relations de fréquences améliore la performance du modèle.

Modèles	**RMSE**
rGCN	1.457
rGCN-Aug (minsup=0.01)	**1.277**
rGCN-Aug (minsup=0.008)	1.707
rGCN-Aug (minsup=0.005)	1.439
GCN	1.711
GCN (minsup=0.008)	1.473

TABLE 3.4 – Résultat de Recommandation

Le tableau (3.4) démontre un échantillon de la prédiction du modèle rGCN-Aug(minsup=0.01). On trouve que pour la plupart des restaurants, le modèle donnait une prédiction proche de l'évaluation réelle de l'utilisateur. On remarque aussi que le modèle a une tendance à surévaluer les restaurants. De plus, pour un exemple (surligné en rouge) il y a un grand décalage entre la prédiction et la réalité. Mais aussi le modèle a bien évalué le restaurant Ucan pour l'utilisateur Deb où le modèle a bien prédit que l'utilisateur ne va pas aimer le restaurant, ce qui est consistant avec classification générale du restaurant de 2 étoiles.

Prediction	Ground Truth	Utilistateur	Restaurant	Etoiles
4	4	Melissa	St-Viateur Bagel & Café	4
5	4	Emilia	Belon	4
4	4	Emilia	Deville Dinerbar	4
4	4	Emilia	Olive & Gourmando	4.5
4	5	Gwen	Milos Restaurant	4.5
4	5	David	Big In Japan	3.5
3	3	David	Taverne Gaspar	3.5
4	4	David	BEVO Bar + Pizzeria	4
4	3	David	La Cage Aux Sports	3
4	5	David	Tommy	4
4	5	David	Maison Christian Faure	4.5
4	5	David	Pastaga	4.5
4	5	David	St-Viateur Bagel & Café	4
1	**2**	**Emmy**	**Ucan**	**2**
5	2	Deb	The Keg Steakhouse + Bar	4
4	5	Deb	Chez Suzette	4
3	4	Eileen	La Société Montréal	4
4	5	Eileen	L'Avenue	4.5
4	5	Eileen	Eggspectation	3.5

FIGURE 3.4 – Exemple de prédiction

Pour la visualisation, nous appliquons TSNE-2D sur les embeddings des arêtes du test et nous colorions les points selon les classes (nombre d'étoiles). On remarque que les embeddings appartenant aux mêmes classes ne sont pas vraiment en cluster. Alors il y a une grande marge d'amélioration du modèle. Mais comme résultats préliminaires le modèle a bien performé.

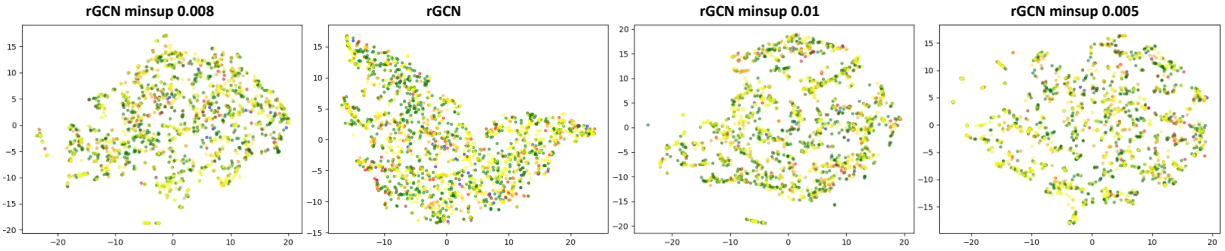

FIGURE 3.5 – Exemple de prédiction

3.6 Conclusion

Dans ce travail nous avons étudié deux taches reliées aux restaurants en utilisant les données offertes par Yelp. Dans la première tache, nous avons prédit la fermeture des restaurants. Dans ce but, nous avons analysé les restaurants de Montréal et nous avons par la suite construit nos attributs à partir des connaissances acquises durant l'analyse. Nous sommes arrivés à une exactitude de 84%. Ce résultat est fiable vu que le statut des restaurants dépend de beaucoup de facteurs (économiques, sociales, modes...) qui sont difficiles à extraire des données. En plus, nous avons fouillé des partons reliés aux tendances de succès des restaurants dans les différentes zones de Montréal. Comme travail futur, nous considérons étudier le problème d'un côté dynamique pour bien capter l'évolution de la performance du restaurant à travers le temps.

Pour la deuxième tache, nous avons développé un modèle qui prédit l'évaluation que donnera un utilisateur à un restaurant. Pour ce but, nous avons modélisé les relations utilisateur/restaurants sous forme d'un graphe hétérogène et nous avons entraîné deux types de modèles d'apprentissage profond sur les graphes, GCN et rGCN, ce dernier est adapté pour les graphes hétérogènes. Nous avons en plus augmenté les connexions des utilisateurs en utilisant FP-Growth pour en extraire les ensembles fréquents de restaurants. Nous sommes arrivés à une erreur de 1.277 et nous avons trouvé que les modèles rGCN augmentés surperformaient les modèles réguliers. Comme travail futur, nous cherchons à améliorer la stratégie d'augmentation en utilisant les règles d'association au lieu des relations de fréquence et à supplémenter les résultats par des d'expérimentations additionnelles sur plusieurs datasets.

Bibliographie

Cassel, D. (2018). can yelp data predict restaurant closures ? Récupéré de https://thenewstack.io/can-yelp-data-predict-restaurant-closures/ 11

Fayyad, U. M., Piatetsky-Shapiro, G. et Smyth, P. (1996). Knowledge discovery and data mining : Towards a unifying framework. Dans E. Simoudis, J. Han, et U. M. Fayyad (dir.). *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*, 82–88. AAAI Press. Récupéré de http://www.aaai.org/Library/KDD/1996/kdd96-014.php 4

Han, J., Pei, J. et Yin, Y. (2000). Mining frequent patterns without candidate generation. Dans W. Chen, J. F. Naughton, et P. A. Bernstein (dir.). *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA*, 1–12. ACM. http://dx.doi.org/10.1145/342009.335372 14

Kipf, T. N. et Welling, M. (2017). Semi-supervised classification with graph convolutional networks. Dans *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. Récupéré de https://openreview.net/forum?id=SJU4ayYgl 19

Schlichtkrull, M. S., Kipf, T. N., Bloem, P., van den Berg, R., Titov, I. et Welling, M. (2018). modeling relational data with graph convolutional networks. Dans A. Gangemi, R. Navigli, M. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, et M. Alam (dir.). *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 de *Lecture Notes in Computer Science*, 593–607. Springer. http://dx.doi.org/10.1007/978-3-319-93417-4_38. Récupéré de https://doi.org/10.1007/978-3-319-93417-4_38 20

Wang, M., Yu, L., Zheng, D., Gan, Q., Gai, Y., Ye, Z., Li, M., Zhou, J., Huang, Q., Ma, C., Huang, Z., Guo, Q., Zhang, H., Lin, H., Zhao, J., Li, J., Smola, A. J. et Zhang, Z. (2019). Deep graph library : Towards efficient and scalable deep learning on graphs. *CoRR*, *abs/1909.01315*. Récupéré de http://arxiv.org/abs/1909.01315 21

wikipedia (2020). Récupéré de https://en.wikipedia.org/wiki/Yelp 2

yelp (2020). yelp dataset. Récupéré de https://www.yelp.com/dataset 3