RAPPORT DE STAGE

Méthodes d'acquisition des données à l'appui des organismes nationaux de statistique des Caraïbes

Master 1 Ingénierie de la décision et Big Data

Sous la direction de Marion Jean-Marie

BELLAY Guillaume

Stage effectué à :

Statistique Canada

100 Promenade du Pré Tunney,

Ottawa, Canada, K1A 0T6

Superviseur : Paul Holness

Faculté des Sciences

Institut de Mathématiques Appliquées

Année universitaire : 2018-2019

UCO ANGERS
UNIVERSITÉ CATHOLIQUE DE L'OUEST

3 place André Leroy | BP 10808- 49008 Angers cedex 01

CHARTE DE NON PLAGIAT

Protection de la propriété intellectuelle

Tout travail universitaire doit être réalisé dans le respect intégral de la propriété intellectuelle d'autrui. Pour tout travail personnel, ou collectif, pour lequel le candidat est autorisé à utiliser des documents (textes, images, musiques, films etc.), celui-ci devra très précisément signaler le crédit (référence complète du texte cité, de l'image ou de la bande-son utilisés, sources internet incluses) à la fois dans le corps du texte et dans la bibliographie. Il est précisé que l'UCO dispose d'un logiciel anti-plagiat dans lms.uco.fr, aussi est-il demandé à tout étudiant de remettre à ses enseignants un double de ses travaux lourds sur support informatique.

Cf. « Prévention des fraudes à l'attention des étudiants »

Je soussigné, Guillaume Bellay, étudiant en Master 1 Ingénierie de la décision et Big Data m'engage à respecter cette charte.

Fait à Ottawa, le 6 Mai 2019.

Signature :

Guillaume Bellay

Certaines données ont volontairement été modifiées par souci de confidentialité.

Remerciements

Je tiens à remercier en premier lieu Paul Holness pour sa disponibilité et ses conseils tout au long de ce stage. Merci également aux membres de l'équipe PRASC pour leur sympathie et leur aide.

Je remercie mon tuteur académique M. Marion pour sa pédagogie et son suivi ainsi que l'ensemble des professeurs de l'Institut de Mathématiques pour leur enseignement de qualité.

Ce stage fut avant tout un travail de collaboration et d'échanges au laboratoire d'innovation et je tiens à exprimer ma gratitude aux personnes suivantes pour leur contribution : Diego Ripley, Russell Gill, Zachary Nick, Raphaël Duteau, Rafael Sobrino, Gabriel Valachi, Brindusa Valachi et Kenneth Chu.

Table des matières

Introduction V
1. Acquisition des données VI
1.1 Google Places API VI
1.1.1 Fonctionnement du programme VI
1.1.2 Résultats VII
1.2 OpenStreetmap VIII
1.2.1 Téléchargement des données VIII
1.2.2 Résultats VIII
2. Préparation des données IX
2.1 Google Places API IX
2.1.1 Uniformisation des données IX
2.1.2 Couplage de données déterministe IX
2.1.3 Couplage de données probabiliste XI
2.2 OpenStreetmap XX
3. Création de la base de données géospatiale XXVI
3.1 Couplage de données probabiliste XXVI
3.2 Visualisation des données et calcul des contributions XXXI
Conclusion XXXIV
Bibliographie XXXV
Annexes XXXVII

Introduction

Le projet régional d'avancement de la statistique dans les Caraïbes (PRASC), financé par le Gouvernement du Canada, est un programme visant à renforcer les capacités et l'infrastructure statistiques dans 14 organismes nationaux de la statistique (ONS) de la région. Dans le cadre du PRASC, quelques projets d'exploration sont menés afin de trouver des méthodes innovantes pour mieux mesurer certains concepts clés pour la comptabilité nationale. Actuellement ces concepts sont peu souvent mesurés dans certains pays des Caraïbes, faute de ressources pour faire de la collecte des données selon les méthodes traditionnelles d'enquête.

L'un de ses objectifs est de déterminer la valeur des investissements non financiers pour 14 pays membres de la Communauté caribéenne (CARICOM) : Antigua-et-Barbuda, la Barbade, Belize, la Dominique, Grenade, la Guyane, la Jamaïque, Montserrat, Saint-Vincent-et-les-Grenadines, Saint-Kitts-et-Nevis, Sainte Lucie, Suriname, les Bahamas, Trinité-et-Tobago. Cette mesure est possible en évaluant l'évolution de la valeur du parc immobilier de chaque pays. Dans un premier temps des registres des bâtiments sont construits à partir de méthodes d'acquisition des données web. Ces bâtiments peuvent être des entreprises, des lieux d'intérêt. Des informations sur ces entreprises et ces lieux d'intérêt telles le nom, l'adresse, les coordonnées géographiques, les activités sont recueillies puis stockées dans une base de données géospatiale. Celle-ci fournit également des cartes montrant la répartition des entreprises et des lieux d'intérêt pour chaque pays. L'application est développée en Python dans Jupyter Notebook. Jupyter Notebook est open source. Cette application peut être donc facilement transmise aux 14 pays cités ci-dessus. Les données sur ces entreprises et lieux d'intérêt proviennent de deux sources : Google Places API et OpenStreetmap.

Google détient une quantité importante de données mais les conditions d'utilisation de leurs services peuvent changer à tout moment. Google peut décider des informations à afficher ou masquer et désormais l'accès à leurs données n'est plus entièrement gratuit. OpenStreetmap est une carte du monde publiée gratuitement et mise à jour régulièrement par sa communauté de plusieurs centaines de milliers de membres. Pour Google l'acquisition des données se fait via une interface de programmation tandis que pour Openstreetmap il suffit de se rendre sur un site de téléchargement.

Dans un premier temps mon travail a consisté à collecter les données de Google Places API et d'Openstreetmap. La qualité des données étant primordiale, ces données ont donc ensuite été nettoyées. Enfin les données de Google Places API et d'Openstreetmap ont été rassemblées afin de créer la base de données géospatiale.

Nous verrons que l'apport des données d'OpenStreetmap a un réel intérêt.

1. Acquisition des données

1.1 Google Places API

1.1.1 Fonctionnement du programme

Les données de Google ont été récupérées grâce à une interface de programmation d'application (API). Une API peut être définie comme un protocole de communication qui permet d'accéder à des services ou des données. L'API reçoit des requêtes de l'utilisateur et renvoie des réponses d'internet. Google est un grand fournisseur d'API. Google Places API retourne des informations sur des lieux, par exemple des entreprises et des lieux d'intérêt. Une requête est construite avec certains paramètres spécifiques.

Dans notre cas nous avons utilisé deux types de requêtes : Google Places Nearby Search et Google Places Details.

Ces deux requêtes requièrent obligatoirement les paramètres suivants : la latitude, la longitude, le rayon de la recherche et une clé API. La clé API permet d'identifier de manière unique un utilisateur qui souhaite envoyer une requête. D'autres paramètres sont optionnels et permettent d'affiner la recherche. Nous recevons comme réponses une liste d'entreprises et de lieux d'intérêts avec leurs caractéristiques. Une requête ne peut toutefois retourner plus de 60 résultats. Par conséquent la zone d'intérêt doit être segmentée en petites régions afin de retrouver un maximum d'entreprises et de lieux d'intérêts.

Un script python permet cette division de la zone géographique du pays en petites régions afin d'éviter de dépasser les limites de l'API. Le script python génère une grille basée sur les coordonnées du pays (latitude maximale, latitude minimale, longitude maximale et longitude minimale) et produit un ensemble de coordonnées (latitude et longitude). Ces coordonnées et le rayon de recherche, stockés dans un fichier texte, sont utilisés comme paramètres dans nos requêtes. Les rayons de recherche ont été choisis arbitrairement. Ils varient selon le nombre d'entreprises de la région d'intérêt. Un rayon de recherche est compris entre 200 et 4000 mètres.

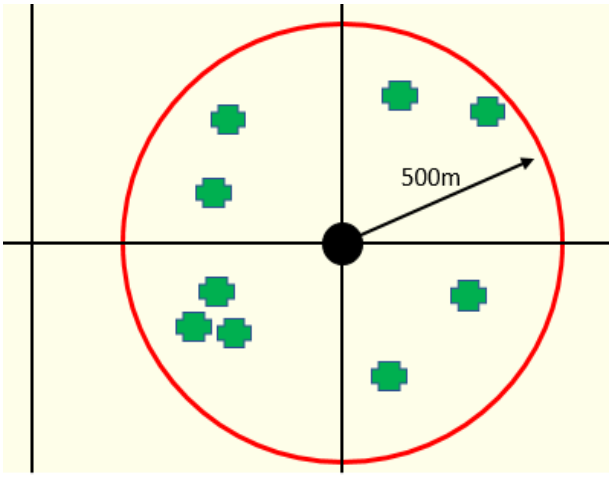

Figure 1 : Recherche des entreprises

Google Places API Nearby Search permet de rechercher des lieux dans une zone spécifiée. Cette requête ne renvoie que des informations basiques sur les entreprises et lieux d'intérêt comme par exemple les coordonnées de l'entreprise (latitude et longitude).
L'attribut 'Place_id' obtenu grâce à cette première requête est ensuite utilisé comme paramètre pour notre seconde requête Google Places Details. Celle-ci nous permet d'avoir des informations plus détaillées sur les entreprises et lieux d'intérêts.

25 attributs ont finalement été retenus : 'Nom', 'Latitude', 'Longitude', 'Adresse', 'Code postal', 'Numéro de téléphone', 'Site internet', 'Activité', 'Place_id', 'Code composé', 'Code global', 'Evaluation', 'Niveau de prix', 'Etage', 'Numéro de rue', 'Route', 'Localité', 'Niveau Administratif', 'Pays', 'Adresse formatée', 'Coord_lat', 'Coord_lng', 'Search_radius', 'Coord_id', 'Fermé définitivement'.

Le fonctionnement de ce programme peut se représenter de la façon suivante :

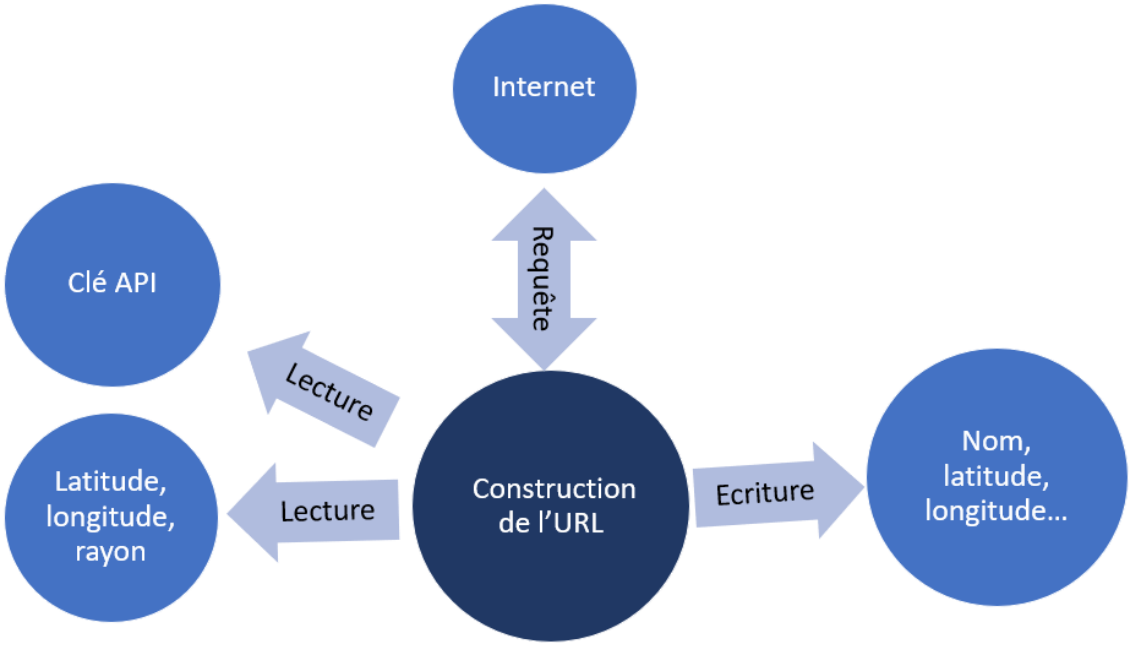

Figure 2 : Fonctionnement du programme

Nous avons obtenu un fichier résultat par pays. L'acquisition des données pour un pays peut prendre plusieurs heures, voire plusieurs jours. Ceci dépend de la taille du pays.

1.1.2 Résultats

Les fichiers obtenus contiennent de nombreuses cellules vides. Nous avons plus prêté attention à certaines variables car celles-ci avaient un nombre de cellules non vides élevé. (Tableau 1)

Tableau 1 : Fréquence de cellules non vides par variable – Google Places API

Variables	**Fréquence cellules non vides**
Nom, latitude, longitude	100%
Adresse, niveau administratif, pays	90-100%
Type, localité	60-80%
Numéro de téléphone, site internet	40-65%
Autres variables	0-5%

Les variables 'Nom', 'Latitude', 'Longitude', 'Adresse', 'Activité', 'Numéro de téléphone', 'Site internet' nous ont été les plus utiles pour la préparation des données.

1.2 OpenStreetmap

1.2.1 Téléchargement des données

Les données ont été téléchargées sur le site data.humdata.org.
Pour chaque pays il suffit de télécharger les dossiers hotosm_kna_buildings_polygons_shp.zip et hotosm_kna_points_of_interest_points_shp.zip. Le premier dossier contient tous les types d'édifices, représentés par des polygones. Le second contient uniquement des lieux d'intérêt, représentés par des points.
Tous les attributs ont été retenus : 'OSM_id', 'Nom', 'Latitude', 'Longitude', 'Commodités', 'Artificiel', 'Magasin', 'Tourisme', 'Horaires d'ouverture', 'Nombre de lits', 'Nombre de chambres', 'Adresse complète', 'Adresse maison', 'Adresse Rue', 'Adresse Ville'.
Deux attributs ('Adresse', 'Activité') ont été créés afin de rassembler les informations de plusieurs attributs. 'Adresse' a remplacé les attributs : 'Adresse complète', 'Adresse maison', 'Adresse Rue', 'Adresse Ville'. 'Activité' a remplacé les attributs : 'Commodités', 'Artificiel', 'Magasin', 'Tourisme'.
Ainsi les attributs finalement retenus sont : 'OSM_id', 'Nom', 'Latitude', 'Longitude', 'Adresse', 'Activité', 'Horaires d'ouverture', 'Nombre de lits', 'Nombre de chambres'.

1.2.2 Résultats

Tableau 2 : Fréquence de cellules non vides par variable - OpenStreetmap

Variables	**Fréquence cellules non vides**
Nom, latitude, longitude, Activité	80-100%
Adresse	30-40%
Autres variables	0-5%

Les variables 'Nom', 'Latitude', 'Longitude', 'Adresse', et 'Activité' nous ont été les plus utiles pour la préparation des données. Nous avons décidé de ne pas conserver les lignes où le nom de l'entreprise (ou lieu d'intérêt) était inconnu.

2. Préparation des données

En observant chaque fichier nous avons remarqué que certaines lignes représentaient la même entité. Il a donc été nécessaire de procéder à un nettoyage des données afin qu'une entreprise (ou lieu d'intérêt) ne soit présente qu'une seule fois dans le fichier.

2.1 Google Places API

2.1.1 Uniformisation des données

Avant de procéder au couplage des données, nous avons procédé à l'uniformisation des données. En effet certaines informations contenaient des erreurs (orthographe, ponctuation, encodage…) et il était important de les corriger. (Tableau 3)
Pour le nom et l'adresse nous avons supprimé les accents, les majuscules, les espaces, les signes de ponctuation et modifié l'encodage afin de ne plus avoir de caractères spéciaux. Les valeurs initiales de ces attributs ont toutefois été sauvegardées : nous avons créé les variables Nom initial et Adresse initiale.
La précision arithmétique des latitude et longitude n'était pas la même. Il a été décidé de retenir 6 chiffres après la virgule.
Pour le numéro de téléphone certaines lignes ne contenaient pas de parenthèses permettant d'identifier l'indicatif téléphonique du pays. Toutes les parenthèses ont donc été supprimées.
Pour le type d'activité nous avons supprimé les espaces, les virgules en trop et le mot 'premise'. Le mot 'premise' n'apporte pas de réelle information quant au type d'activité de l'entreprise (ou lieu d'intérêt).
Pour le site internet certaines lignes ne contenaient pas l'abréviation « www ». Celle-ci a finalement été supprimée de toutes les lignes.

Tableau 3 : Exemples de modifications – Google Places API

Valeur initiale	**Nouvelle valeur**
Super-example's company	superexamplescompany
25.20547	25.205470
(111) 000-0000	111 000-0000
restaurant,, food, premise	restaurant,food
http://www.theexample.com	http://theexample.com

2.1.2 Couplage de données déterministe

Certaines lignes sont identiques à l'exception de 'Coord_lat' (latitude de la requête), 'Coord_lng' (longitude de la requête), 'Search_radius' (rayon de recherche) et 'Coord_id' (identifie de manière unique la requête). (Tableau 4)

Tableau 4 : Exemples de doublons – Google Places API

Nom	Latitude	Longitude	…	Coord_lat	Coord_lng	Search_radius	Coord_id
example	-0.547031	47.462236		-0.547000	47.462000	500	1
example	-0.547031	47.462236		-0.547000	47.462200	500	2

Lorsque nous envoyons une requête via une API, une liste des entreprises et des lieux d'intérêt d'une zone géographique nous est retournée. Cette zone géographique peut avoir été déjà couverte par une autre requête. Une entreprise (ou lieu d'intérêt) peut être retournée par plusieurs requêtes. (Figure 3)

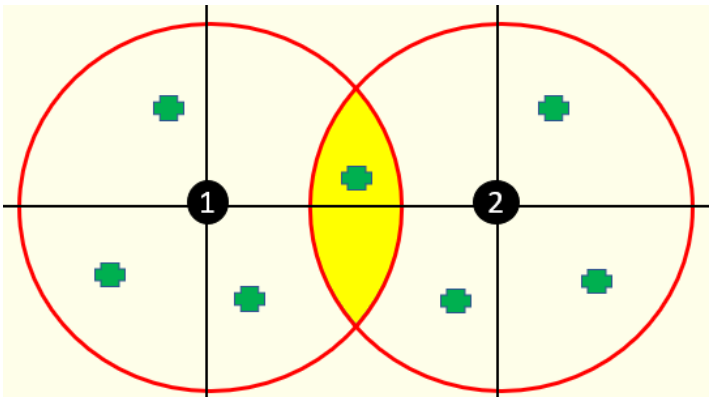

Figure 3 : Intersection des zones de recherche – Google Places API

Une entreprise est identifiable par l'attribut 'Place_id'. Il nous a donc suffi de trouver les lignes qui avaient le même 'Place_id' et de ne conserver qu'une seule ligne par 'Place_id'. Le nombre d'entreprises et de lieux d'intérêt a été mis à jour. (Figure 4)

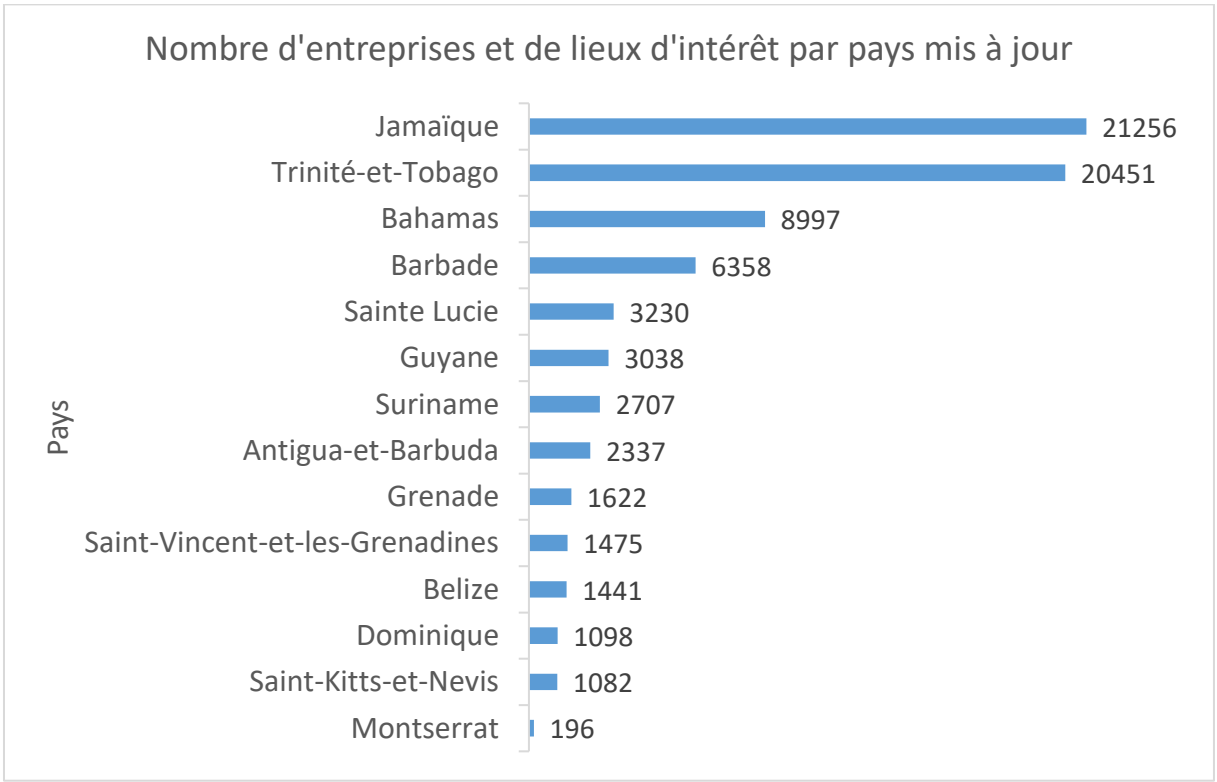

Figure 4 : Nombre d'entreprises et de lieux d'intérêt par pays – Google Places API

2.1.3 Couplage de données probabiliste

Certaines lignes se ressemblent : les attributs 'Nom', 'Latitude', 'Longitude', 'Adresse', 'Numéro de téléphone', 'Activité' et 'Site internet' sont presque identiques. (Tableau 5)

Tableau 5 : Comparaison de deux lignes – Google Places API

Nom	**Latitude**	**Longitude**	**…**	**Activité**	**Place_id**	**…**
example	-0.547020	-47.462230		restaurant	CHijc9r4	
examplestore	-0.547022	-47.462234		restaurant,food	CHabz2t3	

Nous en avons déduit que ces lignes représentaient la même entité. Pour déterminer si deux lignes représentaient la même entité nous avons utilisé la librairie recordlinkage de Python.
Le record linkage (appelé aussi data matching) est une procédure rassemblant des informations d'au moins deux lignes qui semblent appartenir à la même entité. Plusieurs attributs ont été utilisés pour déterminer si les lignes étaient des doublons : 'Nom', 'Latitude', 'Longitude', 'Adresse', 'Numéro de téléphone', 'Activité' et 'Site internet'.

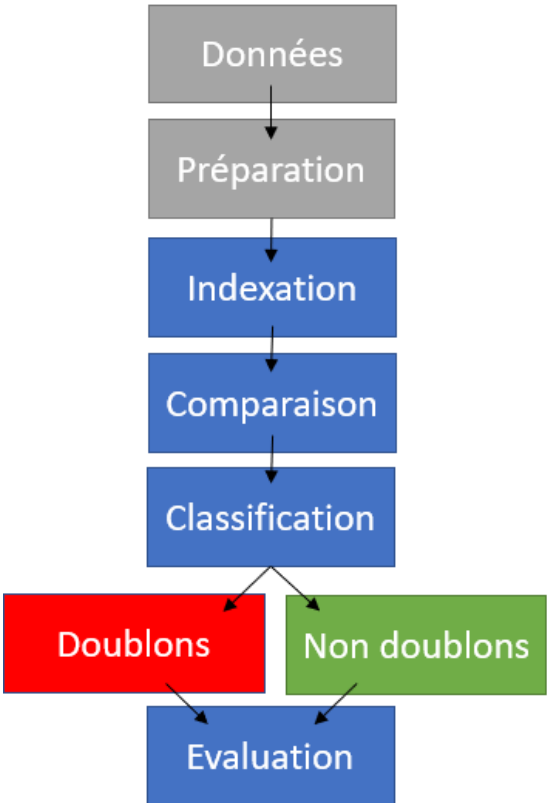

Figure 5 : Le processus de record linkage

Le processus de record linkage se fait en plusieurs étapes (Figure 5). L'étape de l'indexation consiste à coupler toutes les données afin d'obtenir toutes les paires possibles. Lors de l'étape de la comparaison les paires sont comparées par rapport à un ensemble d'attributs. Ces paires sont ensuite classifiées : doublons ou non doublons. Enfin le modèle est évalué.

Lors de l'indexation plusieurs algorithmes peuvent être utilisés. (Tableau 6)

Tableau 6 : Comparaison des algorithmes d'indexation

Algorithmes
FullIndex : tous les attributs sont pris en compte. Cette méthode génère toutes les paires possibles. Dans le cas où nous prenons un fichier de n lignes, nous aurons $n^2/2$ combinaisons.
Block : retourne toutes les paires possibles en se basant sur une variable prise en paramètre.
SortedNeighbourhood : retourne toutes les paires possibles en se basant sur une variable prise en paramètre. Il est moins strict que l'algorithme Block : des erreurs mineures sont acceptées.

L'algorithme FullIndex n'a pas été retenu. Le nombre de comparaisons peut être très important pour des fichiers de grande taille. Pour Saint-Kitts-et-Nevis (un fichier de seulement 1082 lignes) nous obtenons 585362 paires. Le temps d'exécution du programme pour ce fichier dépassait les 10 minutes. L'algorithme SortedNeighbourhood a été sélectionné car nous avions un nombre important d'erreurs d'orthographe.
Lors de l'étape de la comparaison, l'algorithme donne un score compris entre 0 et 1 pour chaque attribut. 1 : les informations sont identiques, 0 : les informations sont très différentes.

Toutefois une seule variable n'était pas suffisante pour déterminer avec exactitude si deux entreprises (ou lieux d'intérêt) représentaient la même entité. 3 variables nous sont semblées indispensables pour trouver des doublons : 'Nom', 'Latitude' et 'Longitude'. Les entreprises (ou lieux d'intérêt) doivent porter un nom similaire et être très proches géographiquement. Les attributs utilisés pour comparer ces lignes sont donc de deux types : des coordonnées géographiques et des chaines de caractères.
L'étape de l'indexation a donc été divisée en plusieurs sous-étapes. Nous n'avons pas eu une étape d'indexation et une étape de comparaison mais 3 étapes d'indexation et 3 étapes de comparaison : 1 étape d'indexation et 1 étape de comparaison par variable ('Nom', 'Latitude', 'Longitude'). Nous avons obtenu 3 fichiers et ces 3 fichiers ont été fusionnés pour n'en former plus qu'un : notre fichier comparaisons.
Plusieurs distances permettent de mesurer la similarité entre deux chaines de caractères : la distance de levenshtein, la distance de damerau-levenshtein et la distance de jarowinkler.
La distance de levenshtein est le nombre minimal de changements (suppressions, insertions, substitutions) nécessaires pour passer d'une chaine de caractères à une autre. La distance de damerau-levenshtein considère en plus les transpositions. L'algorithme de Jaro est une mesure des caractères en commun. Winkler a modifié cet algorithme : les différences au début d'une chaine de caractères ont plus d'importance que les différences à la fin d'une chaine de caractères. Jarowinkler est utile pour comparer de petites chaines de caractères comme les mots, des noms. C'est cet algorithme que nous avons utilisé.

Pour les coordonnées géographiques, la distance d'Haversine est la distance entre deux points d'une sphère définis par leurs longitudes et latitudes.

Tableau 7 : Exemples de combinaisons – Google Places API

Nom 0	**Nom 1**	**Nom score**	**Latitude 0**	**Latitude 1**	**Longitude 0**	…
example	examplestore	0.85	-0.547020	-0.547022	-47.462230	

…	**Longitude 1**	**Geo score**	**Adresse 0**	**Adresse 1**	**Adresse score**	…
	-47.462234	0.97	Example street	Example st	0.9	

…	**Numéro de téléphone 0**	**Numéro de téléphone 1**	**Numéro de téléphone score**	…
	000 000-0000		0	

…	**Site internet 0**	**Site internet 1**	**Site internet score**	…
	http://example.com	http://example.com/	0.98	

…	**Activité 0**	**Activité 1**	**Activité score**
	Restaurant	Restaurant, food	0.75

Dans l'exemple ci-dessus (Tableau 7) les entreprises semblent représenter la même entité.
Il y a donc une redondance d'information.
Lorsque des doublons sont identifiés il faut par la suite supprimer la ligne en trop. Nous conservons la ligne avec le nombre de cellules vides minimal. Dans cet exemple nous conserverions la première entreprise ('example' et non 'examplestore').
Cette étape vient à la fin du processus, lorsque tous les doublons ont été identifiés.

Nous avons aussi testé des algorithmes phonétiques. Ils permettent de convertir des chaines de caractères en codes phonétiques. Il existe 4 algorithmes : Soundex, nysiis, metaphone et match_rating. Ils sont en général très utiles pour les étapes d'indexation et de comparaison. Par exemple 'Smith' et 'Smyth' ont un score de similarité de 1. Dans notre cas nous avons obtenu de moins bons résultats car la plupart des voyelles étaient transformées en 'a'. Ceci eut un impact sur les scores de similarité. Nous n'avons donc pas conservé ces algorithmes.

Après les étapes d'indexation et de comparaison nous avons l'étape de la classification.
Plusieurs modèles de classification supervisée ont été testés : la régression logistique, les arbres de décision, le perceptron.
L'objectif principal de la classification supervisée : nous disposons d'une base d'objets déjà classés et nous devons prévoir la classe d'un nouvel objet.

Dans le dernier fichier obtenu (Tableau 7) nous avons ajouté deux variables : la variable 'Expert' et la variable 'y'. Pour la variable 'Expert', l'expert renseigne si les combinaisons sont des doublons ou non (0 : doublons ; 1 : non doublons). Pour la variable 'y', c'est le modèle qui prédit si ce sont des doublons ou non.

Pour chaque modèle testé les données ont été séparées en deux jeux de données : un jeu de données pour l'apprentissage et un jeu de données test. Le premier permet d'entrainer le modèle, le second permet de le tester et de mesurer l'erreur de ce modèle. Le jeu de données pour l'apprentissage représente 75% du jeu de données initial et le jeu de données test 25%.

Il existe plusieurs mesures de la performance d'un modèle. Dans le cas d'une classification nous utilisons la matrice de confusion, la courbe ROC (Caractéristique de fonctionnement du récepteur) et d'autres indicateurs tels que : Précision (Precision), Rappel (Recall), Exactitude (Accuracy) et F mesure.

La matrice de confusion est une matrice 2x2 (pour 2 classes) qui mesure la qualité d'un modèle de classification. (Figure 6)

Vrais Négatifs (VN) : vrais doublons	Faux Positifs (FP) : faux non doublons
Faux Négatifs (FN) : faux doublons	Vrais Positifs (VP) : vrais non doublons

Figure 6 : Matrice de confusion – Convention scikit learn

La précision est la proportion de doublons qui sont réellement des doublons.
Précision = VP / (VP + FP)
Le rappel est la proportion de doublons qui ont bien été prédits comme tels.
Rappel = VP / (VP + FN)
L'exactitude est la proportion de combinaisons qui ont été bien prédites.
Exactitude = (VP + VN) / (VP + FP + VN + FN)
La F-mesure est la moyenne harmonique de la précision et du rappel.
F-mesure = 2 x Précision x Rappel / (Précision + Rappel)

La courbe ROC est un graphique qui affiche le taux de vrais positifs en fonction du taux de faux positifs pour plusieurs seuils compris entre 0.0 et 1.0.
Taux de vrais positifs = Vrais positifs / (Vrais positifs + Faux négatifs)
Taux de faux positifs = Faux positifs / (Faux positifs + Frais négatifs)
L'aire sous la courbe (AUC : area under the curve) comprise entre 0.0 et 1.0 est un indicateur de la performance du modèle. Plus cette aire est importante plus le modèle est performant.

Les résultats présentés ci-dessous sont ceux de Saint Kitts-et-Nevis.

- Régression logistique :

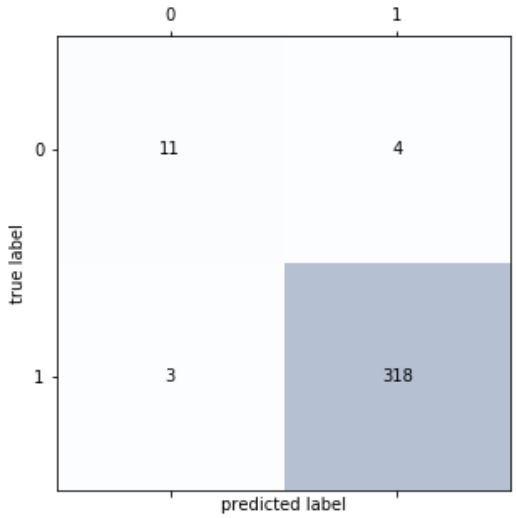

Figure 7 : Matrice de confusion – Régression logistique – Google Places API

Dans notre jeu de données test nous avons 15 doublons. 11 d'entre eux ont été prédits comme doublons et 4 d'entre eux comme non doublons. Nous avons 321 non doublons. 318 d'entre eux ont été prédits comme non doublons et 3 d'entre eux comme doublons. (Figure 7)

Tableau 8 : Indicateurs de performance – Régression logistique – Google Places API

Indicateurs de performance - Régression logistique	
Précision	0.9875776397515528
Exactitude	0.9791666666666666
Rappel	0.9906542056074766
F score	0.9891135303265941

Plus de 98% des combinaisons jugées comme non doublons sont réellement des non doublons. (Précision) Plus de 99% des non doublons ont bien été prédits comme tels (Rappel). Plus de 97% des combinaisons ont été bien prédites (Exactitude). (Tableau 8)

Courbe ROC et AUC :

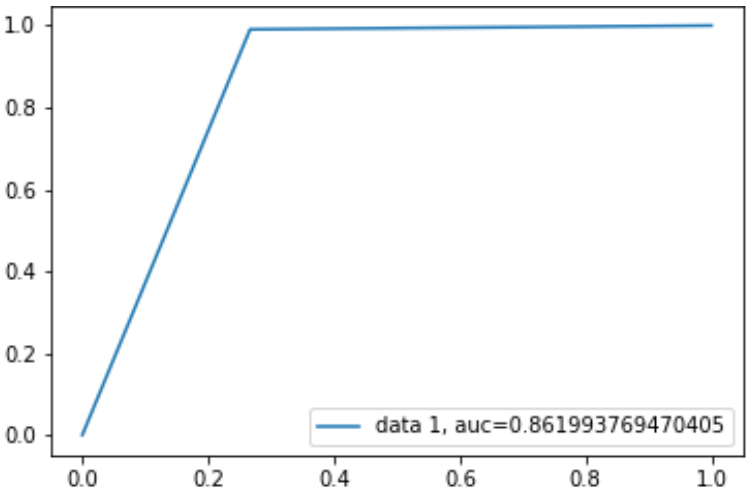

Figure 8 : Courbe ROC – Régression logistique – Google Places API

- Arbre de décision :

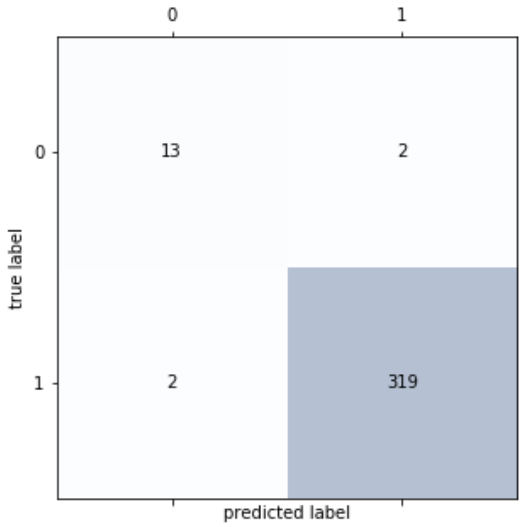

Figure 9 : Matrice de confusion – Arbre de décision – Google Places API

Dans notre jeu de données test nous avons 15 doublons. 13 d'entre eux ont été prédits comme doublons et 2 d'entre eux comme non doublons. Nous avons 321 non doublons. 319 d'entre eux ont été prédits comme non doublons et 2 d'entre eux comme doublons. (Figure 9)

Tableau 9 : Indicateurs de performance – Arbre de décision – Google Places API

Indicateurs de performance – Arbre de décision	
Précision	0.9937694704049844
Exactitude	0.9880952380952381
Rappel	0.9937694704049844
F score	0.9937694704049844

Plus de 99% des combinaisons jugées comme non doublons sont réellement des non doublons. (Précision) Plus de 99% des non doublons ont bien été prédits comme tels (Rappel). Plus de 98% des combinaisons ont été bien prédites (Exactitude). (Tableau 9)

Courbe ROC et AUC :

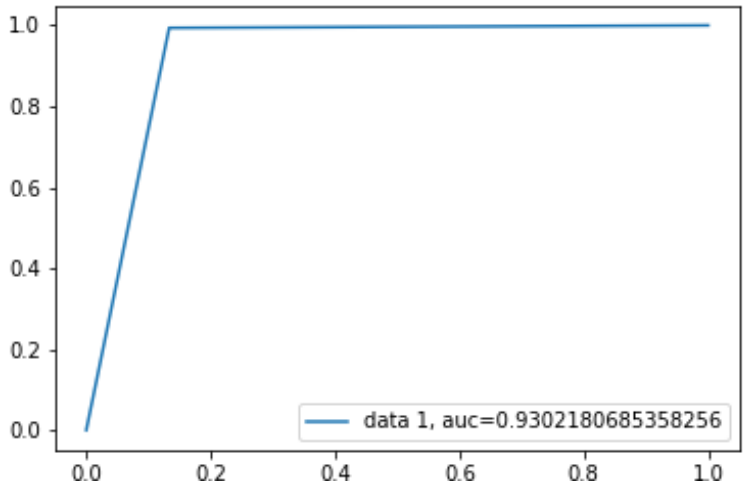

Figure 10 : Courbe ROC – Arbre de décision – Google Places API

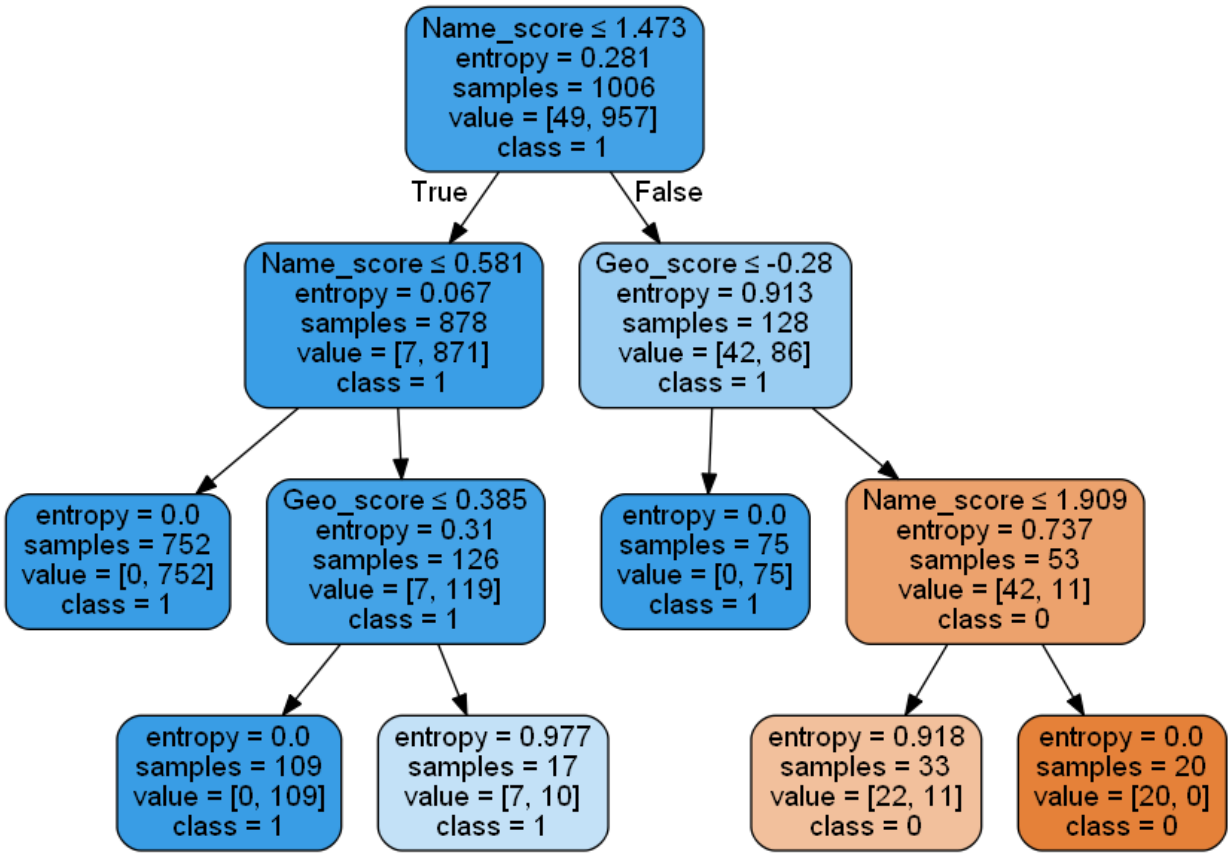

Figure 11 : Arbre de décision – Google Places API

Pour expliquer cet arbre de décision il faut d'abord déstandardiser les données.

Rappelons comment les données ont été standardisées :

Nom score standardisé = (Nom score – moyenne de Nom score) / Ecart type de Nom score

Geo score standardisé = (Geo score – moyenne de Geo score) / Ecart type de Geo score

Moyenne Nom score = 0.6051909263204545 ; Ecart type Nom score = 0.1776568065139753

Moyenne Geo score = 0.7802231768458663 ; Ecart type Geo score = 0.31222934229315896

Nom score standard ≤ 1.473 : nous obtenons Nom score ≤ 0.8668794023155402.

Geo score standard ≤ -0.28 : nous obtenons Geo score ≤ 0.6927989610037818.

Dans le cas où Nom score > 0.8668794023155402 et Geo score > 0.6927989610037818 il s'agit de doublons. Dans le cas contraire il s'agit de non doublons.

- Perceptron :

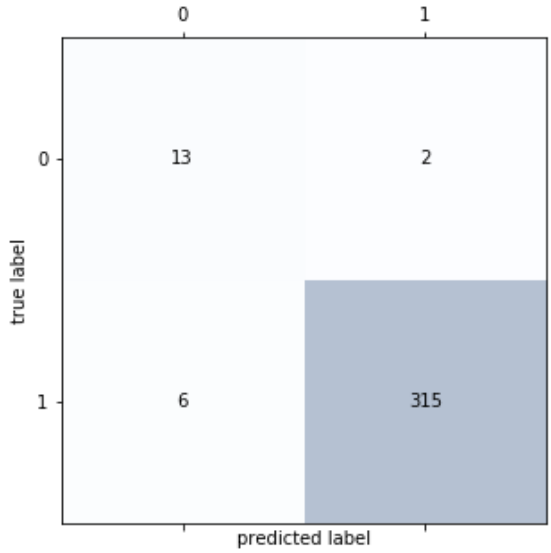

Figure 12 : Matrice de confusion – Perceptron – Google Places API

Dans notre jeu de données test nous avons 15 doublons. 13 d'entre eux ont été prédits comme doublons et 2 d'entre eux comme non doublons. Nous avons 321 non doublons. 315 d'entre eux ont été prédits comme non doublons et 6 d'entre eux comme doublons. (Figure 12)

Tableau 10 : Indicateurs de performance – Perceptron – Google Places API

Indicateurs de performance - Perceptron	
Précision	0.9936908517350158
Exactitude	0.9761904761904762
Rappel	0.9813084112149533
F score	0.987460815047022

Plus de 99% des combinaisons jugées comme non doublons sont réellement des non doublons. (Précision) Plus de 98% des non doublons ont bien été prédits comme tels (Rappel). Plus de 97% des combinaisons ont été bien prédites (Exactitude). (Tableau 10)

Courbe ROC et AUC :

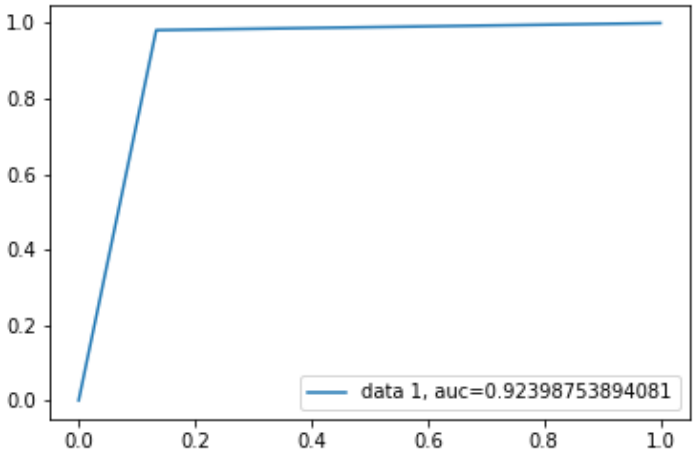

Figure 13 : Courbe ROC – Perceptron – Google Places API

Avec une aire sous la courbe (AUC) de plus de 0.93, le meilleur modèle est l'arbre de décision. Pour Saint Kitts-et-Nevis 60 doublons ont été identifiés.

Lorsque tous les doublons ont été identifiés et supprimés, le nombre d'entreprises et de lieux d'intérêt par pays a été une nouvelle fois mis à jour. (Tableau 11)

Tableau 11 : Nombre d'entreprise par pays après couplage des données – Google Places API

Pays	**Nombre initial d'entreprises et de lieux d'intérêt**	**Nombre d'entreprises et de lieux d'intérêt après le couplage des données probabiliste**
Antigua-et-Barbuda	2337	2303
Bahamas	4817	4768
Barbade	6358	6297
Belize	1441	1427
Dominique	1098	1089
Grenade	1622	1611
Guyane	3038	3020
Jamaïque	21256	20953
Montserrat	196	196
Sainte Lucie	3230	3182
Saint-Kitts-et-Nevis	1082	1022
Saint-Vincent-et-les-Grenadines	1475	1456
Suriname	2707	2697
Trinité-et-Tobago	20451	20191

2.2 OpenStreetmap

La préparation des données d'OpenStreetmap est la même que celle des données de Google Places API.

Il a fallu en premier lieu procéder à l'uniformisation des données.
Les variables 'Nom', 'Adresse', 'Latitude', 'Longitude' et 'Activité' ont subi les mêmes modifications que celles présentées ci-dessus.

Contrairement à Google Places API, les entreprises ne sont pas retournées par plusieurs requêtes. Une entreprise n'est représentée que par une seule ligne (OSM_id uniques). L'étape du couplage de données déterministe n'était donc pas nécessaire.

En revanche certaines lignes se ressemblent fortement : les attributs 'Nom', 'Latitude', 'Longitude', 'Adresse' et 'Activité' sont presque identiques. Comme pour les données de Google Places API nous avons procédé à un couplage de données probabiliste. Dans ce cas seules les variables 'Nom', 'Latitude', 'Longitude', 'Adresse', et 'Activité' ont été utilisées pour déterminer si les lignes étaient des doublons. (Tableau 12)

Tableau 12 : Exemples de combinaisons - OpenStreetmap

Nom 0	**Nom 1**	**Nom score**	**Latitude 0**	**Latitude 1**	**Longitude 0**	…
example	examplestore	0.85	-0.547020	-0.547022	-47.462230	

…	**Longitude 1**	**Geo score**	**Adresse 0**	**Adresse 1**	**Adresse score**	…
	-47.462234	0.97	Example street	Example st	0.9	

…	**Activité 0**	**Activité 1**	**Activité score**
	Restaurant	Restaurant, food	0.75

Les résultats présentés ci-dessous sont ceux de Saint Kitts-et-Nevis.

- Régression logistique:

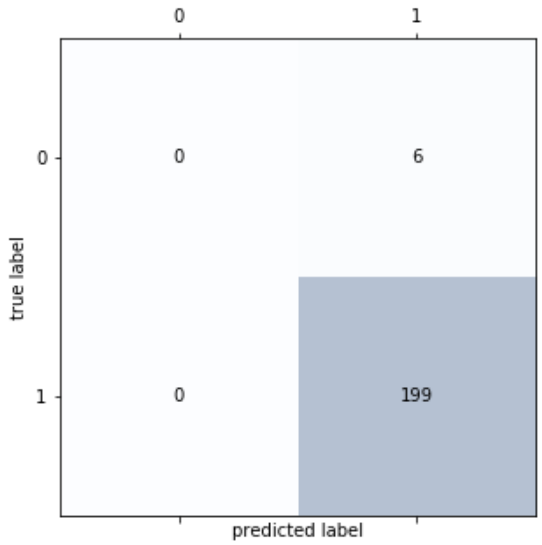

Figure 14 : Matrice de confusion – Régression logistique – OpenStreetmap

Dans notre jeu de données test nous avons 6 doublons. Ils ont tous été prédits comme non doublons. Nous avons 199 non doublons. Ils ont tous été prédits comme non doublons. (Figure 14)

Tableau 13 : Indicateurs de performance – Régression logistique - OpenStreetmap

Indicateurs de performance – Régression logistique	
Précision	0.9707317073170731
Exactitude	0.9707317073170731
Rappel	1.0
F score	0.9851485148514851

Plus de 97% des combinaisons jugées comme non doublons sont réellement des non doublons. (Précision) 100% des non doublons ont bien été prédits comme tels (Rappel). Plus de 97% des combinaisons ont été bien prédites (Exactitude). (Tableau 13)

Courbe ROC et AUC :

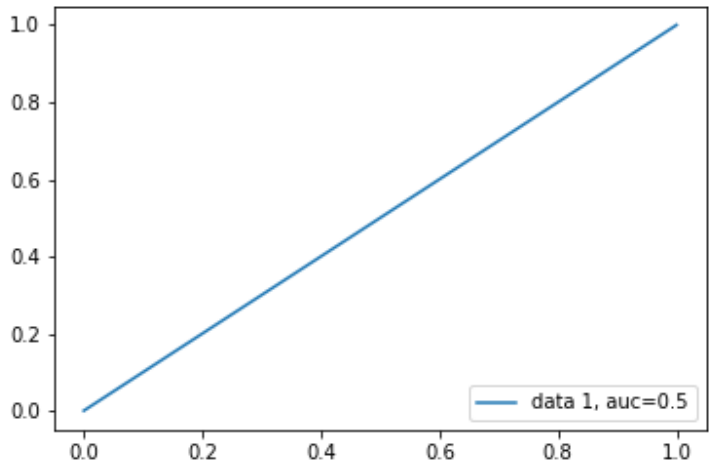

Figure 15 : Courbe ROC – Régression logistique - OpenStreetmap

- Arbre de décision :

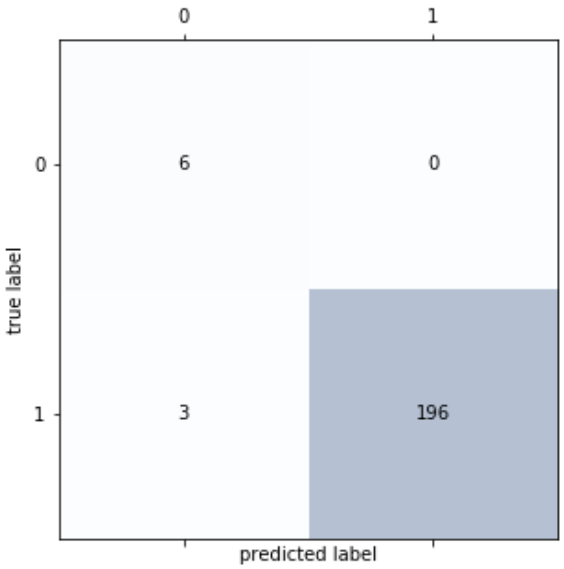

Figure 16 : Matrice de confusion – Arbre de décision – OpenStreetmap

Dans notre jeu de données test nous avons 6 doublons. Ils ont tous été prédits comme doublons. Nous avons 199 non doublons. 196 d'entre eux ont été prédits comme non doublons et 3 d'entre eux comme doublons. (Figure 16)

Tableau 14 : Indicateurs de performance – Arbre de décision - OpenStreetmap

Indicateurs de performance – Arbre de décision	
Précision	1.0
Exactitude	0.9853658536585366
Rappel	0.9849246231155779
F score	0.9924050632911393

100% des combinaisons jugées comme non doublons sont réellement des non doublons. (Précision) Plus de 98% des non doublons ont bien été prédits comme tels (Rappel). Plus de 98% des combinaisons ont été bien prédites (Exactitude). (Tableau 14)

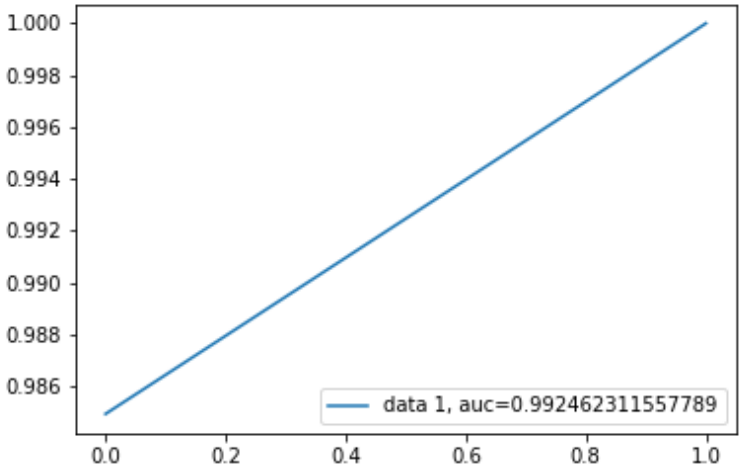

Figure 17 : Courbe ROC – Arbre de décision - OpenStreetmap

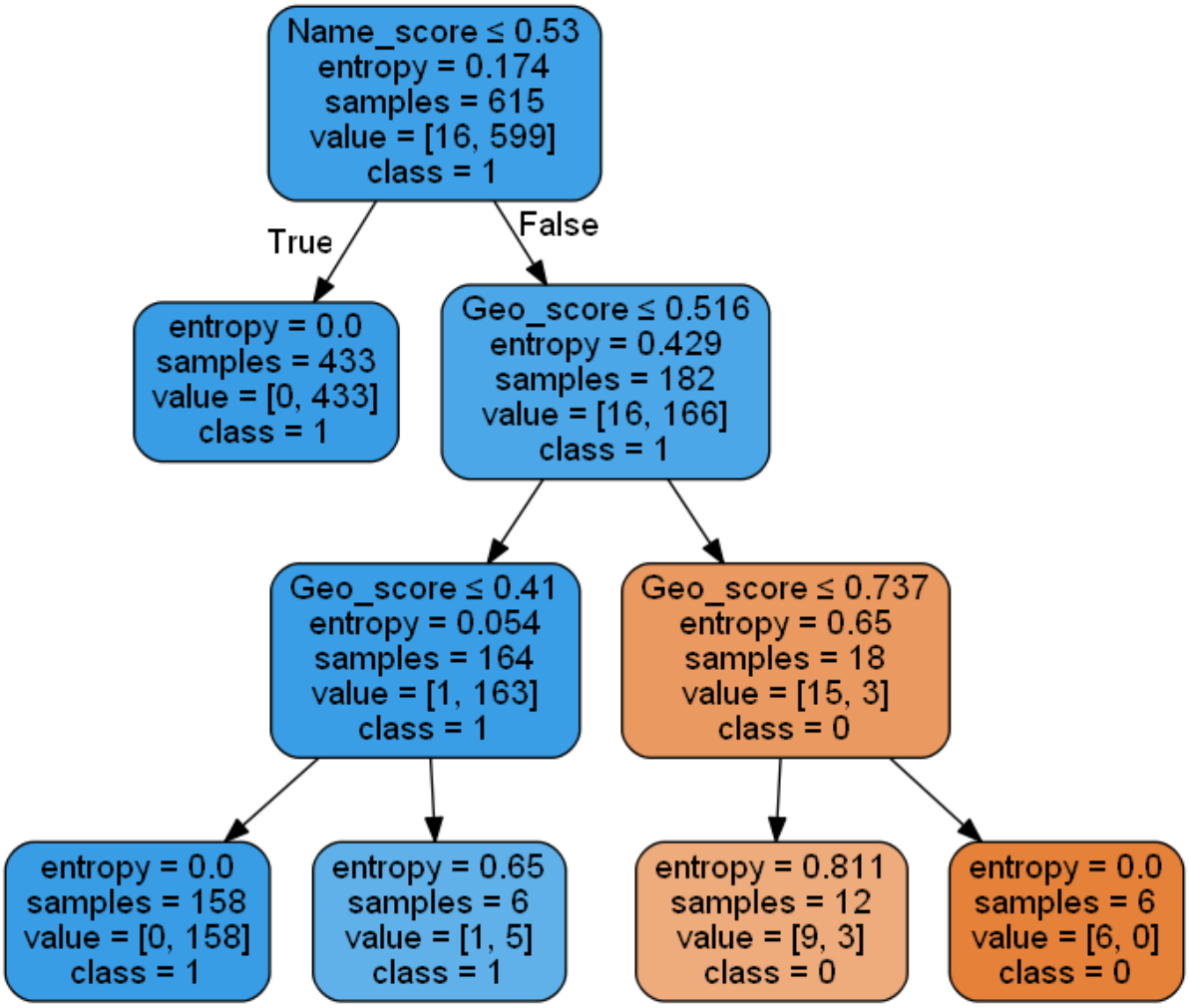

Figure 18 : Arbre de décision – OpenStreetmap

Moyenne Nom score = 0.6360436797941329 ; Ecart type Nom score = 0.23944788362827377

Moyenne Geo score = 0.7169956435915197; Ecart type Geo score = 0.37867859316458463

Nom score standard ≤ 0.53 : nous obtenons Nom score ≤ 0.7629510581171179.

Geo score standard ≤ 0.516 : nous obtenons Geo score ≤ 0.9123937976644454.

Dans le cas où Nom score > 0.7629510581171179 et Geo score > 0.9123937976644454 il s'agit de doublons. Dans le cas contraire il s'agit de non doublons.

- Perceptron :

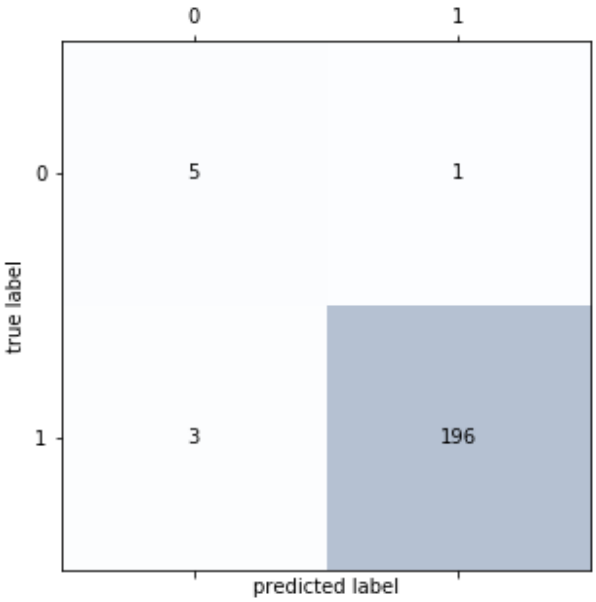

Figure 19 : Matrice de confusion – Perceptron – OpenStreetmap

Dans notre jeu de données test nous avons 6 doublons. 5 d'entre eux ont été prédits comme doublons et 1 seul a été prédit comme non doublon. Nous avons 199 non doublons. 196 d'entre eux ont été prédits comme non doublons et 3 d'entre eux comme doublons. (Figure 19)

Tableau 15 : Indicateurs de performance – Perceptron - OpenStreetmap

Indicateurs de performance - Perceptron	
Précision	0.9949238578680203
Exactitude	0.9804878048780488
Rappel	0.9849246231155779
F score	0.9898989898989898

Plus de 99% des combinaisons jugées comme non doublons sont réellement des non doublons. (Précision) Plus de 98% des non doublons ont bien été prédits comme tels (Rappel). Plus de 98% des combinaisons ont été bien prédites (Exactitude). (Tableau 15)

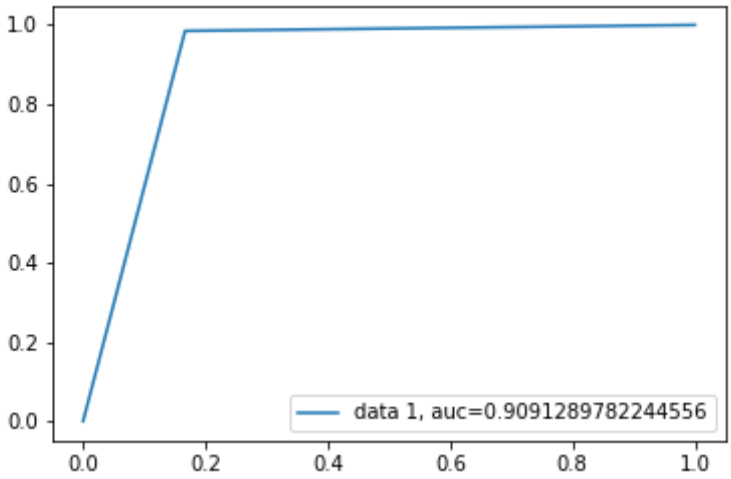

Figure 20 : Courbe ROC – Perceptron - OpenStreetmap

Avec une aire sous la courbe (AUC) de plus de 0.99, le meilleur modèle est l'arbre de décision. Pour Saint Kitts-et-Nevis 21 doublons ont été identifiés.

Lorsque tous les doublons ont été identifiés et supprimés, le nombre d'entreprises et de lieux d'intérêt par pays a été une nouvelle fois mis à jour. (Tableau 16)

Tableau 16 : Nombre d'entreprise par pays après couplage des données – OpenStreetmap

Pays	**Nombre initial d'entreprises et de lieux d'intérêt**	**Nombre d'entreprises et de lieux d'intérêt après le couplage des données probabiliste**
Antigua-et-Barbuda	328	233
Bahamas	1211	871
Barbade	648	542
Belize	2771	2145
Dominique	372	265
Grenade	327	242
Guyane	3121	2365
Jamaïque	2588	2022
Montserrat	42	26
Sainte Lucie	1325	1107
Saint-Kitts-et-Nevis	481	460
Saint-Vincent-et-les-Grenadines	743	562
Suriname	2670	2031
Trinité-et-Tobago	4534	3732

3. Création de la base de données géospatiale

Il faut désormais rassembler les fichiers de Google et d'OpenStreetmap. Le fichier de Google ne contient plus de doublons, tout comme celui d'OpenStreetmap.
Les données ont déjà été uniformisées et les entreprises (ou lieux d'intérêt) sont identifiables par un identifiant unique. En revanche certaines lignes se ressemblent fortement.
Il a donc fallu encore une fois procéder à un couplage des données probabiliste mais cette fois-ci avec deux fichiers.

3.1 Couplage de données probabiliste

Seules les variables 'Nom', 'Latitude', 'Longitude', 'Adresse', et 'Activité' ont été utilisées pour déterminer si les lignes étaient des doublons.

Pour la variable 'Activité' il y a certaines différences entre Google Places API et OpenStreetmap. D'une part les activités ne sont pas les mêmes. Par exemple l'activité 'school' pour Google Places API existe aussi pour OpenStreetmap mais les activités 'kindergarten' et 'university' existent seulement pour OpenStreetmap. D'autre part pour Google Places API une seule entité peut avoir plusieurs activités, ce qui n'est pas le cas pour OpenStreetmap.
Ces différences peuvent avoir un impact sur le score de similarité de la variable 'Activité'.

Les résultats présentés ci-dessous sont ceux de Saint Kitts-et-Nevis.

- Régression logistique :

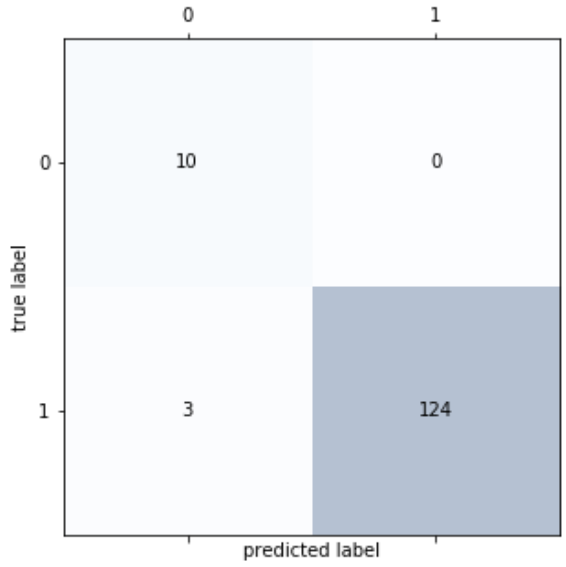

Figure 21 : Matrice de confusion – Régression logistique – OpenStreetmap et Google

Dans notre jeu de données test nous avons 10 doublons. Ils ont tous été prédits comme doublons. Nous avons 127 non doublons. 124 d'entre eux ont été prédits comme non doublons et 3 d'entre eux comme doublons. (Figure 21)

Tableau 17 : Indicateurs de performance – Régression logistique – OpenStreetmap et Google

Indicateurs de performance – Régression logistique	
Précision	1.0
Exactitude	0.9781021897810219
Rappel	0.9763779527559056
F score	0.9880478087649402

100% des combinaisons jugées comme non doublons sont réellement des non doublons. (Précision) Plus de 97% des non doublons ont bien été prédits comme tels (Rappel). Plus de 97% des combinaisons ont été bien prédites (Exactitude). (Tableau 17)

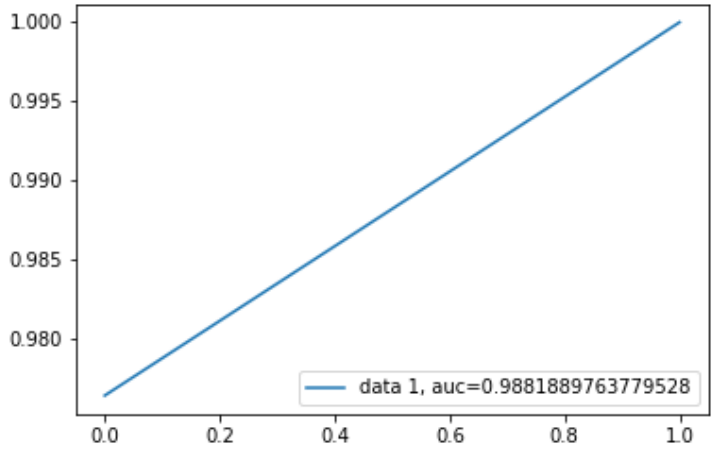

Figure 22 : Courbe ROC – Régression logistique – OpenStreetmap et Google

- Arbre de décision :

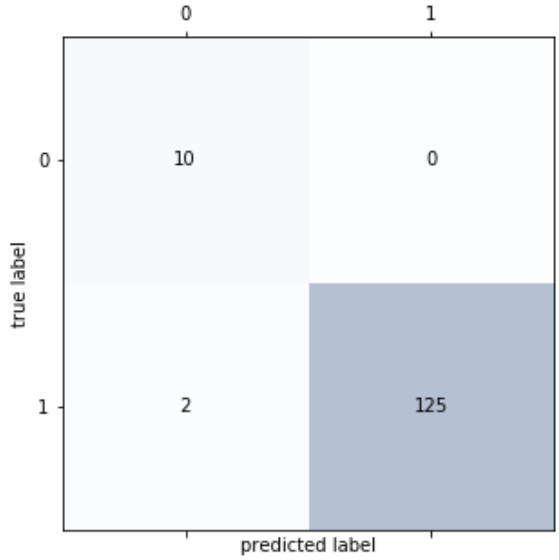

Figure 23 : Matrice de confusion – Arbre de décision – OpenStreetmap et Google

Dans notre jeu de données test nous avons 10 doublons. Ils ont tous été prédits comme doublons. Nous avons 127 non doublons. 125 d'entre eux ont été prédits comme non doublons et 2 d'entre eux comme doublons. (Figure 23)

Tableau 18 : Indicateurs de performance – Arbre de décision – OpenStreetmap et Google

Indicateurs de performance – Arbre de décision	
Précision	1.0
Exactitude	0.9854014598540146
Rappel	0.984251968503937
F score	0.9920634920634921

100% des combinaisons jugées comme non doublons sont réellement des non doublons. (Précision) Plus de 98% des non doublons ont bien été prédits comme tels (Rappel). Plus de 98% des combinaisons ont été bien prédites (Exactitude). (Tableau 18)

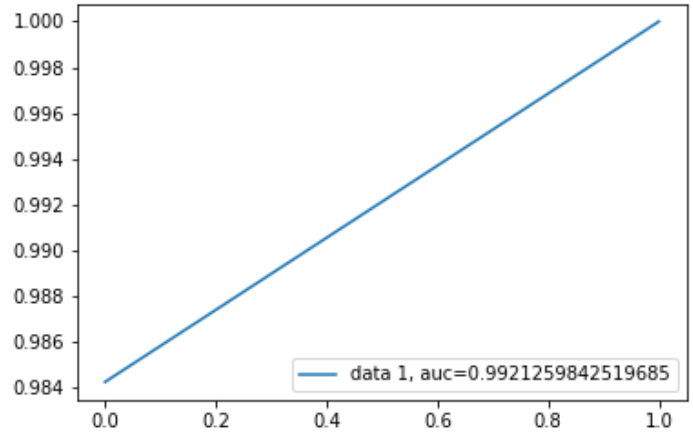

Figure 24 : Courbe ROC – Arbre de décision – OpenStreetmap et Google

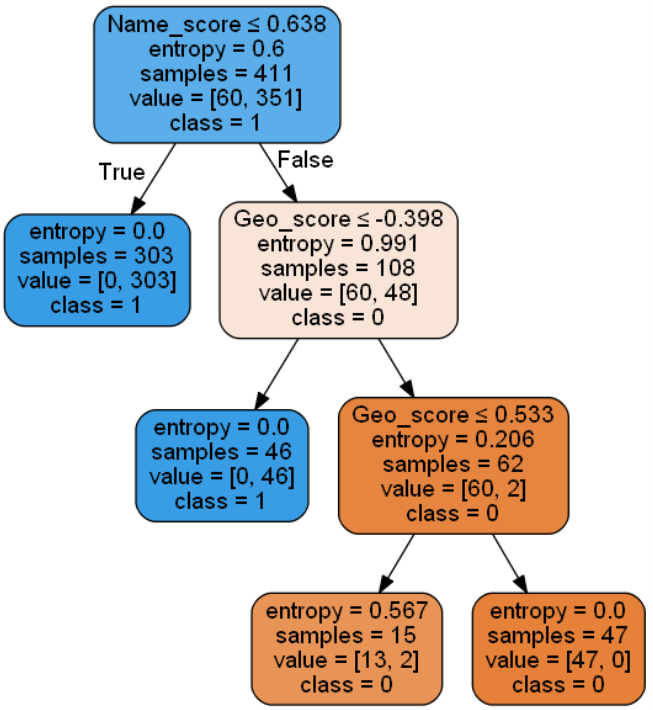

Figure 25 : Arbre de décision – OpenStreetmap et Google

Moyenne Nom score =0.5852051562677193 ; Ecart type Nom score = 0.20393093892706649
Moyenne Geo score =0.820742958452574 ; Ecart type Geo score = 0.2845788569309909

Nom score standard ≤ 0.638 : nous obtenons Nom score ≤ 0.7153130953031877.
Geo score standard ≤ -0.398 : nous obtenons Geo score ≤ 0.7074805733940396.

Dans le cas où Nom score > 0.7153130953031877 et Geo score > 0.7074805733940396 il s'agit de doublons. Dans le cas contraire il s'agit de non doublons.

- Perceptron :

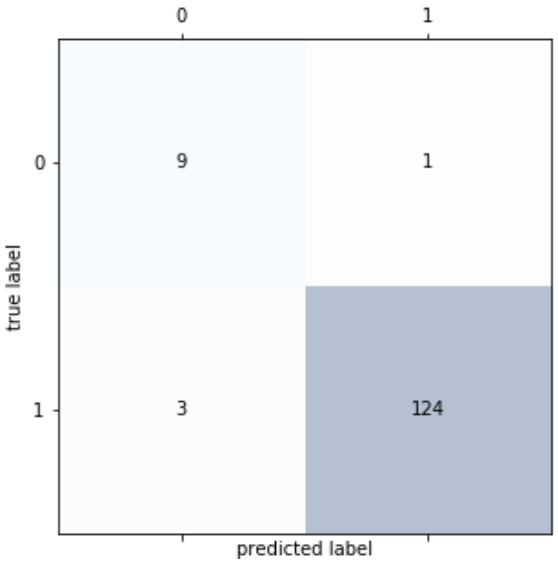

Figure 26 : Matrice de confusion – Perceptron – OpenStreetmap et Google

Dans notre jeu de données test nous avons 10 doublons. 9 d'entre eux ont été prédits comme doublons et 1 d'entre eux a été prédit comme non doublon. Nous avons 127 non doublons. 124 d'entre eux ont été prédits comme non doublons et 3 d'entre eux comme doublons. (Figure 26)

Tableau 19 : Indicateurs de performance – Perceptron – OpenStreetmap et Google

Indicateurs de performance - Perceptron	
Précision	0.992
Exactitude	0.9708029197080292
Rappel	0.9763779527559056
F score	0.9841269841269842

Plus de 99% des combinaisons jugées comme non doublons sont réellement des non doublons. (Précision) Plus de 97% des non doublons ont bien été prédits comme tels (Rappel). Plus de 97% des combinaisons ont été bien prédites (Exactitude). (Tableau 19)

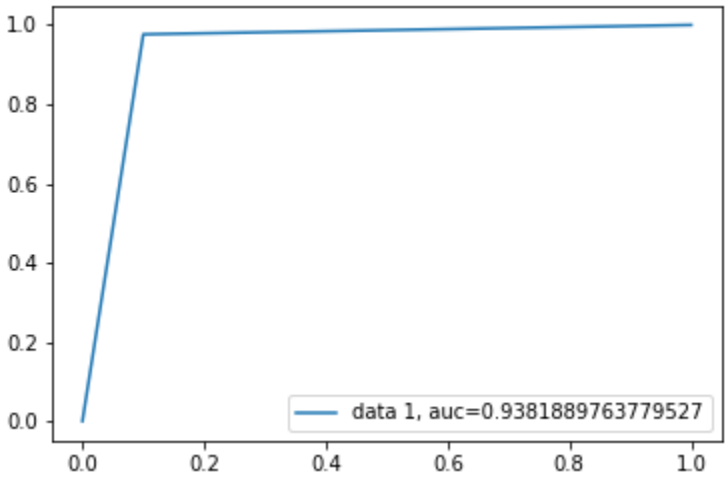

Figure 27 : Courbe ROC – Perceptron – OpenStreetmap et Google

Avec une aire sous la courbe (AUC) de plus de 0.99, le meilleur modèle est l'arbre de décision. Pour Saint Kitts-et-Nevis 70 doublons ont été identifiés.

Lorsque tous les doublons ont été identifiés et supprimés, le nombre d'entreprises et de lieux d'intérêt par pays a été une nouvelle fois mis à jour. (Tableau 20)

Tableau 20 : Nombre d'entreprise par pays après couplage des données – OpenStreetmap et Google

Pays	**Nombre initial d'entreprises et de lieux d'intérêt**	**Nombre d'entreprises et de lieux d'intérêt après le couplage des données probabiliste**
Antigua-et-Barbuda	2536 (2303 + 233)	2409
Bahamas	5639 (4768 + 871)	5434
Barbade	6839 (6297 + 542)	6620
Belize	3572 (1427 + 2145)	3378
Dominique	1354 (1089 + 265)	1302
Grenade	1853 (1611 + 242)	1770
Guyane	5385 (3020 + 2365)	5166
Jamaïque	22975 (20953 + 2022)	22228
Montserrat	222 (196 + 26)	216
Sainte Lucie	4289 (3182 + 1107)	4075
Saint-Kitts-et-Nevis	1482 (1022+460)	1412
Saint-Vincent-et-les-Grenadines	2018 (1456 + 562)	1896
Suriname	4728 (2697 + 2031)	4600
Trinité-et-Tobago	23923 (20191 + 3732)	22770

3.2 Visualisation des données et calcul des contributions

Les données ont été importées sur QGIS. QGIS est un système SIG (système d'information géographique). Il permet de créer, d'éditer, de visualiser et d'analyser des informations géographiques. Une carte avec des données de plusieurs sources est obtenue grâce à la superposition de plusieurs couches (fichiers).

Dans notre cas nous avons d'abord ajouté les données de Google Places API (1ère couche) puis les données d'OpenStreetmap (2ème couche) et enfin la carte de Google Maps obtenue grâce à la fonction Affichage. Ces fichiers se trouvaient sous format csv. Il suffisait seulement de préciser que la latitude sur la carte correspondait à la variable 'Latitude' du fichier et que la longitude sur la carte correspondait à la variable 'Longitude' du fichier.

Pour Saint Kitts-et-Nevis c'est donc à partir des deux cartes ci-dessous (Figure 28 et Figure 29) que nous avons obtenu la carte représentant les 1412 entreprises et lieux d'intérêts du pays (Figure 30).

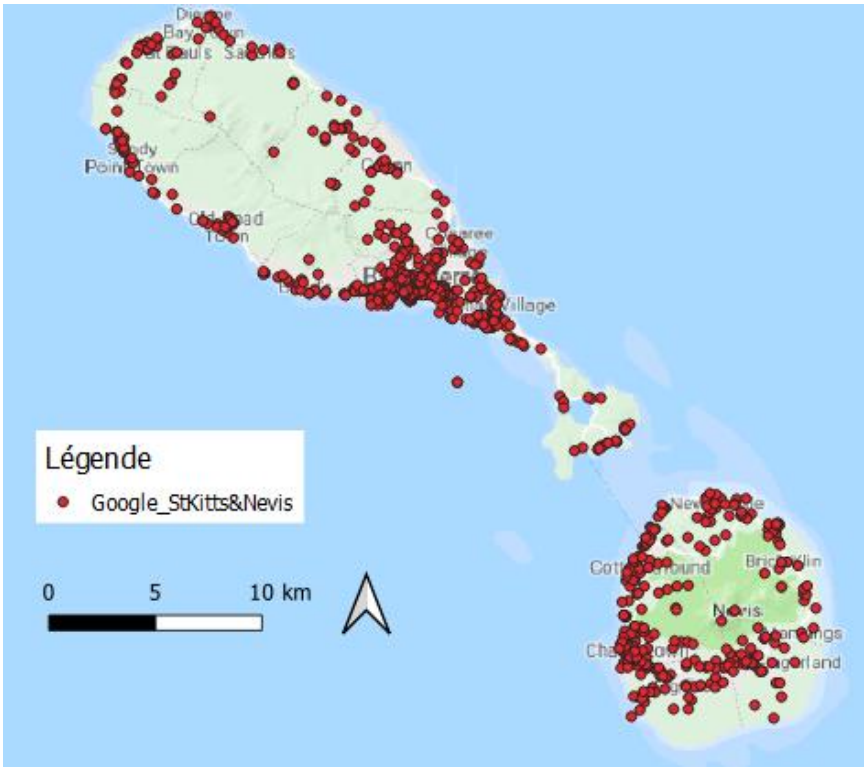

Figure 28 : Carte des entreprises et lieux d'intérêt – Google - Saint Kitts-et-Nevis

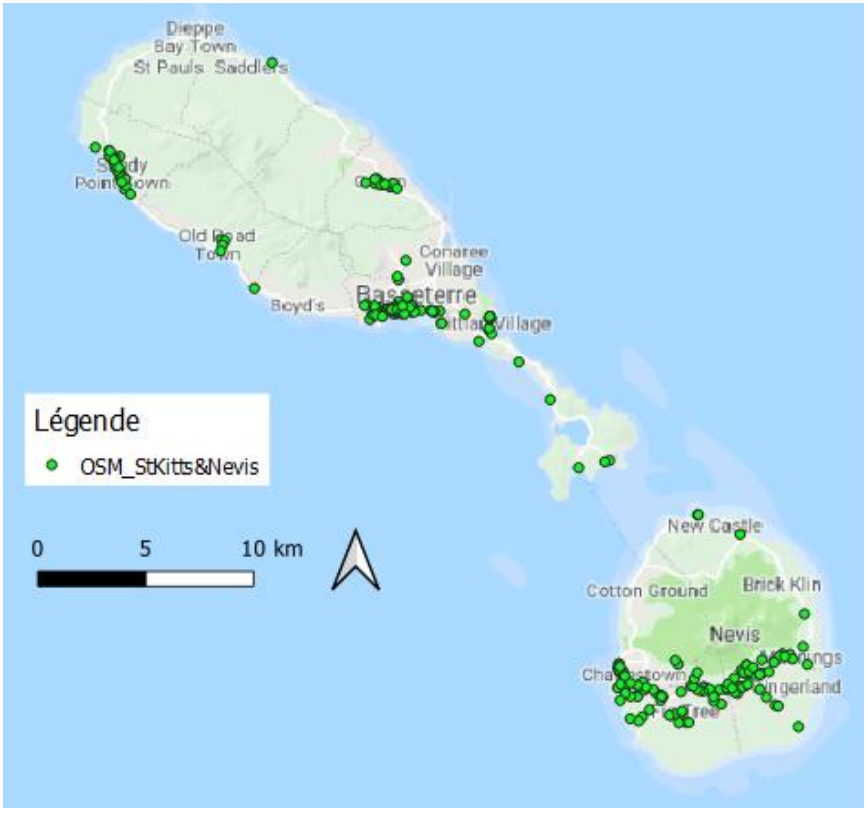

Figure 29 : Carte des entreprises et lieux d'intérêt – OpenStreetmap - Saint Kitts-et-Nevis

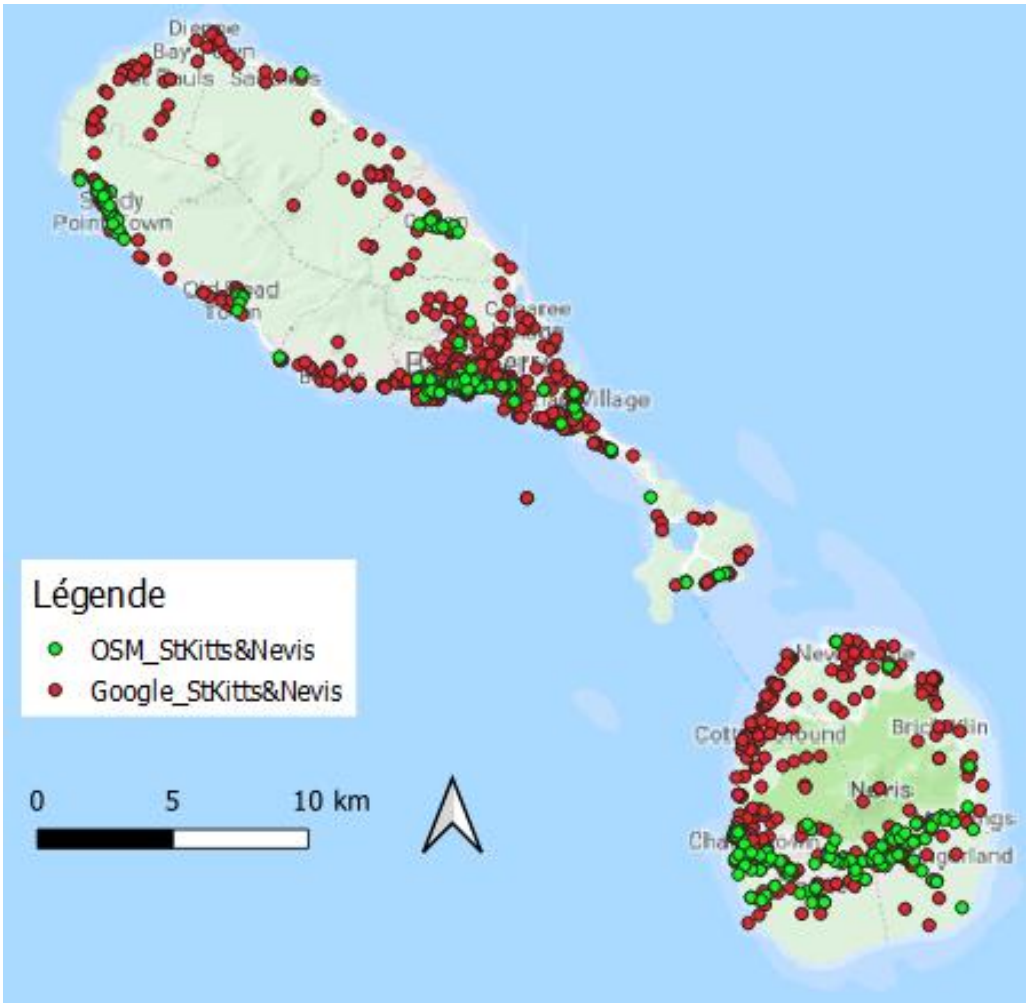

Figure 30 : Carte des entreprises et lieux d'intérêt - Saint Kitts-et-Nevis

Pour Saint Kitts-et-Nevis en ajoutant les données d'OpenStreetmap aux données de Google, nous avons 390 entreprises et lieux d'intérêt supplémentaires. (Figure 31)

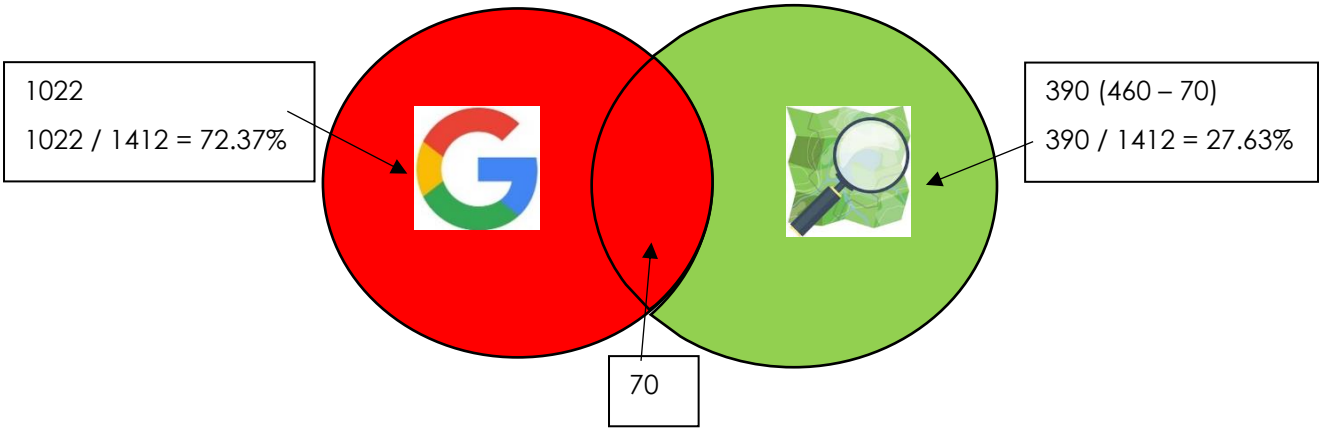

Figure 31 : Contributions de Google Places API et OpenStreetmap

Les données d'OpenStreetmap représentent pour ce fichier plus d'un quart des données.

Tableau 21 : Contributions de Google et d'OpenStreetmap

Pays	**Contribution Google (%)**	**Contribution OpenStreetmap (%)**
Antigua-et-Barbuda	95.60	4.40
Bahamas	87.74	12.26
Barbade	95.12	4.88
Belize	42.24	57.76
Dominique	83.64	16.36
Grenade	91.02	8.98
Guyane	58.46	41.54
Jamaïque	94.26	5.74
Montserrat	90.74	9.26
Sainte Lucie	76.79	23.21
Saint-Kitts-et-Nevis	72.37	27.63
Saint-Vincent-et-les-Grenadines	78.09	21.91
Suriname	58.63	41.37
Trinité-et-Tobago	88.67	11.33

Au final les données d'OpenStreetmap représentent entre 4.40% et 57.76% de l'information.

Conclusion

Dans le cadre du PRASC, des méthodes innovantes sont explorées afin de déterminer si l'obtention de sources de données alternatives pourrait renforcer les méthodes actuelles de mesure de certains indicateurs de la comptabilité nationale pour les pays des Caraïbes. En particulier, un projet exploratoire est en cours pour tenter d'obtenir, à partir du web, une base de données géospatiale de bâtiments. Celle-ci serait ensuite couplée avec d'autres données administratives afin d'évaluer l'évolution de la valeur du parc immobilier de chacun des pays. Ce projet tire parti de plusieurs sources de données, d'applications développées en Python et de méthodes d'apprentissage automatique afin de créer la base de données géospatiale. Il a été démontré à travers ce projet que les méthodes d'acquisition de données web peuvent compléter les méthodes d'enquête traditionnelles et offrent plusieurs avantages, notamment la réduction des coûts, la mise à jour régulière des données et l'efficacité grâce à l'automatisation des processus.

Dans un premier temps les données d'OpenStreetmap ont été ajoutées aux données de Google. Par la suite d'autres sources de données pourront être intégrées. L'un des premiers problèmes que nous avons rencontrés fut la redondance d'informations. Des doublons furent trouvés au sein même de chaque fichier et également lorsque ces fichiers furent rassemblés. Notre application utilise les couplages de données déterministe et probabiliste pour faire à ce problème. Le couplage de données probabiliste s'appuie sur des méthodes d'apprentissage automatique. Il fut montré que l'identification des doublons s'appuie sur trois variables importantes : le nom, la latitude et la longitude de l'entreprise. L'algorithme espérance-maximisation et l'analyse en composantes principales permettront également de mieux comprendre comment trouver ces doublons.
Lorsque les doublons ont été supprimés nous avons pu montrer que l'intégration des données d'OpenStreetmap aux données de Google avait une réelle valeur ajoutée.

Lors de ce stage j'ai développé mes compétences en Python, j'ai découvert QGIS, j'ai approfondi mes connaissances en apprentissage automatique. J'ai régulièrement travaillé au laboratoire d'innovation où j'ai pu notamment échanger, collaborer avec d'autres personnes. J'ai apprécié ce travail d'équipe et j'ai beaucoup appris à leur contact.

Bibliographie

Analytics Vidhya. (2019). *11 Important Model Evaluation Metrics for Machine Learning Everyone should know*. En ligne https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/, consulté le 14 Aout 2019.

Dasylva Abel. (2014). *Améliorations de l'algorithme de maximisation de l'espérance pour le couplage probabiliste*.

Dataquest. (2016). *Working with SQLite Databases using Python and Pandas*. En ligne https://www.dataquest.io/blog/python-pandas-databases/, consulté le 14 Aout 2019.

De Bruin Jonathan. (2015). *Probabilistic record linkage with the Fellegi and Sunter framework*.

De Bruin Jonathan. (2019). *Record Linkage Toolkit Documentation*.

Ensae_teaching_cs. (2019). *Bien démarrer un projet de machine learning*. En ligne https://www.xavierdupre.fr/app/ensae_teaching_cs/helpsphinx/debutermlprojet.html, consulté le 14 Aout 2019.

Grus Joel (2015). *Data Science par la pratique. Fondamentaux avec Python.* Paris : Eyrolles.

Geoawesomeness. (2015). *Why would you use OpenStreetmap if there is Google Maps?* En ligne https://geoawesomeness.com/why-would-you-use-openstreetmap-if-there-is-google-maps/, consulté le 14 Aout 2019.

HDX. En ligne https://data.humdata.org/, consulté le 14 Aout 2019.

Holness Paul. (2014). *Linkage methodology report. Linking canadian patent records from the U.S patent office to statistics canada's business register, 2000 to 2011.*

Machine Learning Mastery. (2018). *How and When to use ROC Curves and Precision-Recall Curves for Classification in Python.* En ligne https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/, consulté le 14 Aout 2019.

Medium. (2018). *How to build a machine learning powered record linkage workflow*. En ligne https://medium.com/@louis_amon/how-to-build-a-machine-learning-powered-record-linkage-workflow-b1890a0eb4ae, consulté le 14 Aout 2019.

Places API. (2019). *Places Search*. En ligne https://developers.google.com/places/web-service/search, consulté le 14 Aout 2019.

Places API. (2019). *Place Details.* En ligne https://developers.google.com/places/web-service/details, consulté le 14 Aout 2019.

Places API. (2019). *Get an API key.* En ligne https://developers.google.com/places/web-service/get-api-key, consulté le 14 Aout 2019.

Pandas 0.24.2 Documentation. (2019). *10 minutes to pandas.* En ligne https://pandas.pydata.org/pandas-docs/stable/getting_started/10min.html, consulté le 14 Aout 2019

Python Machine Learning Tutorial. *Unsupervised Learning: Clustering: Gaussian Mixture Models (GMM)*. En ligne https://www.python-course.eu/expectation_maximization_and_gaussian_mixture_models.php, consulté le 14 Aout 2019.

Raschka Sebastian. (2015). *Python Machine Learning 1st Edition*

Read the Docs. (2019). *Record linkage Toolkit Documentation*. En ligne https://buildmedia.readthedocs.org/media/pdf/recordlinkage/latest/recordlinkage.pdf, consulté le 14 Aout 2019.

Annexes

Annexe 1 : Dictionnaire des données – Google Places API

	Variables	Description
1	Nom	Nom de l'entreprise
2	Latitude	Latitude
3	Longitude	Longitude
4	Adresse	Adresse
5	Code postal	Code postal
6	Numéro de téléphone	Numéro de téléphone
7	Site internet	Site internet
8	Activité	Activités de l'entreprise
9	Place_id	Chaine de caractères qui identifie de manière unique une entreprise.
10	Code composé	Chaine de caractères comprenant un code local et une localisation explicite.
11	Code global	Chaine de caractères comprenant un code de la zone et un code local.
12	Evaluation	Notation de l'entreprise, comprise entre 1.0 et 5.0
13	Niveau de prix	Niveau de prix sur une échelle de 0 à 4. 0. Gratuit 1. Bon marché 2. Normal 3. Cher 4. Très Cher
14	Etage	Hauteur / étage du batiment
15	Numéro de rue	Numéro de rue
16	Route	Nom de la route
17	Localité	Niveau administratif correspondant à la ville
18	Niveau administratif	Niveau administratif correspondant à la province
19	Pays	Pays
20	Adresse formatée	Adresse complète
21	Coordinate_lat	Latitude (paramètre de notre requête)
22	Coord_lng	Longitude (paramètre de notre requête)
23	Search_radius	Rayon de recherche (paramètre de notre requête)
24	Coordinate_id	Identifie de manière unique une requête.
25	Fermé définitivement	Booléen indiquant si l'entreprise est fermée définitivement.

Annexe 2 : Dictionnaire des données – OpenStreetmap

	Variables	Description
1	Nom	Nom de l'entreprise
2	Latitude	Latitude
3	Longitude	Longitude
4	Commodités	Commodités (Restaurant, bar, banque…)
5	Artificiel	Artificiel (eaux usées, réservoir de stockage…)
6	Magasin	Magasin (Vêtements, Sports, alcool…)
7	Tourisme	Tourisme (Hôtel, attraction…)
8	Horaires d'ouverture	Horaires de l'entreprise
9	Nombre de lits	Nombre de lits. Entier
10	Nombre de chambres	Nombre de chambres. Entier
11	Adresse complète	Adresse complète.
12	Adresse maison	Numéro du bâtiment ou nom de rue
13	Adresse Rue	Nom de rue
14	Adresse Ville	Nom de la ville

Résumés et mots clés

RESUME

Le coût croissant des enquêtes traditionnelles, associé à la baisse des taux de réponse, pousse de nombreux offices nationaux de statistique (ONS) à rechercher des méthodes d'acquisition de données alternatives. Le projet régional d'avancement de la statistique dans les Caraïbes (PRASC) s'emploie à renforcer les capacités de 14 pays des Caraïbes.
Le projet a commencé par l'acquisition d'informations sur la localisation et l'activité des entreprises et des lieux d'intérêt des Caraïbes. L'application développée recueille des informations telles que : le nom de l'entreprise, la latitude, la longitude, l'adresse, le numéro de téléphone, le site internet et le type d'activité. Le prototype intègre deux sources de données : Google Places API et OpenStreetmap. Les données d'OpenStreetmap ont été ajoutées afin d'améliorer la couverture globale, de fournir des mises à jour en temps voulu et le tout à moindre coût.

Mots clés : Couplage de données, Base de données géospatiale, Classification, Python, QGIS

ABSTRACT

The rising cost of traditional surveys coupled with the declining response rates force many National Statistical Offices (NSOs) to look for alternative data acquisition methods. The Project for the Regional Advancement of Statistics in the Caribbean (PRASC) is engaged in capacity development in 14 Caribbean countries.
The initiative began with the acquisition of location and activity information for businesses and places of interest in the Caribbean. The application gathers information on the key variables of interest, namely: place name, latitude, longitude, address, phone number, website and type of activity with each entity. The prototype integrates two data sources: Google Places API and OpenStreetmap. OpenStreetmap data was subsequently added to help increase the overall coverage, provide timely updates at less cost.

Keywords: Record linkage, Geospatial database, Classification, Python, QGIS