**HEC MONTRÉAL**

**Analyse de la sévérité moyenne liée aux coûts de pièces d’auto pour une compagnie d’assurance**

**par**

**Fils-Harry Michel**

**Sciences de la gestion**

**(Sciences de données et analytique d’affaires)**

*Rapport de projet supervisé présenté en vue de l’obtention*

*du grade de maîtrise ès sciences*

*(M. Sc.)*

Denis Larocque

HEC Montréal

Directeur de projet

Décembre 2023

© Fils-Harry Michel, 2023

# Résumé

Dans le rapport de projet supervisé qui suit, il sera principalement question de tenter de modéliser, de prédire et d’inférer sur la sévérité moyenne encourue pour une compagnie d’assurances offrant des produits d’assurance auto, dans un contexte de pandémie et hors pandémie. Le premier objectif ici est d’explorer les possibilités qu’un modèle multivarié pourrait offrir quant à la vision globale sur la sévérité totale encourue. Le deuxième objectif est de monter une base de données structurées et des requêtes qui permettront de soutenir d’autres analyses futures.

Pour ce faire, l’analyse se portera principalement sur les informations historiques recueillies au niveau des pièces d’auto à réparer sur les 6 dernières années. Nous chercherons tout d’abord à rassembler différentes variables explicatives qui nous permettrons possiblement d’inférer sur la sévérité totale. Nous ferons donc une étude sur la corrélation entre ces variables.

La sévérité totale est la résultante de la somme des sévérités individuelles de chaque réclamation. Nous définirons ici la sévérité individuelle comme étant le coût encouru pour une unité d’observation, et l’exposition totale par la fréquence observée.

Dans un premier temps, afin de mieux comprendre la sévérité unitaire observée, nous ferons appel à des modèles statistiques tels que les modèles linéaires généralisés gaussiens, ‘’GLM’’ et des modèles d’ensemble tels que les modèles d’arbres de régression.

Dans un deuxième temps, afin de modéliser et d’inférer sur la fréquence, nous tenterons de modéliser la prévision de la demande à l’aide de modèles incorporant des notions de séries temporelles.

D’autres méthodes statistiques seront aussi utilisées dans le processus de la création de notre base de données. En effet, les bases de données sources étant générées par des entrées manuelles, certaines de ces entrées comportent des erreurs, ou ne sont pas dans la même langue. Nous aurons donc recours, par exemple, à des méthodes d’encodage comme du Word2vec afin d’encoder le nom des pièces, et pouvoir par la suite mesurer la similarité entre les noms similaires, et finalement les regrouper. Nous ferons aussi appel à toutes sortes de manipulations telles que des jointures, afin de joindre différentes sources de données, nous créerons des fonctions utiles pour les manipuler.

Nous utiliserons aussi deux langages de programmation dus aux contraintes et limitations de l'entreprise; soit du python pour l’obtention de données externes et l’encodage et du R pour l’obtention des bases de données internes et leur manipulation. Nous devrons alterner d’un langage à un autre en fonction des contraintes et des besoins.

# Table des matières

[Résumé 1](#_Toc1624075415)

[Table des matières 3](#_Toc1078943814)

[Liste des tableaux 5](#_Toc1699306402)

[Liste des figures 5](#_Toc1668710299)

[Liste des abréviations 8](#_Toc513049290)

[Avant-Propos 9](#_Toc1901199574)

[Remerciements 11](#_Toc1459197616)

[Chapitre 1 : Introduction 12](#_Toc1627491719)

[1.1. Mise en contexte 13](#_Toc1554147734)

[1.2. Problèmes et enjeux motivant 15](#_Toc340936345)

[1.3. Objectifs 15](#_Toc1256436772)

[Chapitre 2 : Revue de littérature 16](#_Toc1384330570)

[2.1. Introduction à l’assurance IARD et au concept de la sévérité 16](#_Toc203786252)

[2.2. Représentation vectorielle de variables catégorielles 18](#_Toc1585551927)

[2.2.1. Représentation naïve des variables catégorielles. 18](#_Toc1573920762)

[2.2.2. La vectorisation de mots 20](#_Toc156689203)

[2.3. Regroupement pour des variables à grande dimensionnalité catégorielle 24](#_Toc155559145)

[2.3.1. Méthode de mesure de distance dans un contexte de regroupement de sujets 24](#_Toc436016543)

[2.3.2. Algorithme de regroupement sans nombre de classes prédéterminé 25](#_Toc910690114)

[2.4. Le critère de performance RMSE 28](#_Toc72259612)

[2.5. Le sur-apprentissage, ‘’overfitting’’ 29](#_Toc646924484)

[2.6. Sélection de variables 29](#_Toc1288342264)

[2.7. Modélisation de la sévérité totale en assurance 30](#_Toc931367400)

[2.7.1. Les modèles linéaires généralisés (GLM’s) 30](#_Toc1533769690)

[2.7.2. Les méthodes d’arbres 33](#_Toc802274500)

[2.7.3 Les séries temporelles 39](#_Toc1413293499)

[Chapitre 3 : Méthodologie 43](#_Toc347123972)

[Chapitre 4 : Traitement du jeu de données 44](#_Toc2090251027)

[4.1. Source du jeu de données. 45](#_Toc467286214)

[4.2. Implémentation de fonctions et type de manipulation. 45](#_Toc1672720113)

[4.2.1. Traitement des noms des pièces 46](#_Toc139584415)

[4.2.2 Traitement des données en provenance de sites du gouvernement du Canada 47](#_Toc208629515)

[4.2.3 Traitement des données sur les assurés 48](#_Toc2081299916)

[4.2.4 Traitement des données sur les estimés 51](#_Toc884646078)

[4.2.5 Encodage des données 54](#_Toc370960629)

[4.2.6 Étude de la corrélation inter-variable 54](#_Toc2042903132)

[Chapitre 5 : Analyse des résultats 56](#_Toc26420663)

[Chapitre 6 : Conclusion et discussion 70](#_Toc277098461)

[Bibliographie 75](#_Toc1897013600)

[Annexe 80](#_Toc1012814169)

# Liste des tableaux

Tableau 5.1........................................................................................................57

# Liste des figures

**Aucune entrée de table d'illustration n'a été trouvée.**

Figure 2.1 : Formule de la sévérité totale en actuariat.......................................18

Figure 2.2 : Encodage de mots selon la méthode Bag of Words.......................19

Figure 2.3 : Représentation d’un auto-encodeur.................................................21

Figure 2.4 : Représentation d’un modèle Continuous Bag of Words

(CBOW)…..........................................................................................................25

Figure 2.4 : Représentation d’un modèle Continuous Bag of Words

(CBOW)…..........................................................................................................21

Figure 2.5 : Deuxième représentation du modèle Continuous Bag of Words

(CBOW)…..........................................................................................................22

Figure 2.6 : Représentation vectorielle du mot chat par le modèle word2vec de la librairie Gensim...................................................................................................23

Figure 2.7 : Formule Mathématique du calcul de la distance euclidienne............24

Figure *2.8 :* Calcul de la distance haversienne.....................................................25

Figure 2.9 : Illustration des différents liens inter-clusters.....................................26

Figure 2.10 : Formule mathématique du calcul du RMSE....................................29

Figure 2.11 : Illustration du compromis biais-variance.........................................29

Figure 2.12 : Formules Mathématiques des critères du AIC et du BIC.................30

Figure 2.13: Formule générale abrégée du modèle linéaire généralisé...............31

Figure 2.14 : Formule générale explicite du modèle linéaire généralisé..............31

Figure 2.15 : Équation de la densité de la famille exponentielle...........................32

Figure 2.16 : Tableau sommaire des paramètres de distributions de la famille exponentielle.......................................................................................................32

Figure 2.17 : Équation de la fonction conjointe des probabilités..........................33

Figure 2.18 : Équation de la fonction log-vraisemblance sous forme

additive...............................................................................................................33

Figure 2.19 : Équation de maximisation de la log-vraisemblance........................33

Figure 2.20 : Prédiction du salaire en fonction de l’âge et du sexe par un arbre de régression...........................................................................................................34

Figure 2.21 Équation du critère de partitionnement chez les arbres de

régressions.........................................................................................................35

Figure 2.22 : Formule du critère de pénalisation coût-complexité........................36

Figure 2.23 : Algorithme d’Adaboost...................................................................37

Figure 2.24 : Algorithme du boosting avec perte quadratique..............................38

Figure 2.25 : Algorithme du gradient boosting.....................................................39

Figure 2.26 : Modélisation de l’espérance de Yt..................................................40

Figure 2.27 : Modélisation d’erreurs autorégressives..........................................40

Figure 2.28 : Fonction d’autocorrélation échantillonalle......................................41

Figure 2.29 : Fonction d’autocovariance échantillonalle......................................41

Figure 2.30 : Modélisation récursive de ŷt+1.........................................................42

Figure 2.31 : Représentation du modèle AR pour les erreurs..............................42

Figure 2.32 : Modélisation de l’espérance de Yt avec erreurs

autorégressives …..............................................................................................42

Figure 2.33 Réécriture des composantes du modèle linéaire avec erreurs autorégressives..................................................................................................43

Figure 4.1 : Graphique de la distribution des prix des pièces de

véhicules.............................................................................................................47

Figure 4.2 : Distribution des âges de l’échantillon observé..................................48

Figure 4.3 :  Distribution du sexe dans l’échantillon observée.............................52

Figure 4.4 :  Distribution du statut matrimonial dans l’échantillon

Observé..............................................................................................................50

Figure 4.5 :  Distribution du type de combustible pour voiture dans l’échantillon observé...............................................................................................................52

Figure 4.6 :  Distribution du type de véhicule conduit dans l’échantillon

observé ..............................................................................................................54

Figure 4.7 :  Distribution du type de garage d’où proviennent les

estimés ..............................................................................................................57Figure 4.8 : Graphique de la corrélation entre paires de variables dans le jeu de données..............................................................................................................59

Figure 5.1 : Autocorrélation des Résidus pour le Modèle GLM gaussien pour la prédiction de la sévérité moyenne de la Pièce 6 en Ontario.................................58

Figure 5.2 : Autocorrélation des Résidus pour le Modèle GLM gaussien pour la prédiction de la sévérité moyenne de la Pièce 6 au Québec................................58

Figure 5.3 : Prévisions de la sévérité moyenne hebdomadaire pour la pièce 6 en Ontario par le modèle GLM Gaussien.................................................................59

Figure 5.4 : Prévisions de la sévérité moyenne hebdomadaire pour la pièce 6 au Québec par le modèle GLM Gaussien...............................................................59

Figure 5.5 : Autocorrélation des Résidus pour le Modèle XGB de fonction de pertes quadratique pour la prédiction de la sévérité moyenne de la Pièce 6 en Ontario................................................................................................................60

Figure 5.6 : Autocorrélation des Résidus pour le Modèle XGB de fonction de pertes quadratique pour la prédiction de la sévérité moyenne de la Pièce 6 au Québec...............................................................................................................61

Figure 5.7 : Prévisions de la sévérité moyenne hebdomadaire pour la pièce 6 en Ontario par le modèle XGB avec fonction de perte quadratique........................62

Figure 5.8 : Prévisions de la sévérité moyenne hebdomadaire pour la pièce 6 au Québec par le modèle XGB avec fonction de perte quadratique.......................62

Figure 5.9 : Autocorrélation des Résidus pour le Modèle GLM Poisson pour la prédiction de la demande totale pour la pièce 6 en Ontario...............................63

Figure 5.10 : Autocorrélation des Résidus pour le Modèle GLM Poisson pour la prédiction de la demande totale pour la pièce 6 au Québec..............................63

Figure 5.11 : Prévisions de la demande totale hebdomadaire pour la pièce 6 en Ontario par le modèle GLM Poisson...................................................................64

Figure 5.12 : Prévisions de la demande totale hebdomadaire pour la pièce 6 au Québec par le modèle GLM Poisson..................................................................64

Figure 5.13 : Autocorrélation des Résidus pour le Modèle XGB de fonction de perte Poisson pour la prédiction de la demande totale pour la pièce 6 au Québec...............................................................................................................66

Figure 5.14 : Autocorrélation des Résidus pour le Modèle XGB de fonction de perte Poisson pour la prédiction de la demande totale pour la pièce 6 au Québec...............................................................................................................66

Figure 5.15 : Prévisions de la demande totale pour la pièce 6 en Ontario par le modèle XGB Poisson..........................................................................................67

Figure 5.16 : Prévisions de la demande totale pour la pièce 6 au Québec par le modèle XGB Poisson..........................................................................................67

Figure 5.17 : Niveau d'importances accordées par le modèle XGB à fonction de perte quadratique aux variables d’entrées dans le cadre de la prédiction de la sévérité moyenne hebdomadaire de la pièce 6 en Ontario................................68

Figure 5.18 : Niveau d'importances accordées par le modèle XGB à fonction de perte quadratique aux variables d’entrées dans le cadre de la prédiction de la sévérité moyenne hebdomadaire de la pièce 6 au Québec...............................69

Figure 5.19 : Niveau d'importances accordées par le modèle GLM Poisson aux variables d’entrées dans le cadre de la prédiction de la demande totale hebdomadaire pour la pièce 6 en Ontario..........................................................70

Figure 5.20 : Niveau d'importances accordées par le modèle GLM Poisson aux variables d’entrées dans le cadre de la prédiction de la demande totale hebdomadaire pour la pièce 6 au Québec..........................................................70

# Liste des abréviations

IARD…………………………………………Incendies, Accidents et Risques Divers

BOW…………………………………………………........Bag of words (Sac de mots)

CBOW........................................Continuous Bag of words (Sac de mots continue)

RMSE……………………..Root Mean SquareError (Erreur Quadratique Moyenne)

AIC…………………...Akaike Information Criterion (Critère d’information d’Akaike)

BIC……………….Bayesian Information Criterion (Critère d’Information Bayésien)

GLM………………………………………………………. Modèle linéaire généralisé

GLMP...............................................................Modèle linéaire généralisé Poisson

XGB………………………………………………………..eXtreme gradient Boosting

AR………………………………………………………………...Modèle Autorégressif

MLE……………………………………...Estimateur de maximum de vraisemblance

# Avant-Propos

Dans le cadre de ma maîtrise en sciences de données et analytique d’affaires à l’université du HEC Montréal, j’ai été amené à effectuer un projet de fin d’études portant sur certaines notions acquises tout au long de mon parcours de maîtrise. C’est dans ce contexte que j’ai eu l’opportunité d’effectuer ce projet supervisé avec la compagnie TD Assurance et sous la tutelle de l’ancien Directeur de mon programme de maîtrise et aussi professeur, Denis Larocque. Durant les dernières années, nous avons eu un événement qui a touché une majeure partie de nos activités et notre façon de fonctionner au jour le jour. Cet évènement est la pandémie liée au virus de la COVID-19. Afin d’assurer le bien-être de l’ensemble de la population et son bon fonctionnement, nos gouvernements ont dû adopter des mesures d’urgence passant du couvre-feu à l’encouragement à la distanciation sociale et au travail à distance. C’est dans ce contexte que l’on a pu observer une baisse de la circulation dans les différentes villes du pays pour éviter la propagation du virus. Beaucoup de compagnies ont commencé à encourager leurs employés à opter pour du télétravail lorsque cela était possible au lieu de travailler en présentiel. Il y a aussi eu des périodes de couvre-feu ou les gens étaient fortement encouragés à rester chez eux s’il n’avait pas de motifs valables pour sortir. Cela a pu notamment se traduire par une diminution du nombre de conducteurs sur nos routes et une baisse du trafic urbain dans les grandes villes (Statistiques Canada, 2022). La compagnie d’assurance avec laquelle j’ai eu l’opportunité de travailler offre des produits d’assurances auto. Celle-ci a notamment pu remarquer une différence au niveau des réclamations et donc aussi de la sévérité moyenne pour les années de pandémie en comparaison aux autres années. Pour mieux comprendre le phénomène, on a décidé de commencer une analyse exploratoire, afin de voir s’il serait possible d’utiliser une vision multivariée incorporant de possibles interactions entre divers facteurs pour modéliser des tendances observées au niveau de la sévérité encourue. J’ai donc reçu le mandat de modéliser, de prédire et d’inférer sur différents facteurs pouvant avoir un impact sur la tendance de la sévérité moyenne observée et sur la fréquence. Le défi ici était de trouver une échelle de comparaison valable pour tous les accidents. En effet, il est normal que la sévérité moyenne soit plus élevée pour un véhicule qui, par exemple, ne peut être conduit à la suite d’un accident et un véhicule qui peut être conduit. Afin de remédier à ce problème, nous avons décidé de mener une analyse beaucoup plus granulaire en nous focalisant sur des pièces similaires ayant eu à subir des interventions similaires. Pour ce faire, j’ai décidé de porter mon analyse sur les pièces d’auto sollicitées pour des réparations similaires. L’objectif ici, étant tout d’abord, de structurer une base de données non structurées et d’y rattacher des informations pouvant être pertinentes afin d’y mener une analyse et de finalement modéliser la tendance de la sévérité à l’aide des informations obtenues sur les différentes pièces d’auto et les contextes entourant la réclamation. Ce projet permettra donc à l’entreprise d’amorcer une vision par approche multivariée au lieu d’une vision par approche univariée et de pouvoir comparer ces deux approches. Ce projet permettra aussi de soutenir d’autres équipes en ce qui concerne l’établissement d’une base de données plus structurée qui pourrait possiblement être utilisé dans d’autres analyses et, finalement, d’avoir possiblement un impact sur le processus décisionnel, comme la détermination des réserves potentielles pour de futures réclamations au sein de l’entreprise.

# Remerciements

Mes remerciements vont tout d’abord à TD Assurance qui a accepté que je puisse faire mon projet supervisé avec eux. Entre autres, à mon superviseur immédiat, Marcel Proulx pour sa confiance.

Je profite aussi pour remercier tous les collègues, qui m’ont aidé à remédier à des problèmes liés à de l'information nouvelle pour moi, aux limitations computationnelles et autres.

Je tiens d’autre part à remercier Denis Larocque qui n’a pas hésité une seconde à me prendre sous sa tutelle pour ce projet supervisé. Il s’est montré aussi très patient et très compréhensif en ce qui concerne les enjeux liés au partage de l’information et à la confidentialité des données.

Je tiens finalement à remercier Aurélie Labbe, mon actuelle directrice de programme qui a porté sa voix en ma faveur devant la compagnie avec laquelle je devais faire mon projet supervisé afin de permettre la réalisation de ce projet.

# Chapitre 1 : Introduction

## Mise en contexte

L’année 2020 a marqué un grand tournant dans les esprits de tous les Québécois. Le 27 février 2020, nous recensions un premier cas d’un virus qui nous était jusque-là étranger (INSPQ, 2022). Constatant l’augmentation du nombre de cas de personnes infectées au sein de la population, la santé publique a dû adopter dès le 11 mars 2020 (INSPQ, 2022), une série de mesures afin de contrer la propagation du virus. Passant tout d’abord par l’isolement des voyageurs en provenance de l’étranger à la demande formelle formulée aux vacanciers de revenir de voyage et de rentrer chez eux.

C’est finalement le 15 mars 2020 que l’état d’urgence sanitaire est officiellement déclaré (INSPQ, 2022). Cet évènement est succédé par un ensemble de mesures plus pragmatiques adoptées par la santé publique. Dans ce contexte, on a observé la fermeture de la grande partie des lieux publics et l’adoption de mesures de distanciation physique. S’en est suivie une série de mesures allant jusqu’au confinement et à la limitation des déplacements entre diverses régions jusqu’au 11 juillet 2020, date correspondant à la fin de la première vague (INSPQ, 2022).

S’en est suivi le début de la deuxième vague, où des mesures beaucoup plus musclées seront adoptées pour freiner la pandémie, telles que l’interdiction des rassemblements intérieurs, le retour du confinement, la fermeture des commerces non essentiels, et l’instauration d’un premier couvre-feu obligatoire, et ce, jusqu’au 21 mars 2021 (INSPQ, 2022). Ce qui a marqué la fin de la deuxième vague et le début de la troisième vague notamment marquée par l’apparition et le durcissement de certaines mesures, tel que le couvre-feu par exemple, qui sera devancé à 20h au lieu de 21h30 (INSPQ, 2022). Cette troisième vague prendra finalement fin en juillet 2021 (INSPQ, 2022).

S’en sont suivies une quatrième, une cinquième et une sixième vague, durant lesquelles des ensembles de mesures seront adoptées et levées progressivement. Finalement, le reste de la levée progressive de l’ensemble des mesures sanitaires restantes se fera jusqu’en octobre 2022 (INSPQ, 2022). Les québécois et québécoises pourront finalement commencer à reprendre leurs activités habituelles et circuler plus librement, tout en respectant certaines mesures de base.

C’est dans ce contexte que nous avons pu observer durant cette période une diminution de la circulation de la population en général. N’ayant pas accès à divers lieux publics, étant encouragés à travailler autant que possible à domicile ou étant restreints dans les rassemblements, les déplacements, et les voyages, l’impact s’est fait ressentir dans différents secteurs. Par exemple, selon statistiques Canada, on a pu observer une diminution du nombre de ‘’*navetteurs*’’, personne se déplaçant en voiture, d’au moins 1,7 millions au sein de la population canadienne (Statistiques Canada, 2022). Cela s’est aussi reflété par une baisse du nombre d’accidents sur la route.

Pour les compagnies d’assurance, cela s’est traduit par une baisse de la fréquence des réclamations et ainsi donc une baisse de la sévérité totale encourue.

Afin de maintenir une relation de confiance avec leurs clients et de redistribuer les bénéfices faits durant cette période avec les primes, plusieurs compagnies d’assurance ont décidé de redonner à leurs assurés des ristournes, ou leur ont offert la possibilité d’ajuster leurs primes d’assurance (Charles Mathieu, 2020).

Pour définir les différentes alternatives et options possibles à offrir à leurs clients, il a été important pour ces compagnies de mener des études afin de pouvoir déterminer, voire estimer, les réserves nécessaires à conserver en cas d’éventuelle reprise et retour à la normale.

## Problèmes et enjeux motivant

Dans ce contexte, malgré la diminution de la fréquence des réclamations observées sur l’ensemble de cette période et ainsi donc de la diminution de la sévérité totale encourue observée, la compagnie avec laquelle j’aie eu la chance de mener cette analyse a remarqué une augmentation de la sévérité moyenne encourue depuis les 5 dernières années.

En d’autres mots, le traitement d’une réclamation donnée coûterait de plus en plus cher d’année en année. Cela peut notamment s’expliquer par plusieurs facteurs tels que la hausse de la rémunération globale des employés devant traiter la réclamation, incluant les employés internes et externes comme les garagistes. Le type d’accident ou de véhicule impliqué dans l’accident, ou bien même, la hausse des prix des pièces.

C’est dans ce contexte qu’une étude préliminaire avait été menée avec ma participation afin d’avoir un début d’aperçu des facteurs ayant possiblement le plus d’impact sur la sévérité moyenne observée. Dans cette étude, il a été question de comparer deux provinces, soit l’Ontario et le Québec sur les périodes de 2018 à 2022. Nous avons ensuite essayé de segmenter les sévérités moyennes observées en fonction de leurs différents types de dépenses (paie des employés, paiement de la main d’œuvre, paiement des pièces, paiement pour location de voitures de courtoisies, etc.) afin de mieux comprendre ce qui semblait impacté cette tendance observée au niveau de la sévérité moyenne. À la suite de cette étude, nous avons pu élaborer un tableau de bord nous permettant de dégager la principale source de dépense, soit les coûts liés aux pièces d’auto.

## Objectifs

Afin de pouvoir de développer une meilleure compréhension de la sévérité et d’adopter une vision beaucoup plus granulaire de l’ensemble notre problème, nous avons eu l’idée de développer une analyse et des outils nous permettant de mieux comprendre la sévérité moyenne et la demande totale liée aux pièces d’auto.

Plusieurs questions se posaient alors, quelles sont les pièces d’auto ayant le plus d’impact sur notre sévérité observée? Peut-on en prédire la demande et le coût ? Sur un ordre plus général, ces questions avaient pour but de mieux prévoir les futures dépenses de la compagnie et par le fait même, les réserves à mettre de côté afin de traiter les futures réclamations.

L’analyse de la sévérité au niveau des pièces d’auto pour une compagnie d’assurance auto s’est toujours avérée être une tâche ardue. Tout d’abord, en fonction d’un fournisseur à un autre, les noms attribués à une même pièce peuvent varier. Certains noms de pièces peuvent tout aussi être entrées dans les bases de données en français, ou en anglais par exemple. Puisque ce sont des entrées manuelles, certaines entrées de noms de pièces comportent des erreurs. Le prix unitaire d’une pièce peut varier en fonction de la marque et du modèle de véhicule, ce qui amène un enjeu de quantité d’informations. Une grande partie de ce travail a donc portée sur la création de la table de données utilisée pour notre analyse.

De plus, afin de de mieux comprendre la sévérité moyenne observée, nous tenterons d’ajuster deux différentes familles de modèle statistique pour modéliser la sévérité et modéliser la fréquence observée pour certaines pièces d’auto. Finalement, nous comparerons les performances des deux modèles. Nous conclurons ce rapport en revenant sur les nombreux enjeux rencontrés lors de ce projet, et les éléments intéressants qui auraient pu y être apportés afin d’améliorer les résultats obtenus.

# Chapitre 2 : Revue de littérature

## 2.1. Introduction à l’assurance IARD et au concept de la sévérité

L’acronyme I.A.R.D est une abréviation que l’on retrouve dans le domaine de l’assurance signifiant : « Incendies, Accidents et Risques Divers » (Jean-Philippe Boucher, 2019).

En général, une manière simple de voir l’assurance est de le concevoir comme un transfert de risque entre deux parties (Jean-Philippe Boucher, 2019). Dans notre contexte plus précisément, une tierce personne paiera à une compagnie une prime afin que celle-ci se charge de couvrir \* son risque. Dans le contexte présent on parle de payer en cas de réclamation.

Lorsqu'une compagnie d’assurance accepte de couvrir un risque, son principal défi devient de calculer le montant qu’il charge à l’assuré en échange de ce transfert de risque. Ce montant qui est chargé à l’assuré est appelé une prime d’assurance. Celle-ci peut différer d’un assuré à un autre en fonction du profil de l’assuré et des termes du contrat d'assurance (Jean-Philippe Boucher, 2019).

Pour cela, dans un premier temps, il est donc important que le montant d’une réclamation potentielle soit mesurable, voire estimable. C’est ce que nous qualifierons ici de sévérité. D’autre part, il faut aussi que la prime chargée par la compagnie pour la couverture du risque soit justifiable et raisonnable (Jean-Philippe Boucher, 2019).

C’est dans ce contexte qu’il devient impératif pour une compagnie d’assurance de tout d’abord assurer la profitabilité de l’entreprise tout en cherchant à attirer le plus de clients possibles.

Afin d’estimer la prime juste à charger à un assuré, il nous faut d’abord avoir une idée juste du potentiel risque que celui-ci représente. Donc en d’autres termes, une estimation juste de la valeur d’un sinistre potentiel mène à un calcul beaucoup plus juste de la prime d’un assuré (Jean-Philippe Boucher, 2019). Car la prime est fonction du risque. Il est donc important d’essayer d’obtenir une estimation de ce que nous avons qualifié comme étant la sévérité plus tôt.

La sévérité totale en actuariat se définit comme suit :

*Figure 2.1 : Formule de la sévérité totale en actuariat*

Une image contenant texte, blanc, Police, écriture manuscrite

Description générée automatiquement

Source : Nugent, Ugo [2019] ACT5400 : Crédibilité UQAM, 96 p.

La sévérité totale a donc deux composantes nécessaires à son estimation. D’une part, les coûts des sinistres individuels et d’autre part, le nombre total de sinistres.

On a donc dans la formule présentée ci-dessus S étant une variable aléatoire composée, représentant ici la Sévérité totale observée d’un portefeuille d’assurés.

Les Xi pour i=1, … N, sont ici, des variables aléatoires représentant le coût individuel d’un sinistre donné.

N est ici aussi une variable aléatoire représentant le nombre total de sinistres, autrement dit, la fréquence.

À mentionner que nous supposons que les Xi sont indépendants et identiquement distribués.

## 2.2. Représentation vectorielle de variables catégorielles

### 2.2.1. Représentation naïve des variables catégorielles.

Dans le cadre de ce projet, certaines des variables utilisées étaient numériques et d'autres catégorielles. Il est cependant souvent nécessaire de trouver une représentation numérique à nos variables catégorielles afin de pouvoir utiliser certains modèles et obtenir certaines statistiques descriptives.

Une approche naïve serait : ‘’Bag of Words’’, ou bien ‘’BOW’’. Cette approche consiste à compter le nombre d’apparition dans une phrase donnée de chaque mot distinct présent dans l’ensemble du texte (Nada Belaidi, 2022).

*Figure 2.2 : Encodage de mots selon la méthode Bag of Words*

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Source : https://blent.ai/blog/a/word-embedding-nlp-definition

Par exemple, la phrase [‘’cat’’, ‘’eats’’, ‘’dog’’] est encodée ici de la manière suivante : [0, 0, 1, 1, 1, 0]. Si la phrase avait été [‘’cat’’, ‘’eats’’, ‘’dog’’, ‘’cat’’], la représentation vectorielle aurait été [0, 0, 1, 2, 1, 0].

Une autre forme de représentation similaire qui sera présenté dans ce rapport est le ‘’One-Hot Encoding’’, qui part du même principe, mais au lieu de compter l'occurrence de l’apparition d’un mot, cette méthode ne fait qu’indiquer la présence ou non de ce mot par les valeurs 1 ou 0 respectivement (Nada Belaidi, 2022). Nous obtiendrions donc à la phrase suivante : [‘’cat’’, ‘’eats’’, ‘’dog’’, ‘’cat’’], la représentation vectorielle suivante : [0, 0, 1, 1, 1, 0].

Cependant lorsque le corpus de texte est très grand ou que le nombre de catégories au sein d’un vecteur est très grand, les vecteurs de représentations deviennent eux aussi très grands.

### 2.2.2. La vectorisation de mots

Le ‘’Word Embedding’’, autrement appelé, la ‘’vectorisation de mot’’ ou ‘’le plongement lexical’’ est un processus consistant à représenter chaque mot d’un corpus de texte sous la forme d’un vecteur numériques.

Ce concept est dérivé de la théorie de Zellig Harris portant sur la distribution sémantique (Data Analytics Post, 2023). Cette théorie postule que des mots apparaissant dans des contextes semblables ont aussi des significations apparentées. À l’origine, cette méthode était principalement utilisée pour réduire la dimensionnalité d’un jeu de données.

Une méthode très réputée faisant appel à ce concept est la méthode word2vec développée par une équipe de recherche de google (Data Analytics Post, 2023).

*Figure 2.3 : Représentation d’un auto-encodeur*

Une image contenant diagramme, ligne, Tracé, origami

Description générée automatiquement

Source : https://blent.ai/blog/a/word-embedding-nlp-definition

Deux variantes de cette méthode existent, soit le modèle du ‘’Continuous Bag Of Words’’, CBOW, qui cherche à prédire un mot en fonction des mots avoisinants, et un autre modèle nommé ‘’skip-gram’’, cherchant à partir du mot central, à prédire les mots avoisinants (Nada Belaidi, 2022).

Dans notre cas, nous utiliserons les fonctions de la librairie ‘’Gensim’’ en python. Le modèle par défaut utilisé par cette librairie est le ‘’CBOW’’, se reflétant par l’initiation de l’hyperparamètre sg à 0.

Dans cette méthode on prend un mot en position i et on sélectionne par la suite un certain nombre de voisins à gauche et à droite du mot. Ce sont alors nos éléments de notre vecteur d’entrée représenté par le ‘’input Layer dans l’image qui suit’’ :

*Figure 2.4 : Représentation d’un modèle Continuous Bag of Words (CBOW)*

Une image contenant texte, capture d’écran, ligne, diagramme

Description générée automatiquement

Source : https://blent.ai/blog/a/word-embedding-nlp-definition

Les données en entrée représentées par un vecteur que nous nommerons X, seront alors transformées par une étape de multiplication matricielle entre le vecteur X et une matrice de poids d’entrée que nous nommerons W. Cette opération vise à obtenir une représentation intermédiaire h, qui est la projection du vecteur X sur le vecteur W. C’est ce que l’on appelle l’étape d’encodage (Nada Belaidi, 2022).

*Figure 2.5 : Deuxième représentation du modèle Continuous Bag of Words (CBOW)*

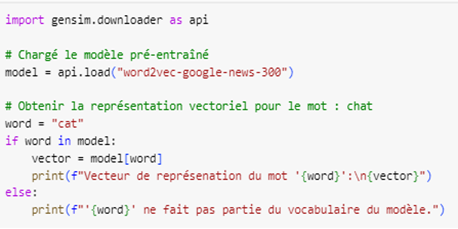
Une image contenant texte, ligne, diagramme, Police

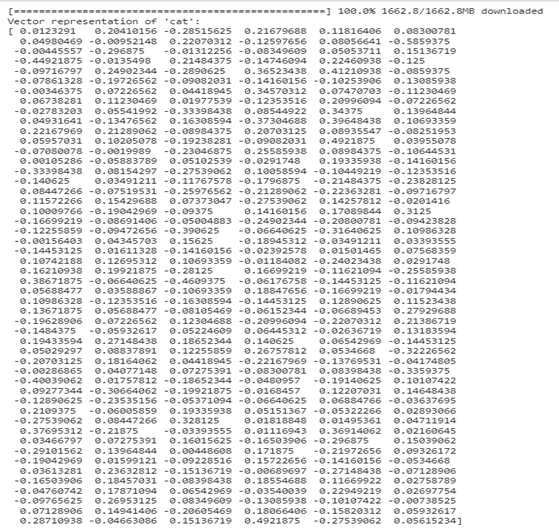
Description générée automatiquement

Source: https://openclassrooms.com/fr/courses/4470541-analysez-vos-donnees-textuelles/4855006-effectuez-des-plongements-de-mots-word-embeddings

Cette méthode nous permet d’obtenir une représentation du mot dans l’espace. Comme une sorte de positionnement cartésien. Par exemple, une représentation plausible du mot chat pourrait être celle du modèle pré-entraîné word2vec de la librairie Gensim :

*Figure 2.6 : Représentation vectorielle du mot chat par le modèle word2vec de la librairie Gensim*





## 2.3. Regroupement pour des variables à grande dimensionnalité catégorielle

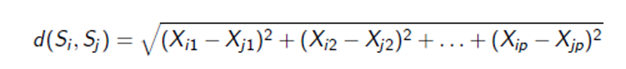
Un des grands enjeux en modélisation reste la gestion de la dimensionnalité de nos données en fonction de la capacité computationnelle disponible. C’est pourquoi, en apprentissage statistique, nous faisons appel à des algorithmes de regroupement autrement appelés méthodes de ‘’clustering’’. Ces méthodes nous permettent d’affecter à des observations ayant des caractéristiques semblables, un groupe. D’autre part, nous cherchons aussi à maximiser la différence entre chaque groupe. En d’autres mots, minimiser les différences intra-groupes, et maximiser les différences inter-groupes.

### 2.3.1. Méthode de mesure de distance dans un contexte de regroupement de sujets

Afin de déterminer les assignations des observations à un groupe, il devient important de définir une mesure de distance entre les différents sujets.

L’une des mesures de distance sur un plan les plus connues et populaires sur un plan est la distance euclidienne. Supposons que nous ayons Si et Sj, deux observations ayant pour représentation vectorielle Si = [Xi1,……….Xip] et Sj = [Xj1,……….Xjp] respectivement. Le calcul de la distance euclidienne entre ces deux observations est alors donné par :

*Figure 2.7 : Formule Mathématique du calcul de la distance euclidienne*



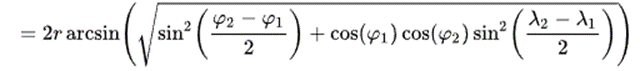
Source : Thioub, M. [2021] 60603 *Apprentissage statistique Analyse de regroupement*, HEC Montréal, 64 p.

Lorsque nous cherchons à déterminer les distances entre deux points caractérisés par une longitude et une latitude se trouvant sur un globe, l’utilisation de la distance euclidienne devient alors obsolète. La distance d’haversienne est une mesure prenant en compte le caractère sphérique de la distance séparant deux points. Celle-ci est composée de 4 composantes :

* Le rayon de la sphère en question
* π afin de ramener le tout en radian
* φ la latitude
* λ la longitude

La distance haversienne est alors calculée comme suit :

*Figure 2.8 : Calcul de la distance haversienne*



Source: <https://fr.wikipedia.org/wiki/Formule_de_haversine#:~:text=La%20formule%20de%20ha> versine,-En%20prenant%20deux&text=d%20est%20la%20distance%20du,du%20point%202%2C%20en%20radians

### 2.3.2. Algorithme de regroupement sans nombre de classes prédéterminé

Une fois une mesure de distance déterminée afin de mesurer les similarités et les dissimilarités entre divers sujets, il devient maintenant important de déterminer le nombre de groupes que l’on désire former. À titre d’exemple, pour 10000 observations, si nous désirons former 100 groupes, notre nombre de catégories passera de 10000 à 100, résultant ainsi en une diminution dans la dimensionnalité de nos données. Le choix du nombre de groupes est crucial, car ce choix influencera la qualité de la représentativité de nos données.

Un trop grand nombre de catégories pourrait occasionner du sur-apprentissage. C’est-à-dire, que notre ensemble de catégorie serait trop semblable à notre ensemble que nous essayons de représenter, plus particulièrement, en termes de dimensionnalité. Il ne serait alors pas intéressant d'utiliser des méthodes de regroupement. D’autre part, si le nombre catégories est trop faible, nous ferons désormais face à du sous-ajustement où notre groupement n’arriverait plus à représenter la variabilité observée dans nos données.

Donc comment déterminer le bon nombre de groupements (clusters) en fonction de notre jeu de données? Certaines méthodes ont été développées, telles que le clustering hiérarchique afin de déterminer le nombre de groupes optimaux à former en fonction de notre jeu de données (Mamadou-Yamar Thioub, 2021).

Cette méthode, comme mentionnée plus haut, a pour avantage de ne pas nécessiter la détermination du nombre de clusters au préalable. En temps normal, une simple analyse graphique du dendrogramme peut nous permettre de déterminer le nombre de regroupement idéal à maintenir (Mamadou-Yamar Thioub, 2021). De plus, à l’instar des k-means, l’interprétation des scissions de classe au niveau des variables catégorielles est plus facile (Mamadou-Yamar Thioub, 2021). Cependant, l’un des inconvénients majeurs que nous pouvons rencontrer arrive lorsque cet algorithme de clustering doit traiter un large jeu de données. En effet, à chaque itération de l’algorithme, les distances deux à deux de chaque regroupement doivent être recalculées. La complexité de cette étape est notre goulot d’étranglement (bottleneck) qui fait en sorte qu’il est beaucoup plus complexe que l’algorithme du k-means qui a été présenté précédemment (Mamadou-Yamar Thioub, 2021). L’algorithme est aussi très sensible au bruit. Plus les données sont distantes dans l’espace, plus l'algorithme est sujet à l’erreur lors de la désignation de regroupement à distance minimum (Mamadou-Yamar Thioub, 2021). À titre informatif, dans la suite de ce rapport, les mots cluster et regroupement seront considérés comme interchangeables.

Afin de déterminer l’assignation de chaque observation à chacun des groupes, on utilise la distance euclidienne afin de déterminer les observations se trouvant à la plus petite distance les unes des autres. Les deux regroupements se trouvant à plus petites distances l’un de l’autre sont alors regroupés en un seul groupe et leurs nouvelles distances inter-groupes deviennent la distance minimale de chaque composante par rapport aux autres composantes des autres groupes. Pour calculer la distance inter-groupe, il existe plusieurs formes de liens, soit le lien simple, le lien ‘’average’, le lien complet, le lien centroïde à centroïde, et le lien en utilisant la méthode de Ward. Ces liens indiquent la manière dont sera calculée la distance inter-cluster. Une illustration des différents liens vous est fournie ci-dessous.

*Figure 2.9 : Illustration des différents liens inter-clusters*

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

Source : https://www.mygreatlearning.com/blog/hierarchical-clustering/

Dans ce projet, nous avons plutôt fait appel à l’algorithme ‘’DBSCAN’’.

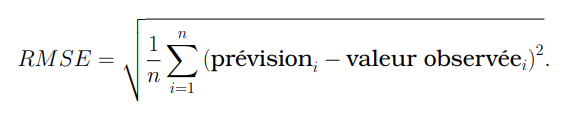
Cet algorithme a été développé par Martin Ester, Hans-Peter Kriegel, Jorg Sander et Xiaowei Xu. Leur travail a été publié dans un article ayant pour titre : ‘’A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise’’ (Esther et al., 1996). Son but est simple, divisé en k groupes homogènes les n observations. L’algorithme se résume comme suit. Pour chaque point, on regarde le nombre de voisins ne se trouvant à pas plus d’une distance que nous qualifierons de d. Chaque distance inter-observations est calculée à partir de la distance euclidienne que nous avons décrite précédemment. Par la suite, si un des points a beaucoup de voisins, il est alors considéré comme étant le point central d’un regroupement. Lorsqu’une observation n’a aucun voisin, on dit de cette observation qu’elle constitue une anomalie (Robert J., 2023).

La question à se poser maintenant est, quelle est donc cette distance d? Cette distance est souvent déterminée par essai-erreur ou par optimisation. En général, on essaie de fixer cette distance de façon que 90% à 95% des points aient au moins un voisin se trouvant à moins d’une distance d (Robert J., 2023).

## 2.4. Le critère de performance RMSE

Afin d’évaluer la performance d’un modèle, on a souvent recours à la méthode du RMSE, “Root Mean Square Error”. Cette méthode consiste à calculer la racine carrée de la moyenne du carré de la différence entre les estimations et leurs vraies valeurs observée par le modèle. Le but étant de chercher à diminuer la somme des erreurs au carré. On cherche ici le modèle qui minimisera la valeur du RMSE (Debbie J. Dupuis, 2020).

*Figure 2.10 : Formule mathématique du calcul du RMSE*



Source : Debbie, J. Dupuis [2020]. *Méthodes de prévision Séance 2 – Idées et outils de base,* HEC Montréal, 32 p.

## 2.5. Le sur-apprentissage, ‘’overfitting’’

Le sur-apprentissage ou erreur de généralisation survient lorsque nous tentons d’ajuster un modèle à nos données et que celui-ci se spécialise sur l’ensemble d’entraînement, et ne parvient pas à performer correctement sur un ensemble test (des données non utilisées pour l’entraînement du modèle). On dit alors que le modèle ne parvient pas à généraliser. Une erreur courante survient lorsque nous tentons d’inclure trop de paramètres à notre modèle. Bien que le biais du modèle tende à diminuer, l’estimation de chaque paramètre occasionne une incertitude supplémentaire faisant ainsi augmenter la variance et par conséquent, l’erreur de généralisation (Laurent Charlin, 2021).

*Figure 2.11 : Illustration du compromis biais-variance*

Une image contenant texte, ligne, diagramme, Tracé

Description générée automatiquement

Source : Charlin, Laurent [2021]. *Apprentissage Automatique I 80-629*HEC, 62 p.

## 2.6. Sélection de variables

Pour réduire l’erreur de généralisation il est donc important de faire une sélection de variables dans notre modèle. Dans le cadre du modèle linéaire, des algorithmes tels que le ‘’stepwise’’ peuvent être utilisés pour la sélection de variables. Le principe est simple, à chaque itération, on ajoute et/ou on enlève des variables du modèle et on calcule la performance du modèle résultant. Cet algorithme repose le plus souvent sur des critères de performances tels que le AIC et le BIC, qui sont souvent utilisés pour pénaliser les modèles en fonction de leur complexité. Voici leurs formules respectives :

*Figure 2.12 : Formules Mathématiques des critères du AIC et du BIC*

Une image contenant Police, texte, blanc, ligne

Description générée automatiquement

Source : Denis Larocque, Aurélie Labbe [2022]. *80619 Méthodes avancées en exploitation de données Introduction,* HEC Montréal, 14 p.

Où, LL(ϴMLE) représente la fonction de log-vraisemblance, ϴMLE estl’estimateur du paramètre de maximum de vraisemblance, p représente le nombre de paramètres et n représente le nombre d’observations dans l’échantillon.

## 2.7. Modélisation de la sévérité totale en assurance

En pratique, dans le domaine de l’assurance, les modèles les plus souvent utilisés et les plus complexes ne dépassent souvent pas la portée des GLM (General Linear Models). Ces modèles sont les plus couramment utilisés, tout d’abord, car ils sont relativement faciles à interpréter. Ils nous permettent de voir plus clairement les effets des variables explicatives sur la variable que nous tentons de prédire, contrairement à d’autres modèles qui sont qualifiés de ‘’boîtes noires’’.

### 2.7.1. Les modèles linéaires généralisés (GLM’s)

Comme expliqué précédemment, Il est possible de décomposer la sévérité totale en assurance en deux composantes indépendantes. La première composante est la sévérité individuelle, et la deuxième composante est la fréquence. Il est possible de modéliser ces deux composantes à partir de différentes familles de loi de GLM. Une forme générale du modèle connue est la suivante :

*Figure 2.13 Formule générale abrégée du modèle linéaire généralisé*

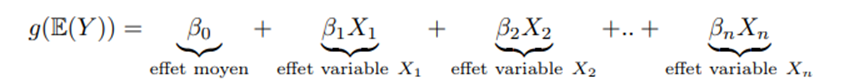
Une image contenant Police, blanc, calligraphie, texte

Description générée automatiquement

Source : <https://archipel.uqam.ca/11765/1/M15665.pdf>

Ou de manière plus explicite,

*Figure 2.14 : Formule générale explicite du modèle linéaire généralisé*



Source : https://www.institutdesactuaires.com/docs/mem/f7ec303ec9df8c23bcb407657d109184.pdf

Où le terme à gauche de l’égalité représente une fonction *g* appliqué à µi, l’espérance de la variable aléatoire Y, qui est la variable d’intérêt. Quant aux termes se trouvant à droite de l’égalité, nous retrouvons une combinaison linéaire des paramètres, également appelés coefficients du modèle (β0, ……..,βn) et leurs variables aléatoires respectives (X1,….., Xn). D’après ce modèle, l’espérance de la prédiction se traduit par la somme des effets individuels des variables aléatoires du modèles.

Avec ce modèle, on suppose que les Yi sont mutuellement indépendants et que leur distribution doit faire partie d’une famille de loi exponentielle (Carlos. F., 2018). Cette famille de loi exponentielle a une densité de la forme :

*Figure 2.15 : Équation de la densité de la famille exponentielle*

Une image contenant Police, texte, blanc, écriture manuscrite

Description générée automatiquement

Source : <https://archipel.uqam.ca/11765/1/M15665.pdf>

Où φ est le paramètre de dispersion, ωi est un poids, et ϴ est le paramètre canonique.

Dans le tableau qui suit, quelques exemples de distribution de la famille de loi exponentielle et leurs paramètres y sont présentés.

*Figure 2.16 : Tableau sommaire des paramètres de distributions de la famille exponentielle*

Une image contenant texte, Police, nombre, reçu

Description générée automatiquement

Source : <https://archipel.uqam.ca/11765/1/M15665.pdf>

Afin d’estimer les valeurs des coefficients, et donc, le modèle en soi, on a souvent recours à la méthode du maximum de vraisemblance. Cette méthode consiste à déterminer les valeurs des coefficients β qui permettent de maximiser la vraisemblance avec l’échantillon des Yi observées. Nous déterminons tout d’abord, l’équation conjointe de probabilité à l’aide de l’équation suivante :

*Figure 2.17 : Équation de la fonction conjointe des probabilités*

Une image contenant Police, typographie, blanc, calligraphie

Description générée automatiquement

Source : <https://archipel.uqam.ca/11765/1/M15665.pdf>

Afin de simplifier les calculs, nous transformons l’équation précédente en une forme additive au lieu de multiplicative en prenant le logarithme des deux côtés de l’équation. Ainsi, nous obtenons la *log-vraisemblance* sous la forme suivante :

*Figure 2.18 : Équation de la fonction log-vraisemblance sous forme additive*



Source : Denis Larocque, Aurélie Labbe [2022]. *80619 Méthodes avancées en exploitation de données Introduction,* HEC Montréal, 14 p.

Finalement, nous obtenons la log-vraisemblance en mettant le dérivé de la partie droite de l’équation égale à 0. Cela revient à résoudre l’équation suivante :

*Figure 2.19 : Équation de maximisation de la log-vraisemblance*

Une image contenant Police, écriture manuscrite, blanc, texte

Description générée automatiquement

Source : https://www.ressources-actuarielles.net/C1256F13006585B2/0/457A36A8ECC541AEC1257D740067EEC4/%24FILE/GLM\_FP.pdf

Dans le domaine de l’assurance IARD, on suppose souvent que la distribution de la fréquence des sinistres suit une loi de poisson, qui est une loi de comptage. En ce qui a trait à la sinistralité individuelle, on suppose souvent qu’elle suit une loi log-normal. Il nous est possible d’utiliser les GLM et leurs familles de loi exponentielle de Poisson et gaussienne pour modéliser ces deux distributions. Il est également à noter qu’en pratique on utilise souvent le lien log afin d’inclure un effet multiplicatif.

### 2.7.2. Les méthodes d’arbres

#### 2.7.2.1. Les arbres

Le second type de modèle de régression que nous aborderons dans ce rapport est le modèle des arbres de régression. Le but de ces types de modèles est de partitionner le domaine respectif des prédicteurs de façon à regrouper tous les sous-ensembles ayant les mêmes caractéristiques dans un même partitionnement de l’espace et ainsi améliorer le pouvoir prédictif de l’arbre (Denis L., 2022). En exemple, voici un arbre de régression présentant le partitionnement permettant de prédire le salaire en fonction de l’âge et du sexe.

*Figure 2.20 : Prédiction du salaire en fonction de l’âge et du sexe par un arbre de régression*

Une image contenant texte, diagramme, ligne, Tracé

Description générée automatiquement

Source : Denis Larocque, Aurélie Labbe [2022]. *80619 Méthodes avancées en exploitation de données Méthodes basées sur les arbres et forêts aléatoires*, *(RF)*, HEC Montréal, 101 p.

L’algorithme de CART est un algorithme assez intéressant permettant d’avoir une première idée sur le partitionnement au sein des méthodes d’arbres. C’est un procédé itératif utilisant pour la plupart du temps le partitionnement binaire, c’est-à-dire, qu’il opère uniquement deux divisions sous chaque nœud (Denis L., 2022).

Il existe en général deux types de partitionnements. Lorsque le prédicteur est continu, la valeur de X est comparée à une valeur constante. Dans l’exemple présenté à la figure ci-dessus, pour la variable ‘’Age’’, nous pouvons voir qu’au niveau du premier nœud si une personne est âgée de plus de 30 ans on tombe dans le partitionnement gauche de l’arbre, sinon, dans le partitionnement de droite. Pour les variables catégorielles, on regarde si la valeur que prend la variable, tombe dans un sous-ensemble des valeurs possibles de cette variable. Dans l’exemple illustré à la figure de l’exemple ci-dessus, une scission à lieu au niveau de la variable sexe ayant seulement deux niveaux représentant chacun un sous-ensemble respectivement des valeurs possibles pour la scission. Si le sexe est mâle, on se retrouve dans le sous-ensemble de gauche, sinon dans le sous-ensemble de droite (Denis L., 2022).

Le but revient à trouver le meilleur partitionnement parmi l’ensemble des partitionnements possibles. Pour comparer les partitionnements entre eux et en déterminer le meilleur, on a souvent recours à l’erreur quadratique dans le cadre de la régression (Denis L., 2022). On cherchera donc à minimiser l’équation suivante :

*Figure 2.21 Équation du critère de partitionnement chez les arbres de régressions*

Une image contenant texte, Police, ligne, blanc

Description générée automatiquement

Source : Denis Larocque, Aurélie Labbe [2022]. *80619 Méthodes avancées en exploitation de données Méthodes basées sur les arbres et forêts aléatoires*, *(RF)*, HEC Montréal, 101 p.

Où ῩR et ῩL sont les moyennes des observations des nœuds se trouvant à droite et à gauche respectivement. D’autre part, tL et tR représentent l’ensemble des indices des observations qui se retrouvant dans le nœud de gauche et de droite respectivement.

Afin d’éviter le sur-apprentissage, il est possible d’imposer une contrainte sur le nombre minimal d’observations par nœud, par exemple (Denis L., 2022). D’autres processus tels que l’élagage, permettent de pénaliser la formation de grands arbres en établissant une relation entre la performance et la complexité de l’arbre grâce à la formule suivante :

*Figure 2.22 : Formule du critère de pénalisation coût-complexité*



Source : Denis Larocque, Aurélie Labbe [2022]. *80619 Méthodes avancées en exploitation de données Méthodes basées sur les arbres et forêts aléatoires*, *(RF)*, HEC Montréal, 101 p.  

Où SSE(T) représente la somme des carrés de l’erreur de l’échantillon, α représente la force de pénalité associée à la complexité du modèle et NT est le paramètre représentant la complexité du modèle.

#### 2.7.2.2. Le boosting

Dans cette section, nous aborderons le sujet du boosting avec perte quadratique. En soi, l’algorithme de boosting et les modèles de même famille ont commencé à se faire connaître grâce au recherche et développement de Freund (1996) et son algorithme Adaboost (Denis L., 2022).

L'algorithme vous est présenté dans la figure qui suit :

*Figure 2.23 : Algorithme d’Adaboost*

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Source : Denis Larocque, Aurélie Labbe [2022]. *80619 Méthodes avancées en exploitation des données boosting*. HEC Montréal, 68 p.

Le principe du boosting avec perte quadratique suit un principe se situant dans le même ordre d’idée.

On choisit tout d’abord taux d’apprentissage ϵ > 0 mais très petit. Ce paramètre peut être vu comme une forme de contrôle sur le poids que représente chaque ajout d’un nouvel arbre au modèle. On sélectionne par la suite le nombre d’itérations désiré et un apprenant g qui servira à modéliser les erreurs de prédictions. On initialise finalement un classifieur Ĝ(X) à 0 et un vecteur de résidus égale aux valeurs des y.

On itère par la suite de m=1 au nombre total d’itération (M).

1. À chaque itération on cherchera à minimiser la somme des erreurs, représentée par le carré de la soustraction de la valeur se trouvant dans le vecteur résiduel à la position i et la valeur correspondante à la même position dans le vecteur de prédiction de l’apprenant.
2. On met à jour le classificateur de l’itération m en additionnant le classificateur m-1 au produit de notre taux d’apprentissage avec la valeur de l’apprenant.

Le modèle final correspondra à classificateur de la M ième itération. L’algorithme est résumé dans la figure ci-dessous :

*Figure 2.24 : Algorithme du boosting avec perte quadratique*

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Source : Denis Larocque, Aurélie Labbe [2022]. *80619 Méthodes avancées en exploitation des données boosting*. HEC Montréal, 68 p.

Une amélioration de cette technique est le boosting avec gradient. Cette méthode utilise un processus nommé la descente de gradient qui consiste à itérer, sur les différentes valeurs de la fonction de perte du modèle. À chaque itération i, la valeur de la perte est comparée à la valeur de la perte calculée au pas précédent (i-1). On itère ainsi jusqu’à ce qu’un critère d’arrêt soit atteint. C’est ainsi qu’il nous est possible de trouver un minimum local, soit la valeur de perte minimale (Denis L., 2022). Une description de l’algorithme vous est présentée à la figure suivante :

*Figure 2.25 : Algorithme du gradient boosting*

*Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement*

Source : Denis Larocque, Aurélie Labbe [2022]. *80619 Méthodes avancées en exploitation des données boosting*. HEC Montréal, 68 p.

### 2.7.3 Les séries temporelles

Puisque nous nous intéressons à prédire la tendance de la sévérité totale observée, nous n'avons d’autres choix que d’aborder des notions de séries temporelles. Les séries temporelles sont un ensemble de données recueillies à un intervalles réguliers. Ce sujet pourrait être par exemple, le cours d’une action à la bourse ou bien la demande en électricité sur une période donnée. Les séries temporelles permettent de mener une étude sur les tendances possibles qui peuvent être observées à travers le temps et même, de repérer des cycles.

Dans notre cas à nous, il s’agirait plutôt du niveau de la demande pour une pièce d’auto donné et de l'évolution du prix moyen à travers le temps.

Comme abordé plus tôt, nous avons vu qu’il est possible de modéliser la sévérité et la demande à l’aide de modèles linéaires généralisés par exemple.

L’estimation de Yt, la valeur de la sévérité au temps t, pourrait alors se modéliser de la manière suivante :

*Figure 2.26 : Modélisation de l’espérance de Yt*



Source : Debbie, J. Dupuis [2021]. *Méthodes de prévision Séances 6 & 7 – Régression linéaire multiple I & II,* HEC Montréal, 27 p.

Cependant, il est possible de supposer à partir de cette formulation que les erreurs sont autorégressives. C’est-à-dire, que les erreurs ∈ au temps t serait fonction des erreurs de prédictions passées à un pas t-∆ (Debbie J. D., 2021). En supposant que ∆=1, nous retrouvons la formulation suivante:

*Figure 2.27 : Modélisation d’erreurs autorégressives*

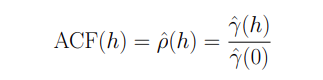


Source : Debbie, J. Dupuis [2021]. *Méthodes de prévision Séances 6 & 7 – Régression linéaire multiple I & II,* HEC Montréal, 27 p.

où, ∅ représente un paramètre de lissage et wt, représente le bruit blanc supposé être indépendant et identiquement distribué, de loi normale d’espérance 0 et de variance σ2.

Afin de supposer que wt représente une série bruit blanc, il est important de vérifier l’état de stationnarité de notre. C’est-à-dire de vérifier si les observations de wt à décalage de temps ∆ ne sont pas corrélées entre elles (Debbie J. D., 2021). Afin d’évaluer ces corrélations entre paires d’observations, on a recours à la fonction d’autocorrélation suivante :

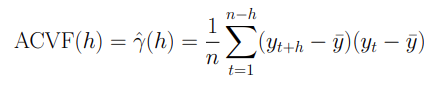
*Figure 2.28 : Fonction d’autocorrélation échantillonalle*



Source : Debbie, J. Dupuis [2020]. *Méthodes de prévision Séance 2 – Idées et outils de base,* HEC Montréal, 32 p.

Où, h représente les pas de décalage de temps, 𝜸(0) est la variance échantillonnalle, et où 𝜸(h) se retrouve grâce à la formule suivante:

*Figure 2.29 : Fonction d’autocovariance échantillonalle*

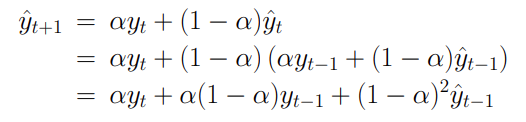


Source : Debbie, J. Dupuis [2020]. *Méthodes de prévision Séance 2 – Idées et outils de base,* HEC Montréal, 32 p.

Où, y barre est la moyenne. En pratique, l’observation d’un corrélogramme nous permet d’avoir une idée sur les autocorrélations échantillonnalles de la série en question.

De façon régressive nous pouvons voir ∈t représenté sous une forme de lissage exponentielle, comme dans la formulation suivante:

*Figure 2.30 : Modélisation récursive de ŷt+1*



Source : Debbie, J. Dupuis [2022]. *Méthodes de prévision Séance 4 – Lissage exponentiel I,* HEC Montréal, 20 p.

En utilisant la notation autorégressive suivante:

Φ( B) = 1 – φ1B – φ2B2,

Il nous est possible de réécrire le modèle autorégressif des erreurs sous la forme suivante :

*Figure 2.31 : Représentation du modèle AR pour les erreurs*



Source : Debbie, J. Dupuis [2021]. *Méthodes de prévision Séances 6 & 7 – Régression linéaire multiple I & II,* HEC Montréal, 27 p.

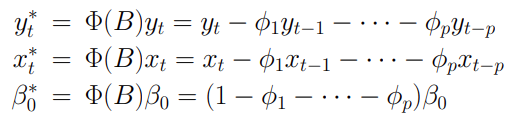
Le modèle devient donc :

*Figure 2.32 : Modélisation de l’espérance de Yt avec erreurs autorégressives*



Source : Debbie, J. Dupuis [2021]. *Méthodes de prévision Séances 6 & 7 – Régression linéaire multiple I & II,* HEC Montréal, 27 p.

Il nous devient donc possible de réécrire les composantes du modèle linéaire sous les formats suivants:

*Figure 2.33 Réécriture des composantes du modèle linéaire avec erreurs autorégressives*   


Source : Debbie, J. Dupuis [2021]. *Méthodes de prévision Séances 6 & 7 – Régression linéaire multiple I & II,* HEC Montréal, 27 p.

# Chapitre 3 : Méthodologie

Ce projet est la continuité d’une analyse précédemment menée sur la tendance de la sévérité moyenne observée pour les provinces d’Ontario et du Québec entre les années 2018 et 2022. Dans cette même optique, j’ai décidé de sélectionner des données portant sur des accidents ayant eu lieu entre le premier janvier 2018 et la fin du troisième trimestre de 2023.

Dans une deuxième phase, il a été question de collecter et de rassembler les données provenant de différentes sources afin de constituer une base de données structurée pour l’analyse.

Dans un troisième temps, il a été question de mener une analyse exploratoire sur la base de données bâtie afin de distinguer les informations pertinentes et non pertinentes. Par la suite, plusieurs traitements ont dû être appliqués aux données avant de procéder à l’ajustement des modèles : imputation ou retrait de valeurs manquantes, normalisation de variables continues, vectorisation de mots réduction de dimensionnalité. Finalement, les 5 pièces d’auto présentant le plus haut rapport sévérité-fréquence ont été sélectionnées pour l’analyse. L’étude a été effectuée sur une base hebdomadaire, de façon à prendre en compte la proportion de demande pour une pièce donnée.

Dans un quatrième temps, le jeu de données a été divisé en un ensemble test, correspondant aux accidents ayant eu lieu à partir de l’année 2023 et un ensemble d’entraînement portant sur les données d’accident ayant eu lieu entre le premier janvier 2018 et le 31 décembre 2022. Afin de réduire la dimensionnalité des données, des méthodes de sélection de variables comme le ‘’stepwise’’ et la validation croisée ont été mises en œuvre.

Finalement, deux familles de modèles ont été ajustées aux données afin de modéliser la sévérité et la fréquence observée pour les pièces sélectionnées. La première famille de modèle sélectionnée est le modèle linéaire généralisé n’incluant aucune interaction entre les variables. La deuxième famille de modèle choisie est la famille des arbres de régressions. Afin de modéliser la sévérité, des liens gaussiens ont été utilisés tandis que pour la fréquence, des liens de Poisson ont été privilégiés. Finalement, le RMSE a été utilisé pour évaluer la performance des modèles sur les ensembles de validation et de test. En guise de synthèse, les résultats vous seront présentés dans les sections suivantes.

# Chapitre 4 : Traitement du jeu de données

## 4.1. Source du jeu de données.

Le jeu de données utilisé a été constitué à partir de plusieurs tables de données non-structurées fournies par la compagnie contenant des informations sur le profil des assurés et sur les estimés des réparations pour des réclamations couvrant les années 2018 à 2023. À ces données, nous avons joint des informations sur la température et sur les indices de prix à la consommation provenant de sources du gouvernement du Canada.

## 4.2. Implémentation de fonctions et type de manipulation.

Comme mentionné plus tôt, afin de bâtir la table de données il a fallu réaliser une série de manipulation telles que des jointures de tables avec des sources de données structurées et non-structurées. Il a également été nécessaire de créer des fonctions pour collecter, filtrer, standardiser et encoder les données afin qu’elles soient exploitables et manipulables dans l’objectif de notre analyse. Pour ce faire, on a dû recourir à trois types de langage de programmation en raison de certaines limitations internes. Premièrement, du SQL pour joindre les informations non-structurées provenant des différentes tables de la compagnie. Ce langage facilitait la jointure de grosses bases de données, contrairement aux deux autres langages. Cependant ce langage offrait beaucoup de limitations fonctionnelles. Deuxièmement, du R pour tout d’abord extraire les données des tables à partir du pipeline, puis pour le traitement des données et la création de fonctions. Cependant il arrivait que l’encodement de certaines fonctionnalités en R, n’étaient pas aussi efficaces qu’en python. Ce qui était souvent limitant en fonction de la capacité computationnelle qui nous était alloué. Finalement, du python pour la création de fonctions et la performance des modèles. Il aura donc été régulièrement nécessaire d’alterner d’un langage à un autre en fonctions des avantages et inconvénients de chacun des langages.

### 4.2.1. Traitement des noms des pièces

Comme mentionné plus tôt dans ce rapport, l’analyse qui suit vise à comprendre la sévérité moyenne observée au niveau des pièces d’auto. Un traitement au niveau des pièces a donc été important. Grâce aux tables d’estimés, il a été possible d’obtenir le corpus des différents noms de pièces, qui ont dû être réparées entre janvier 2018 et octobre 2023. Toutefois, ces entrées manuelles présentaient certains défis : fautes d'orthographe, variations linguistiques, Incohérence dans la mise en forme (‘’casting’’) et présence de caractères spéciaux ou de chiffres compliquant le regroupement. Pour remédier à ces problèmes, il a tout d’abord fallu créer une fonction permettant de gérer les caractères spéciaux, les espaces inutiles et de normaliser la “caste” des mots. Par la suite, il a fallu traiter les cas des différences linguistiques. Certaines librairies existent en R ou python mais n'étaient pas adaptées aux besoins de la cause. D’autre part, il n’était pas envisageable d’avoir recours à un API de google pour de la traduction automatique par soucis de limitations internes et de politiques de confidentialité des données. La dernière option, sans doute la plus longue en termes de temps dans tout ce projet, a été de manuellement entrer toutes les pièces dans une interface en ligne afin d’en obtenir la traduction complète en anglais. Enfin, afin de réduire la dimensionnalité des données en regroupant les pièces similaires, il a fallu faire appel à un algorithme de wor2vec pour vectoriser le nom des pièces, pondéré par la valeur standardisée des prix, l’algorithme DBSCAN afin d’attribuer à chaque nom de pièce un groupe en particulier. Ainsi, il a été possible de regrouper 92 331 noms de pièces d’auto différents en 4723 groupes. Bien que la méthode de regroupement ne soit pas parfaite, elle a tout de même permis de réduire grandement la dimensionnalité de cette variable.

Voici une première vue de la distribution du prix par pièces la figure ci-dessous :

*Figure 4.1 : Graphique de la distribution des prix des pièces de véhicules*

Une image contenant texte, diagramme, capture d’écran, Tracé

Description générée automatiquement

En regardant le graphique ci-dessus, on peut voir que la distribution du prix des pièces semble suivre une distribution de densité log-normale. Afin de normaliser les données, Il a donc été intéressant d’envisager utiliser le log du prix des pièces plutôt que le prix en soi.

Comme mentionné plus tôt dans ce rapport, puisqu’il était impossible pour des raisons computationnelles de gérer autant de catégorie, il a fallu concentrer l’entraînement sur 5 pièces seulement. Ces pièces ont été sélectionnées en fonction de leur rapport sévérité-fréquence. Les termes groupe 6, groupe 9, groupe 20, groupe 37 et groupe 40 seront utilisés pour faire allusion à ces pièces.

### 4.2.2 Traitement des données en provenance de sites du gouvernement du Canada

Afin d’obtenir d’autres potentiels prédicteurs de la valeur des pièces contenues dans les estimés recueillis, des informations en provenance de statistiques Canada portant sur les indices de prix de certains biens ayant un possible rapport avec la composition des pièces ont été jointes au jeu de données. Les indices de prix du caoutchouc et plastique, du bois de sciage, du pétrole, des métaux ferreux et non ferreux, des produits métalliques et des produits électriques ont été sélectionnés.

Dans un deuxième temps, des informations sur la météo en provenance du site ‘’climate.weather.gc.ca’’ pour les périodes visées ont été rajoutées. Afin de joindre ces informations à la base de données, Il a été nécessaire de standardiser les codes postaux des assurés afin de les convertir dans un format uniforme et d’en extraire le ‘’FSA’’, ou secteur de tri d’acheminement, représenté par les trois premières lettres de chaque code postal, identifiant une région spécifique. Grâce à ses codes, les informations concernant les longitudes et latitudes respectives ont pu être ajoutées à la table de données. Les données manquantes de longitude et de latitude ont été complétées par les coordonnées centrales des FSA, selon les informations disponibles. Après cela, les données sur les différentes stations météorologiques au Québec et en Ontario ont été collectées. Une fonction en R a été développée pour calculer la distance haversienne entre chaque station et lieu d’accident. En s’appuyant sur les données associées aux stations et aux accidents, une fonction en python a été conçue pour récupérer les informations météorologiques journalières comme la température maximale, la température minimale, la température moyenne, le HDD, le CDD, la quantité totale de précipitation, la quantité totale de neige, le nombre de centimètres de neige sur le sol, la direction du vent et la vitesse maximale du vent, à partir de la date et de l’identifiant de la station. Enfin, des méthodes d’imputation par la moyenne et de troncation à zéro ont été appliquées pour traiter les données météorologiques manquantes.

### 4.2.3 Traitement des données sur les assurés

Dans le cadre de notre analyse nous n’avons utilisé que trois variables portant sur le profil du conducteur.

La première variable est l’âge de l’assuré. Je suppose ici que le risque d'occurrences d’un accident n’est pas le même en fonction de l'âge. La distribution de l’âge des assurés ayant eu un accident est présentée dans la figure ci-dessous:

*Figure 4.2 : Distribution des âges de l’échantillon observé*

Une image contenant capture d’écran, texte, Tracé, diagramme

Description générée automatiquement

À partir du graphique ci-dessus, il est possible d’observer que la distribution de l’âge des assurés ayant eu un accident semble être d’une densité log-normal. Pour imputer les valeurs manquantes de la variable âge, des valeurs ont été simulées selon une distribution normale à partir des paramètres de la fonction de densité log-normale. Par la suite, des groupes d’âge ont été formés afin de créer une nouvelle variable catégorielle avec moins de dimensions. Les âges ont été regroupées en quatre catégories : soit les assurés âgés de moins de 25 ans, les assurés âgés entre 25 à 35 ans, les assurés âgés de 35 à 50 ans et les assurés âgés de plus de 50 ans.

La deuxième variable est en lien avec le sexe des assurés prenant deux valeurs possibles : ‘F’ pour femme, et ‘M’ pour homme. Le graphique ci-dessous montre la distribution des sexes dans le jeu de données:

*Figure 4.3 : Distribution du sexe dans l’échantillon observée*

Une image contenant texte, capture d’écran, diagramme, Police

Description générée automatiquement

On peut observer que la variable semble suivre une distribution discrète avec comme probabilité d’être une femme de p(f) = Nf/(Nf+Nh) ou d’être un homme avec p(h) = Nh/(Nf+Nh). Les deux probabilités étant relativement proches, des simulations de valeurs ont été réalisées afin d’imputer les données manquantes en utilisant la loi discrète de probabilités présentée dans la figure ci-dessus.

La troisième variable est en lien avec le statut matrimonial de l’assuré. L’ensemble des valeurs possibles pour cette variable sont ‘D’pour divorcé(e), ‘M’pour marié(e), ‘S’ pour célibataire, ‘U’ pour non répertorié(e), ‘W’ pour veuf(ve) et ‘C’ pour conjoint(e) de fait. Le graphique ci-dessous, on peut voir la distribution du statut matrimonial dans nos données :

*Figure 4.4 : Distribution du statut matrimonial dans l’échantillon observé*

Une image contenant texte, capture d’écran, diagramme, Tracé

Description générée automatiquement

Afin de remplacer les valeurs manquantes, Le jeu de données a été redistribué en fonction de l’âge et le genre des assurés. En supposant que la distribution du statut matrimonial varie en fonction du genre et de l’âge, la fonction de masse de la probabilité d’appartenance à un statut a été déterminée. Finalement, les valeurs manquantes ont été imputées par les valeurs simulées de cette fonction de masse de probabilité.

### 4.2.4 Traitement des données sur les estimés

En ce qui concerne les données sur les estimés, 14 variables ont été sélectionnées. Soit tout d’abord, la région, le nombre de cylindre sur le véhicule, la puissance, le poids du véhicule par puissance, le type de liquide de combustion, le statut du véhicule, véhicule hybride ou non, l’année du véhicule, la marque du véhicule, la valeur actuelle nette du véhicule, l’âge du véhicule, le millage sur le véhicule, la description du véhicule et le type de garage.

Premièrement, en ce qui concerne les variables portant sur la marque du véhicule et l’année du véhicule, les observations comportant des valeurs manquantes ont été enlevé du jeu de données. Deuxièmement, pour la variable de la région et celle du véhicule hybride, aucune imputation n’a pas été nécessaire.

Pour ce qui est de la puissance du véhicule, le poids du véhicule par unité de puissance, la valeur actuelle nette du véhicule et le millage, les valeurs manquantes ont été imputées par la moyenne. Pour les trois premières variables les données ont été regroupées en fonction de l’année, de la marque et du modèle du véhicule. Par la suite, il a fallu calculer la moyenne par regroupement pour les deux variables et remplacer les valeurs manquantes par ces valeurs. En ce qui concerne la quatrième variable, les données ont été regroupées par âge de véhicule. La moyenne du millage a ensuite été calculé par âge de véhicule. Finalement, les valeurs manquantes de la variable millage ont été remplacées par la moyenne de millage en fonction de l’âge du véhicule qui y était associé.

Pour la variable portant sur le nombre de cylindres par véhicule, Les valeurs manquantes ont été imputées par la valeur médiane de la variable au lieu de la moyenne. De plus, la valeur médiane était équivalente au mode.

En ce qui a trait aux valeurs manquantes de la variable statut du véhicule, les véhicules d’âge 0 ou -1, ont été catégorisés comme ‘’NEW’. Les autres observations ayant des valeurs manquantes pour la variable se sont fait attribuer la valeur “USED”.

Ensuite, pour les variables portant sur le type d'essence, la description du véhicule et le type de garage, les valeurs manquantes ont été imputées par leurs valeurs modales respectives. Les graphiques ci-dessous nous illustre les distributions de ces trois variables aléatoires :

*Figure 4.5 : Distribution du type de combustible pour voiture dans l’échantillon observé*

Une image contenant texte, capture d’écran, Rectangle

Description générée automatiquement

*Figure 4.6 : Distribution du type de véhicule conduit dans l’échantillon observé*

Une image contenant texte, capture d’écran, diagramme, Police

Description générée automatiquement

*Figure 4.7 : Distribution du type de garage d’où proviennent les estimés*

Une image contenant texte, capture d’écran, diagramme, Tracé

Description générée automatiquement

Donc selon les graphiques ci-dessus, les valeurs modales sont ‘’Gasoline’’ pour le type d’essence, ‘‘Passenger Car’’ pour la variable type de véhicule et ‘Other’ pour la variable du type de garage.

### 4.2.5 Encodage des données

Afin de rendre les données interprétables par certains modèles et afin de tenir compte du caractère catégoriel de certaines variables, il a été nécessaire d’opérer une dernière manipulation aux données.

C’est dans ce contexte qu’il a fallu avoir recours à une dernière technique abordée dans la section de la revue littéraire, soit, le ‘One-Hot-Encoding’. Cette technique transforme les valeurs possibles d’une variable catégorielle donnée en nouvelles variables binaires prenant les valeurs 0 ou 1 en fonction de l’absence ou de la présence de la catégorie en question pour une observation donnée. Il a donc fallu encoder les variables catégorielles telles que la région, la marque du véhicule, le type d’essence, le statut matrimonial, le statut du véhicule, le type de garage et la description du véhicule.

Puis, toutes les variables numériques continues ont dû être normalisé. Cela inclut donc : la valeur actuelle nette du véhicule, le millage, le prix étant déjà normalisé.

Après tout cela, notre jeu de données comportait 112 variables explicatives incluant toutes les nouvelles variables binaires générées par le processus du ‘One-Hot-Encoding.

### 4.2.6 Étude de la corrélation inter-variable

Finalement pour conclure le traitement des données, une analyse de la corrélation inter-variable a été menée. À des fins d’interprétabilité visuel, les variables ont été renommées. Le résultat est présenté dans le graphique ci-dessous:

*Figure 4.8 : Graphique de la corrélation entre paires de variables dans le jeu de données*

Une image contenant texte, Tracé, ligne, diagramme

Description générée automatiquement

Le graphique ci-dessus est une représentation imagée de la force de corrélation des variables entre elles. Une coloration bleue, indique une corrélation positive entre deux variables, une coloration rouge indique une corrélation négative entre deux variables. L’intensité de la couleur communique de l’information sur la force de la corrélation existant entre deux variables. À partir de l’analyse de la corrélation entre les variables et le graphique ci-dessus, Il a été possible de retirer certaines variables portant sur les indices de prix, sur le temps, sur les niveaux de marques de voitures. Cette analyse a permis de réduire la dimensionnalité du jeu de données en passant de 112 variables explicatives à 101 variables explicatives. Finalement, afin de rendre la série plus stationnaire, trois variables décalages par rapport à la variable d’intérêt ont été rajoutées.

Il est à noter que les modèles ont été entraînés sur 304 observations chacun, représentant les semaines de janvier 2018 à octobre 2023. Aussi, le jeu de données du Québec contenait d’avantages de valeurs manquantes que celui de l’Ontario. Donc, dans les cas où des valeurs manquaient, celle-ci était imputée.

# Chapitre 5 : Analyse des résultats

Dans un premier temps, deux modèles ont été ajustés aux données afin de modéliser la sévérité moyenne. Ces deux familles de modèles ont été utilisées sur des données portant sur deux régions différentes et 5 pièces d’auto. Dans les lignes qui suivent, les résultats pour une seules pièces d’auto seront présentés.

La pièce choisie sera appelée pièce 6. Cette pièce est celle ayant le plus grand rapport coût-sévérité, et par le fait même celle ayant le plus grand nombre de données. De plus, elle semblait bien représenter les résultats observés pour les 4 autres pièces. Les résultats de performance sont présentés dans le tableau ci-dessous.

*Tableau 5.1 : Résultats de performances des modèles GLM et XGB pour la prédiction de la sévérité moyenne et de la fréquence en Ontario et au Québec pour la pièce 6*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Modèles | GLM Sévérité | | XGB Sévérité | | GLM Fréquence | | XGB Fréquence | |
| Régions | ON | QC | ON | QC | ON | QC | ON | QC |
| Score de Validation | 117,68 | 249,76 | 61,70 | 55,85 | 71,43 | 181,28 | 186,4 | 10,78 |
| Score de Test | 161,48 | 568,44 | 66,88 | 74,86 | 27,47 | 178,09 | 116,39 | 12,77 |
| Score d’Autocorrélation | 0,62 | 0,45 | 0,0048 | 0,000000035 | 0,45 | 0,001 | 0,8 | ~0 |

Tout d’abord, concernant les résultats obtenus en termes de performances des modèles sur la sévérité moyenne observée par semaine pour la pièce 6, il se trouve que pour l’Ontario et le Québec, le GLM obtient un score de 117,68 en validation et de 161,48 en test pour l’Ontario et un score de 249,76 en validation et 568,44 en test pour le Québec. Cela laisse à penser que l’augmentation du RMSE représenterait l’erreur de généralisation qui pourrait être liée à du surajustement. De plus pour le Québec, on constate une plus grande variation du RMSE, ce qui laisse à penser que le modèle semble avoir eu plus de difficulté à s’ajuster aux nouvelles données. C’est d’ailleurs ce que présentent les deux corrélogrammes obtenus à la suite de l’ajustement de ces modèles.

*Figure 5.1 : Autocorrélation des Résidus pour le Modèle GLM gaussien pour la prédiction de la sévérité moyenne de la Pièce 6 en Ontario*

**Une image contenant texte, capture d’écran, ligne, Tracé

Description générée automatiquement**

*Figure 5.2 : Autocorrélation des Résidus pour le Modèle GLM gaussien pour la prédiction de la sévérité moyenne de la Pièce 6 au Québec*

**Une image contenant texte, capture d’écran, Police, Tracé

Description générée automatiquement**

Il est observable que dans le corrélogramme de l’Ontario, deux pics dépassent l'intervalle de confiance à 95%. Le premier pic représente la corrélation de la série avec elle-même, on s’attend donc à ce que la valeur soit de 1, car la série devrait être parfaitement corrélée avec elle-même. En ce qui concerne le deuxième pic, il semble indiquer qu’il existe une autocorrélation significative. Dans le graphique du Québec, 4 pics dépassent l’intervalle de confiance à 95%. Comme pour le premier graphique, le premier pic représente l’autocorrélation de la série avec elle-même. Quant aux trois autres pics, ils semblent indiquer la présence d’autocorrélations significatives dans le modèle. Finalement, les scores d’autocorrélation obtenu pour le modèle GLM gaussien sont de 0,68 pour l’Ontario et de 0,45 pour le Québec. Il est à rappeler qu’un bon score d’autocorrélation devrait se rapprocher de 0, plus le score en est éloigné et plus il ne devient pas possible de valider le choix du modèle en question et ses hypothèses portant sur les erreurs autorégressives comme mentionné au chapitre 2 dans la section portant sur les séries temporelles. Les prédictions du modèle linéaire généralisé pour la sévérité moyenne hebdomadaire de la pièce 6 pour l’Ontario et le Québec vous sont présentées dans les deux figures qui suivent :

*Figure 5.3 : Prévisions de la sévérité moyenne hebdomadaire pour la pièce 6 en Ontario par le modèle GLM Gaussien*

Une image contenant texte, capture d’écran, Police, Tracé

Description générée automatiquement

*Figure 5.4 : Prévisions de la sévérité moyenne hebdomadaire pour la pièce 6 au Québec par le modèle GLM Gaussien*

**Une image contenant texte, capture d’écran, Police, Tracé

Description générée automatiquement**

Cette vue des prédictions du modèle linéaires généralisé gaussien sur les deux ensembles de données permet de mieux inférer sur les scores d’autocorrélations obtenus. Il est observable que les vraies valeurs se retrouvent à l’extérieur de l’intervalle de confiance à 95% des prédictions. Ce qui indique un problème d’ajustement du modèle. Pour les prédictions sur les deux ensembles, les modèles semblent avoir capturer les tendances mais pas le niveau des vraies valeurs.

En ce qui concerne le modèle XGB, les résultats montrent que pour l’Ontario, le score de validation est de 61,70 contre 66,88 en test. Le modèle semble considérablement mieux performer que le modèle de base, soit le GLM sans interactions. Il reste toutefois à mentionner que le RMSE du modèle est plus élevé en test qu'en validation, ce qui pourrait être aussi lié à du surajustement. Pour ce qui est du Québec le score de validation est de 55,85 et le score de test est de 74,86. L’erreur de généralisation pour le Québec, ici représentée par une augmentation de la valeur du RMSE, est plus grande que celle pour le jeu de données de l’Ontario.

*Figure 5.5 : Autocorrélation des Résidus pour le Modèle XGB de fonction de pertes quadratique pour la prédiction de la sévérité moyenne de la Pièce 6 en Ontario*

**Une image contenant texte, capture d’écran, ligne, Tracé

Description générée automatiquement**

*Figure 5.6 : Autocorrélation des Résidus pour le Modèle XGB de fonction de pertes quadratique pour la prédiction de la sévérité moyenne de la Pièce 6 au Québec*

**Une image contenant texte, capture d’écran, ligne, Tracé

Description générée automatiquement**

Dans le corrélogramme de l’Ontario, 1 pic dépasse l’intervalle de confiance à 95%. C’est le pic qui représente la corrélation de la série avec elle-même, on s’attend donc à ce que la valeur soit de 1, car la série devrait être parfaitement corrélée avec elle-même. Pour le reste, il ne semble pas y avoir de d’autocorrélation significative, ce qui est bon signe car cela permet de valider le choix d’ajustement du modèle XGB avec des erreurs autorégressives. Dans le corrélogramme du Québec, 2 pics dépassent l’intervalle de confiance à 95%. Comme pour le premier corrélogramme, le premier pic représente l’autocorrélation de la série avec elle-même. Quant au deuxième pic, il semble indiquer la présence d'autocorrélations significatives dans le modèle. Finalement, les scores obtenus avec l’ajustement du modèle XGB avec perte quadratique sont de 0,0048 pour l’Ontario et de 0,000000035 pour le Québec. Ce qui semble être de très bon score en comparaisons aux ajustements du modèle linéaire généralisé gaussien pour les mêmes jeux de données. Les prédictions du modèle XGB avec perte quadratique pour la sévérité moyenne hebdomadaire de la pièce 6 en Ontario et au Québec sont présentées dans les deux figures qui suivent :

*Figure 5.7 : Prévisions de la sévérité moyenne hebdomadaire pour la pièce 6 en Ontario par le modèle XGB avec fonction de perte quadratique*

**Une image contenant texte, capture d’écran, Tracé, ligne

Description générée automatiquement**

*Figure 5.8 : Prévisions de la sévérité moyenne hebdomadaire pour la pièce 6 au Québec par le modèle XGB avec fonction de perte quadratique*

**Une image contenant texte, capture d’écran, Tracé, ligne

Description générée automatiquement**

Dans le cas du modèle XGB avec perte quadratique, il était difficile d’obtenir des intervalles de prédictions car ce genre de modèle fait appels des méthodes d’ensembles. Nous ne commenterons ici que les observations sur les courbes. Les courbes des prédictions dans les deux graphiques semblent saisir les tendances et les niveaux des courbes des valeurs réelles. Le modèle champion pour la modélisation de la prévision de la sévérité moyenne hebdomadaire pour la pièce 6 est donc le modèle XGB à fonction de perte quadratique.

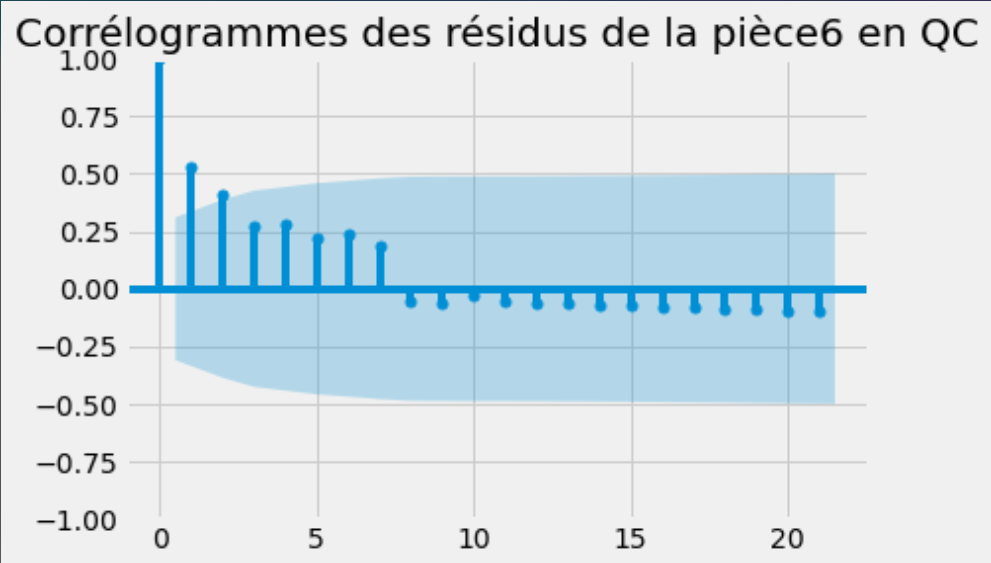
Deuxièmement, il sera question d’analyser les résultats de performances des deux familles de modèles sur la fréquence moyenne observée par semaine pour la pièce 6. Pour le modèle linéaire généralisé avec famille de loi de poisson, le score de validation est de 71,43 et le score sur l’ensemble test est de 27,47. La baisse de la valeur du RMSE du score de validation au score de test semble indiquer une bonne capacité au modèle à généraliser. Pour le Québec, le score de validation est de 181,28 pour l’ensemble de validation et de 178,09 pour l’ensemble test, ce qui est bien supérieur aux RMSE respectifs obtenus sur l’ensemble de données de l'Ontario. Cela démontre une plus grande difficulté du modèle à s’adapter aux jeux de données du Québec que celui de l’Ontario. Cependant, on observe aussi une baisse du RMSE lorsqu’on compare le score de validation et le score sur l’ensemble test. Cela semble indiquer que le modèle semble tout de même pouvoir généraliser correctement pour le Québec.

*Figure 5.9 : Autocorrélation des Résidus pour le Modèle GLM Poisson pour la prédiction de la demande totale pour la pièce 6 en Ontario*

**Une image contenant texte, capture d’écran, ligne, Police

Description générée automatiquement**

*Figure 5.10 : Autocorrélation des Résidus pour le Modèle GLM Poisson pour la prédiction de la demande totale pour la pièce 6 au Québec*

****

Le corrélogrammes de l’Ontario a deux pics qui dépassent l’intervalle de confiance à 95%. Le premier pic représente la corrélation de la série avec elle-même, on s’attend donc à ce que la valeur soit de 1, car la série devrait être parfaitement corrélée avec elle-même. Pour le deuxième pic, il semble indiquer l’existence d’une autocorrélation significative au sein de la série. Cela ne permet donc pas de valider le choix d’ajustement du modèle GLMP avec des erreurs autorégressives. Dans le corrélogramme du Québec, 2 pics dépassent l’intervalle de confiance à 95%. Comme pour le premier graphique, le premier pic représente l’autocorrélation de la série avec elle-même. Quant au deuxième pic, il semble indiquer la présence d’autocorrélations significatives dans le modèle. Finalement, un score de 0,45 et de 0,001 respectivement est obtenu pour l’ajustement du modèle Poisson aux jeux de données de l’Ontario et du Québec. Les prédictions du modèle linéaire généralisé Poisson pour la demande totale hebdomadaire de la pièce 6 en Ontario et au Québec sont présentées dans les deux figures qui suivent :

*Figure 5.11 : Prévisions de la demande totale hebdomadaire pour la pièce 6 en Ontario par le modèle GLM Poisson*

**Une image contenant texte, Tracé, capture d’écran, ligne

Description générée automatiquement**

*Figure 5.12 : Prévisions de la demande totale hebdomadaire pour la pièce 6 au Québec par le modèle GLM Poisson*

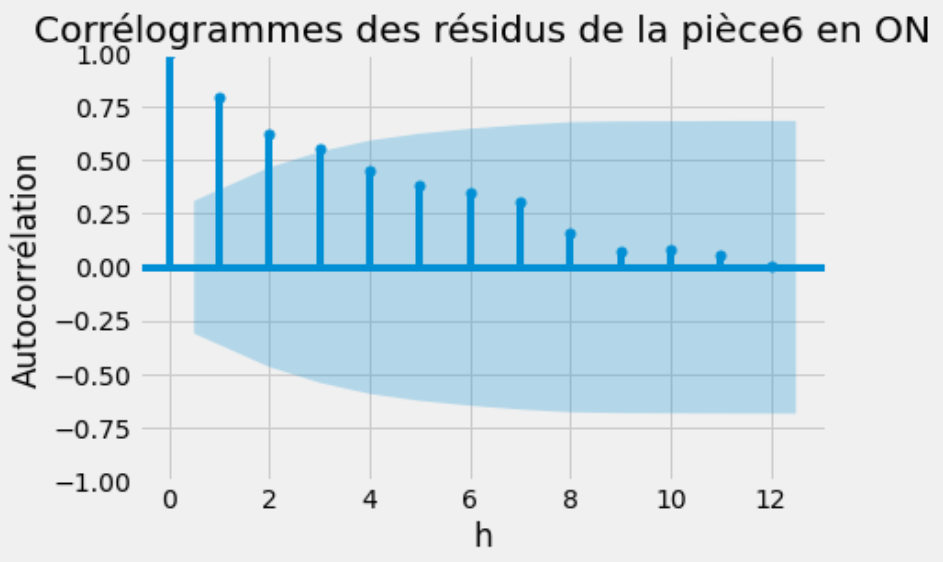
**Une image contenant texte, capture d’écran, Tracé, ligne

Description générée automatiquement**

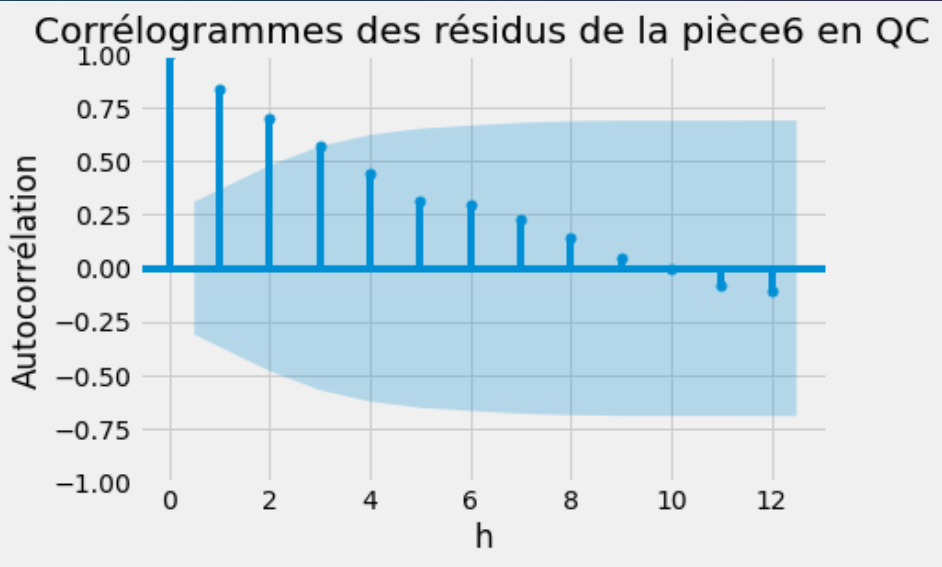
Cette vue des prédictions du modèle linéaire généralisé Poisson sur les deux ensembles de données permet de mieux inférer sur les scores d’autocorrélations obtenus. Il est à se rappeler que le modèle avait obtenir un score de 0,45 sur l’ensemble de jeu de données de l’Ontario et un score de 0,001 sur l’ensemble de jeu de données du Québec. Ce qui aurait dû indiquer que le modèle s’était mieux ajusté aux données du Québec qu’au données de l’Ontario. Cependant, les scores de RMSE obtenus sur les ensembles de validation et de test pour l’Ontario étaient beaucoup plus élevés pour le Québec, que pour l’Ontario. Il y’avait donc là une forme de contradiction. En ayant un aperçu du graphique de prévisions de chacun, il est possible d’observer que les données sur l’ensemble Québec, semblent toutes très proche de 0 ou à 0, d’où la bonne performance du modèle sur le score d’autocorrélation, mais pas sur celui du RMSE. Cela peut s’expliquer par l’imputation à 0 des valeurs manquantes qui a été fait dû à un manque de données afin de pouvoir entraîner le modèle.

Pour ce qui est du modèle XGB avec fonction de perte de poisson, le modèle obtient un score de 186,4 en validation et de 116,39 en test. Une augmentation par rapport aux résultats obtenus pour les mêmes ensembles respectifs avec le GLM. Cela semble indiquer que le modèle n’est pas arrivé à s’ajuster adéquatement aux données de l’ensemble Ontario. Malgré tout, une baisse significative du RMSE de l’ensemble de validation à l’ensemble test indique que le modèle arrive somme toute à généraliser correctement. Pour ce qui est de la performance du modèle sur les données du Québec, on a obtenu un score de 10,78 en validation et de 12,77 en test. C’est de loin le meilleur résultat obtenu sur les différents ensembles d’entraînement par les deux modèles. Cependant, l’augmentation de la valeur du RMSE de l’ensemble de validation à l’ensemble test, semble indiquer un possible sur ajustement du modèle aux données d’entraînement.

*Figure 5.13 : Autocorrélation des Résidus pour le Modèle XGB de fonction de perte Poisson pour la prédiction de la demande totale pour la pièce 6 au Québec*

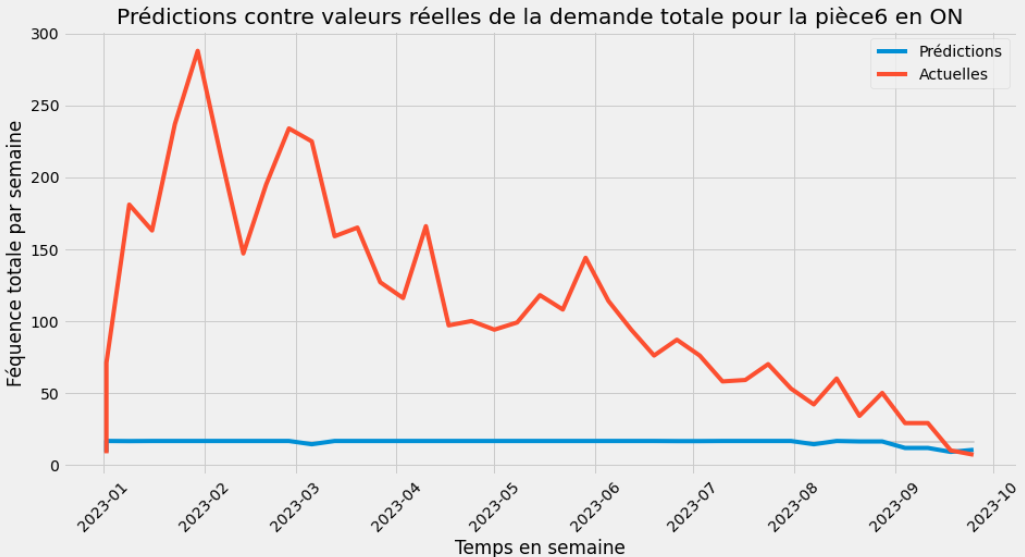
****

*Figure 5.14 : Autocorrélation des Résidus pour le Modèle XGB de fonction de perte Poisson pour la prédiction de la demande totale pour la pièce 6 au Québec*

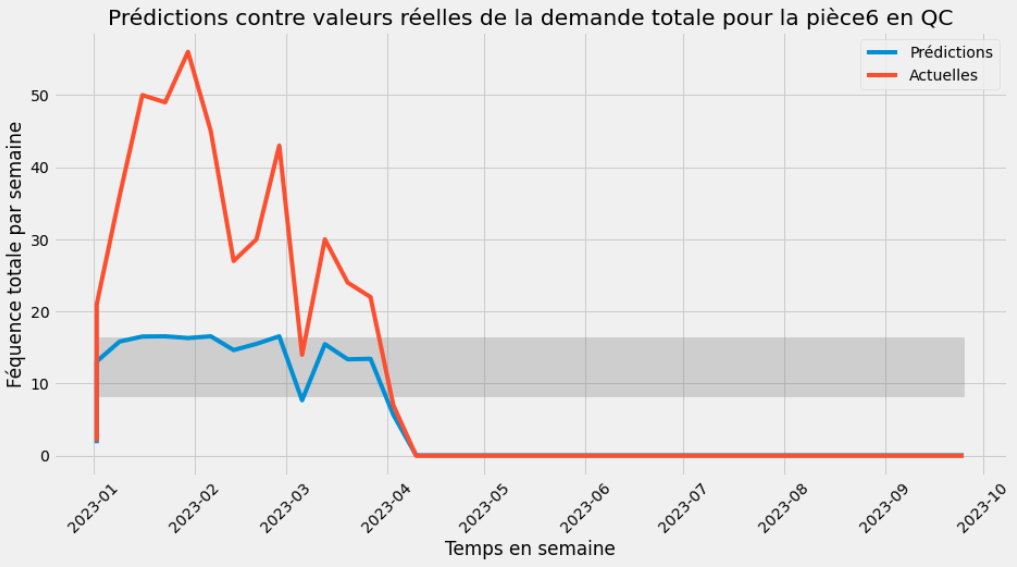
****

Dans le corrélogrammes de l’Ontario, 3 pics dépassent l’intervalle de confiance à 95%. Le premier est le pic qui représente la corrélation de la série avec elle-même, on s’attend donc à ce que la valeur soit de 1, car la série devrait être parfaitement corrélée avec elle-même. Les deux autres semblent indiquer la présence d’autocorrélations significatives dans les erreurs du modèle. Cela permet donc pas de valider le choix d’ajustement du modèle XGBP avec erreur autorégressives. Dans celui du Québec, 3 pics dépassent l’intervalle de confiance à 95%. La même analyse est faite que pour le graphique de l’Ontario. Finalement, un score de 0,8 et de 0 sont respectivement obtenus pour l’ajustement du modèle XGB avec fonction de perte de poisson aux jeux de données de l’Ontario et celui du Québec. 0.8 étant la pire valeur de scores d’autocorrélation obtenues jusqu’à présent. Les prédictions du modèle XGB avec fonction de perte de poisson pour la demande totale hebdomadaire de la pièce 6 en Ontario et au Québec sont présentées dans les deux figures qui suivent :

*Figure 5.15 : Prévisions de la demande totale pour la pièce 6 en Ontario par le modèle XGB Poisson*

****

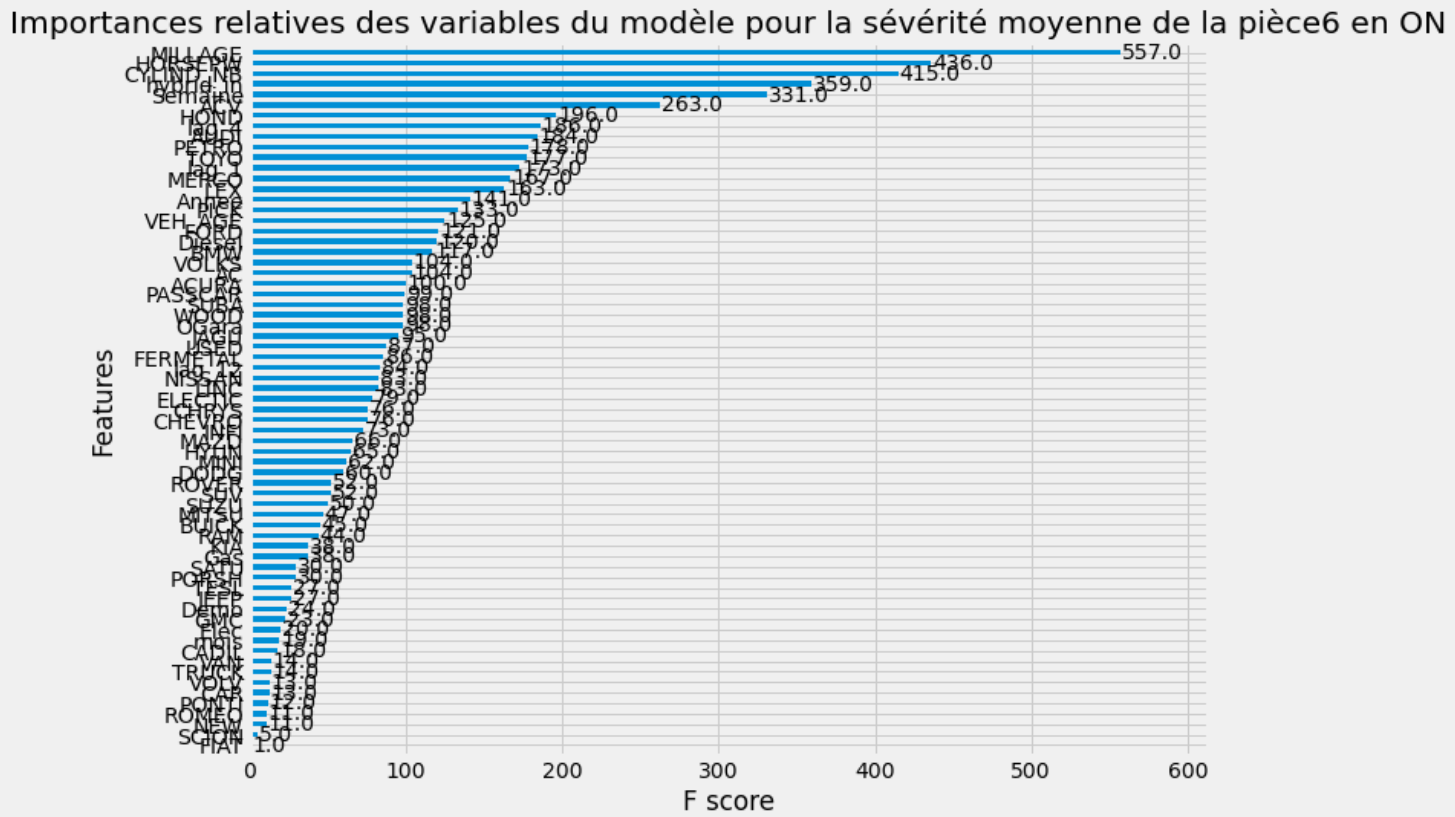
*Figure 5.16 : Prévisions de la demande totale pour la pièce 6 au Québec par le modèle XGB Poisson*

****

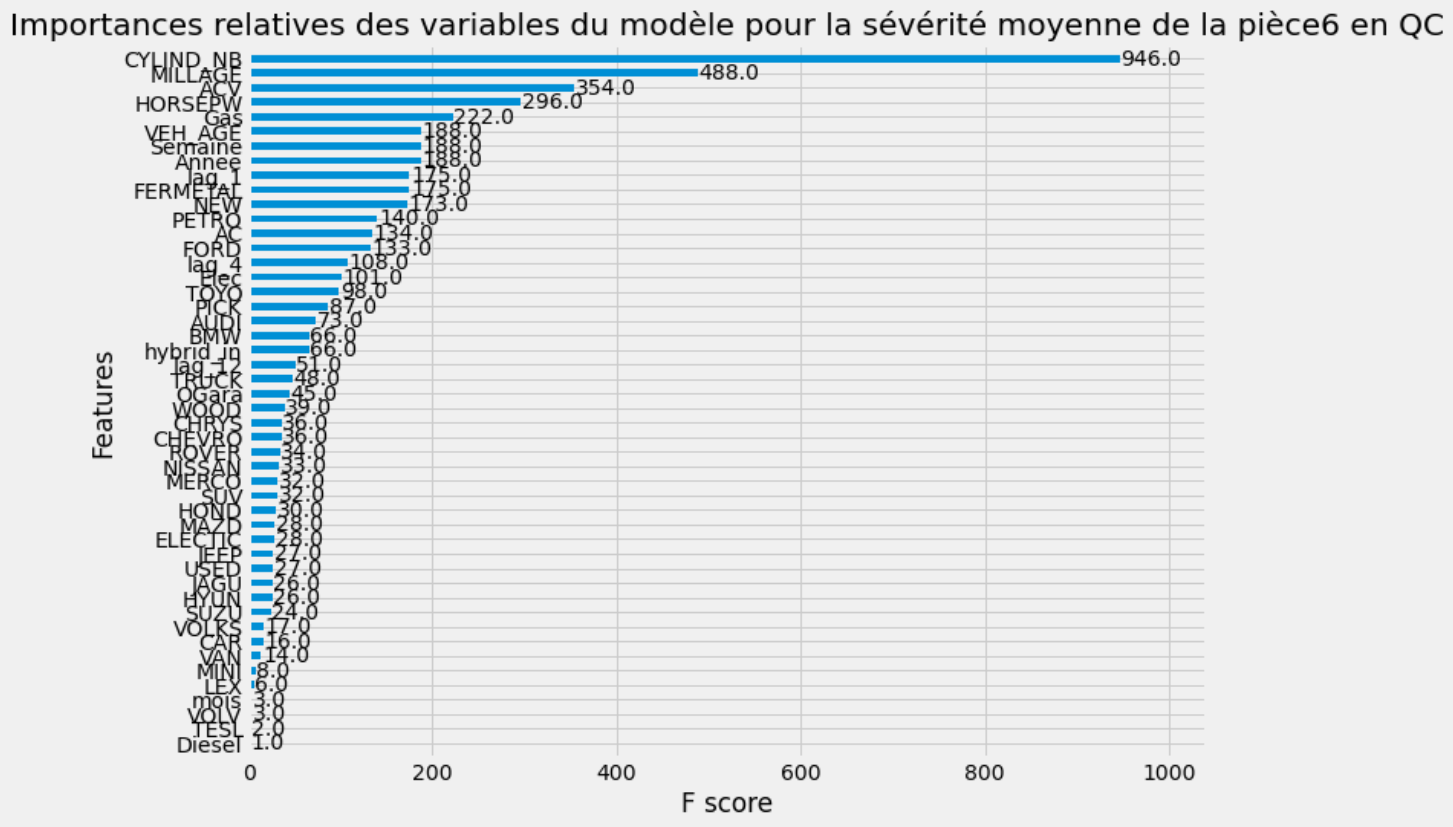
Dans les deux graphiques, les courbes de prédictions ne semblent capturer ni la tendance, ni le niveau des données réelles. Le modèle champion pour la modélisation de la prévision de la demande totale pour la pièce 6 est donc le modèle linéaire généralisé Poisson.

Enfin, la dernière partie de l’analyse portera sur les facteurs ayant eu le plus d’impact sur les modèles gagnants. Le modèle gagnant pour la sévérité est le XGB avec fonction de perte quadratique. La représentation graphique des résultats est présentée ci-dessous.

*Figure 5.17 : Niveau d'importances accordées par le modèle XGB à fonction de perte quadratique aux variables d’entrées dans le cadre de la prédiction de la sévérité moyenne hebdomadaire de la pièce 6 en Ontario*

****

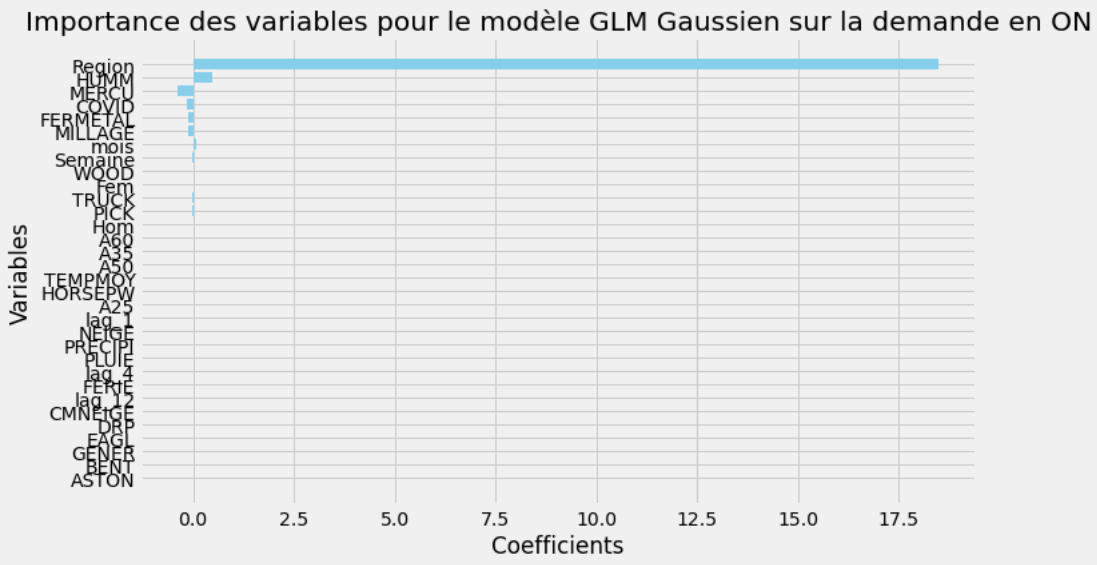
*Figure 5.18 : Niveau d'importances accordées par le modèle XGB à fonction de perte quadratique aux variables d’entrées dans le cadre de la prédiction de la sévérité moyenne hebdomadaire de la pièce 6 au Québec*

****

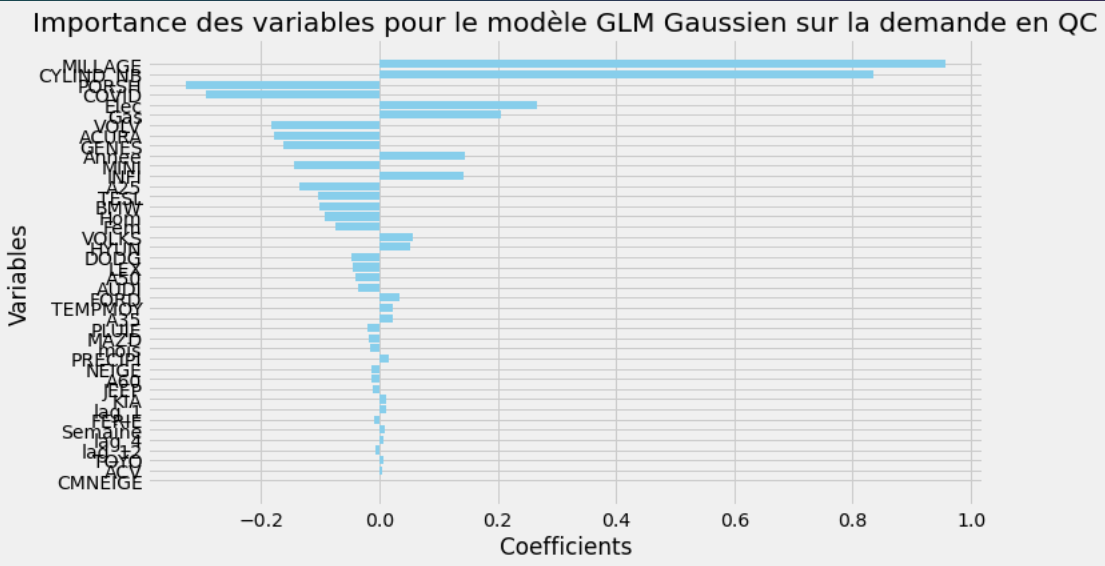
Les caractéristiques semblant le plus souvent revenir en tête de fil sont : le nombre de cylindre d’un véhicule, le millages parcourus, la puissance en ‘’horsepower’’, l’âge du véhicule, certaines marques, le cout du prix du des métaux ferreux, et les garages de type ‘’AC’’ pour Centre Auto.

Pour ce qui est des caractéristiques dominantes du modèle gagnant pour la demande totale hebdomadaire de la pièce 6 dans les deux régions, les résultats sont présentés dans les prochaines figures pour l’Ontario, d’abord et pour le Québec ensuite :

*Figure 5.19 : Niveau d'importances accordées par le modèle GLM Poisson aux variables d’entrées dans le cadre de la prédiction de la demande totale hebdomadaire pour la pièce 6 en Ontario*



*Figure 5.20 : Niveau d'importances accordées par le modèle GLM Poisson aux variables d’entrées dans le cadre de la prédiction de la demande totale hebdomadaire pour la pièce 6 au Québec*



Les caractéristiques semblant le plus souvent revenir en tête de fil sont : le millage parcourus encore une fois, la covid est aussi bel et bien reflétée dans les résultats, étant corrélé négativement avec la demande on en déduit que l’impact des semaine covid a occasionné une baisse de la demande pour la pièce 6. Les femmes semblent avoir un moins grand impact que les hommes sur la demande de façon générale. Quelques caractéristiques sur la période de l’année comme la semaine, le mois et la température moyenne semble aussi avoir une importance sur la demande de la pièce 6. Au Québec, les véhicules électriques semblent avoir un fort impact sur la demande pour la pièce 6. Finalement, certaines marques de véhicules comme Infinity ou Hyundai semble aussi avoir un impact.

# Chapitre 6 : Conclusion et discussion

En conclusion, lors de ce projet, il a été question de modéliser et d’inférer sur la sévérité moyenne et la fréquence des réclamations encourues depuis le premier janvier 2018. Pour ce faire, un jeu de données a tout d’abord été construit à l’aide de données collectées provenant de différentes sources de données non structurées. Par la suite, les données ont été jointes et structurées pour faciliter le reste de l’analyse. Dans un troisième temps, les données ont été nettoyées et agrégé de manière à pouvoir y appliquer différents outils statistiques de modélisation disponibles dans les environnements de R et de Python.

Une analyse a ensuite été menée sur les corrélations entre les données ce qui a mené à opérer une série de techniques de réduction de la dimensionalité des données afin de ne pas surcharger les modèles. Les données portant sur 5 pièces distincts ont ensuite été sélectionnées pour les régions du Québec et de l’Ontario par soucis de capacité computationnelle. Dans un troisième temps, deux types de famille de modèle ont été ajustés aux jeux de données. La première famille de modèle était la famille des modèles linéaires généralisés et la deuxième famille était celle des régressions avec méthodes d’arbres.

Ces deux familles de modèles ont été ajustées de manière à être utilisées comme modèle de prévision de la fréquence totale hebdomadaire et de la sévérité moyenne hebdomadaire avec erreurs autorégressives. Les performances des modèles ont ensuite été comparées grâce au score du RMSE et le score de l’autocorrélation.

Finalement, les résultats de ce projet ont été consignés dans ce rapport. Il a été obtenu dans un premier temps que le modèle champion pour modéliser la sévérité moyenne hebdomadaire était celui du XGB avec fonction de perte quadratique. Ce modèle avait des scores en validation et en test pour le critère du RMSE bien meilleur que celui du modèle linéaire généralisé gaussien. Et de plus, les scores d’autocorrélation étant très proche de 0 étaient de loin les meilleurs. Cependant, il est à noter que ce modèle semblait avoir quelques problèmes de surajustement car la valeur du score du RMSE augmentait de l’ajustement à l’ensemble de validation à l’ajustement sur l’ensemble test. Le modèle aurait donc possiblement des problèmes de généralisation.

En ce qui concerne le modèle champion pour la fréquence, le modèle linéaire généralisé poisson avait mieux performer sur les jeux de données que le modèle XGB avec fonction de perte de poisson. En termes de score sur les ensembles de validation et de test, les résultats semblaient mitigés. Le modèle linéaire Poisson semblait mieux performer que le modèle XGB à fonction de perte poisson sur l’ensemble de jeu de données de l’Ontario, mais pour l’ensemble de jeu de données du Québec, le XGBP semblait cette fois-ci mieux performer. En revanche, en regardant les scores d’autocorrélation il devenait évident que quelque chose n’allait pas avec le modèle XGBP qui obtenait un score de 0,8, soit le pire score de tous les modèles ajustés.

Finalement, les modèles champions ont été utilisés afin de répondre à l’objectif initial étant de mener une analyse permettant de mieux comprendre les facteurs ayant un impact sur les tendances de la sévérité moyenne, et la fréquence observée au cours des dernières années. Pour la sévérité, les facteurs obtenus sont : le nombre de cylindre d’un véhicule, le millage parcouru, la puissance en ‘’horsepower’’, l’âge du véhicule, certaines marques, le cours du prix des métaux ferreux, et les garages de type ‘’AC’’ pour centre auto. Pour ce qui est de la fréquence, les facteurs semblant avoir un plus grand impact sur les prévisions sont : le millage parcouru encore une fois, quelques caractéristiques sur la période de l’année comme la semaine et le mois et la température moyenne. Finalement certaines marques de véhicule au Québec comme les Infinity et certains types de véhicules en Ontario comme les trucks et les pickups semblent avoir un effet significatif sur la prévision de la demande totale hebdomadaire.

La réalisation de ce projet a suscité plusieurs grands défis majeurs.

Les accès :

Depuis l'adoption de la loi 25 portant sur la législation en matière de protection des renseignements personnels en septembre 2021, toutes les entreprises opérant sur le sol québécois ont dû se conformer et s’adapter à une série de nouvelles mesures afin de protéger les données de leurs consommateurs et clients (Journal de Montréal, 2023). Une première vague de mesures a eu lieu en septembre 2022. Récemment, une deuxième vague de mesures est tombée en septembre 2023. Les entreprises ont été appelées à évaluer les risques de la collecte d’informations personnelles, à obtenir à nouveau le consentement de leurs utilisateurs pour la collecte de données et à supprimer ou masquer les informations personnelles collectées (Journal de Montréal, 2023). TD étant très soucieuse et consciente de l'importance de protéger les données de ces utilisateurs, a récemment resserrée sa politique à l’interne sur l’accès à certaines informations.

D’autres défis concernant les problèmes d’accès étaient aussi notables. Il n’était souvent pas possible d’avoir accès à certaines librairies qui n’étaient pas encore reconnus par la compagnie. Il a fallu avoir recours à l’installation d’environnement virtuel et d’autres méthodes afin de pouvoir utiliser certains d’entre eux. Parfois, il arrivait que l’accès aux tables de données utilisées dans les requêtes soient temporairement non disponibles. Ce qui engendrait beaucoup de retard.

Les limites computationnelles :

Dans un autre ordre d’idée, nous avons rencontré aussi certains défis portant sur les ressources computationnelles qui m’étaient allouées. Le premier défi a été de pouvoir trouver une façon de nettoyer les noms des pièces d’auto et de les regrouper en catégorie afin de réduire la dimensionnalité du jeu de données. En effet, les noms des pièces étaient en français ou en anglais et comportaient souvent des typos ou des noms génériques propres à une marque ou un modèle de véhicule. Dû à des contraintes de l’entreprise quant aux accès des api google, il ne m’était pas non plus possible de créer un code automatisant la traduction et la correction des noms de pièces automatiquement. J’ai donc dû monitorer manuellement la traduction des pièces d’auto et utiliser des techniques d’apprentissage automatiques afin d’arriver à regrouper les pièces similaires entre elles. De plus, il arrivait parfois qu’il y ait des valeurs aberrantes. Un exemple de cas est lorsqu’une entrée de pièce correspondait à la valeur totale de l’estimer des réparations pour les pièces. Il m’a donc fallu trouver une solution pour repérer ses valeurs aberrantes et les exclure du jeu de données.

Un autre défi de taille a été de gérer la dimensionnalité des données en fonction des capacités computationnelles de la compagnie. Il aurait été intéressant d’entraîner les modèles sur plus de semaines d’observations afin d’obtenir possiblement un meilleur ajustement. Cependant, créer des requêtes avec des données et essayer d’effectuer des tests en même temps afin de valider l’exactitude des manipulations de structurations et de nettoyages de données effectués relève d’un défi de taille lorsque nous manipulons de grande base de données et sommes limités en termes de capacité computationnelle. Afin de d’éviter de surcharger le pipeline de l’entreprise, il était souvent nécessaire d’attendre après les heures de bureau ou très tard la nuit afin de mettre en marche les requêtes créer pour ce projet. Il devenait difficile d’ajuster et d’optimiser les paramètres de ceux-ci car cela nécessitait souvent un trop grand coût computationnel pour ce qui m’était alloué. Il ne nous était pas non plus possible de faire appel à une capacité computationnelle supplémentaire comme il est possible de le faire avec des interfaces comme ‘’google colab’’, il m’arrivait souvent de dépasser la limite de CPU qui m’était alloué lors de l’entraînement de mes modèles. Nous recevions souvent des avertissements de l’équipe de maintenance du pipeline pour que nous arrêtions le déroulement de nos requêtes de code. Il arrivait parfois même qu’ils arrêtent eux-mêmes les exécutions des requêtes sans prévenir.

Plus d’observations sur les semaines nous auraient permis potentiellement de créer plus de variables de décalages afin de ‘’stationnariser’’ notre série temporelle et possiblement d’améliorer les scores d’autocorrélation obtenus. On aurait aussi par le fait même, diminué le nombre de de valeurs manquantes pour certaines semaines dans notre jeu de données et dans certains cas, évité les imputations à 0 qui ont été faites et qui ont possiblement impactées les capacités prédictives des modèles ajustés. Cela aurait grandement permis aux modèles de mieux saisir les niveaux et les tendances des séries observées, d’établir de meilleures prévisions et d’ainsi améliorer les scores en validation et en test.

Recommandations et améliorations possible :

Il aurait été intéressant d’avoir eu accès à une plus grande capacité computationnelle afin dans un premier temps, de pouvoir agréger plus de données historiques. Cela aurait possiblement permis un meilleur entraînement des modèles déployés. De plus, cela aurait aussi permis d’améliorer le processus de recherche de paramètres optimaux pour nos modèles.

Il aurait aussi été intéressant de générer des interactions entre certaines variables dans notre modèle de base, soit le GLM. Cela aurait possiblement permis d’obtenir de meilleurs résultats en entraînement et en test.

Finalement, Il aurait été aussi intéressant d’incorporer au sein de l’entreprise une méthodologie afin de mieux monitorer l’entrée des informations. Cela aurait permis d'augmenter la précision et l’exactitude de nos résultats.

# Bibliographie

AXA. [s.d.]. *Qu’entend-on par la notion d’IARD ? Lexique Assurance Auto.* Récupéré de

<https://www.axa.fr/assurance-auto/iard.html>

Belaidi, N. [s.d.]. *Word Embedding & NLP* : *Définition, exemples. Formation Tech et Data en ligne | Blent.ai.* Récupéré de

<https://blent.ai/blog/a/word-embedding-nlp-definition>

Boucher, Jean-Phillipe [2019]. *ACT2040 Chapitre 1 : Introduction à l’assurance IARD*. UQAM, 21 p.

Bucci, Silvia [2021]. *Étude et implémentation de techniques d’analyse de sensibilité dans les modèles de tarification Non-Vie. Application à la tarification à l’adresse.*,216 p. Récupéré de

<https://www.institutdesactuaires.com/docs/mem/f7ec303ec9df8c23bcb407657d109184.pdf>

Carlos Felippe Rostand Koetz [2018]. *Analyse et évaluation de la méthode de tarification a posteriori avec les données de panel,* 131 p. Récupéré de<https://archipel.uqam.ca/11765/1/M15665.pdf>

Charlin, Laurent [2021]. *Apprentissage Automatique I 80-629* HEC, 62 p.

Charpentier, Arthur [2013]. Actuariat *IARD - ACT2040 Partie 6 - modélisation des coûts individuels de sinistres,* 56 p. Récupéré de

<https://freakonometrics.hypotheses.org/files/2013/10/slides-2040-6-A2013.pdf>

Contributeurs aux projets Wikimédia. [2021]. *Formule de Haversine*. Récupéré de <https://fr.wikipedia.org/wiki/Formule_de_haversine#:~:text=La%20formule%20de%20haversine,-En%20prenant%20deux&text=d%20est%20la%20distance%20du,du%20point%202%2C%20en%20radians>

Data Analytics Post. [2018]. *Word2VEC - Data Analytics Post.* Récupé de <https://dataanalyticspost.com/Lexique/word2vec/>

Debbie, J. Dupuis [2020]. *Méthodes de prévision Séance 2 – Idées et outils de base,* HEC Montréal, 32 p.

Debbie, J. Dupuis [2021]. *Méthodes de prévision Séance 3 – Outils de base, opinions d’experts et prévisions naïves,* HEC Montréal, 26 p.

Debbie, J. Dupuis [2022]. *Méthodes de prévision Séance 4 – Lissage exponentiel I,* HEC Montréal, 20 p.

Debbie, J. Dupuis [2021]. *Méthodes de prévision Séances 6 & 7 – Régression linéaire multiple I & II,* HEC Montréal, 27 p.

Denis Larocque, Aurélie Labbe [2022]. *80619 Méthodes avancées en exploitation des données boosting*. HEC Montréal, 68 p.

Denis Larocque, Aurélie Labbe [2022]. *80619 Méthodes avancées en exploitation de données Introduction,* HEC Montréal, 54 p.

Denis Larocque, Aurélie Labbe [2022]. *80619 Méthodes avancées en exploitation de données Méthodes basées sur les arbres et forêts aléatoires*, *(RF)*, HEC Montréal, 101 p.

Denis Larocque, Aurélie Labbe [2022]. *80619 Méthodes avancées en exploitation de données Régularisation et sélection de variables pour les modèles de régression*, HEC Montréal, 156 p.

Ester et al. [1996]. *A Density-Based Algorithm for Discovering Clusters,* 6 p.Récupéré de

<https://cdn.aaai.org/KDD/1996/KDD96-037.pdf>

Government of Canada, Statistics Canada [2022]. *Le Quotidien-La pandémie de COVID-19 a-t-elle changé les tendances du navettage pour de bon ?* Récupéré de

<https://www150.statcan.gc.ca/n1/daily-quotidien/221130/dq221130c-fra.htm>

Institut national de santé publique du Québec [2022]. *Ligne du temps COVID-19 au Québec| Institut national de santé publique du Québec*. Récupéré de <http://www.inspq.qc.ca/covid-19/donnees/ligne-du-temps>

Le Journal de Montréal [s.d]. *Loi 25 : Des données personnelles à protéger davantage.* Récupéré de

<https://www.journaldemontreal.com/2023/09/21/loi-25--des-donnees-personnelles-a-proteger-davantage>

Li, Z. [2021]. *A Beginner’s guide to Word embedding with Gensim Word2VEC model. Medium.* Récupéré de

<https://towardsdatascience.com/a-beginners-guide-to-word-embedding-with-gensim-word2vec-model-5970fa56cc92>

Mathieu, C. [2021]. *COVID-19 : une opportunité à saisir pour les assureurs automobiles. Portail de l’assurance*. Récupéré de

<https://portail-assurance.ca/dommages/covid-19-une-opportunite-a-saisir-pour-les-assureurs-automobiles/>

Miseray, Antoine et Frédéric Planchet [2017]. *Tarification IARD Introduction aux techniques avancées*, 90 p. Récupéré de

<https://www.ressources-actuarielles.net/C1256F13006585B2/0/457A36A8ECC541AEC1257D740067EEC4/%24FILE/GLM_FP.pdf>

Nugent, Ugo [2019]. *ACT5400 : Crédibilité, UQAM*, 96 p.

OpenClassrooms[s.d]. *Effectuez des plongements de mots (word embeddings)*. Récupéré de

<https://openclassrooms.com/fr/courses/4470541-analysez-vos-donnees-textuelles/4855006-effectuez-des-plongements-de-mots-word-embeddings>

Pierre-Olivier Goffard [2019]. *Modélisation Charge Sinistre M2 Actuariat,* 42 p. Récupéré de

<https://pierre-olivier.goffard.me/Teaching/MCS_ISFA/Chap3_modele_collectif.pdf>

Planchet, Frédéric et Guillaume Serdeczny [2014]. *Modèles fréquence – coût : Quelles perspectives d’évolution?*, 45 p. Récupéré de

<https://www.institutdesactuaires.com/global/gene/link.php?doc_id=1231&fg=1>

Robert, J. [2023]. *Machine Learning & Clustering: Focus sur l’Algorithme DBSCAN. Formation Data Science | DataScientest.com.* Récupéré de

<https://datascientest.com/machine-learning-clustering-dbscan#:~:text=Le%20DBSCAN%20est%20un%20algorithme,J%C3%B6rg%20Sander%20et%20Xiawei%20Xu>

Spoendlin, F. [2023]. *An overview of clustering algorithms | Oxford Protein Informatics Group*. Récupéré de

<https://www.blopig.com/blog/2023/04/an-overview-of-clustering-algorithms/>

Thioub, M. [2021]. 60603 *Apprentissage statistique – Chapitre 11 Analyse de regroupement*, HEC Montréal, 64 p.

# Annexe

**Code R (Traitement des données)**

*library(DBI)*

*library(dplyr)*

*library(dbplyr)*

*library(dfcr)*

*library(tcltk)*

*library(stringr)*

*library(reshape2)*

*library(lubridate)*

*library(chron)*

*library(RecordLinkage)*

*library(stringdist)*

*library(text)*

*library(tidytext)*

*library(cluster)*

*library(dbscan)*

*library(slam)*

*library(lsa)*

*library(textTinyR)*

*library(fpc)*

*library(tm)*

*library(proxy)*

*library(factoextra)*

*library(cluster)*

*library(textcat)*

*library(corrplot)*

*library(caret)*

*library(xgboost)*

*library(rpart)*

*library(adabag)*

*library(gbm)*

*library(ggplot2)*

*library(catboost)*

*# Connection au pipline*

*setwd('~/work/RPROJECT')*

*doConn("odbc")*

*sc <- dbConnect(odbc::odbc(),"Impala")*

*############ Création de fonctions utiles ################*

*# Fonction pour nettoyer des strings*

*rm\_special\_char = function(phrase){*

*phrase = gsub("[\r\n]", " ", phrase)*

*phrase = toupper(phrase)*

*phrase = gsub('-M', '', phrase)*

*#phrase = gsub('\\b[LR]\\b','',phrase)*

*phrase = sub("\\s{2,}.\*","",phrase)*

*phrase = gsub('PRIOR DAMAGE:', '', phrase)*

*phrase = gsub('PRIOR DAMAGES', '', phrase)*

*phrase = gsub('DAMMAGE:', '', phrase)*

*phrase = gsub('DAMMAGES - ', '', phrase)*

*phrase = gsub('DAMMAGES: ', '', phrase)*

*phrase = sub('(^| )?(R|L|D|G)( |$)?', '', phrase)*

*phrase = sub(' +', ' ', phrase)*

*phrase = sub('^ | $', '', phrase)*

*phrase = gsub(' G ', '', phrase)*

*phrase = str\_replace\_all(phrase, "[^[:alnum:]]", " ")*

*phrase = iconv(phrase, from = 'UTF-8', to = 'ASCII//TRANSLIT')*

*phrase = gsub('[0-9]+', '', phrase)*

*phrase = str\_replace\_all(phrase, " P ", "")*

*phrase = str\_replace\_all(phrase, " R ", "")*

*phrase = str\_replace\_all(phrase, " N ", "")*

*phrase = str\_replace\_all(phrase, " T ", "")*

*phrase = str\_replace\_all(phrase, " S ", "")*

*phrase = str\_replace\_all(phrase, " R ", "")*

*phrase = str\_replace\_all(phrase, " L ", "")*

*phrase = str\_replace\_all(phrase, " D ", "")*

*phrase = str\_replace\_all(phrase, " G ", "")*

*phrase = str\_trim(phrase)*

*return(phrase)*

*}*

*####################################################*

*# Fonction pour nettoyer des strings (pièces d'auto)*

*rm\_special\_char2 = function(phrase){*

*phrase = str\_extract(phrase,'\\(([^\\)]+)\\)')*

*phrase = gsub('[()]','',phrase)*

*phrase = gsub("[\r\n]", " ", phrase)*

*phrase = toupper(phrase)*

*phrase = gsub('-M', '', phrase)*

*#phrase = gsub('\\b[LR]\\b','',phrase)*

*phrase = sub("\\s{2,}.\*","",phrase)*

*phrase = gsub('PRIOR DAMAGE:', '', phrase)*

*phrase = gsub('PRIOR DAMAGES', '', phrase)*

*phrase = gsub('DAMMAGE:', '', phrase)*

*phrase = gsub('DAMMAGES - ', '', phrase)*

*phrase = gsub('DAMMAGES: ', '', phrase)*

*phrase = sub('(^| )?(R|L|D|G)( |$)?', '', phrase)*

*phrase = sub(' +', ' ', phrase)*

*phrase = sub('^ | $', '', phrase)*

*phrase = gsub(' G ', '', phrase)*

*phrase = str\_replace\_all(phrase, "[^[:alnum:]]", " ")*

*phrase = iconv(phrase, from = 'UTF-8', to = 'ASCII//TRANSLIT')*

*phrase = gsub('[0-9]+', '', phrase)*

*phrase = str\_replace\_all(phrase, "FRONT", "")*

*phrase = str\_replace\_all(phrase, "REAR", "")*

*phrase = str\_replace\_all(phrase, "UPPER", "")*

*phrase = str\_replace\_all(phrase, "LOWER", "")*

*phrase = str\_replace\_all(phrase, "UNDER", "")*

*phrase = str\_replace\_all(phrase, "CENTER", "")*

*phrase = str\_replace\_all(phrase, "SIDE", "")*

*phrase = str\_replace\_all(phrase, "LEFT", "")*

*phrase = str\_replace\_all(phrase, "RIGHT", "")*

*phrase = str\_replace\_all(phrase, "OUTER", "")*

*phrase = str\_replace\_all(phrase, "INSIDE", "")*

*phrase = str\_replace\_all(phrase, "EXTERIOR", "")*

*phrase = str\_replace\_all(phrase, "LATERAL", "")*

*phrase = str\_replace\_all(phrase, "YES", "")*

*phrase = str\_replace\_all(phrase, "NOT AN AUTO PART", "")*

*phrase = str\_replace\_all(phrase, "NO", "")*

*phrase = str\_replace\_all(phrase, "MIDDLE", "")*

*phrase = str\_replace\_all(phrase, "INTERIOR", "")*

*phrase = str\_replace\_all(phrase, "EXTERIOR", "")*

*phrase = str\_replace\_all(phrase, "INNER", "")*

*phrase = str\_replace\_all(phrase, "CENTRAL", "")*

*phrase = str\_replace\_all(phrase, "A C", "AC")*

*phrase = str\_replace\_all(phrase, "INTERNAL", "")*

*phrase = str\_replace\_all(phrase, "EXTERNAL", "")*

*phrase = str\_replace\_all(phrase, " O ", "")*

*phrase = str\_replace\_all(phrase, " X ", "")*

*phrase = str\_replace\_all(phrase, "FRT ", "")*

*phrase = str\_replace\_all(phrase, " LT ", "")*

*phrase = str\_replace\_all(phrase, " F ", "")*

*phrase = str\_replace\_all(phrase, " M ", "")*

*phrase = str\_replace\_all(phrase, "SIDE", "")*

*phrase = str\_replace\_all(phrase, "LTTIRE", "TIRE")*

*phrase = str\_replace\_all(phrase, " LAT ", "")*

*phrase = str\_replace\_all(phrase, " AV ", "")*

*phrase = str\_replace\_all(phrase, " ALUM ", "ALUMINIUM")*

*phrase = str\_replace\_all(phrase, " RT ", "")*

*phrase = str\_replace\_all(phrase, " RF ", "")*

*phrase = str\_replace\_all(phrase, " D ", "")*

*phrase = str\_replace\_all(phrase, " LFT ", "")*

*phrase = str\_replace\_all(phrase, " LIEFT ", "")*

*phrase = str\_replace\_all(phrase, " L ", "")*

*phrase = str\_replace\_all(phrase, " LH ", "")*

*phrase = str\_replace\_all(phrase, " RR ", "")*

*phrase = str\_replace\_all(phrase, " SL ", "")*

*phrase = str\_replace\_all(phrase, " INT ", "")*

*phrase = str\_replace\_all(phrase, " AR ", "")*

*phrase = str\_replace\_all(phrase, " EXT ", "")*

*phrase = str\_replace\_all(phrase, " ELEV ", "")*

*phrase = str\_replace\_all(phrase, " TOWING ", "")*

*phrase = str\_replace\_all(phrase, " LOWER ", "")*

*phrase = str\_replace\_all(phrase, " RH ", "")*

*phrase = str\_replace\_all(phrase, " LT ", "")*

*phrase = str\_replace\_all(phrase, "QTR ", "")*

*phrase = str\_replace\_all(phrase, " LWR ", "")*

*phrase = str\_replace\_all(phrase, " AT ", "")*

*phrase = str\_replace\_all(phrase, " T ", "")*

*phrase = str\_replace\_all(phrase, " RE ", "")*

*phrase = str\_replace\_all(phrase, " INSTALL ", "")*

*phrase = str\_replace\_all(phrase, " AND ", "")*

*phrase = str\_replace\_all(phrase, " EA ", "")*

*phrase = str\_replace\_all(phrase, "VISION D ", "")*

*phrase = str\_replace\_all(phrase, "TOYO H ", "")*

*phrase = str\_replace\_all(phrase, " PR ", "")*

*phrase = str\_replace\_all(phrase, " K ", "")*

*phrase = str\_replace\_all(phrase, " GLUE ", "")*

*phrase = str\_replace\_all(phrase, " D ", "")*

*phrase = str\_replace\_all(phrase, " PRE ", "")*

*phrase = str\_replace\_all(phrase, " POST ", "")*

*phrase = str\_replace\_all(phrase, " B ", "")*

*phrase = str\_replace\_all(phrase, " R ", "")*

*phrase = str\_replace\_all(phrase, " MLR ", "")*

*phrase = str\_replace\_all(phrase, " OTR ", "")*

*phrase = str\_replace\_all(phrase, " PRO ", "")*

*phrase = str\_replace\_all(phrase, " V ", "")*

*phrase = str\_replace\_all(phrase, "CTR ", "")*

*phrase = str\_replace\_all(phrase, " RF ", "")*

*phrase = str\_replace\_all(phrase, " RR ", "")*

*phrase = str\_replace\_all(phrase, " DR ", "")*

*phrase = str\_replace\_all(phrase, " A ", "")*

*phrase = str\_replace\_all(phrase, "TAPE ", "")*

*phrase = str\_replace\_all(phrase, "ADHESIVE ", "")*

*phrase = str\_replace\_all(phrase, "REPLACE ", "")*

*phrase = str\_replace\_all(phrase, "ORD ", "")*

*phrase = str\_replace\_all(phrase, "FABRIC", "")*

*phrase = str\_replace\_all(phrase, " LH ", "")*

*phrase = str\_replace\_all(phrase, "U NUT", "")*

*phrase = str\_replace\_all(phrase, "I M LABLE", "")*

*phrase = str\_replace\_all(phrase, "GARDE PLANCHER GAUCHE", "")*

*phrase = str\_replace\_all(phrase, "GARDE PLANCHER DROIT", "")*

*phrase = str\_replace\_all(phrase, "INSTALLATION CONDUITE", "")*

*phrase = str\_replace\_all(phrase, "INSTALLATION PALIER", "")*

*phrase = str\_replace\_all(phrase, "LHL", "")*

*phrase = str\_replace\_all(phrase, "CLOISON", "")*

*phrase = str\_replace\_all(phrase, "JEEP", "")*

*phrase = str\_replace\_all(phrase, " RT ", "")*

*phrase = str\_replace\_all(phrase, "REPAIR", "")*

*phrase = str\_replace\_all(phrase, " A ", "")*

*phrase = str\_replace\_all(phrase, " A ", "")*

*phrase = str\_trim(phrase)*

*return(phrase)*

*}*

*####################################################*

*# Fonction pour calculer le mode*

*le\_mode = function(colonne){*

*unique\_value = unique(colonne)*

*counts = table(colonne)*

*mode\_nb = max(counts)*

*mode = unique\_value[counts == mode\_nb]*

*return(mode)*

*}*

*####################################################*

*data\_station = read.csv('Canadawide\_stationlist.csv',sep=',',header=TRUE)*

*data\_station\_ON = data\_station %>% filter(province\_cd == 'ON')*

*data\_station\_QC = data\_station %>% filter(province\_cd == 'QC')*

*####################################################*

*# Fonction de haversine pour calculer la distaces entre deux coorodonées*

*distance\_haversine <- function(latitude1, longitude1, latitude2,longitude2){*

*# convertir les donnees en donnees radian*

*latitude1 = latitude1 \* pi / 180*

*latitude2 = latitude2 \* pi / 180*

*longitude1 = longitude1 \* pi / 180*

*longitude2 = longitude2 \* pi / 180*

*rayon\_terre = 6371000*

*diff\_lat = latitude2-latitude1*

*diff\_long = longitude2 - longitude1*

*a = sin(diff\_lat/2)^2+cos(latitude1)\*cos(latitude2)\*sin(diff\_long/2)^2*

*c = 2 \* atan2(sqrt(a),sqrt(1-a))*

*distance = rayon\_terre \* c*

*return(distance)*

*}*

*####################################################*

*# Fonction permettant de déterminer les stations les plus proches (premières phases)*

*station\_proche <- function(latitude,longitude,region,data\_station\_ON,data\_station\_QC){*

*latitude=as.numeric(latitude)*

*longitude=as.numeric(longitude)*

*if(is.na(latitude)){*

*return(0)*

*}else if(region == 'ON'){*

*distances <- sapply(1:nrow(data\_station\_ON), function(i){*

*distance\_haversine(latitude, longitude, data\_station\_ON[i,'latitude'], data\_station\_ON[i,'longitude'])*

*})*

*index\_station = as.integer(which.min(distances))*

*station\_id\_proche = data\_station\_ON[index\_station,'station\_id']*

*return(station\_id\_proche)*

*}else if(region == 'QC'){*

*distances <- sapply(1:nrow(data\_station\_QC), function(i){*

*distance\_haversine(latitude, longitude, data\_station\_QC[i,'latitude'],data\_station\_QC[i,'longitude'])*

*})*

*index\_station = as.integer(which.min(distances))*

*station\_id\_proche = data\_station\_QC[index\_station,'station\_id']*

*return(station\_id\_proche)*

*}*

*}*

*####################################################*

*# Fonction permettant de déterminer les stations les plus proches (omission des stations sans données)*

*station\_proche\_2 <- function(loss\_dt,latitude,longitude,region,data\_station\_ON,data\_station\_QC,station\_sans\_donnee){*

*latitude=as.numeric(latitude)*

*longitude=as.numeric(longitude)*

*if(is.na(latitude)){*

*return(0)*

*}else if(region == 'ON'){*

*sans\_donnees = station\_sans\_donnee %>% filter(clm\_loss\_dt == loss\_dt, clm\_loss\_location\_state\_cd == 'ON') %>% select(station\_id\_p)*

*data\_station\_ONe = data\_station\_ON %>% anti\_join(sans\_donnees, by=c('station\_id'='station\_id\_p'))*

*distances <- sapply(1:nrow(data\_station\_ONe), function(i){*

*distance\_haversine(latitude, longitude, data\_station\_ONe[i,'latitude'], data\_station\_ONe[i,'longitude'])*

*})*

*index\_station = as.integer(which.min(distances))*

*station\_id\_proche = data\_station\_ONe[index\_station,'station\_id']*

*return(station\_id\_proche)*

*}else if(region == 'QC'){*

*sans\_donnees = station\_sans\_donnee %>% filter(clm\_loss\_dt == loss\_dt, clm\_loss\_location\_state\_cd == 'QC') %>% select(station\_id\_p)*

*data\_station\_QCe = data\_station\_QC %>% anti\_join(sans\_donnees, by=c('station\_id'='station\_id\_p'))*

*distances <- sapply(1:nrow(data\_station\_QCe), function(i){*

*distance\_haversine(latitude, longitude, data\_station\_QCe[i,'latitude'],data\_station\_QCe[i,'longitude'])*

*})*

*index\_station = as.integer(which.min(distances))*

*station\_id\_proche = data\_station\_QCe[index\_station,'station\_id']*

*return(station\_id\_proche)*

*}*

*}*

*####################################################*

*# importation de la table créer en SQL*

*data\_finance1 = sc %>% tbl(tblsan("michelf\_tdd\_claim\_financial")) %>%*

*group\_by(clm\_claim\_no) %>%*

*filter(paid\_am\_fin > 0) %>%*

*mutate(veh\_latitude\_no = as.numeric(veh\_latitude\_no),*

*veh\_longitude\_no = as.numeric(veh\_longitude\_no)) %>%*

*# Patch pour les données sans longitude ou lattitude (calcule de la coordonnée centrale par fsa)*

*left\_join(sc %>%*

*tbl(tblsan("codepostaux\_table")) %>%*

*filter((prov == 'QC'|| prov == 'ON')),*

*by=c('veh\_postal\_code\_rating\_cd'='pc\_2018')) %>%*

*mutate(fsa = case\_when(!is.na(fsa)~fsa,*

*TRUE~substr(veh\_postal\_code\_rating\_cd,1,3)),*

*latitude = case\_when(!is.na(as.numeric(latitude))~as.numeric(latitude),*

*TRUE~veh\_latitude\_no),*

*longitude = case\_when(!is.na(as.numeric(longitude))~as.numeric(longitude),*

*TRUE~veh\_longitude\_no)) %>%*

*ungroup() %>%*

*group\_by(fsa) %>%*

*mutate(fsa\_lat = mean(latitude,na.rm=TRUE),*

*fsa\_long= mean(longitude,na.rm =TRUE),*

*longitude1 = case\_when(!is.na(longitude)~longitude,*

*TRUE~fsa\_long),*

*latitude1 = case\_when(!is.na(latitude)~latitude,*

*TRUE~fsa\_lat)) %>%*

*ungroup() %>% distinct() %>%*

*mutate(checknumber = case\_when(!(checknumber == '') ~checknumber,TRUE ~'NA'),*

*checkno = case\_when(!is.na(checkno) ~checkno,TRUE ~'NA')) %>%*

*select(clm\_loss\_dt,*

*annee,*

*mois,*

*jour,*

*claim\_key,*

*clm\_claim\_no,*

*clm\_loss\_location\_state\_cd,*

*clm\_accident\_fiscal\_year\_nb,*

*veh\_make\_tx,*

*veh\_price\_client\_am,*

*veh\_credit\_score\_no,*

*veh\_vicc\_engine\_cylinder\_nb,*

*veh\_vicc\_weight\_by\_hp\_pc,*

*veh\_vicc\_engine\_fuel\_tx,*

*veh\_vicc\_engine\_hybrid\_in,*

*veh\_vicc\_horsepower\_nb,*

*dri\_age\_nb,dri\_gender\_cd,*

*dri\_marital\_status\_cd,*

*veh\_age\_nb,*

*veh\_status\_tx,*

*clm\_state\_cd,*

*clmpye\_contact\_type\_tx,*

*clm\_loss\_location\_city\_tx,*

*fsa,*

*longitude1,*

*latitude1) %>% as.data.frame() %>% distinct() %>%*

*####################################################*

*# à ne pas faire rouler*

*# data\_finance2 = data\_finance1 %>%*

*# rowwise() %>%*

*# mutate(station\_id\_p = station\_proche(latitude1,*

*# longitude1,*

*# clm\_loss\_location\_state\_cd,*

*# data\_station\_ON,*

*# data\_station\_QC)) %>%*

*# filter(station\_id\_p !=0)*

*# station\_download = data\_finance1 %>% ungroup() %>%*

*# select(station\_id\_p,annee) %>%*

*# group\_by(station\_id\_p) %>% arrange(station\_id\_p,annee) %>% distinct()*

*# write.csv(station\_download, 'stations\_a\_download.csv')*

*####################################################*

*# data\_finance2 = data\_finance2 %>%*

*# left\_join(read.csv('temperature\_info.csv',sep=',',header=TRUE) %>%*

*# select(-year,-X,-mois,-jour),*

*# by=c('clm\_loss\_dt'='date','station\_id\_p'='station'))*

*########## Chargement des stations sans données ########*

*station\_sans\_donnee24 = read.csv('sans\_donnee.csv',sep=',',header=TRUE) %>% select(-X)*

*##### Première détermination des station les plus proches ####*

*data\_finance2 = data\_finance1 %>%*

*rowwise() %>%*

*mutate(station = station\_proche\_2(clm\_loss\_dt,*

*latitude1,*

*longitude1,*

*clm\_loss\_location\_state\_cd,*

*data\_station\_ON ,*

*data\_station\_QC,station\_sans\_donnee24)) %>%*

*left\_join(read.csv('temperature\_info.csv',sep=',',header=TRUE) %>%*

*select(-year,-X,-mois,-jour),*

*by=c('clm\_loss\_dt'='date','station'))*

*#### Jointure informations sur la température #####*

*data\_finance2 = data\_finance2 %>%*

*left\_join(read.csv('temperature\_info.csv',sep=',',header=TRUE) %>%*

*select(-year,-X,-mois,-jour),*

*by=c('clm\_loss\_dt'='date','station'))*

*### Dernières patch des stationsans données par imputation par point central du fsa ###*

*station\_sans\_donnee25= data\_finance2 %>% filter(is.na(max\_temp)) %>% ungroup() %>%*

*select(station\_id\_p,clm\_loss\_dt,clm\_loss\_location\_state\_cd,fsa,clm\_loss\_location\_city\_tx) %>%*

*distinct() %>% mutate(clm\_loss\_location\_city\_tx = toupper(clm\_loss\_location\_city\_tx)) %>%*

*ungroup() %>%*

*select(clm\_loss\_dt,clm\_loss\_location\_city\_tx) %>% distinct()*

*data\_finance2 = data\_finance2 %>% mutate(clm\_loss\_location\_city\_tx = toupper(clm\_loss\_location\_city\_tx))*

*station\_sans\_donnee25 = station\_sans\_donnee25 %>%*

*left\_join(data\_finance2 %>% filter(!is.na(max\_temp)) %>% ungroup() %>%*

*select(clm\_loss\_dt,clm\_loss\_location\_city\_tx,station),*

*by=c('clm\_loss\_dt','clm\_loss\_location\_city\_tx')*

*) %>% group\_by(clm\_loss\_dt,clm\_loss\_location\_city\_tx) %>%*

*mutate(idx=row\_number()) %>% filter(idx==1) %>%*

*mutate(station2 =station) %>% select(-idx,-station) %>% filter(!is.na(station2))*

*data\_finance2 = data\_finance1 %>%*

*rowwise() %>%*

*mutate(station = station\_proche\_2(clm\_loss\_dt,*

*latitude1,*

*longitude1,*

*clm\_loss\_location\_state\_cd,*

*data\_station\_ON ,*

*data\_station\_QC,station\_sans\_donnee24)) %>%*

*left\_join(station\_sans\_donnee25,by=c('clm\_loss\_dt','clm\_loss\_location\_city\_tx')) %>%*

*mutate(station = case\_when(!is.na(station2)~station2,*

*TRUE~station)) %>% select(-station2) %>%*

*left\_join(read.csv('temperature\_info.csv',sep=',',header=TRUE) %>%*

*select(-year,-X,-mois,-jour),*

*by=c('clm\_loss\_dt'='date','station'))*

*write.csv(data\_finance2, 'mit4.csv')*

*# Jointure de l'information liée à l'indice de prix et aux estimés #*

*index\_price = read.csv('indice\_prix.csv',sep=';',header=TRUE)*

*data\_finance3 = data\_finance2 %>%*

*left\_join(read.csv('indice\_prix.csv',sep=';',header=TRUE) %>%*

*mutate(mois = case\_when(mois < 10 ~paste0(0,as.character(mois)),*

*TRUE~as.character(mois)),*

*annee = as.character(annee)),by=c('annee','mois')) %>%*

*left\_join(sc %>%*

*tbl(tblsan("catdigi1\_dai.g0051\_\_mitchell\_veh\_appraisal\_header")) %>%*

*filter(loss\_dt >= '2018-01-01',*

*loss\_dt <= '2023-10-01',*

*total\_loss\_flag == 'N') %>%*

*mutate(pod\_date=str\_c(substr(podium\_delivery\_date,1,4),"-",*

*substr(podium\_delivery\_date,5,6),"-",substr(podium\_delivery\_date,7,8)," ",*

*substr(podium\_delivery\_date,9,10),":",substr(podium\_delivery\_date,11,12),":",*

*substr(podium\_delivery\_date,13,14))) %>%*

*group\_by(claim\_num) %>%*

*mutate(*

*## Obtenir le dernier estimé ###*

*clm\_claim\_no = substr(claim\_num,1,9),*

*max\_splmt\_num = max(splmt\_num),*

*max\_pod\_date = max(pod\_date)*

*) %>%*

*filter(max\_splmt\_num == splmt\_num ,max\_pod\_date == pod\_date) %>% ungroup() %>%*

*group\_by(clm\_claim\_no,claim\_num,claim\_id) %>%*

*#Enlever les observations sans valeurs*

*mutate(l = length(repairfacilityzip\_code),l\_max = max(l)) %>%*

*filter(l == l\_max) %>%*

*select(clm\_claim\_no,*

*claim\_num,*

*claim\_id,*

*max\_splmt\_num,*

*veh\_year,*

*veh\_make\_desc,*

*veh\_model\_desc,*

*veh\_engine,*

*doors,*

*veh\_mileage\_cnt,*

*veh\_drivable\_flag,*

*prim\_point\_of\_impact\_desc,*

*acv\_amt,*

*total\_part\_amt,*

*veh\_type\_desc,*

*repairfacilityzip\_code) %>%*

*left\_join(sc %>%*

*tbl(tblsan("catdigi1\_dai.g0051\_\_mitchell\_veh\_appraisal\_line\_content")) %>%*

*group\_by(claim\_id,line\_num) %>%*

*mutate(max\_pod\_datel = max(podium\_delivery\_date),line\_num=as.numeric(line\_num)) %>%*

*filter(max\_pod\_datel==podium\_delivery\_date) %>%*

*select(claim\_id,*

*line\_num,*

*part\_qty,*

*part\_num,*

*part\_num\_dsc,*

*unit\_part\_price,oper\_type\_desc),by= c('claim\_id')) %>%*

*left\_join(sc %>%*

*tbl(tblsan("catdigi1\_dai.prod30tdd\_\_tdd\_claim\_vendor\_auto\_mitchell\_snap")) %>%*

*mutate(vdatk\_auto\_center\_yn = toupper(vdatk\_auto\_center\_yn),*

*vdavap\_drp\_yn = toupper(vdavap\_drp\_yn),*

*garage\_type = case\_when( vdatk\_auto\_center\_yn == 'YES'~'AC',*

*vdatk\_auto\_center\_yn != 'YES' & vdavap\_drp\_yn == 'YES'~'DRP',*

*TRUE~'Other')) %>%*

*select(vdavap\_claim\_no,garage\_type),by= c('clm\_claim\_no' = 'vdavap\_claim\_no'))%>%*

*distinct() %>% ungroup() %>% arrange(clm\_claim\_no,claim\_num) %>%*

*group\_by(clm\_claim\_no,claim\_num) %>% arrange(line\_num) %>% as.data.frame() ,by = c('clm\_claim\_no'))%>%*

*mutate(prix= as.numeric(unit\_part\_price))*

*# Pré-filtre sur les pièces et les valeurs extrêmes ou aberrantes #*

*data\_content = data\_finance3 %>%*

*filter(part\_qty > 0 , unit\_part\_price > 0,*

*oper\_type\_desc =='Operation - Remove Replace') %>%*

*distinct() %>%*

*mutate(part\_num\_dsc = rm\_special\_char(part\_num\_dsc),*

*unit\_part\_price= as.numeric(unit\_part\_price)) %>%*

*ungroup() %>%*

*select(-claim\_key,-claim\_id,-line\_num,-veh\_credit\_score\_no) %>%*

*group\_by(claim\_num) %>%*

*mutate(nb\_pieces = max(row\_number()),*

*omit\_parts =*

*case\_when(*

*unit\_part\_price > 4500 |unit\_part\_price < 75 ~1,*

*TRUE~0),*

*exception\_flag = case\_when(omit\_parts == 1 | (nb\_pieces < 5 & unit\_part\_price >4000)~1 , TRUE~0)) %>%*

*filter(!(str\_detect(part\_num\_dsc,'ESTIMATE') | str\_detect(part\_num\_dsc,'INVOICES') |*

*str\_detect(part\_num\_dsc,'SUBLET') | str\_detect(part\_num\_dsc, 'PART') |*

*str\_detect(part\_num\_dsc,'DEALER') | str\_detect(part\_num\_dsc, 'CHECK') |*

*str\_detect(part\_num\_dsc,'TOTAL') | str\_detect(part\_num\_dsc,'OEM') |*

*str\_detect(part\_num\_dsc, 'FINAL') | str\_detect(part\_num\_dsc,'SUPP') |*

*str\_detect(part\_num\_dsc,'LABOUR') | exception\_flag ==1 |*

*str\_detect(part\_num\_dsc,'ESTIMATION') |*

*str\_length(part\_num\_dsc) < 4))*

*## traitement du nom des pièces*

*data\_content2 = data\_content %>%*

*left\_join(read.csv('pièces\_auto\_2.csv',sep=';',header=TRUE), by=c('part\_num\_dsc'='origin\_name')) %>%*

*left\_join(read.csv('trie.csv',sep=';',header=TRUE) %>%*

*mutate(corrected\_name2=paste0("(",corrected\_name,")")) %>% select(-corrected\_name),*

*by=c('part\_num\_dsc')) %>% filter(!is.na(claim\_num)) %>%*

*mutate(corrected\_name= case\_when(!is.na(corrected\_name)~corrected\_name,*

*!is.na(corrected\_name2)~corrected\_name2,*

*TRUE~part\_num\_dsc)) %>% select(-corrected\_name2) %>%*

*mutate(corrected\_name = rm\_special\_char2(corrected\_name))*

*write.csv(data\_content2, 'data\_content.csv')*

*### Traitement du nom des pièces ######*

*data\_group\_pour\_py = data\_content2 %>%*

*ungroup() %>% select(prix,corrected\_name) %>% distinct()*

*write.csv(data\_group\_pour\_py ,'data\_out.csv')*

*## à rouler*

*data\_content2= data\_content2 %>%*

*left\_join(read.csv('parts\_grouping.csv',sep=',',header=TRUE),*

*by=c('corrected\_name','prix')) %>% select(-X)*

*write.csv(data\_content2, 'df\_final.csv')*

*###### Partie Analyse exploratoire #######*

*## chargement du jeu de données*

*data\_content2= read.csv('df\_final.csv',sep=',',header=TRUE)*

*data\_content2 = data\_content2 %>% mutate(semaine = week(clm\_loss\_dt),*

*prix = log(unit\_part\_price)) %>% select(-X)*

*#prix\_data = data\_content2 %>% filter(prix > 0) %>% ungroup() %>% select(prix)*

*ggplot(data\_content2,aes(x=unit\_part\_price))+geom\_histogram(binwidth = 100, fill = 'skyblue', color = 'black')+*

*labs(x='Prix',y = 'Nombre de pièces', title = 'Distribution des prices des pièces de véhicules')+*

*xlim(0,4500)+ ylim(0,250000)*

*#Choix des pièces ayant des valeurs pour les observations #*

*sev\_summarise = data\_content2 %>%*

*group\_by(groupe, clm\_loss\_location\_state\_cd,clm\_accident\_fiscal\_year\_nb) %>%*

*summarise(sev = round(sum(prix, na.rm=TRUE),2),*

*n\_frequence = n()) %>% filter(n\_frequence > 50) %>% ungroup() %>%*

*group\_by(groupe,clm\_loss\_location\_state\_cd) %>%*

*mutate(compteur = max(row\_number())) %>% filter(compteur == 6) %>% ungroup() %>%*

*group\_by(groupe,clm\_accident\_fiscal\_year\_nb) %>%*

*mutate(compteur2 = max(row\_number())) %>%filter(compteur2 == 2) %>% ungroup() %>%*

*group\_by(groupe,clm\_loss\_location\_state\_cd) %>%*

*mutate(sev\_tot = sum(sev), freq\_tot = sum(n\_frequence))*

*pieces\_choisies = sev\_summarise %>% group\_by(groupe,clm\_loss\_location\_state\_cd) %>%*

*summarise(sev\_tot = max(sev\_tot), freq\_tot = max(freq\_tot)) %>%*

*pivot\_wider(names\_from = c(clm\_loss\_location\_state\_cd),*

*values\_from = c(sev\_tot,freq\_tot)) %>% na.omit*

*data\_pieces = pieces\_choisies %>% select(groupe) %>% left\_join(data\_content2, by=c('groupe'))*

*######## Traitement de la base de données ##########*

*## Distributions des regions*

*sum(is.na(data\_pieces$clm\_loss\_location\_state\_cd))*

*comptes\_region = data\_pieces %>% group\_by(annee, clm\_loss\_location\_state\_cd) %>%*

*summarise(Occurences = n())*

*# visualisation*

*ggplot(comptes\_region,aes(x = annee, y = Occurences, fill = clm\_loss\_location\_state\_cd))+*

*geom\_bar(stat = "identity", position = "dodge")+*

*labs(title = 'Histogramme par annee et par region', x = NULL, y= NULL)+*

*theme(axis.text.y = element\_blank())*

*####### veh\_vicc\_engine\_cylinder\_nb ######*

*sum(is.na(data\_pieces$veh\_vicc\_engine\_cylinder\_nb))*

*mode\_engine\_cylinder = le\_mode(data\_pieces$veh\_vicc\_engine\_cylinder\_nb)*

*median\_engine\_cylinder = median(data\_pieces$veh\_vicc\_engine\_cylinder\_nb, na.rm = TRUE)*

*veh\_ajut = data\_pieces %>% filter(!is.na(veh\_vicc\_engine\_cylinder\_nb)) %>%*

*ungroup() %>% select(veh\_make\_desc,veh\_model\_desc,veh\_vicc\_engine\_cylinder\_nb) %>% distinct()*

*veh\_ajut1 = data\_pieces %>% filter(is.na(veh\_vicc\_engine\_cylinder\_nb)) %>%*

*ungroup() %>% select(veh\_make\_desc,veh\_model\_desc,veh\_vicc\_engine\_cylinder\_nb) %>% distinct()*

*### imputation par la médiane*

*veh\_ajut2 = veh\_ajut1 %>%*

*left\_join(veh\_ajut %>% mutate(veh\_vicc\_engine\_cylinder\_nb2=veh\_vicc\_engine\_cylinder\_nb) %>%*

*select(-veh\_vicc\_engine\_cylinder\_nb), by=c('veh\_make\_desc','veh\_model\_desc')) %>%*

*mutate(veh\_vicc\_engine\_cylinder\_nb3 = case\_when(!is.na(veh\_vicc\_engine\_cylinder\_nb2)~veh\_vicc\_engine\_cylinder\_nb2,*

*veh\_vicc\_engine\_cylinder\_nb2 > 12 ~as.integer(6),*

*veh\_make\_desc=='Harley-Davidson'~ as.integer(2),*

*veh\_make\_desc=='BMW'~as.integer(6),*

*TRUE~as.integer(median\_engine\_cylinder)))*

*data\_pieces = data\_pieces %>%*

*left\_join(veh\_ajut2 %>%*

*select(veh\_make\_desc,veh\_model\_desc,veh\_vicc\_engine\_cylinder\_nb3),*

*by=c('veh\_make\_desc','veh\_model\_desc'))%>%*

*mutate(veh\_vicc\_engine\_cylinder\_nb=case\_when(is.na(veh\_vicc\_engine\_cylinder\_nb)~veh\_vicc\_engine\_cylinder\_nb3,*

*TRUE~veh\_vicc\_engine\_cylinder\_nb),*

*veh\_vicc\_engine\_cylinder\_nb = case\_when(veh\_vicc\_engine\_cylinder\_nb >12~as.integer(round(veh\_vicc\_engine\_cylinder\_nb/10,0)),*

*TRUE~veh\_vicc\_engine\_cylinder\_nb)*

*) %>%*

*select(-veh\_vicc\_engine\_cylinder\_nb3) %>% distinct()*

*##### veh\_vicc\_weight\_by\_hp\_pc ######*

*sum(is.na(data\_pieces$veh\_vicc\_weight\_by\_hp\_pc))*

*veh\_ajut = data\_pieces %>% filter(!is.na(veh\_vicc\_weight\_by\_hp\_pc)) %>%*

*ungroup() %>% select(veh\_make\_desc,veh\_model\_desc,veh\_vicc\_weight\_by\_hp\_pc) %>% distinct()*

*veh\_ajut1 = data\_pieces %>% filter(is.na(veh\_vicc\_weight\_by\_hp\_pc)) %>%*

*ungroup() %>% select(veh\_make\_desc,veh\_model\_desc,veh\_vicc\_weight\_by\_hp\_pc) %>% distinct()*

*veh\_ajut2 = veh\_ajut1 %>%*

*left\_join(veh\_ajut %>% mutate(veh\_vicc\_weight\_by\_hp\_pc2=veh\_vicc\_weight\_by\_hp\_pc) %>%*

*select(-veh\_vicc\_weight\_by\_hp\_pc), by=c('veh\_make\_desc','veh\_model\_desc')) %>%*

*group\_by(veh\_make\_desc,veh\_model\_desc) %>% mutate(m = mean(veh\_vicc\_weight\_by\_hp\_pc2)) %>%*

*select(-veh\_vicc\_weight\_by\_hp\_pc2,-veh\_vicc\_weight\_by\_hp\_pc) %>% distinct()*

*data\_pieces2 = data\_pieces %>%*

*left\_join(veh\_ajut2 %>%*

*select(veh\_make\_desc,veh\_model\_desc,m),*

*by=c('veh\_make\_desc','veh\_model\_desc'))%>%*

*mutate(veh\_vicc\_weight\_by\_hp\_pc=case\_when(is.na(veh\_vicc\_weight\_by\_hp\_pc)~m,*

*TRUE~veh\_vicc\_weight\_by\_hp\_pc)) %>%*

*select(-m) %>% distinct() %>% filter(!is.na(veh\_vicc\_weight\_by\_hp\_pc))*

*# on perd 94 observations*

*# même en utilisant les donees historiques, il est difficiles de de patcher toutes les observations*

*# regardons combien ces observations représentent dans nos donnees.*

*###################### veh\_vicc\_engine\_fuel\_tx ##########################*

*sum(is.na(data\_pieces$veh\_vicc\_engine\_fuel\_tx))*

*## On va tronquer par gazoline le mode*

*comptes\_fuel = data\_pieces %>% group\_by(veh\_vicc\_engine\_fuel\_tx) %>%*

*summarise(Occurences = n())*

*ggplot(comptes\_fuel,aes(x = veh\_vicc\_engine\_fuel\_tx, y = Occurences))+*

*geom\_bar(stat = "identity", position = "dodge")+*

*#geom\_text(aes(label = Occurences), vjust = -0.5,position = position\_dodge(width = 0.9))*

*labs(title = 'Histogramme type de fuel', x = 'fuel', y= 'Occurences')*

*data\_pieces2 = data\_pieces2 %>%*

*mutate(veh\_vicc\_engine\_fuel\_tx =*

*case\_when(is.na(veh\_vicc\_engine\_fuel\_tx)~'Gasoline',*

*TRUE~veh\_vicc\_engine\_fuel\_tx))*

*########## garage\_type ###########*

*sum(is.na(data\_pieces$garage\_type))*

*comptes\_garages = data\_pieces %>% group\_by(garage\_type) %>%*

*summarise(Occurences = n())*

*ggplot(comptes\_garages,aes(x =garage\_type , y = Occurences,fill=garage\_type))+*

*geom\_bar(stat = "identity", position = "dodge")+*

*#geom\_text(aes(label = Occurences), vjust = -0.5,position = position\_dodge(width = 0.9))*

*labs(title = 'Type de garage', x = 'garage', y= 'Occurences')*

*data\_pieces2 = data\_pieces2 %>%*

*mutate(garage\_type =*

*case\_when(is.na(garage\_type)~'DRP',*

*TRUE~garage\_type))*

*############# veh\_vicc\_horsepower\_nb #############*

*sum(is.na(data\_pieces2$veh\_vicc\_horsepower\_nb))*

*veh\_ajut = data\_pieces2 %>% filter(!is.na(veh\_vicc\_horsepower\_nb)) %>%*

*ungroup() %>% select(veh\_make\_desc,veh\_model\_desc,veh\_vicc\_horsepower\_nb) %>% distinct()*

*veh\_ajut1 = data\_pieces2 %>% filter(is.na(veh\_vicc\_horsepower\_nb)) %>%*

*ungroup() %>% select(veh\_make\_desc,veh\_model\_desc,veh\_vicc\_horsepower\_nb) %>% distinct()*

*veh\_ajut2 = veh\_ajut1 %>%*

*left\_join(veh\_ajut %>% mutate(veh\_vicc\_horsepower\_nb2=veh\_vicc\_horsepower\_nb) %>%*

*select(-veh\_vicc\_horsepower\_nb), by=c('veh\_make\_desc','veh\_model\_desc')) %>%*

*group\_by(veh\_make\_desc,veh\_model\_desc) %>% mutate(m = as.integer(mean(veh\_vicc\_horsepower\_nb2))) %>%*

*select(-veh\_vicc\_horsepower\_nb2,-veh\_vicc\_horsepower\_nb) %>% distinct()*

*data\_pieces2 = data\_pieces2 %>%*

*left\_join(veh\_ajut2 %>%*

*select(veh\_make\_desc,veh\_model\_desc,m),*

*by=c('veh\_make\_desc','veh\_model\_desc'))%>%*

*mutate(veh\_vicc\_horsepower\_nb=case\_when(is.na(veh\_vicc\_horsepower\_nb)~m,*

*TRUE~veh\_vicc\_horsepower\_nb)) %>%*

*select(-m) %>% distinct() %>% filter(!is.na(veh\_vicc\_horsepower\_nb))*

*############# dri\_age\_nb #############*

*sum(is.na(data\_pieces2$dri\_age\_nb))*

*comptes\_age = data\_pieces2 %>% group\_by(dri\_age\_nb) %>%*

*summarise(Occurences = n())*

*ggplot(comptes\_age,aes(x = dri\_age\_nb, y = Occurences, fill = dri\_age\_nb))+*

*geom\_bar(stat = "identity", position = "dodge")+*

*# geom\_text(aes(label = Occurences), vjust = -0.5,position = position\_dodge(width = 0.9))*

*labs(title = 'Distribution de la fréquence accident par age ', x = 'Age', y= 'Occurences')*

*data\_pieces2$log\_age = log(data\_pieces2$dri\_age\_nb)*

*mean\_log = mean( data\_pieces2$log\_age, na.rm=TRUE)*

*sd\_log = sd( data\_pieces2$log\_age, na.rm = TRUE)*

*data\_pieces2 = data\_pieces2 %>%*

*mutate(dri\_age\_nb = case\_when(is.na(dri\_age\_nb)~as.integer(round(rlnorm(1, meanlog = mean\_log, sdlog = sd\_log))),*

*TRUE~dri\_age\_nb))*

*sum(is.na(data\_pieces2$dri\_age\_nb))*

*############# dri\_gender\_cd ############*

*sum(is.na(data\_pieces2$dri\_gender\_cd))*

*comptes\_genre = data\_pieces2 %>% group\_by(dri\_gender\_cd) %>%*

*summarise(Occurences = n()) %>% na.omit %>% ungroup() %>%*

*mutate(tot\_genres = sum(Occurences), prob =Occurences/tot\_genres )*

*ggplot(comptes\_genre,aes(x =dri\_gender\_cd, y = Occurences, fill = dri\_gender\_cd))+*

*geom\_bar(stat = "identity", position = "dodge")+*

*# geom\_text(aes(label = Occurences), vjust = -0.5,position = position\_dodge(width = 0.9))*

*labs(title = 'Histogramme de la distribution du sexe', x = 'sexe', y= 'Occurences')*

*## fonction discrete*

*genres = c('M','F')*

*prob = comptes\_genre$prob*

*data\_pieces2 = data\_pieces2 %>%*

*mutate(dri\_gender\_cd = case\_when(is.na(dri\_gender\_cd)~sample(genres,size = 1,prob = prob),*

*TRUE~dri\_gender\_cd))*

*################ veh\_status\_tx #############*

*sum(is.na(data\_pieces2$veh\_status\_tx))*

*veh\_ajut1 = data\_pieces2 %>% filter(is.na(veh\_status\_tx)) %>%*

*ungroup() %>% select(veh\_make\_desc,veh\_model\_desc,veh\_status\_tx,veh\_year,annee) %>%*

*mutate(veh\_age =annee-veh\_year,veh\_status\_tx2 = case\_when(veh\_age == 0 ~'New',*

*TRUE~'Used') ) %>%*

*select(-veh\_age) %>% distinct()*

*data\_pieces2 = data\_pieces2 %>%*

*left\_join(veh\_ajut1 %>%*

*select(veh\_make\_desc,veh\_model\_desc,veh\_year,annee,veh\_status\_tx2),*

*by=c('veh\_make\_desc','veh\_model\_desc','veh\_year','annee'))%>%*

*mutate(veh\_status\_tx=case\_when(is.na(veh\_status\_tx)~veh\_status\_tx2,*

*TRUE~veh\_status\_tx)) %>%*

*select(-veh\_status\_tx2) %>% distinct()*

*############ veh\_vicc\_engine\_hybrid\_in ############*

*sum(is.na(data\_pieces2$veh\_vicc\_engine\_hybrid\_in))*

*veh\_ajut = data\_pieces2 %>% filter(!is.na(veh\_vicc\_engine\_hybrid\_in)) %>%*

*ungroup() %>% select(veh\_make\_desc,veh\_model\_desc,veh\_vicc\_engine\_hybrid\_in) %>% distinct()*

*veh\_ajut1 = data\_pieces2 %>% filter(is.na(veh\_vicc\_engine\_hybrid\_in)) %>%*

*ungroup() %>% select(veh\_make\_desc,veh\_model\_desc,veh\_vicc\_engine\_hybrid\_in) %>% distinct()*

*veh\_ajut2 = veh\_ajut1 %>%*

*left\_join(veh\_ajut %>% mutate(veh\_vicc\_engine\_hybrid\_in2=veh\_vicc\_engine\_hybrid\_in) %>%*

*select(-veh\_vicc\_engine\_hybrid\_in), by=c('veh\_make\_desc','veh\_model\_desc')) %>%*

*group\_by(veh\_make\_desc,veh\_model\_desc) %>%*

*mutate(veh\_vicc\_engine\_hybrid\_in3= as.integer(max(veh\_vicc\_engine\_hybrid\_in2))) %>%*

*select(-veh\_vicc\_engine\_hybrid\_in2,-veh\_vicc\_engine\_hybrid\_in) %>% distinct()*

*data\_pieces2 = data\_pieces2 %>%*

*left\_join(veh\_ajut2 %>%*

*select(veh\_make\_desc,veh\_model\_desc,veh\_vicc\_engine\_hybrid\_in3),*

*by=c('veh\_make\_desc','veh\_model\_desc'))%>%*

*mutate(veh\_vicc\_engine\_hybrid\_in=case\_when(is.na(veh\_vicc\_engine\_hybrid\_in)~veh\_vicc\_engine\_hybrid\_in3,*

*TRUE~veh\_vicc\_engine\_hybrid\_in)) %>%*

*select(-veh\_vicc\_engine\_hybrid\_in3) %>% distinct()*

*sum(is.na(data\_pieces2$veh\_vicc\_engine\_hybrid\_in))*

*########################### dri\_marital\_status\_cd #######################*

*sum(is.na(data\_pieces2$dri\_marital\_status\_cd))*

*comptes\_marital = data\_pieces2 %>% group\_by(dri\_age\_nb ,dri\_gender\_cd ,dri\_marital\_status\_cd) %>%*

*summarise(Occurences = n()) %>% na.omit %>% ungroup() %>%*

*group\_by(dri\_age\_nb ,dri\_gender\_cd ) %>%*

*mutate(tot\_genres\_age = sum(Occurences), prob =Occurences/tot\_genres\_age )*

*ggplot(comptes\_marital,aes(x =dri\_marital\_status\_cd, y = Occurences, fill = dri\_marital\_status\_cd))+*

*geom\_bar(stat = "identity", position = "dodge")+*

*# geom\_text(aes(label = Occurences), vjust = -0.5,position = position\_dodge(width = 0.9))*

*labs(title = 'Histogramme de la distribution du statut matrimonial', x = 'matriarcal', y= 'Occurences')*

*marital\_fun = function(dri\_age\_nb,dri\_gender\_cd,data2){*

*data1 = data.frame(dri\_age\_nb,dri\_gender\_cd)*

*data1 = data1 %>% left\_join(data2 %>% select(-Occurences,-tot\_genres\_age),*

*by=c('dri\_age\_nb','dri\_gender\_cd')) %>% distinct()*

*return(sample(data1$dri\_marital\_status\_cd,size = 1,prob = data1$prob))*

*}*

*data\_pieces2a = data\_pieces2 %>% filter(is.na(dri\_marital\_status\_cd)) %>%*

*rowwise() %>%*

*mutate(dri\_marital\_status\_cd2 = marital\_fun(dri\_age\_nb ,dri\_gender\_cd,comptes\_marital))*

*data\_pieces3 = data\_pieces2 %>%*

*left\_join(data\_pieces2a %>%*

*select(groupe,dri\_age\_nb,dri\_gender\_cd,clm\_loss\_dt,claim\_num,dri\_marital\_status\_cd2),*

*by=c('clm\_loss\_dt', 'claim\_num','groupe','dri\_age\_nb','dri\_gender\_cd')) %>%*

*mutate(dri\_marital\_status\_cd = case\_when(is.na(dri\_marital\_status\_cd)~dri\_marital\_status\_cd2,*

*TRUE~dri\_marital\_status\_cd)) %>%*

*select(-dri\_marital\_status\_cd2) %>% distinct()*

*############ veh\_age\_nb ############*

*data\_pieces3 = data\_pieces3 %>% select(-veh\_age\_nb)*

*data\_pieces2 = data\_pieces2 %>% select(-veh\_make\_tx,-veh\_price\_client\_am)*

*################ clm\_state\_cd ######################*

*data\_pieces3 = data\_pieces3 %>% select(-clm\_state\_cd)*

*################ clmpye\_contact\_type\_tx #############*

*data\_pieces3 = data\_pieces3 %>% select(-clmpye\_contact\_type\_tx)*

*####################### FSA ###################*

*sum(is.na(data\_pieces3$fsa))*

*essai = data\_pieces3[which(is.na(data\_pieces3$fsa)),]*

*ess = data\_content2[which(!is.na(data\_content2$fsa)),]*

*essai2 = essai %>% group\_by(clm\_loss\_location\_city\_tx) %>%*

*select(clm\_loss\_dt ,clm\_loss\_location\_city\_tx,fsa)%>% distinct() %>%*

*mutate(clm\_loss\_location\_city\_tx = toupper(clm\_loss\_location\_city\_tx)) %>%*

*left\_join(ess %>%mutate(longitude2=longitude1,latitude2=latitude1,*

*max\_temp2=max\_temp,min\_temp2=min\_temp,*

*moy\_temp2=moy\_temp,H\_deg2=H\_deg,*

*C\_deg2=C\_deg,tot\_rain2=tot\_rain,*

*tot\_snow2=tot\_snow,tot\_precipi2=tot\_precipi,*

*snow\_ground2=snow\_ground,*

*clm\_loss\_location\_city\_tx = toupper(clm\_loss\_location\_city\_tx),*

*fsa2 = fsa*

*) %>%*

*select(clm\_loss\_dt,clm\_loss\_location\_city\_tx,fsa2,longitude2,*

*latitude2,max\_temp2,min\_temp2,moy\_temp2,H\_deg2, C\_deg2,*

*tot\_rain2, tot\_snow2,tot\_precipi2, snow\_ground2) %>%*

*distinct(),*

*by=c('clm\_loss\_dt','clm\_loss\_location\_city\_tx')) %>%*

*mutate(idx = row\_number()) %>% filter(idx==1)*

*data\_pieces3 = data\_pieces3 %>%*

*mutate(clm\_loss\_location\_city\_tx = toupper(clm\_loss\_location\_city\_tx)) %>%*

*left\_join(essai2 %>% select(-idx,-fsa), by=c('clm\_loss\_dt','clm\_loss\_location\_city\_tx')) %>%*

*mutate(fsa = case\_when(is.na(fsa)~fsa2,*

*TRUE~fsa),*

*longitude = case\_when(is.na(longitude1)~longitude2,*

*TRUE~longitude1),*

*latitude = case\_when(is.na(latitude1)~latitude2,*

*TRUE~latitude1),*

*max\_temp = case\_when(is.na(max\_temp)~max\_temp2,*

*TRUE~max\_temp),*

*min\_temp = case\_when(is.na(min\_temp)~min\_temp2,*

*TRUE~min\_temp),*

*moy\_temp = case\_when(is.na(moy\_temp)~moy\_temp2,*

*TRUE~moy\_temp),*

*H\_deg = case\_when(is.na(H\_deg)~H\_deg2,*

*TRUE~H\_deg),*

*C\_deg = case\_when(is.na(C\_deg)~C\_deg2,*

*TRUE~C\_deg),*

*tot\_rain = case\_when(is.na(tot\_rain)~tot\_rain2,*

*TRUE~tot\_rain),*

*tot\_snow = case\_when(is.na(tot\_snow)~tot\_snow2,*

*TRUE~tot\_snow),*

*tot\_precipi = case\_when(is.na(tot\_precipi)~tot\_precipi2,*

*TRUE~tot\_precipi),*

*snow\_ground = case\_when(is.na(snow\_ground)~snow\_ground2,*

*TRUE~snow\_ground)) %>%*

*select(-fsa2,-longitude2,-longitude1, -latitude1,*

*-latitude2,-max\_temp2,-min\_temp2,-moy\_temp2,-H\_deg2, -C\_deg2,*

*-tot\_rain2, -tot\_snow2,-tot\_precipi2, -snow\_ground2) %>% distinct()*

*data\_pieces3 = data\_pieces3 %>% filter(!is.na(fsa))*

*############### clm\_loss\_location\_city\_tx ##############*

*data\_pieces4 = data\_pieces3 %>% na\_if("")*

*data\_pieces4 = data\_pieces4 %>% select(-clm\_loss\_location\_city\_tx)*

*############### max\_splmt\_num ##############*

*data\_pieces4 = data\_pieces4 %>% select(-max\_splmt\_num)*

*############## veh\_year #############*

*sum(is.na(data\_pieces4$veh\_year))*

*data\_pieces4 = data\_pieces4 %>% filter(!is.na(veh\_year))*

*############ veh\_make\_desc ##########*

*sum(is.na(data\_pieces4$veh\_make\_desc))*

*data\_pieces4 = data\_pieces4 %>% filter(!is.na(veh\_make\_desc))*

*############ acv\_amt ############*

*sum(is.na(data\_pieces4$acv\_amt))*

*sum(data\_pieces4$acv\_amt==0)*

*## patch*

*data\_acv1 = data\_pieces4 %>% filter(acv\_amt==0) %>%*

*ungroup() %>% select(veh\_make\_desc, veh\_year,veh\_mileage\_cnt)*

*data\_acv2 = data\_content2 %>% ungroup() %>%*

*mutate(veh\_age = annee-veh\_year) %>%*

*group\_by(veh\_make\_desc, veh\_model\_desc, veh\_age) %>%*

*mutate(m = as.integer(mean(acv\_amt,na.rm=TRUE)),*

*m2 = as.integer(mean(veh\_price\_client\_am,na.rm=TRUE))) %>%*

*ungroup() %>% group\_by(veh\_make\_desc,veh\_type\_desc,veh\_age) %>%*

*mutate(m3 = as.integer(mean(acv\_amt,na.rm=TRUE))) %>%*

*select(veh\_make\_desc,veh\_model\_desc, veh\_year,veh\_type\_desc,veh\_age,m,m2,m3) %>% distinct() %>%*

*mutate(m4 = case\_when(m > 100 ~m,*

*m2 > 200 ~m2,*

*TRUE~m3))*

*data\_pieces5 = data\_pieces4 %>% mutate(veh\_age = annee-veh\_year) %>%*

*left\_join(data\_acv2 %>%*

*select(veh\_make\_desc,veh\_model\_desc, veh\_year,veh\_type\_desc,veh\_age,m4) %>%*

*distinct(), by=c('veh\_make\_desc','veh\_model\_desc', 'veh\_year','veh\_type\_desc','veh\_age')) %>%*

*mutate(acv\_amt = case\_when(acv\_amt==0~as.numeric(m4),*

*TRUE~acv\_amt)) %>% select(-m4)*

*########### veh\_model\_desc ############*

*sum(is.na(data\_pieces5$veh\_model\_desc))*

*data\_pieces5 = data\_pieces5 %>% select(-veh\_model\_desc)*

*############## veh\_engine ###########*

*sum(is.na(data\_pieces5$veh\_engine))*

*data\_pieces5 = data\_pieces5 %>% select(-veh\_engine)*

*############### doors #################*

*sum(is.na(data\_pieces5$doors))*

*data\_pieces5 = data\_pieces5 %>% select(-doors)*

*############## veh\_mileage\_cnt ##########*

*sum(is.na(data\_pieces5$veh\_mileage\_cnt))*

*data\_pieces5 = data\_pieces5 %>% mutate(veh\_age = annee - veh\_year)*

*comptes\_mileage = data\_pieces5 %>% ungroup() %>%*

*select(veh\_age,veh\_mileage\_cnt) %>% group\_by(veh\_age) %>%*

*summarise(m = as.integer(mean(veh\_mileage\_cnt,na.rm =TRUE)))*

*data\_pieces5 = data\_pieces5 %>%*

*left\_join(comptes\_mileage %>%*

*select(veh\_age,m),*

*by=c('veh\_age'))%>%*

*mutate(veh\_age = case\_when(is.na(veh\_age)~m,*

*TRUE~veh\_age)) %>%*

*select(-m) %>% distinct()*

*#################### veh\_drivable\_flag ##############*

*data\_pieces5 = data\_pieces5 %>% select(-veh\_drivable\_flag)*

*################# prim\_point\_of\_impact\_desc ########*

*data\_pieces5 = data\_pieces5 %>% select(-prim\_point\_of\_impact\_desc)*

*############## veh\_type\_desc ####################*

*sum(is.na(data\_pieces5$veh\_type\_desc))*

*comptes\_ty = data\_pieces %>% group\_by(veh\_type\_desc) %>%*

*summarise(Occurences = n())*

*ggplot(comptes\_ty,aes(x = veh\_type\_desc, y = Occurences, fill=veh\_type\_desc))+*

*geom\_bar(stat = "identity")+*

*theme\_minimal()+*

*#geom\_text(aes(label = veh\_type\_desc), vjust = -0.5,size=4)+*

*labs(title = 'type de véhicule', x = 'Type véhicule', y= 'Occurences')+*

*theme(axis.text.x = element\_text(angle = 90,hjust = 0.8), plot.margin = margin(2,0,-1,0,'cm'))*

*data\_pieces5 = data\_pieces5 %>%*

*mutate(veh\_type\_desc = case\_when(is.na(veh\_type\_desc)~'Passenger Car',TRUE~veh\_type\_desc))*

*############# repairfacilityzip\_code ##########*

*data\_pieces5 = data\_pieces5 %>% select(-repairfacilityzip\_code,-part\_qty, -part\_num,-oper\_type\_desc)*

*data\_pieces5 = data\_pieces5 %>% select(- nb\_pieces, -omit\_parts)*

*data\_pieces5 = data\_pieces5 %>% select(-exception\_flag)*

*data\_pieces5 = data\_pieces5 %>% select(-log\_age)*

*####################################################*

*data\_Holidays = read.csv('Holidays\_Calendar.csv',sep=';',header=TRUE) %>%*

*filter(Region=='ALL'|Region=='QC'|Region =='ON') %>% select(Date) %>% distinct() %>% mutate(holiday =1)*

*################ variables severite #########*

*data= data\_pieces5 %>% left\_join(data\_Holidays, by=c('clm\_loss\_dt'='Date')) %>%*

*#filter(groupe==6|groupe==37|groupe==20|groupe==40|groupe==9|groupe==90) %>%*

*mutate(veh\_vicc\_engine\_fuel\_tx = as.factor(veh\_vicc\_engine\_fuel\_tx),*

*veh\_make\_desc = toupper(veh\_make\_desc),*

*veh\_make\_desc = str\_replace\_all(veh\_make\_desc, "TRUCK/VAN", ""),*

*veh\_make\_desc = str\_replace\_all(veh\_make\_desc, "TRUCK/V", ""),*

*veh\_make\_desc = str\_replace\_all(veh\_make\_desc, "TRUCK", ""),*

*veh\_make\_desc = str\_replace\_all(veh\_make\_desc, " A", ""),*

*veh\_make\_desc = str\_trim(veh\_make\_desc),*

*#veh\_make\_desc = as.factor(veh\_make\_desc),*

*#veh\_type\_desc = as.factor(veh\_type\_desc),*

*#garage\_type = as.factor(garage\_type),*

*#clm\_loss\_location\_state\_cd = as.factor(clm\_loss\_location\_state\_cd),*

*veh\_status\_tx = toupper(veh\_status\_tx),*

*holiday = case\_when(is.na(holiday)~0,TRUE~1),*

*#veh\_status\_tx = as.factor(veh\_status\_tx),*

*#fsa = as.factor(fsa),*

*prix= unit\_part\_price,*

*#standardisation des données*

*acv\_amt =(acv\_amt- mean(acv\_amt, na.rm=TRUE))/sd(acv\_amt, na.rm = TRUE),*

*Plastic\_rubber = (Plastic\_rubber- mean(Plastic\_rubber, na.rm=TRUE))/sd(Plastic\_rubber, na.rm = TRUE),*

*Lumber\_wood = (Lumber\_wood- mean(Lumber\_wood, na.rm=TRUE))/sd(Lumber\_wood, na.rm = TRUE),*

*Energy\_petroleum = (Energy\_petroleum- mean(Energy\_petroleum, na.rm=TRUE))/sd(Energy\_petroleum, na.rm = TRUE),*

*Ferrous\_metal = (Ferrous\_metal- mean(Ferrous\_metal, na.rm=TRUE))/sd(Ferrous\_metal, na.rm = TRUE),*

*non\_ferrous\_metal = (non\_ferrous\_metal- mean(non\_ferrous\_metal, na.rm=TRUE))/sd(non\_ferrous\_metal, na.rm = TRUE),*

*fabricated\_metal\_product = (fabricated\_metal\_product- mean(fabricated\_metal\_product, na.rm=TRUE))/sd(fabricated\_metal\_product, na.rm = TRUE),*

*Motorized.and.recreational.vehicles = (Motorized.and.recreational.vehicles- mean(Motorized.and.recreational.vehicles, na.rm=TRUE))/sd(Motorized.and.recreational.vehicles, na.rm = TRUE),*

*electrical\_product = (electrical\_product- mean(electrical\_product, na.rm=TRUE))/sd(electrical\_product, na.rm = TRUE),*

*veh\_mileage\_cnt =(veh\_mileage\_cnt- mean(veh\_mileage\_cnt, na.rm=TRUE))/sd(veh\_mileage\_cnt, na.rm = TRUE),*

*#groupe = as.factor(groupe)*

*dri\_age\_nb = case\_when(dri\_age\_nb <=25~'age\_25',*

*dri\_age\_nb <= 35 ~'age\_35',*

*dri\_age\_nb <= 50 ~'age\_50',*

*TRUE ~'age\_60'),*

*demande = 1*

*) %>%*

*select(annee,*

*semaine,*

*clm\_loss\_location\_state\_cd ,*

*groupe,*

*veh\_vicc\_engine\_hybrid\_in,*

*veh\_vicc\_engine\_cylinder\_nb,*

*veh\_vicc\_horsepower\_nb,*

*veh\_vicc\_engine\_fuel\_tx,*

*veh\_status\_tx,*

*veh\_type\_desc,*

*veh\_age,*

*veh\_mileage\_cnt,*

*acv\_amt,*

*veh\_make\_desc,*

*garage\_type,*

*dri\_gender\_cd,*

*dri\_age\_nb,*

*dri\_marital\_status\_cd,*

*tot\_rain,*

*tot\_snow,*

*tot\_precipi,*

*snow\_ground,*

*holiday,*

*covid\_in,*

*demande,*

*max\_temp,*

*min\_temp,*

*moy\_temp,*

*Lumber\_wood,*

*Energy\_petroleum,*

*Ferrous\_metal,*

*electrical\_product,*

*prix) %>%*

*rename(### sum*

*'x0'='annee',*

*"x1"="semaine",*

*'x2'= 'clm\_loss\_location\_state\_cd',*

*'x3'= 'groupe',*

*'x4' = 'veh\_vicc\_engine\_hybrid\_in',*

*'x5'='veh\_vicc\_engine\_cylinder\_nb',*

*'x6' = 'veh\_vicc\_horsepower\_nb',*

*'x7'= 'veh\_vicc\_engine\_fuel\_tx',*

*'x8'='veh\_status\_tx',*

*'x9'='veh\_type\_desc',*

*'x10'='veh\_age',*

*'x11'='veh\_mileage\_cnt',*

*'x12'='acv\_amt',*

*'x13'='veh\_make\_desc',*

*'x14'='garage\_type',*

*'x15'= 'dri\_gender\_cd',*

*'x16'='dri\_age\_nb',*

*'x17'='dri\_marital\_status\_cd',*

*'x18'='tot\_rain',*

*'x19'='tot\_snow',*

*'x20'='tot\_precipi',*

*'x21'='snow\_ground',*

*'x22'='holiday',*

*'x23'= 'covid\_in',*

*'x24' ='demande',*

*#### Moyenne ##########*

*'x25'= 'max\_temp',*

*'x26'= 'min\_temp',*

*'x27' ='moy\_temp',*

*'x28' = 'Lumber\_wood',*

*'x29' = 'Energy\_petroleum',*

*'x30' = 'Ferrous\_metal',*

*'x31' = 'electrical\_product',*

*'x32' = 'prix'*

*) %>%*

*na.omit*

*############### Encodage des variables ##########*

*data\_severite = data %>% filter(x3==6|x3==9|x3==20|x3==37|x3==40)*

*# gender*

*gender\_encoded = model.matrix(~x15-1, data = data\_severite)*

*#age*

*age\_encoded = model.matrix(~x16-1, data = data\_severite)*

*#marital*

*marital\_encoded = model.matrix(~x17-1, data=data\_severite)*

*#combustion*

*combust\_encoded = model.matrix(~x7 - 1, data = data\_severite)*

*#garage*

*garage\_encoded = model.matrix(~x14 - 1, data = data\_severite)*

*#veh\_status*

*status\_encoded = model.matrix(~x8 - 1, data = data\_severite)*

*#veh\_type*

*type\_encoded = model.matrix(~x9 - 1, data = data\_severite)*

*#make*

*make\_encoded = model.matrix(~x13 - 1, data = data\_severite)*

*data\_severite2 = data\_severite %>% ungroup() %>%*

*select(-x7,-x8,-x9,-x13,-x14,-x15,-x16,-x17)*

*autres\_num = as.matrix(data\_severite2)*

*data\_severite2 = cbind(gender\_encoded,age\_encoded,*

*marital\_encoded,combust\_encoded,*

*status\_encoded,garage\_encoded,*

*make\_encoded,type\_encoded,autres\_num)*

*data\_severite2 = data.frame(data\_severite2)*

*data\_severite2 = data\_severite2 %>% mutate(x0 = as.numeric(x0),*

*x1 = as.numeric(x1),*

*x2 = case\_when(x2=='ON'~1,TRUE~0),*

*x3 = as.numeric(x3))*

*#%>% select(-X.Intercept.)*

*data\_sum = data\_severite2[,c(1:80,85:105)]*

*data\_groupe = data\_severite2[,c(81,82,83,84)]*

*df = cbind(data\_groupe,data\_sum)*

*df= df %>% mutate(across(everything(), as.numeric))*

*n\_c = sapply(df ,is.numeric)*

*####### Aggrégation par semaine ##############*

*df1 = df %>% mutate(Annee = x0,Semaine = x1, Region = x2, groupe = x3) %>%*

*group\_by(Annee,Semaine,Region,groupe) %>%*

*dplyr::summarise(*

*hybrid\_in =sum(x4,na.rm = TRUE),*

*CYLIND\_NB =mean(x5,na.rm = TRUE),*

*HORSEPW =mean(x6,na.rm = TRUE),*

*VEH\_AGE = mean(x10,na.rm = TRUE),*

*MILLAGE = mean(x11,na.rm=TRUE),*

*ACV = mean(x12,na.rm=TRUE),*

*PLUIE = sum(x18,na.rm = TRUE),*

*NEIGE = sum(x19, na.rm = TRUE),*

*PRECIPI = sum(x20,na.rm = TRUE),*

*CMNEIGE = sum(x21,na.rm = TRUE),*

*FERIE = sum(x22,na.rm = TRUE),*

*COVID = max(x23),*

*Fem = sum(x15F,na.rm = TRUE),*

*Hom = sum(x15M,na.rm = TRUE),*

*A25 = sum(x16age\_25,na.rm = TRUE),*

*A35 = sum(x16age\_35,na.rm = TRUE),*

*A50 = sum(x16age\_50,na.rm = TRUE),*

*A60 = sum(x16age\_60,na.rm = TRUE),*

*Conj = sum(x17C,na.rm = TRUE),*

*Marie = sum(x17M,na.rm = TRUE),*

*Celib = sum(x17S,na.rm = TRUE),*

*Veuf= sum(x17W,na.rm = TRUE),*

*SMI = sum(x17U,na.rm = TRUE),*

*Diesel = sum(x7Diesel,na.rm = TRUE),*

*Elec= sum(x7Electric,na.rm = TRUE),*

*Gas = sum(x7Gasoline,na.rm = TRUE),*

*NGaz = sum(x7Natural.Gas,na.rm = TRUE),*

*Demo = sum(x8DEMO,na.rm = TRUE),*

*NEW = sum(x8NEW,na.rm = TRUE),*

*USED =sum(x8USED,na.rm = TRUE),*

*AC = sum(x14AC, na.rm = TRUE),*

*DRP = sum(x14DRP, na.rm = TRUE),*

*OGara = sum(x14Other, na.rm = TRUE),*

*ACURA = sum(x13ACURA,na.rm = TRUE),*

*ROMEO = sum(x13ALFA.ROMEO,na.rm = TRUE),*

*ASTON = sum(x13ASTON.MARTIN,na.rm = TRUE),*

*AUDI = sum(x13AUDI,na.rm = TRUE),*

*BENT = sum(x13BENTLEY,na.rm = TRUE),*

*BMW = sum(x13BMW,na.rm = TRUE),*

*BUICK =sum(x13BUICK,na.rm = TRUE),*

*CADIL= sum(x13CADILLAC,na.rm = TRUE),*

*CHEVRO = sum(x13CHEVROLET,na.rm = TRUE),*

*CHRYS = sum(x13CHRYSLER,na.rm = TRUE),*

*DODG = sum(x13DODGE,na.rm = TRUE),*

*EAGL = sum(x13EAGLE,na.rm = TRUE),*

*FIAT= sum(x13FIAT,na.rm = TRUE),*

*FORD = sum(x13FORD,na.rm = TRUE),*

*GENER = sum(x13GENERIC,na.rm = TRUE),*

*GENES = sum(x13GENESIS,na.rm = TRUE),*

*GMC = sum(x13GMC,na.rm = TRUE),*

*HOND = sum(x13HONDA,na.rm = TRUE),*

*HUMM = sum(x13HUMMER,na.rm = TRUE),*

*HYUN = sum(x13HYUNDAI,na.rm = TRUE),*

*INFI = sum(x13INFINITI,na.rm = TRUE),*

*JAGU = sum(x13JAGUAR,na.rm = TRUE),*

*JEEP = sum(x13JEEP,na.rm = TRUE),*

*KIA = sum(x13KIA,na.rm = TRUE),*

*LAMBO = sum(x13LAMBORGHINI,na.rm = TRUE),*

*ROVER = sum(x13LAND.ROVER,na.rm = TRUE),*

*LEX = sum(x13LEXUS,na.rm = TRUE),*

*LINC =sum(x13LINCOLN,na.rm = TRUE),*

*MATSE = sum(x13MASERATI,na.rm = TRUE),*

*MAZD = sum(x13MAZDA,na.rm = TRUE),*

*MCLA = sum(x13MCLAREN,na.rm = TRUE),*

*MERCO = sum(x13MERCEDES.BENZ,na.rm = TRUE),*

*MERCU = sum(x13MERCURY,na.rm = TRUE),*

*MINI = sum(x13MINI,na.rm = TRUE),*

*MITSU = sum(x13MITSUBISHI,na.rm = TRUE),*

*NISSAN = sum(x13NISSAN,na.rm = TRUE),*

*PONTI =sum(x13PONTIAC,na.rm = TRUE),*

*PORSH = sum(x13PORSCHE,na.rm = TRUE),*

*RAM = sum(x13RAM,na.rm = TRUE),*

*SAAB = sum(x13SAAB,na.rm = TRUE),*

*SATU = sum(x13SATURN,na.rm = TRUE),*

*SCION = sum(x13SCION,na.rm = TRUE),*

*SMART = sum(x13SMART,na.rm = TRUE),*

*SUBA = sum(x13SUBARU,na.rm = TRUE),*

*SUZU = sum(x13SUZUKI,na.rm = TRUE),*

*TESL = sum(x13TESLA,na.rm = TRUE),*

*TOYO =sum(x13TOYOTA,na.rm = TRUE),*

*VOLKS = sum(x13VOLKSWAGEN,na.rm = TRUE),*

*VOLV = sum(x13VOLVO,na.rm = TRUE),*

*CAR = sum(x9Car,na.rm = TRUE),*

*MOTO = sum(x9Motorcycle,na.rm = TRUE),*

*PASSCAR = sum(x9Passenger.Car,na.rm = TRUE),*

*PICK = sum(x9Pickup,na.rm = TRUE),*

*SUV = sum(x9SUV,na.rm = TRUE),*

*TRUCK = sum(x9Truck,na.rm = TRUE),*

*VAN = sum(x9Van,na.rm = TRUE),*

*MAXTEMP = mean(x25,na.rm=TRUE),*

*MINTEMP = mean(x26,na.rm=TRUE),*

*TEMPMOY = mean(x27,na.rm=TRUE),*

*WOOD = mean(x28,na.rm=TRUE),*

*PETRO = mean(x29,na.rm=TRUE),*

*FERMETAL = mean(x30,na.rm=TRUE),*

*ELECTIC = mean(x31,na.rm=TRUE),*

*demande = sum(x24,na.rm = TRUE),*

*prix= mean(x32,na.rm=TRUE))*

*dateref = df1 %>%ungroup() %>% select(Annee,Semaine) %>% distinct()*

*### Séparation des jeux de données par pièces et régions ###*

*data6ON = dateref %>%left\_join(df1 %>% filter(groupe==6,Region==1),by=c('Annee','Semaine'))*

*data9ON = dateref %>%left\_join(df1 %>% filter(groupe==9,Region==1),by=c('Annee','Semaine'))*

*data20ON = dateref %>%left\_join(df1 %>% filter(groupe==20,Region==1),by=c('Annee','Semaine'))*

*data37ON = dateref %>%left\_join(df1 %>% filter(groupe==37,Region==1),by=c('Annee','Semaine'))*

*data40ON = dateref %>%left\_join(df1 %>% filter(groupe==40,Region==1),by=c('Annee','Semaine'))*

*data6QC = dateref %>%left\_join(df1 %>% filter(groupe==6,Region==0),by=c('Annee','Semaine'))*

*data9QC = dateref %>%left\_join(df1 %>% filter(groupe==9,Region==0),by=c('Annee','Semaine'))*

*data20QC = dateref %>%left\_join(df1 %>% filter(groupe==20,Region==0),by=c('Annee','Semaine'))*

*data37QC = dateref %>%left\_join(df1 %>% filter(groupe==37,Region==0),by=c('Annee','Semaine'))*

*data40QC = dateref %>%left\_join(df1 %>% filter(groupe==40,Region==0),by=c('Annee','Semaine'))*

*write.csv(data6ON, 'data6ON.csv')*

*write.csv(data9ON, 'data9ON.csv')*

*write.csv(data20ON, 'data20ON.csv')*

*write.csv(data37ON, 'data37ON.csv')*

*write.csv(data40ON, 'data40ON.csv')*

*write.csv(data6QC,'data6QC.csv')*

*write.csv(data9QC,'data9QC.csv')*

*write.csv(data20QC,'data20QC.csv')*

*write.csv(data37QC,'data37QC.csv')*

*write.csv(data40QC,'data40QC.csv')*

**Code Python (Entraînement des modèles et fonctions utiles)**

*import numpy as np*

*import pandas as pd*

*import sklearn*

*import scipy.sparse as sp*

*from langdetect import detect*

*import gensim*

*import requests*

*from pathlib import Path*

*import requests*

*from statsmodels.tsa.stattools import adfuller*

*import statsmodels.api as sm*

*from statsmodels.formula.api import glm*

*import pandas as pd*

*from mlxtend.feature\_selection import SequentialFeatureSelector*

*from sklearn.linear\_model import LinearRegression*

*from sklearn.model\_selection import KFold*

*from statsmodels.graphics.tsaplots import plot\_acf, plot\_pacf*

*from scipy.sparse import csr\_matrix*

*from sklearn.metrics.pairwise import cosine\_similarity*

*from sklearn.metrics.pairwise import cosine\_distances*

*from sklearn.feature\_extraction.text import CountVectorizer*

*from sklearn.feature\_extraction.text import TfidfVectorizer*

*from sklearn.cluster import DBSCAN, AgglomerativeClustering*

*from sklearn.preprocessing import OneHotEncoder, MinMaxScaler*

*import scipy*

*from scipy.sparse import csr\_matrix, dia\_matrix*

*from sklearn.preprocessing import normalize*

*from xgboost import plot\_importance*

*import seaborn as sns*

*color\_pal = sns.color\_palette()*

*import gc*

*gc.collect()*

*from sklearn.model\_selection import TimeSeriesSplit*

*from sklearn.preprocessing import LabelEncoder*

*from sklearn.model\_selection import cross\_val\_score, RandomizedSearchCV , train\_test\_split*

*import xgboost as xgb*

*from scipy.stats import randint, uniform*

*import torch*

*import torch.nn as nn*

*import torch.nn.functional as F*

*import torch.utils.data as data*

*from torch.utils.data import Dataset, DataLoader*

*import torch.optim as optim*

*import math*

*import copy*

*import random*

*from tqdm.notebook import tqdm*

*import sklearn.metrics as metrics*

*from sklearn.metrics import mean\_squared\_error*

*import os*

*from itertools import product*

*import matplotlib.pyplot as plt*

*plt.style.use('fivethirtyeight')*

*import transformers*

*from transformers import BertModel, BertTokenizer, AdamW, get\_linear\_schedule\_with\_warmup*

*device = torch.device('cuda:0') if torch.cuda.is\_available() else torch.device('cpu')*

*####################################################*

*# Partie Web*

*filepath = r'C:/Users/TAB2218/Downloads/data\_out.csv'*

*data\_content = data\_content[['corrected\_name','unit\_part\_price']]*

*data\_content.head(10)*

*data\_content = data\_content.dropna()*

*## fonction pour encoder les noms de pieces*

*part\_name = data\_content[['corrected\_name']]*

*sentences = [str(item).split() for item in part\_name['corrected\_name'].tolist()]*

*model = gensim.models.Word2Vec(sentences, vector\_size=50, window =5, min\_count= 1, workers=4)*

*def get\_vector(name):*

*words = name.split()*

*vectors = [model.wv[word] for word in words if word in model.wv]*

*return np.mean(vectors, axis=0) if vectors else np.zeros((model.vector\_size,))*

*vectors = np.array([get\_vector(word) for word in part\_name['corrected\_name']])*

*### Traitement des prix*

*price\_scaler = MinMaxScaler()*

*prices\_normalized = price\_scaler.fit\_transform(data\_content[['unit\_part\_price']])*

*### Jointures des encodages*

*features = np.concatenate([prices\_normalized, vectors], axis=1)*

*name\_weight = 1.1*

*#region\_weight = 2*

*#features[:, :regions\_encoded.shape[1]] \*= region\_weight*

*features[:, 1:1+vectors.shape[1]] \*= name\_weight*

*#### Obtention des clusters*

*db = DBSCAN(eps= 0.015, min\_samples=1).fit(features)*

*clusters = db.labels\_*

*## Assigantion des groupes*

*data\_content.loc[:,'groupe']= clusters*

*data\_sorted = data\_content.sort\_values(by= 'groupe')*

*# falcutative*

*data\_sorted.to\_csv('parts\_grouping.csv')*

*############ Téléchargement des stations ################*

*# Obtenir les stations utilisées*

*stationpath = r'C:/Users/TAB2218/OneDrive - The Toronto-Dominion Bank/Documents/stations\_a\_download.csv'*

*# donnees sur les stations canadiennes visees et leurs longitudes et latitudes*

*data\_station = pd.read\_csv(stationpath,sep=",", encoding='cp1252')*

*data\_station2 = data\_station[['station\_id\_p','annee']]*

*# liens de telechargement segmenter pour les besoins*

*lien\_tel\_1='https://climate.weather.gc.ca/climate\_data/bulk\_data\_e.html?format=csv&stationID='*

*lien\_tel\_2='&Year='*

*lien\_tel\_3='&Month=10'*

*lien\_tel\_4='&timeframe=2'*

*### fonction de téléchargement*

*list\_dfs = []*

*for i in data\_station2.index:*

*annee = data\_station2.loc[i,'annee']*

*station\_id = data\_station2.loc[i,'station\_id\_p']*

*download\_url = f'{lien\_tel\_1}{station\_id}{lien\_tel\_2}{annee}{lien\_tel\_3}{lien\_tel\_4}'*

*df\_small = pd.read\_csv(download\_url)*

*df\_small['station'] = station\_id*

*list\_dfs.append(df\_small)*

*df = pd.concat(list\_dfs, axis=0)*

*df2 = df[['Date/Time','Year','Month','Day','Max Temp (°C)','Min Temp (°C)','Mean Temp (°C)','Heat Deg Days (°C)','Cool Deg Days (°C)','Total Rain (mm)','Total Snow (cm)','Total Precip (mm)','Snow on Grnd (cm)','Dir of Max Gust (10s deg)','Spd of Max Gust (km/h)','station']]*

*df2.columns = ['date', 'year',*

*'mois', 'jour',*

*'max\_temp', 'min\_temp',*

*'moy\_temp', 'H\_deg',*

*'C\_deg', 'tot\_rain',*

*'tot\_snow', 'tot\_precipi',*

*'snow\_ground','direc\_ghust','max\_speed\_ghust','station']*

*df2.to\_csv('temperature\_info.csv')*

*################## Test Bert ###################*

*help(BertTokenizer.from\_pretrained)*

*pre\_trained\_model = 'bert-base-cased'*

*path\_of\_cache = r'C:/Users/TAB2218/Downloads/oscar\_package/models--bert-base-cased'*

*tokenizer = BertTokenizer.from\_pretrained(path\_of\_cache)*

*model = BertModel.from\_pretrained(path\_of\_cache)*

*input\_text = 'Hello, Bert!'*

*inputs = tokenizer(input\_text, return\_tensors = 'pt')*

*outputs = model(\*\*inputs)*

*print(outputs.last\_hidden\_state)*

*####################################################*

*filepath1 = r'C:/Users/TAB2218/Downloads/data6ON.csv'*

*filepath2 = r'C:/Users/TAB2218/Downloads/data9ON.csv'*

*filepath3 = r'C:/Users/TAB2218/Downloads/data20ON.csv'*

*filepath4 =r'C:/Users/TAB2218/Downloads/data37ON.csv'*

*filepath5 = r'C:/Users/TAB2218/Downloads/data40ON.csv'*

*filepath6 = r'C:/Users/TAB2218/Downloads/data6QC.csv'*

*filepath7 = r'C:/Users/TAB2218/Downloads/data9QC.csv'*

*filepath8 = r'C:/Users/TAB2218/Downloads/data20QC.csv'*

*filepath9 =r'C:/Users/TAB2218/Downloads/data37QC.csv'*

*filepath10 = r'C:/Users/TAB2218/Downloads/data40QC.csv'*

*dataframes = [pd.read\_csv(filepath1,sep=",", encoding='cp1252').drop(columns=['Unnamed: 0']),pd.read\_csv(filepath2,sep=",", encoding='cp1252').drop(columns=['Unnamed: 0']),pd.read\_csv(filepath3,sep=",", encoding='cp1252').drop(columns=['Unnamed: 0']),pd.read\_csv(filepath4,sep=",", encoding='cp1252').drop(columns=['Unnamed: 0']),pd.read\_csv(filepath5,sep=",", encoding='cp1252').drop(columns=['Unnamed: 0']),pd.read\_csv(filepath6,sep=",", encoding='cp1252').drop(columns=['Unnamed: 0']),pd.read\_csv(filepath7,sep=",", encoding='cp1252').drop(columns=['Unnamed: 0']),pd.read\_csv(filepath8,sep=",", encoding='cp1252').drop(columns=['Unnamed: 0']),pd.read\_csv(filepath9,sep=",", encoding='cp1252').drop(columns=['Unnamed: 0']),pd.read\_csv(filepath10,sep=",", encoding='cp1252').drop(columns=['Unnamed: 0'])]*

*data = pd.read\_csv(filepath6,sep=",", encoding='cp1252').drop(columns=['Unnamed: 0'])*

*################## Sévérité ################*

*severites\_variables = ['Annee', 'Semaine','Region', 'groupe', 'hybrid\_in','CYLIND\_NB','HORSEPW',*

*'VEH\_AGE','MILLAGE','ACV','Diesel',*

*'Elec','Gas','NGaz','Demo','NEW','USED','AC','DRP','OGara','CAR','MOTO','PASSCAR',*

*'PICK','SUV','TRUCK','VAN','WOOD','PETRO','FERMETAL','ELECTIC','ACURA','ROMEO',*

*'ASTON','AUDI','BENT','BMW','BUICK','CADIL','CHEVRO','CHRYS','DODG','EAGL','FIAT',*

*'FORD','GENER','GENES','GMC','HOND','HUMM','HYUN','INFI','JAGU','JEEP','KIA','LAMBO',*

*'ROVER','LEX','LINC','MATSE','MAZD','MCLA','MERCO','MERCU','MINI','MITSU','NISSAN',*

*'PONTI','PORSH','RAM','SAAB','SATU','SCION','SMART','SUBA','SUZU','TESL','TOYO','VOLKS','VOLV','prix']*

*severite\_variables=['Annee', 'Semaine','Region', 'groupe', 'hybrid\_in','CYLIND\_NB','HORSEPW',*

*'VEH\_AGE','MILLAGE','ACV','Diesel','Elec','Gas','NGaz','Demo','NEW','USED','AC','DRP',*

*'OGara','CAR','MOTO','PASSCAR','PICK','SUV','TRUCK','VAN','WOOD','PETRO','FERMETAL','ELECTIC']*

*index = [6,9,20,37,40,6,9,20,37,40]*

*regions = ['QC','ON']*

*predictions= {}*

*train\_scores = {}*

*IC\_scores = {}*

*test\_scores = {}*

*corr\_scores\_sev\_glm= {}*

*# Boucle pour itérer sur les dataframes*

*for i, data in enumerate(dataframes, start=1):*

*print(f"Traitement du dataframe {i}...")*

*print(i)*

*idx = index[i-1]*

*if i>5:*

*re = regions[0]*

*else:*

*re = regions[1]*

*data.fillna(0,inplace =True)*

*data['groupe'] = idx*

*Y = data['prix']*

*X = data[severites\_variables].drop(columns=['prix'])*

*## Sélection du modèle*

*lr = LinearRegression()*

*selector =SequentialFeatureSelector(lr,k\_features='best',forward=False,verbose= 2,scoring='neg\_mean\_squared\_error',cv = 3)*

*selector.fit(X,Y)*

*indices = np.array(selector.k\_feature\_idx\_)*

*final\_X = X.iloc[:,indices]*

*## Variables que nous voulons absolument avoir dans le modèle*

*final\_X['hybrid\_in'] = X['hybrid\_in']*

*final\_X['CYLIND\_NB'] = X['CYLIND\_NB']*

*final\_X['HORSEPW'] =X['HORSEPW']*

*final\_X['VEH\_AGE'] = X['VEH\_AGE']*

*final\_X['MILLAGE'] = X['MILLAGE']*

*final\_X['PICK'] = X['PICK']*

*final\_X['SUV'] = X['SUV']*

*final\_X['VAN'] = X['VAN']*

*final\_X['PETRO'] = X['PETRO']*

*final\_X['FERMETAL'] = X['FERMETAL']*

*final\_X['AC'] = X['AC']*

*final\_X['DRP'] = X['DRP']*

*final\_X['OGARA'] = X['OGara']*

*final\_X['ELECTIC'] = X['ELECTIC']*

*final\_X['prix'] = data['prix']*

*## Rajouter des caractéristiques temporelles*

*final\_X['Annee']= X['Annee']*

*final\_X['Semaine']= X['Semaine']*

*final\_X['date'] = pd.to\_datetime(final\_X['Annee'].astype(str)+final\_X['Semaine'].astype(str)+'1',format='%Y%W%w')*

*final\_X.set\_index('date', inplace =True)*

*final\_X['mois'] = final\_X.index.month*

*final\_X=final\_X.sort\_index()*

*## Rajouter des décalages*

*final\_X['lag\_12'] = final\_X['prix'].shift(12)*

*final\_X['lag\_4'] = final\_X['prix'].shift(4)*

*final\_X['lag\_1'] = final\_X['prix'].shift(1)*

*## Sépartion des ensembles*

*data\_train= final\_X.loc[final\_X.index < '01-01-2023'].dropna()*

*data\_test = final\_X.loc[final\_X.index >= '01-01-2023'].dropna()*

*## Initialisation du split en partitions égales pour l'entraînement*

*tss = TimeSeriesSplit(n\_splits=3)*

*## Ordonner le dataframe*

*data\_train = data\_train.sort\_index()*

*## Entrainement du modele*

*cle\_pred = f'pred\_sev\_{re}\_{idx}'*

*cle\_scrore\_train = f'scrore\_train\_{re}\_{idx}'*

*cle\_IC = f'IC\_{re}\_{idx}'*

*cle\_scrore\_test = f'scrore\_test\_{re}\_{idx}'*

*cle\_score\_corr = f'score\_corr\_{re}\_{idx}'*

*#fold = 0*

*for train\_idx, val\_idx in tss.split(data\_train):*

*print(train\_idx)*

*train = data\_train.iloc[train\_idx]*

*validation = data\_train.iloc[val\_idx]*

*X\_train = train.drop(columns=['prix'])*

*y\_train = train['prix']*

*X\_validation = validation.drop(columns=['prix'])*

*y\_validation = validation['prix']*

*glm = sm.GLM(y\_train, sm.add\_constant(X\_train),family = sm.families.Gaussian())*

*modelsevglm = glm.fit()*

*y\_pred = modelsevglm.predict(sm.add\_constant(X\_validation))*

*train\_scores[cle\_scrore\_train] = np.sqrt(mean\_squared\_error(y\_validation, y\_pred))*

*## Prédictions y\_test*

*X\_test = data\_test.drop(columns=['prix'])*

*IC\_severite = modelsevglm.get\_prediction(sm.add\_constant(X\_test)).conf\_int()*

*IC\_scores[cle\_IC] = IC\_severite*

*y\_test= modelsevglm.predict(sm.add\_constant(X\_test))*

*predictions[cle\_pred] = y\_test*

*y\_real = data\_test['prix']*

*residus=y\_test-y\_real*

*test\_scores[cle\_scrore\_test] = np.sqrt(mean\_squared\_error(y\_real, y\_test))*

*corr\_scores = adfuller(residus)*

*corr\_scores\_sev\_glm[cle\_score\_corr] = corr\_scores[1]*

*## Visualisation*

*plt.plot(X\_test.index, y\_test, label = 'Prédictions')*

*plt.fill\_between(X\_test.index, IC\_severite[:, 0],IC\_severite[:,1],color = 'gray', alpha = 0.3)*

*plt.plot(X\_test.index, y\_real, label = 'Actuelles')*

*plt.title(f'Prédiction contre valeur actuelle de la sévérité moyenne pour la pièce{idx} en {re}')*

*plt.xlabel('Temps en semaine')*

*plt.xticks(rotation=45)*

*plt.ylabel('Sévérité moyenne par semaine')*

*plt.legend()*

*plt.show()*

*plot\_acf(residus, lags=12)*

*plt.title(f'Corrélogrammes des résidus de la pièce{idx} en {re}')*

*plt.show()*

*########### GLM Frequence ####################*

*frequence\_variables = [ "Annee" ,"Semaine" ,"Region", "hybrid\_in", "CYLIND\_NB", "HORSEPW","VEH\_AGE" ,"MILLAGE",*

*"ACV", "PLUIE" ,"NEIGE", "PRECIPI","CMNEIGE" ,"FERIE" ,"COVID" ,"Fem" , "Hom",*

*"A25","A35" ,"A50","A60", "Conj" ,"Marie" , "Celib", "Veuf" ,"SMI",*

*"Diesel" ,"Elec","Gas" , "NGaz","Demo" ,"NEW" ,"USED", "AC", "DRP",*

*"OGara","ACURA" , "ROMEO","ASTON" , "AUDI" , "BENT","BMW","BUICK","CADIL",*

*"CHEVRO","CHRYS" ,"DODG" ,"EAGL","FIAT","FORD","GENER","GENES","GMC",*

*"HOND" , "HUMM" ,"HYUN" ,"INFI" ,"JAGU" ,"JEEP","KIA" ,"LAMBO" ,"ROVER","LEX",*

*"LINC","MATSE" ,"MAZD", "MCLA" ,"MERCO" ,"MERCU" ,"MINI","MITSU","NISSAN",*

*"PONTI","PORSH", "RAM" ,"SAAB" ,"SATU","SCION", "SMART","SUBA","SUZU",*

*"TESL","TOYO" ,"VOLKS", "VOLV", "CAR", "MOTO", "PASSCAR","PICK",*

*"SUV","TRUCK","VAN", "MAXTEMP","MINTEMP","TEMPMOY","WOOD","PETRO","FERMETAL","ELECTIC",'demande']*

*predictions\_glm\_freq= {}*

*train\_scores\_glm\_freq = {}*

*IC\_scores\_glm\_freq = {}*

*test\_scores\_glm\_freq = {}*

*corr\_scores\_freq\_glm= {}*

*# Boucle pour itérer sur les dataframes*

*for i, data in enumerate(dataframes, start=1):*

*print(f"Traitement du dataframe {i}...")*

*print(i)*

*idx = index[i-1]*

*if i>5:*

*re = regions[0]*

*else:*

*re = regions[1]*

*data.fillna(0,inplace =True)*

*#data['groupe'] = idx*

*data = data[frequence\_variables]*

*Y = data['demande']*

*X = data[frequence\_variables]*

*## Sélection du modèle*

*model\_initial = glm('demande ~ '+ '+'.join(frequence\_variables),data=data,family = sm.families.Poisson()).fit()*

*selection = [var for var in frequence\_variables if model\_initial.pvalues[var]<0.05]*

*model\_2 = glm('demande ~ '+ '+'.join(selection),data=data,family = sm.families.Poisson()).fit()*

*selection2 = [var for var in selection if model\_2.pvalues[var]<0.05]*

*model\_3 = glm('demande ~ '+ '+'.join(selection2),data=data,family = sm.families.Poisson()).fit()*

*selection3 = [var for var in selection2 if model\_3.pvalues[var]<0.05]*

*final\_X = data[selection3]*

*## Variables que nous voulons absolument avoir dans le modèle*

*final\_X['COVID'] = data['COVID']*

*final\_X['Annee'] = data['Annee']*

*final\_X['Semaine'] =data['Semaine']*

*final\_X['FERIE'] = data['FERIE']*

*final\_X['Hom'] = data['Hom']*

*final\_X['Fem'] = data['Fem']*

*final\_X['PLUIE'] = data['PLUIE']*

*final\_X['NEIGE'] = data['NEIGE']*

*final\_X['PRECIPI'] = data['PRECIPI']*

*final\_X['A25'] = data['A25']*

*final\_X['A35'] = data['A35']*

*final\_X['A50'] = data['A50']*

*final\_X['A60'] = data['A60']*

*final\_X['TEMPMOY'] = data['TEMPMOY']*

*final\_X['MILLAGE'] = data['MILLAGE']*

*final\_X['demande'] = data['demande']*

*## Rajouter des caractéristiques temporelles*

*final\_X['date'] = pd.to\_datetime(final\_X['Annee'].astype(str)+final\_X['Semaine'].astype(str)+'1',format='%Y%W%w')*

*final\_X.set\_index('date', inplace =True)*

*final\_X['mois'] = final\_X.index.month*

*final\_X=final\_X.sort\_index()*

*## Rajouter des décalages*

*final\_X['lag\_12'] = final\_X['demande'].shift(12)*

*final\_X['lag\_4'] = final\_X['demande'].shift(4)*

*final\_X['lag\_1'] = final\_X['demande'].shift(1)*

*## Sépartion des ensembles*

*final\_X = sm.add\_constant(final\_X)*

*data\_train= final\_X.loc[final\_X.index < '01-01-2023'].dropna()*

*data\_test = final\_X.loc[final\_X.index >= '01-01-2023'].dropna()*

*## Initialisation du split en partitions égales pour l'entraînement*

*tss = TimeSeriesSplit(n\_splits=3)*

*## Ordonner le dataframe*

*data\_train = data\_train.sort\_index()*

*## Entrainement du modele*

*cle\_pred = f'pred\_freq\_{re}\_{idx}'*

*cle\_scrore\_train = f'scrore\_train\_{re}\_{idx}'*

*cle\_IC = f'IC\_{re}\_{idx}'*

*cle\_scrore\_test = f'scrore\_test\_{re}\_{idx}'*

*cle\_score\_corr = f'score\_corr\_{re}\_{idx}'*

*for train\_idx, val\_idx in tss.split(data\_train):*

*print(train\_idx)*

*train = data\_train.iloc[train\_idx]*

*validation = data\_train.iloc[val\_idx]*

*X\_train = train.drop(columns=['demande'])*

*y\_train = train['demande']*

*X\_validation = validation.drop(columns=['demande'])*

*y\_validation = validation['demande']*

*model = sm.GLM(y\_train, X\_train,family = sm.families.Poisson())*

*modelfreqglm = model.fit()*

*y\_pred = modelfreqglm.predict(X\_validation)*

*print(sm.add\_constant(X\_validation).shape)*

*print(X\_train.shape)*

*train\_scores\_glm\_freq[cle\_scrore\_train] = np.sqrt(mean\_squared\_error(y\_validation, y\_pred))*

*## Prédictions y\_test*

*X\_test = data\_test.drop(columns=['demande'])*

*IC\_severite = modelfreqglm.get\_prediction(X\_test).conf\_int()*

*IC\_scores\_glm\_freq[cle\_IC] = IC\_severite*

*y\_test= modelfreqglm.predict(X\_test)*

*predictions\_glm\_freq[cle\_pred] = y\_test*

*y\_real = data\_test['demande']*

*test\_scores\_glm\_freq[cle\_scrore\_test] = np.sqrt(mean\_squared\_error(y\_real, y\_test))*

*residus = y\_real-y\_test*

*corr\_scores = adfuller(residus)*

*corr\_scores\_freq\_glm[cle\_score\_corr] = corr\_scores[1]*

*## Visualisation*

*plt.plot(X\_test.index, y\_test, label = 'Prédictions')*

*plt.fill\_between(X\_test.index, IC\_severite[:, 0],IC\_severite[:,1],color = 'gray', alpha = 0.3)*

*plt.plot(X\_test.index, y\_real, label = 'Actuelles')*

*plt.title(f'Prédictions contre valeurs actuelles de la demande totale pour la pièce{idx} en {re}')*

*plt.xlabel('Temps en semaine')*

*plt.xticks(rotation=45)*

*plt.ylabel('Demande Totale par semaine')*

*plt.legend()*

*plt.show()*

*plot\_acf(residus, lags=21)*

*plt.title(f'Corrélogrammes des résidus de la pièce{idx} en {re}')*

*plt.ylabel('Autocorrélation')*

*plt.xlabel('h')*

*plt.show()*

*############ Partie GBM Severite #########*

*index = [6,9,20,37,40,6,9,20,37,40]*

*regions = ['QC','ON']*

*predictions\_Sev\_B= {}*

*train\_scores\_Sev\_B = {}*

*IC\_scores\_Sev\_B = {}*

*test\_scores\_Sev\_B = {}*

*corr\_scores\_sev\_gbm= {}*

*lower = []*

*upper = []*

*# Boucle pour itérer sur les dataframes*

*for i, data in enumerate(dataframes, start=1):*

*print(f"Traitement du dataframe {i}...")*

*print(i)*

*idx = index[i-1]*

*if i>5:*

*re = regions[0]*

*else:*

*re = regions[1]*

*Y = data['prix']*

*X = data[severites\_variables]*

*## Variables que nous voulons absolument avoir dans le modèle*

*final\_X = X*

*## Rajouter des caractéristiques temporelles*

*final\_X['Annee']= data['Annee']*

*final\_X['Semaine']= data['Semaine']*

*final\_X['date'] = pd.to\_datetime(final\_X['Annee'].astype(str)+final\_X['Semaine'].astype(str)+'1',format='%Y%W%w')*

*final\_X.set\_index('date', inplace =True)*

*final\_X['mois'] = final\_X.index.month*

*final\_X=final\_X.sort\_index()*

*## Rajouter des décalages*

*final\_X['lag\_12'] = final\_X['prix'].shift(12)*

*final\_X['lag\_4'] = final\_X['prix'].shift(4)*

*final\_X['lag\_1'] = final\_X['prix'].shift(1)*

*## Sépartion des ensembles*

*data\_train= final\_X.loc[final\_X.index < '01-01-2023'].dropna()*

*data\_test = final\_X.loc[final\_X.index >= '01-01-2023'].dropna()*

*## Initialisation du split en partitions égales pour l'entraînement*

*tss = TimeSeriesSplit(n\_splits=4, test\_size=4\*12\*1, gap=2)*

*## Ordonner le dataframe*

*data\_train = data\_train.sort\_index()*

*## Entrainement du modele*

*cle\_pred = f'pred\_sev\_{re}\_{idx}'*

*cle\_scrore\_train = f'scrore\_train\_{re}\_{idx}'*

*cle\_IC = f'IC\_{re}\_{idx}'*

*cle\_scrore\_test = f'scrore\_test\_{re}\_{idx}'*

*cle\_score\_corr = f'score\_corr\_{re}\_{idx}'*

*#fold = 0*

*for train\_idx, val\_idx in tss.split(data\_train):*

*print(train\_idx)*

*train = data\_train.iloc[train\_idx]*

*validation = data\_train.iloc[val\_idx]*

*X\_train = train.drop(columns=['prix'])*

*y\_train = train['prix']*

*X\_validation = validation.drop(columns=['prix'])*

*y\_validation = validation['prix']*

*model\_sev\_xgb = xgb.XGBRegressor(base\_score=0.5,*

*booster='gbtree',*

*n\_estimators=500,*

*early\_stopping\_rounds=150,*

*objective='reg:squarederror',*

*max\_depth=6,*

*learning\_rate=0.01)*

*model\_sev\_xgb.fit(X\_train, y\_train,eval\_set=[(X\_train, y\_train), (X\_validation, y\_validation)],verbose=25)*

*y\_pred = model\_sev\_xgb.predict(X\_validation)*

*lower\_quartile = np.percentile(y\_pred, (0.05/2)\*100)*

*upper\_quartile = np.percentile(y\_pred, (1-0.05/2)\*100)*

*lower.append(lower\_quartile)*

*upper.append(upper\_quartile)*

*score\_valid = np.sqrt(mean\_squared\_error(y\_validation, y\_pred))*

*train\_scores\_Sev\_B[cle\_scrore\_train] = score\_valid*

*intervalle\_lower = np.mean(lower,axis=0)*

*intervalle\_upper = np.mean(upper, axis=0)*

*IC = pd.DataFrame({'lower':lower, 'upper':upper}).to\_numpy()*

*IC\_scores\_Sev\_B[cle\_IC]=IC*

*X\_test = data\_test.drop(columns=['prix'])*

*y\_test = model\_sev\_xgb.predict(X\_test)*

*predictions\_Sev\_B[cle\_pred]=y\_test*

*y\_real = data\_test['prix']*

*score\_test = np.sqrt(mean\_squared\_error(y\_real, y\_test))*

*test\_scores\_Sev\_B[cle\_scrore\_test] = score\_test*

*residus = y\_real-y\_test*

*corr\_scores = adfuller(residus)*

*corr\_scores\_sev\_gbm[cle\_score\_corr] = corr\_scores[1]*

*## Visualisation*

*plt.figure(figsize=(14,7))*

*plt.plot(X\_test.index, y\_test, label = 'Prédictions')*

*plt.fill\_between(X\_test.index,intervalle\_lower ,intervalle\_upper,color = 'gray', alpha = 0.3)*

*#*

*plt.plot(X\_test.index, y\_real, label = 'Actuelles')*

*plt.title(f'Prédictions contre valeurs réelles de la sévérité moyenne pour la pièce{idx} en {re}')*

*plt.xlabel('Temps en semaine')*

*plt.xticks(rotation=45)*

*plt.ylabel('Sévérité moyenne par semaine')*

*plt.legend()*

*plt.show()*

*fig, ax = plt.subplots(figsize=(10, 8))*

*plot\_importance(model\_sev\_xgb, ax=ax, height=0.5)*

*plt.show()*

*plot\_acf(residus, lags=7)*

*plt.title(f'Corrélogrammes des résidus de la pièce{idx} en {re}')*

*plt.show()*

*########### Partie GBM Frequence ######################*

*index = [6,9,20,37,40,6,9,20,37,40]*

*regions = ['QC','ON']*

*predictions\_freq\_B= {}*

*train\_scores\_freq\_B = {}*

*IC\_scores\_freq\_B = {}*

*test\_scores\_freq\_B = {}*

*corr\_scores\_freq\_gbm = {}*

*lower = []*

*upper = []*

*# Boucle pour itérer sur les dataframes*

*for i, data in enumerate(dataframes, start=1):*

*print(f"Traitement du dataframe {i}...")*

*print(i)*

*idx = index[i-1]*

*if i>5:*

*re = regions[0]*

*else:*

*re = regions[1]*

*Y = data['demande']*

*X = data[frequence\_variables]*

*## Variables que nous voulons absolument avoir dans le modèle*

*final\_X = X*

*## Rajouter des caractéristiques temporelles*

*final\_X['Annee']= data['Annee']*

*final\_X['Semaine']= data['Semaine']*

*final\_X['date'] = pd.to\_datetime(final\_X['Annee'].astype(str)+final\_X['Semaine'].astype(str)+'1',format='%Y%W%w')*

*final\_X.set\_index('date', inplace =True)*

*final\_X['mois'] = final\_X.index.month*

*final\_X=final\_X.sort\_index()*

*## Rajouter des décalages*

*final\_X['lag\_12'] = final\_X['demande'].shift(12)*

*final\_X['lag\_4'] = final\_X['demande'].shift(4)*

*final\_X['lag\_1'] = final\_X['demande'].shift(1)*

*## Sépartion des ensembles*

*data\_train= final\_X.loc[final\_X.index < '01-01-2023'].dropna()*

*data\_test = final\_X.loc[final\_X.index >= '01-01-2023'].dropna()*

*## Initialisation du split en partitions égales pour l'entraînement*

*tss = TimeSeriesSplit(n\_splits=4, test\_size=4\*12\*1, gap=2)*

*## Ordonner le dataframe*

*data\_train = data\_train.sort\_index()*

*## Entrainement du modele*

*cle\_pred = f'pred\_sev\_{re}\_{idx}'*

*cle\_scrore\_train = f'scrore\_train\_{re}\_{idx}'*

*cle\_IC = f'IC\_{re}\_{idx}'*

*cle\_scrore\_test = f'scrore\_test\_{re}\_{idx}'*

*cle\_score\_corr = f'score\_corr\_{re}\_{idx}'*

*#fold = 0*

*for train\_idx, val\_idx in tss.split(data\_train):*

*print(train\_idx)*

*train = data\_train.iloc[train\_idx]*

*validation = data\_train.iloc[val\_idx]*

*X\_train = train.drop(columns=['demande'])*

*y\_train = train['demande']*

*X\_validation = validation.drop(columns=['demande'])*

*y\_validation = validation['demande']*

*model\_sev\_xgb = xgb.XGBRegressor(base\_score=0.5,*

*booster='gbtree',*

*n\_estimators=500,*

*early\_stopping\_rounds=100,*

*objective='count:poisson',*

*max\_depth=6,*

*learning\_rate=0.01)*

*model\_sev\_xgb.fit(X\_train, y\_train,eval\_set=[(X\_train, y\_train), (X\_validation, y\_validation)],verbose=25)*

*y\_pred = model\_sev\_xgb.predict(X\_validation)*

*lower\_quartile = np.percentile(y\_pred, (0.05/2)\*100)*

*upper\_quartile = np.percentile(y\_pred, (1-0.05/2)\*100)*

*lower.append(lower\_quartile)*

*upper.append(upper\_quartile)*

*score\_valid = np.sqrt(mean\_squared\_error(y\_validation, y\_pred))*

*train\_scores\_freq\_B[cle\_scrore\_train] = score\_valid*

*intervalle\_lower = np.mean(lower,axis=0)*

*intervalle\_upper = np.mean(upper, axis=0)*

*IC = pd.DataFrame({'lower':lower, 'upper':upper}).to\_numpy()*

*IC\_scores\_freq\_B[cle\_IC]=IC*

*X\_test = data\_test.drop(columns=['demande'])*

*y\_test = model\_sev\_xgb.predict(X\_test)*

*predictions\_freq\_B[cle\_pred]=y\_test*

*y\_real = data\_test['demande']*

*residus = y\_real-y\_test*

*score\_test = np.sqrt(mean\_squared\_error(y\_real, y\_test))*

*test\_scores\_freq\_B[cle\_scrore\_test] = score\_test*

*corr\_scores = adfuller(residus)*

*corr\_scores\_freq\_gbm[cle\_score\_corr] = corr\_scores[1]*

*## Visualisation*

*plt.figure(figsize=(14,7))*

*plt.plot(X\_test.index, y\_test, label = 'Prédictions')*

*plt.fill\_between(X\_test.index,intervalle\_lower ,intervalle\_upper,color = 'gray', alpha = 0.3)*

*#*

*plt.plot(X\_test.index, y\_real, label = 'Actuelles')*

*plt.title(f'Prédictions contre valeurs réelles de la demande totale pour la pièce{idx} en {re}')*

*plt.xlabel('Temps en semaine')*

*plt.xticks(rotation=45)*

*plt.ylabel('Féquence totale par semaine')*

*plt.legend()*

*plt.show()*

*fig, ax = plt.subplots(figsize=(10, 8))*

*plot\_importance(model\_sev\_xgb, ax=ax, height=0.5)*

*plt.show()*

*plot\_acf(residus, lags=12)*

*plt.xlabel('h')*

*plt.ylabel('Autocorrélation')*

*plt.title(f'Corrélogrammes des résidus de la pièce{idx} en {re}')*

*plt.show()*