Date : 20 mars 2018
Auteur : Fabien ROUILLON
Vincent CHAMBRIN

Projet Apprentissage Statistique

ChallengeData : Prediction of transaction claims status By PRICEMINISTER RAKUTEN

Encadrant : Bertrand MICHEL

Sommaire

Le but de ce projet est de participer au challenge Prediction of transaction claims status By PRICEMINISTER RAKUTEN de la plateforme Data Challenge. Ce projet s'inscrit dans le cadre de l'enseignement Machine Learning : pratique et compléments de l'option Mathématiques et Applications de l'École Centrale de Nantes. L'objectif principal est ici de mettre en pratique les méthodes de classification vues en cours dans un contexte industriel.

Dans un premier temps nous présenterons le challenge et le jeu de données. Ensuite, nous décrirons de manière exhaustive le jeu de données avant de détailler les étapes de feature engineering mises en place. Enfin, nous présenterons les résultats obtenus avec les différents modèles de machine leanring testés avant de conclure.

Mots-clés : *machine learning*, challenge data science, Python.

Table des matières

1 Introduction

1.1 Présentation du challenge

PriceMinister est une entreprise et plateforme d'achat-vente en ligne jouant le rôle d'intermédiaire entre acheteurs et vendeurs comprenant 17 millions d'utilisateurs inscrits en 2016. Avec environ 50 000 transactions par jour de produits classés an 14 rubriques (Informatique, électroménager, loisirs, beauté, ...) c'est l'une des plateformes de commerce en ligne majeure en France.

Ce challenge vise à prédire si une transaction passée en ligne via le site PriceMinister a des chances d'aboutir à une réclamation de la part de l'acheteur, par exemple un colis non reçu ou endommagé, et le cas échéant prédire le type de réclamation. Améliorer la capacité de prédiction des réclamations est un des enjeux majeurs de l'e-commerce car il permet non seulement d'améliorer l'expérience utilisateur, mais aussi d'augmenter le chiffre d'affaires en anticipant d'éventuels coûts en se focalisant sur les transactions à risques.

1.2 Présentation du jeu de données

Il s'agit donc d'un problème de classification multi-classe dont les classes à prédire sont :

- '-' : pas de réclamation
- 'WITHDRAWAL' : retrait de la commande
- 'SELLER_CANCEL_POSTERIORI' : le vendeur annule la commande
- 'NOT_RECEIVED' : colis non reçu
- 'DIFFERENT' : produit différent de l'annonce
- 'DAMAGED' : produit endommagé
- 'FAKE' : le produit est un faux (arnaque)
- 'UNDEFINED' : réclamation d'un autre type

Le jeu de données comprend des données hétérogènes telles que le moyen de paiement, les départements ou pays du vendeur et de l'acheteur, le type de produit acheté :

- ID : identifiant de la commande
- SHIPPING_MODE : méthode de transport
- SHIPPING_PRICE : coût du transport, si existant
- WARRANTIES_FLG : True si une garantie a été prise par l'acheteur
- WARRANTIES_PRICE : prix de la garantie si existante
- CARD_PAYEMENT : indicatrice du moyen de paiement par carte bancaire
- COUPON_PAYEMENT : indicatrice du moyen de paiement avec un coupon discount
- RSP_PAYEMENT : indicatrice du moyen de paiement avec des Rakuten Super Points
- WALLET_PAYMENT : indicatrice du moyen de paiement avec PriceMinister-Rakuten wallet
- PRICECLUB_STATUS : status de l'acheteur
- REGISTRATION_DATE : année d'enregistrement de l'acheteur
- PURCHASE_COUNT : nombre d'achats précédemment réalisés par l'acheteur
- BUYER_BIRTHDAY_DATE : année de naissance de l'acheteur
- BUYER_DEPARTMENT : département de l'acheteur
- BUYING_DATE : année et mois de l'achat
- SELLER_SCORE_COUNT : nombre de ventes précédemment réalisées par le vendeur
- SELLER_SCORE_AVERAGE : score du vendeur sur le site PriceMinister-Rakuten
- SELLER_COUNTRY : pays du vendeur
- SELLER_DEPARTMENT : département français du vendeur (-1 si vendeur à l'étranger)
- PRODUCT_TYPE : type de produit commandé

— PRODUCT_FAMILY : famille du produit commandé
— ITEM_PRICE : prix du produit acheté

Plusieurs problématiques sont soulevées par ce jeu de données. Dans un premier temps, il est essentiel de mettre en relief l'ancienneté et la fiabilité à la fois du vendeur et de l'acheteur en croisant les données en rapport avec les statuts, le nombre d'années d'inscription sur le site et le nombre de ventes/commandes réalisées. De plus, les nombreuses données géographiques devront être exploitées pour mettre an avant la distance parcouru par la commande. Le type et la famille de produit peuvent aussi s'avérer déterminant dans la prédiction de réclamations étant données que certains produits doivent être plus sujets à certains types de plaintes telles que la casse. De la même manière certains types de transports peuvent être plus sujets à la casse ou au retard de livraison. Enfin, le mode de paiement et l'achat d'une garantie peuvent aussi d'une manière moins évidente influencer les réclamations.

2 Description du jeu de données

Dans cette partie nous nous attacherons de détailler l'étude réalisée sur le jeu de données. Cette étude détaillée permet notamment de faire ressortir les corrélations entre les variables et de mettre en avant les variables les plus déterminantes dans la prédiction du type de réclamation. De plus c'est à cette occasion que la réflexion sur la transformation des variables existantes ou la création de nouveaux features va émerger.

2.1 Résumé statistique et première observations

Le jeu de données de test est composée de 99 995 observations. Le jeu de données d'entraînement est composé de 100 000 observations dont les fréquences de chacun des types de réclamations sont les suivantes :

- '-' : 50%
- 'NOT_RECEIVED' : 14,8%
- 'SELLER_CANCEL_POSTERIORI' : 13,8%
- 'WITHDRAWAL' : 7%
- 'DAMAGED' : 5,9%
- 'DIFFERENT' : 4,3%
- 'UNDEFINED' : 4,1%
- 'FAKE' : 0,2%

On note qu'il y a donc 50% de réclamations et 50% de non réclamation sur les données d'entraînement. De plus, les fréquences des types de réclamations ne sont pas équilibrées, avec notamment une fréquence très basse pour la déclaration de faux (0,2%). Pour comparer nos modèles nous utiliserons donc une métrique de type AUC, plus adaptée à ce type de problème.

La plupart des variables sont catégorielles :

- Variables nominales :
  - SHIPPING_MODE (11 niveaux)
  - BUYER_DEPARTMENT (100 niveaux)
  - BUYING_DATE (10 niveaux)
  - SELLER_COUNTRY (39 niveaux)
  - SELLER_DEPARTMENT (98 niveaux)
  - PRODUCT_TYPE (137 niveaux)
  - PRODUCT_FAMILY (12 niveaux)
- Variables dichotomiques/binaires :
  - WARRANTIES_FLG
  - CARD_PAYMENT
  - COUPON_PAYMENT
  - RSP_PAYMENT
  - WALLET_PAYMENT
- Variables ordinales :
  - SHIPPING_PRICE (5 niveaux)
  - WARRANTIES_PRICE (5 niveaux)
  - PRICECLUB_STATUS (5 niveaux)
  - PURCHASE_COUNT (6 niveaux)
  - SELLER_SCORE_COUNT (5 niveaux)
  - ITEM_PRICE (8 niveaux)
- Variables quantitatives :
  - REGISTRATION_DATE (17 valeurs distinctes)

— BUYER_BIRTHDAY_DATE (107 valeurs distinctes)
— SELLER_SCORE_AVERAGE (31 valeurs distinctes)

Cependant, ces variables quantitatives pourront être transformées en variables catégorielles. La date de naissance de l'acheteur pourra par exemple être convertie en catégories d'âge.

Nous pouvons de plus noter que certaines variables ont beaucoup de valeurs manquantes, il faudra donc réfléchir à la méthode employée pour combler ces valeurs manquantes :

— WARRANTIES_PRICE : 96603 / 96.6% de valeurs manquantes
— SHIPPING_PRICE : 67610 / 67.6% de valeurs manquantes
— BUYER_BIRTHDAY_DATE : 5836 / 5.8% de valeurs manquantes

Aussi, nous notons que les variables binaires ne sont pas équilibrées :

— WARRANTIES_FLG : 96603 / 96.6% de 0
— COUPON_PAYMENT : 94271 / 94.3% de 0
— CARD_PAYMENT : 10407 / 89,6% de 1

Certaines variables catégorielles ont beaucoup de niveaux, il faudra donc envisager l'éventualité de les modifier avant de les transformer en tableaux d'indicatrices, ce qui augmenterait considérablement le nombre de variables :

— PRODUCT_TYPE : 137 valeurs distinctes
— BUYER_DEPARTMENT 100 valeurs distinctes
— SELLER_DEPARTMENT 98 valeurs distinctes

Nous remarquons de plus que le jeu de données possède 3238 lignes dupliquées. Les données ont été rendues anonymes, on ne peut donc pas identifier les acheteurs ou les vendeurs mais il est cependant envisageable que certains individus apparaissent dans plusieurs transactions similaires.

2.2 Description exhaustive des variables

SHIPPING_MODE et SHIPPING_PRICE :

Les moyens de livraisons peu populaires ont en moyenne plus de réclamations que les autres. Pour exemple le moyen de livraison PICKUP avec 73,3% de réclamations bien plus élevé que les 50% de moyenne.

MONDIAL_RELAY_PREPAYE (46%), SUIVI (46%) and NORMAL (47%) sont les moyens de livraison les plus fiables.

La livraison NORMAL a le taux le plus élevé de réclamations NOT_RECEIVED (16%) mais a les taux de WITHDRAWAL (4%) et de UNDEFINED (2%) les plus bas

CHRONOPOST a le plus haut taux de réclamations DAMAGED.

Paradoxalement, le taux de réclamations évolue positivement avec le coût de la livraison. En particulier les réclamations UNDEFINED et WITHDRAWAL sont plus probables lorsque le prix du transport est élevé.

SHIPPING_PRICE	<1	1<5	5<10	10<20	>20
SHIPPING_MODE					
CHRONOPOST	0	14	0	125	3
EXPRESS_DELIVERY	13	23	103	111	1
MONDIAL_RELAY	668	963	155	28	10
MONDIAL_RELAY_PREPAYE	38	889	286	0	0
NORMAL	4499	12648	2506	88	13
RECOMMANDE	870	558	1653	818	194
SO_POINT_RELAIS	16	385	923	51	0
SO_RECOMMANDE	2	58	368	54	2
SUIVI	499	1463	1198	55	2

De plus, il y a une corrélation claire entre SHIPPING_MODE et SHIPPING_PRICE. Par exemple, un SHIPPING_PRICE de plus de 20 a de grandes chances d'être de type RECOMMANDE.

WARRANTIES_FLG et WARRANTIES_PRICE :

Les personnes possédant une garantie sont plus susceptibles de porter des réclamations, en particulier de type WITHDRAWAL. Ceci est logique étant donné que le retrait peu être facilité par la prise d'une garantie.

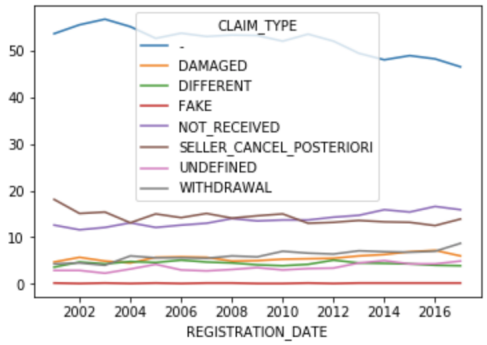
Nous n'observons pas de corrélation claires entre le prix de la garantie et le type de réclamation.

PRICECLUB_STATUS :

Cette variable est liée au nombre de points accumulés en effectuant des actions telles que vendre des produits, parrainer un ami ou encore utiliser l'application PrimeMinister. Avec ces points l'utilisateur peut occasionnellement bénéficier d'offres et de réductions.

Il n'y a pas de lien clair entre PRICECLUB_STATUS et les réclamations.

REGISTRATION_DATE et PURCHASE_COUNT :

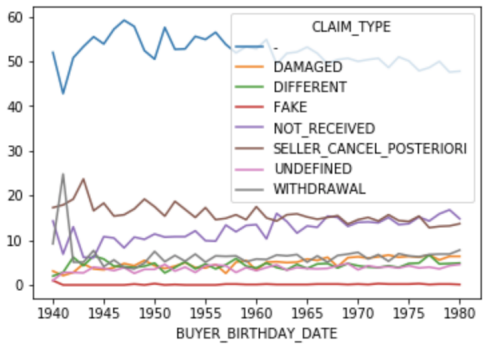


Nous observons que les récents utilisateurs ont plus tendance à porter de réclamations. Il est notable que les acheteurs avec de l'expérience sur le site sont moins susceptibles de porter des réclamations WITHDRAWAL et UNDEFINED. Cela fait sens car un utilisateur habitué a moins de chance de s'être trompé dans sa commande et de la retirer. Cependant ces acheteurs ont aussi moins tendance à ne pas recevoir leur colis ou à recevoir un colis endommagé. Ceci peut s'expliquer par les habitudes des consommateurs qui avec le temps n'achètent qu'auprès des vendeurs en qui ils ont confiance.

De manière prévisible, il y a une corrélation importante entre l'année d'inscription de l'utilisateur et le nombre de commandes passées.

De plus, nous pouvons observer un fossé clair entre les utilisateurs très récents (<5 items achetés) et les autres. Pour cette raison il peut être intéressant de créer une variable supplémentaire pour mettre en avant le fait qu'un utilisateur soit novice ou non.

BUYER_BIRTHDAY_DATE :



Une corrélation peu évidente peut être observée entre l'âge et les réclamations. En effet, les individus plus jeunes ont une probabilité légèrement plus grande que la moyenne de porter une réclamation de type NOT_RECEIVED.

BUYER_DEPARTMENT, SELLER_DEPARTMENT et SELLER_COUNTRY :

Pour observer plus facilement l'influence de la localisation des acheteurs et vendeurs nous avons créé une variable supplémentaire permettant d'identifier les utilisateurs par leur région au lieu de leur département.

Il n'y a cependant pas d'influence visible entre la région de l'acheteur et le type de réclamation si l'acheteur vit en France. Nous pouvons seulement noter qu'en Ile de France et dans la région PACA les colis sont légèrement plus susceptibles de ne pas arriver à destination.

Bien entendu, nous pouvons observer que les colis venant de l'étranger sont plus sujets à réclamation. Il peut donc être intéressant de créer une variable binaire supplémentaire pour mettre l'accent sur le fait que le colis vienne de l'étranger ou non.

On peut noter cependant que certains pays européens ont des taux de réclamations proches de la moyenne voir meilleurs (GERMANY : 39,1%, BELGIUM : 41,8%, LUXEMBOURG : 41,5%), alors que les colis provenant d'autres pays européens proches ont un taux de réclamations pour colis non reçu beaucoup plus élevé que la moyenne (19,2% pour UNITED KINGDOM). Il semble donc difficile de regrouper ces pays en catégories et il sera nécessaire de créer une indicatrice pour chacun de ces pays lors de la phase de feature engineering.

BUYING_DATE :

Cette variable est codé au format 'MM/AAAA' dont nous extrayons uniquement le mois qui est ici l'élément qui nous intéressent car toutes les ventes sont réalisées sur l'année 2017.

Nous remarquons d'ailleurs que les données s'arrêtent au mois d'octobre, ce qui est regrettable puisque la période de Noël, particulièrement prolifique pour les sites de e-commerce, n'est pas représentée. Il aurait été intéressant de visualiser l'impact de cette période sur les réclamations.

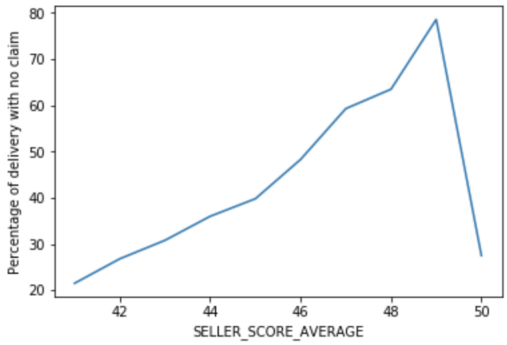
Dans ce jeu de données, le mois de Janvier est celui où le plus de commandes sont réalisées, et c'est aussi le mois où le plus de réclamations sont faites. De même, le mois d'Octobre est le plus calme en terme de ventes mais aussi en terme de pourcentage de réclamations faites. La tendance est donc que plus il y a de flux de commandes plus il risque d'y avoir des réclamations.

SELLER_SCORE_COUNT et SELLER_SCORE_AVERAGE :

Logiquement, nous observons que le nombre de produits vendus par le vendeur est négativement corrélé avec

le taux de réclamations portées. De même, les vendeurs avec les meilleurs notes moyennes sont les plus fiables. En particulier, les vendeurs avec une note moyenne de 49/50 sont peu nombreux (3994) et au dessus du lot en terme de fiabilité (seulement 12,4% de réclamations). Il peut donc être interessant de leur attribuer une variable binaire pour renforcer cet écart de fiabilité.

Cependant, il faut noter qu'un très petit nombre de vendeurs (51) ont une note moyenne maximale de 50/50 et ont cependant un taux de réclamations moyen très élevé (72,5%). Ces vendeurs sont peut-être des comptes fake qui ont réussis à obtenir la note maximale pour attirer des clients, ou bien des anomalies. Dans tous les cas, il ne faut pas les classer parmi les bons vendeurs. Il ne faut pas non plus les considérer comme des outliers et les retirer du jeu de données car le taux de réclamations qui est lié à ces vendeurs est significatif et peut contribuer à améliorer notre prédicteur.



Nous observons de plus une corrélation importante et prévisible entre SELLER_SCORE_COUNT et SELLER_SCORE_AVERAGE.

PRODUCT_TYPE et PRODUCT_FAMILY :

Nous remarquons que les produits appartenant à la famille ELECTRONICS ont le taux le plus élevé de réclamations DAMAGED (10,8% contre 5,1% en moyenne). En effet les produits électroniques tels qu'une télévision sont plus fragiles et susceptibles d'être endommagé lors du transport.

ITEM_PRICE :

Les produits les moins chers sont moins susceptibles de mener à une réclamation. En particulier les réclamations pour FAKE sont quasiment toujours des produits peu chers. De même ce sont surtout les achats peu chers qui risquent de ne pas arriver à destination. Les produits coûteux ont en revanche plus de chance de mener à un retrait de commande.

Les produits dont le prix est compris dans l'intervalle 100<500 ont typiquement plus de chance d'arriver endommagé. Cela correspond au prix des appareils électroniques qui comme nous l'avons observé précédemment sont plus souvent endommagé lors du transport.

3 Feature Engineering

Cette section présente les principales transformations du jeu de données mises en oeuvre. Cela comprend en particulier la transformation des variables catégorielles en indicatrices ou dummy variables (SHIPPING_-MODE, PRODUCT_FAMILY,...), le mapping des variables ordinales en variables d'entiers (SHIPPING_-PRICE, PURCHASE_COUNT, ...), compléter les valeurs manquantes et enfin créer de nouvelles variables.

Variables transformées :

- REGISTRATION_DATE est transformée en BUYER_SENIORITY indiquant le nombre d'années d'ancienneté de l'utilisateur sur le site plutôt qu'une année
- BUYER_BIRTHDAY_DATE est remplacée par BUYER_AGE pour la même raison

Variables créées :

- UNEXPERIENCED_BUYER pour les individus qui ont effectué moins de 5 achats sur le site
- VIP_SELLER pour les vendeurs ayant une note de 49/50
- BUYER_REGION et SELLER_REGION représentant les régions des vendeurs et acheteurs. Ces variables viennent renforcer les variables BUYER_DEPARTMENT et SELLER_DEPARTMENT.
- BUYER_IS_ABROAD et SELLER_IS_ABROAD indiquant que le vendeur/acheteur est à l'étranger.
- SAME_REGION_BUYER_SELLER et SAME_DEPARTMENT_BUYER_SELLER pour indiquer si l'acheteur et le vendeur d'une commande sont issus da la même région ou du même département.
- SELLER_BUYER_REGION_DISTANCE distance en kms entre la ville principale de la région du vendeur et la ville principale de la région de l'acheteur. Cette variable, obtenue en réalisant une matrice de distance relatives entre chaque région permet d'apporter une information supplémentaire quant à la distance approximativement parcourue lors de la livraison lorsque le vendeur et l'acheteur sont en France.
- SELLER_COUNTRY_DISTANCE indiquant la distance relative entre le pays du vendeur et la France. Cette distance est échelonnée suivant une échelle de 0 à 4, 0 étant pour FRANCE METROPOLITAINE et 4 pour les pays les plus éloignées tels que JAPON, AUSTRALIE. Cette variable vient donc prolonger la variable SELLER_BUYER_REGION_DISTANCE lorsque le vendeur est à l'étranger.

Remplacement des valeurs manquantes :

- BUYER_AGE : méthode ffill
- WARRANTIES_PRICE : les valeurs manquantes correspondent au cas où l'utilisateur n'a pas pris de garantie (même quantité de valeurs manquantes qu'il n'y a de False dans la variable WARRANTIES_-FLG) et sont donc mises à 0.
- SHIPPING_PRICE : nous avons considéré que les valeurs nulles correspondent au cas où les frais de livraisons sont gratuits et donc mis ces valeurs à 0

Cas particulier :

PRICECLUB_STATUS peut être vu comme une variable catégorielle nominale et être transformée en variables binaires. Cependant, il y a une hiérarchie entre les différents status : UNSUBSCRIBED < REGULAR < PLATINUM < SILVER < GOLD. Nous avons donc préféré transformer cette variable en une variable d'entiers (0 pour UNSUBSCRIBED jusqu'à 4 pour GOLD).

4 Prédicteurs et résultats

4.1 Comparaison des modèles mis en place

Les modèles présentés ci-dessous ont été testés en cross validation avec des paramètres par défaut parmi les algorithmes suivants : Random Forest, Gradient Boosting avec XGBoost, Multilayer Perceptron, KNN. Si les performances en cross validation étaient similaires entre les différents algorithmes testés, nous utilisions alors un GridSearchCV pour optimiser les paramètres des modèles et déterminer le meilleur. Si au contraire un modèle était d'emblée au dessus des autres en termes de performances, seul cet algorithme était alors optimisé avec GridSearchCV.

1- Le premier modèle mis en place avait pour but d'éviter la surcharge de variables et de synthétiser l'information dans des variables plus parcimonieuses. C'est dans ce but que les variables BUYER_REGION et SELLER_REGION ont été créées à la place de BUYER_DEPARTMENT et SELLER_DEPARTMENT. Par exemple, la variable BUYING_DATE avait été synthétisée en seulement 3 groupes significatifs représentant les 3 périodes de l'année où les pourcentages de réclamations et le nombre de ces réclamations étaient similaires. De même, la variable PRODUCT_TYPE qui possède 137 niveau n'avait été restreinte qu'aux types ayant un impact visible sur le type de réclamation.

AUC : 0,57 algorithme : random forest (n_estimators=100).

2- Le deuxième modèle mis en place a consisté cette fois-ci à garder le plus d'information possible quitte à risquer d'avoir un peu de sur-apprentissage. Nous avons ainsi à la fois gardé les informations liées aux départements mais aussi celles liées aux régions et transformé l'ensemble des niveaux de la variable PRODUCT_TYPE en variables binaires.

AUC : 0,60 algorithme : random forest (n_estimators=100)

AUC	Model
0.601	Random Forest
0.586	MLP
0.571	Gradient Boosting
0.569	KNN

3- L'avancée majeure de notre modèle vient avec cette troisième phase où nous avons tenter de prédire dans un premier temps la présence ou l'absence de réclamation et le cas échéant, prédire le type de réclamation avec un second modèle entrainé uniquement sur les données où une réclamation a été portée. Les variables restent les mêmes qu'au modèle précédent. La motivation derrière ce changement est que la classe '-' représente 50% de l'effectif total et correspond aussi au cas singulier de l'absence de réclamation, alors que les autres labels correspondent à un type de réclamation. C'est avec cette amélioration que nous nous sommes hissés dans le top 10 du classement.

AUC : 0,63 algorithme de prédiction claim/no claim : random forest (n_estimators=200). algorithme de prédiction du type de claim : random forest (n_estimators=100).

4- Une étape importante a ensuite été de modifier l'encodage du vecteur des réclamations qui était à présent encodé avec un labelBinarizer qui est typiquement utilisé dans les problèmes de classification multiclasses pour obtenir des variables binaires. Nous avons utilisé ici un encodage différent en associant un entier à chaque type de réclamation, classés par ordre de fréquence d'apparition dans le jeu de donné : 0 lorsqu'il n'y a pas de réclamation jusqu'à 7 pour FAKE. Cet encodage a apporté un gain non négligeable dans notre score avec les mêmes algorithmes.

AUC : 0,64 algorithme de prédiction claim/no claim : random forest (n_estimators=200). algorithme de prédiction du type de claim : random forest (n_estimators=100).

AUC	Model 1	Model 2
0.643677	Random Forest	Random Forest
0.633005	Random Forest	GBM
0.629802	Random Forest	MLP
0.627927	GBM	Random Forest
0.600046	MLP	Random Forest

5- Les dernières modifications apportées à notre modèle résident dans la création des features liés aux distances relatives entre acheteur et vendeur SELLER_BUYER_REGION_DISTANCE et SELLER_COUNTRY_-DISTANCE.

AUC : 0,649 algorithme de prédiction claim/no claim : random forest (n_estimators=200). algorithme de prédiction du type de claim : random forest (n_estimators=100).

Avec cette dernière modification nous avons alors atteint la 2nde place du classement, le premier groupe ayant un score de 0,650 à cet instant.

4.2 Analyse des résultats

Nous pouvons noter que l'algorithme de random forest s'est toujours avéré le plus efficace dans tous les types de modèles testés.

Parmi les paramètres testés avec GridSearchCV seul le paramètre n_estimators semble déterminant pour notre problème, les autres paramètres étant systématiquement optimisés avec leur valeur par défaut :

- max_features='auto'
- criterion='Gini'
- max_depth=None

Un des changement déterminant dans l'amélioration de notre score a été le passage d'un modèle de prédiction simple à un modèle en deux temps, avec un premier algorithme de classification binaire claim/no claim. Nous pouvons observer avec les matrices de confusion ci-dessous la différence majeure qui réside dans ce changement de méthode.

Matrice de confusion du modèle 2 :

	-	WITHDRAWAL	DAMAGED	DIFFERENT	SELLER_CANCEL_POSTERIORI	NOT_RECEIVED	UNDEFINED	FAKE
-	14384	39	42	20	210	254	31	1
WITHDRAWAL	1771	145	6	5	56	36	5	0
DAMAGED	1639	14	60	5	15	43	6	1
DIFFERENT	1155	5	9	69	17	14	3	0
SELLER_CANCEL_POSTERIORI	3345	15	6	8	735	42	8	0
NOT_RECEIVED	3503	12	17	10	74	875	18	1
UNDEFINED	1008	9	14	2	20	40	125	0
FAKE	53	0	0	0	0	0	0	0

Matrice de confusion du modèle 3 :

	-	WITHDRAWAL	DAMAGED	DIFFERENT	SELLER_CANCEL_POSTERIORI	NOT_RECEIVED	UNDEFINED	FAKE
-	10030	105	3351	46	608	785	54	2
WITHDRAWAL	684	193	913	6	126	95	7	0
DAMAGED	695	20	877	10	42	121	17	1
DIFFERENT	556	16	505	79	50	63	3	0
SELLER_CANCEL_POSTERIORI	1595	36	1296	9	1075	136	12	0
NOT_RECEIVED	1509	24	1465	17	176	1297	20	2
UNDEFINED	364	17	552	2	51	101	131	0
FAKE	6	1	38	1	1	4	0	2

En effet ce changement réside dans le fait que le modèle 2 a plus tendance a prédire qu'il n'y aura pas de réclamation (biais conservateur) alors que le modèle 3 au contraire est plus susceptible de prédire qu'il y a réclamation (biais libéral). Le modèle 2 prédit majoritairement '-' qui est la classe majoritaire mais reste tout de même performant sur les autres classes. En effet, en dehors des prédictions '-' le modèle 2 prédit majoritairement les bons types de réclamations. Cependant, le modèle 3 qui "prend plus de risque" est plus performant car il prédit mieux chacun des types de réclamations.

5 Conclusion

Ce projet nous a permis de mettre en pratique nos acquis en machine learning et en data science en nous confrontons a un problème issu de l'industrie. Nous avons notamment réaliser au cours de cette étude toute l'importance du feature engineering. L'ajout d'une seule variable peut s'avérer déterminante, comme cela a été le cas avec les derniers features de comparaison des distances entre acheteurs et vendeurs. Cette étape est aussi celle qui demande le plus de temps et d'effort et ne doit pas être bâclée en se contentant de transformer les variables catégorielles du jeu de donnée en variables binaires. De plus, nous avons pu nous rendre compte que l'intuition et l'expérimentation est aussi essentielle en machine learning. Beaucoup d'idées peuvent sembler bonnes et suivre une certaine logique mais il faut avant tout tester ces idées pour voir émerger un modèle efficace. C'est dans cette démarche que nous avons pu constater l'efficacité de passer sur un modèle de prédiction en deux temps.

Pour améliorer notre modèle nous aurions pu pousser plus loin l'étude des distance entre vendeurs en acheteurs en calculant les distance entre départements plutôt qu'entre région. Nous aurions pu aussi tenter de mettre en place une méthode plus avancée telle que le stacking qui est souvent utilisée pour gagner les challenge de data science. Cependant, la deuxième place que nous occupons à l'heure actuelle est très encourageante.