UNIVERSITÉ LAVAL

Faculté des sciences et de génie
Département d'informatique et de génie logiciel

GLO - 4027
Analyse et traitement de données massives

Partie 4 : Résultat final

Microsoft Malware Prediction

https://www.kaggle.com/c/microsoft-malware-prediction/data

1er Cycle

Membres de l'équipe réalisatrice:

- **William Kirouac-Samson**
- **Meryem Chafry**

1. Introduction

Suite aux suggestions proposées dans le dernier rapport et aux problèmes détectés, le travail effectué dans cette dernière étape de ce projet consiste à corriger et à améliorer les points suivants:

- Réduire l'ensemble d'entraînement de données tout en appliquant nos algorithmes sur un ensemble de données représentatif.
- Combiner les attributs fortement corrélés.
- Explorer plus d'algorithmes qui peuvent porter des solutions à notre projet.

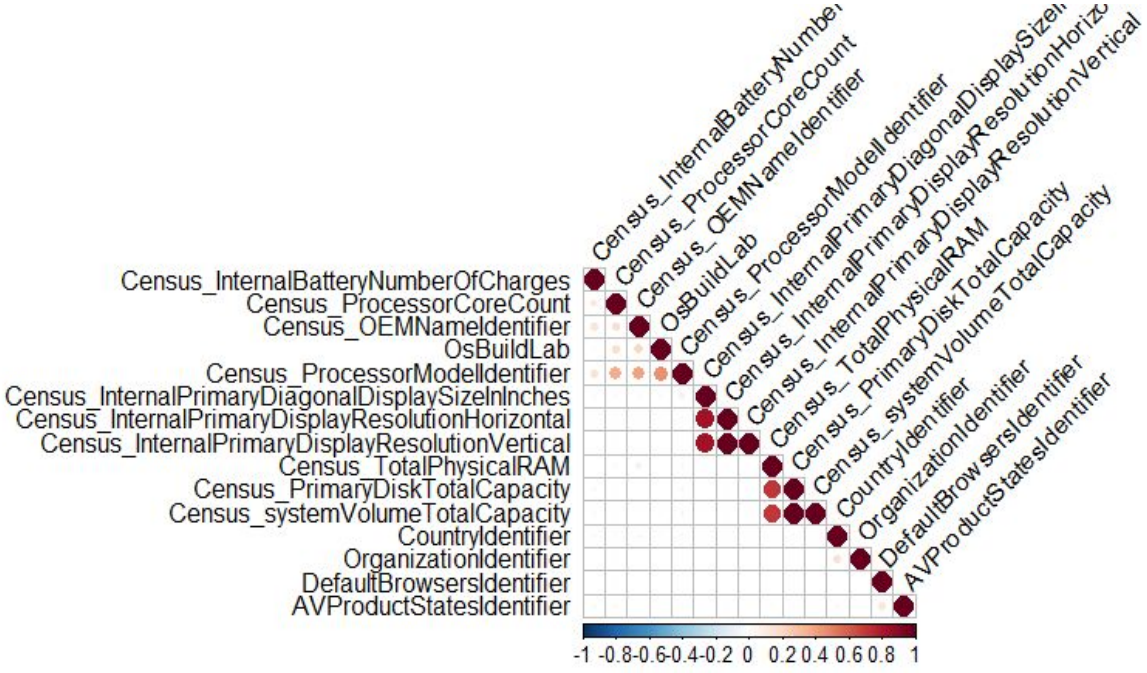
À ce fait, nous présenterons dans ce rapport les différents algorithmes implémentés afin d'effectuer le traitement de données. Nous détaillerons également leurs fonctionnements et les différents résultats obtenus.

Ainsi, ce rapport consiste à comparer les différentes solutions trouvées, les discuter et les évaluer.

Une rétrospective sera offerte à la fin du rapport afin de remettre le point sur le plan initial et d'expliquer les choses qu'on aurait pu changer si le projet était à refaire.

2. Prétraitement de données

Vu le problème du temps de calcul traité dans le rapport précédent, dû non seulement à la taille du dataset mais aussi aux dimensions du problème, nous avons choisi dans un premier temps d'aller chercher les attributs numériques fortement corrélés.



Comme on peut le constater à partir de la visualisation graphique de la matrice de corrélation des attributs numériques, représentée ci-dessus, les attributs qui représentent par exemple la taille de l'écran (*Census_InternalPrimaryDiagonalDisplaySizeInches,Census_InternalPrimaryDisplayResolutionHorizontal, Census_InternalPrimaryDisplayResolutionVerical)* sont très fortement corrélés (valeur de corrélation positive très proche de 1). Ce qui nous a menés à combiner les trois attributs dans une seule variable représentative *PixelsPerInches.*

On retrouve la presque la même corrélation pour les deux attributs (*Census_PrimaryDiskTotalCapacity Census_systemVolumeTotalCapacity*) auxquels on reprend la même décision, mais cette fois-ci avec une variable représentant le logarithme des rapports des deux attributs vus qu'elles sont des larges quantités.

3. Algorithmes implémentés

Les différents modèles testés sont la régression logistique, les glms boostés et l'arbre de décisions avec et sans pruning. D'après les critères de l'AIC et l'aire sous la courbe ROC (AUC) nous avons comparé les différents modèles de régression logistique. La performance des autres modèles a été comparée seulement sur la base de l'AUC. La solution retenue est un modèle de régression logistique avec un choix très pointu d'attributs. Ce modèle correspond au meilleur compromis entre la performance AUC et AIC ainsi que la complexité du modèle et de sa robustesse.

3.1 régression logistique

Tout d'abord, l'exercice de prétraitement des données a donné lieu à un problème dans l'estimation des paramètres du modèle complet. Le problème a été rapidement cerné alors que nous avons retiré du modèle les variables nominales. Ces variables, souvent à très haute cardinalité, ont été regroupées suivant l'ensemble de l'échantillon. Toutefois, en procédant à l'extraction d'environ 1% des données complètes, certaines variables se trouvaient à avoir des modalités à très petites fréquences. Ceci donnait lieu à des variables à facteur où certaines modalités donnaient tout le temps lieu à des probabilités de 1 ou 0 sur la variable réponse. Les paramètres prenaient de très petites valeurs, car il tentait de déduire l'équation du type :

$$0 = Pr(y_i = 1 \mid x_j) = e^{\beta_j x_{ij}} \qquad (3.1)$$

Où la seule solution était de faire tendre β_j vers $-\infty$ car la variable x_{ij} est une variable dummy donc prend une valeur de 0 ou 1. Pour régler ce problème, un deuxième prétraitement a été appliqué afin d'éliminer ces facteurs à faibles fréquences suite à l'échantillonnage.

3.1.a Modèle complet

L'ensemble des variables disponible suite au prétraitement des données a été inclus dans ce modèle. Les variables continues ont été standardisées par :

$$x_i^* = \frac{x_i - \overline{x}}{\sigma} \quad (3.2)$$

Où x_i est la variable d'intérêt, $\overline{x}$ est la moyenne et σ l'écart-type. De plus, chaque variable nominale a été transformée en variables indicatrices sur chacun des facteurs.

Après toutes ces transformations, le nombre de variables retenues au modèle en excluant la variable réponse est 115. Le modèle produit un AIC (*Critère d'akaike*) de 125 911 et un AUC de 0.694. Ces résultats sont comparables au résultat des autres participants Kaggle.

3.1.b Modèle avec modifications sur les variables *Target encoding*

Lors du processus de prétraitement des données, nous avons encodé certaines variables à haute cardinalité selon le processus de targes encoding. Soit, en remplaçant chacun des facteurs par sa moyenne dans l'échantillon. Ces variables sont fortement associées avec la probabilité de la variable réponse. Au final, 7 variables de ce type ont été retenues pour le modèle dans 3.1.a. afin de produire une forme de robustesse au modèle à suivre, nous avons décidé d'encoder ces 7 variables avec une Analyse en composantes principales. La projection de ces variables sur l'espace propre de ces 7 variables mène à 7 variables non corrélées entre elles.

Ceci aidera énormément, car les 7 variables ont été construites selon la variable réponse et sont donc associées entre elles. De plus, en gardant un sous-échantillon des composantes principales, ont réduira la variance totale de ces 7 variables et fournira une forme de régularisation sur la solution finale.

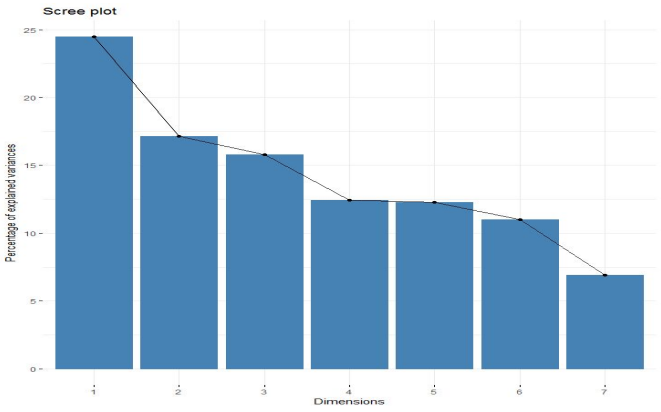


Figure 2 : Proportion de la variance expliquée par chaque composante.

La figure 2 permet de voir la proportion de variances expliquées par chacune des composantes principales. Entre autres, la première composante représente près de 25% de la variance. La deuxième composante représente 17% et la troisième composante près de 16%. Après la troisième composante, la proportion de variances chute brusquement nous donnant une indication sur le nombre de composantes à récupérer.

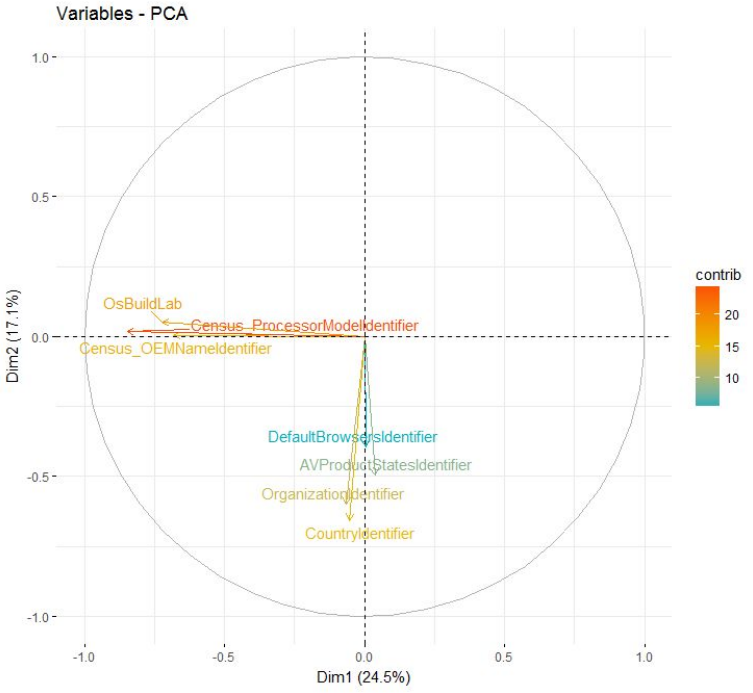


Figure 3 : Projection des variables dans les deux premières composantes principales.

La figure 3 montre la contribution de chaque variable dans les 2 premières composantes. La dimension 1 représentant 24,5 % de la variabilité totale est fortement associée aux variables *CountryIdentifier* ainsi que *OrganisationIdentifier*.

Ces deux variables sont fortement associées, car la compagnie est localisée dans un pays en particulier. Ainsi, les deux variables mènent à des probabilités de machine défectueuses dans une Organisation/Pays similaire. La dimension 2 représentant 17,1% de la variance est expliquée majoritairement par la variable *Census_ProcessorModelIdentifier*. Dans la description des variables, cet attribut n'est pas explicité. Toutefois, on peut déduire qu'il s'agit d'un modèle de processeur de l'ordinateur. Cette composante va dans le même sens que *OsBuildLab* et *Census_OEMNameIdentifier*. Le *OSBuildLab* correspond à un laboratoire où le système d'exploitation a été généré. On peut déduire que la deuxième composante est associée à certaines spécificités de la machine.

Finalement, le modèle a été testé avec l'ensemble des composantes principales n'offrant pas nécessairement de différences avec le modèle complet autre qu'une certaine régularisation afin de limiter le surentraînement des données. C'est pourquoi l'aire sous la courbe ROC et l'AIC demeure inchangée face au modèle complet.

3.1.c Modèle en se limitant sur les 5 premières composantes principales des 7 variables

Telle qu'expliqué plus tôt, le but de projeter les variables hautement corrélées sous son espace propre permet de dé corrélés les variables, réduire la variance et offrir une régularisation sur la solution finale offrant une ainsi une certaine robustesse.
Le fait de garder les 7 composantes principales, nous n'avons pas nécessairement de gains. En limitant sur les 5 premières composantes, la proportion de variances récupérées est d'environ 85%. L'aire sous la courbe ROC est alors de 0.694 avec un AIC de 126 950. Le fait de réduire le nombre de composantes n'affecte pas le ROC, mais augmente l'AIC.

3.1.d Modèle en se limitant sur les 3 premières composantes principales des 7 variables

Les 3 premières composantes sont associées à une variance de 58%. L'aire sous la courbe ROC est alors de 0.689 avec un AIC de 126 824. Cette combinaison nous donne une certaine assurance sur la contribution des composantes principales dans le modèle. Le fait de projeter 7 variables sur 3 dimensions sans trop affecter l'aire sous la courbe ROC nous dit que le modèle ne repose pas seulement sur les 7 variables. Les solutions suivantes seront testées avec les 3 premières composantes principales.

3.1.e Modèle en se limitant sur les 3 premières composantes principales des 7 variables et en excluant les variables n'étant pas significatives

Les variables n'étant pas explicatives selon le test :

$$H_0 : \widehat{\beta_j} = 0 \; ; \; H_1 : \widehat{\beta_j} \neq 0 \qquad (3.3)$$

Selon un seuil établie à 5%, 15 variables ont été retirées. Le retrait de ces variables permettra de réduire la complexité de la solution et de permettre une interprétation plus globale au problème.

Le nombre de variables incluses au modèle est alors de 30 variables. Après la transformation des variables facteurs en dummy variables, le nombre de paramètres à estimer était de 74. À cette étape, l'aire sous la courbe ROC est de 0.689 et un AIC de 126 827. Ceci correspondra à notre dernier modèle de régression logistique.

3.2 GLM additif par descente du gradient

Nos premiers tests avec les Glm boostés on produit de plus faibles performances que les Glm non boostés. L'estimation des paramètres a été effectuée par validation croisée en subdivisant l'échantillon en 5 partitions. La meilleure itération parmi les 100 a produit une aire sous la courbe ROC de 0.687 ne représentant pas de gains significatifs face au modèle de régression logistique.

De plus, en regardant l'influence des variables incluses dans le modèle, nous nous sommes aperçus que les performances s'appuyaient grandement sur l'influence de 3 variables uniquement. Ces 3 variables expliquaient 88% de la variabilité totale du modèle.

Ceci n'offrant pas nécessairement de robustesse à notre solution nous à poussé d'abandonner les modèles par descente du gradient.

3.3 Arbre de décision

La dernière solution testée est un arbre de décision. Dans les rapports précédents, nous avions l'intuition que la meilleure prévision s'appuierait sur des relations plus complexes entre les variables et les facteurs. Toutefois, les prévisions obtenues n'étaient pas aussi fortes que les glm et les modèles additifs.

En testant diverse méthode d'optimisation, par exemple l'élagage, les meilleurs résultats obtenus ont été une aire sous la courbe ROC d'environ 0.61. l'arbre construit s'appuyait encore une fois sur un faible échantillon des variables incluses et ne montrait aucune complexité s'appuyant sur les facteurs.

3.4 Solution retenue

Voici un résumé des différentes solutions testées expliquées dans la section 3 du rapport.

Modèle	AIC	AUC	Nb Paramètres
(1) GLM : Complet	125 911	0,694	115
(2) GLM : ACP (Toutes les composantes)	125 911	0,694	115

(3) GLM : ACP (5 premières composantes)	126 950	0,694	113
(4) GLM : ACP (3 premières composantes)	126 824	0,689	111
(5) GLM : (4) avec variables significatives.	126 827	0,689	74
(6) GLBM	-	0,687	11
(7) Arbre de décision	-	0,610	4

Tableau 1 : Résumé des solutions testées dans la section (3)

D'après le tableau 1, la solution optimale semble est le Glm avec seulement les variables significatives. Ce choix est le meilleur compromis entre complexité et performance du modèle.

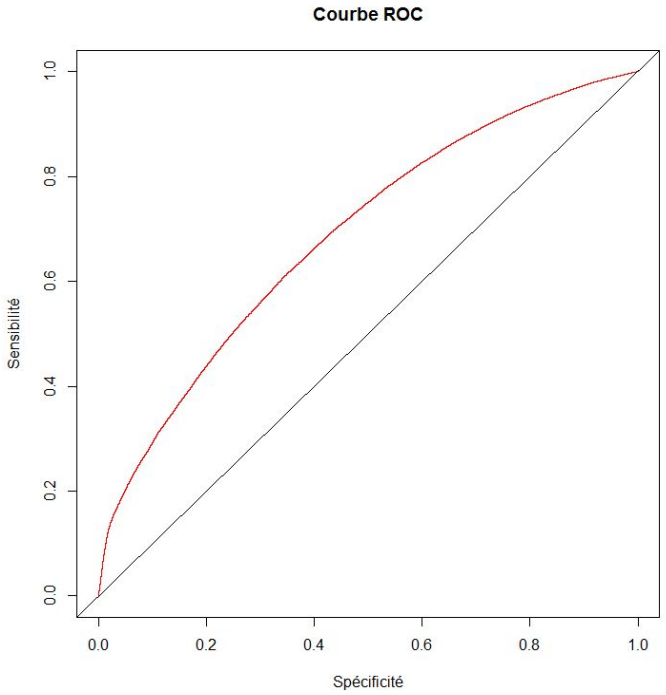


Figure 4 : Courbe ROC du modèle retenu

La courbe ROC présentée à la figure 4 ne montre aucune asymétrie ce qui est une bonne indication que le modèle prend soin de tenir le taux de faux positifs au même que le taux de vrai positif. Ce qui correspond à l'objectif fixé dans les rapports précédents.

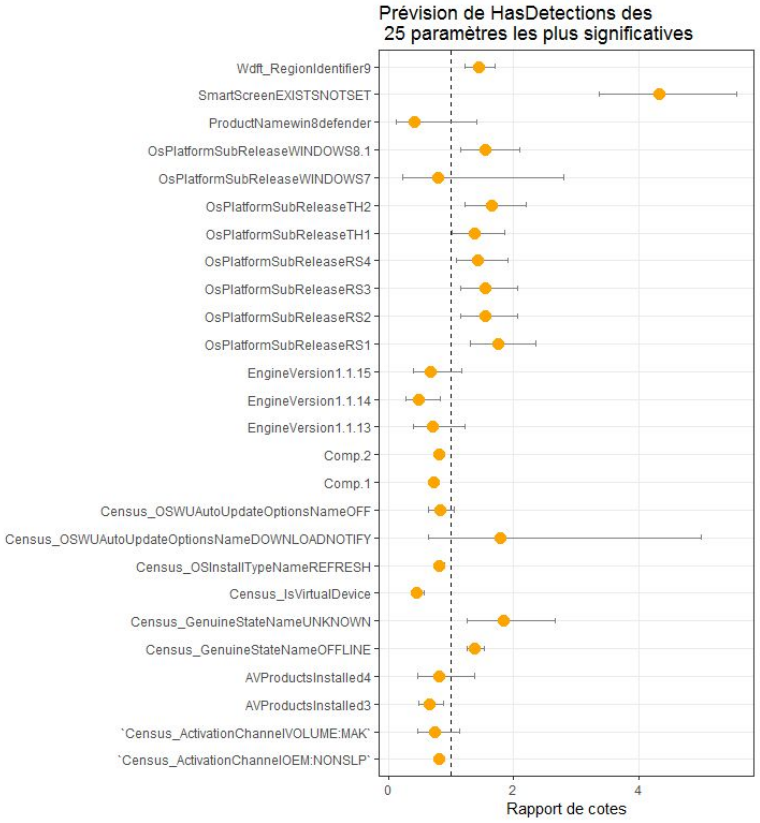


Figure 5 : Rapport de cotes

Les rapports de cotes présentés dans la figure 5 montrent l'impact de chacune des variables sur la probabilité d'avoir une machine infectée. Les 25 variables les plus significatives ont été tracées. Entre autres, quand la variable *SmartScreenExistsNotSet* est fixée à 1, la probabilité d'avoir une machine infectée est 4.33 fois plus élevée que lorsque cette variable est à 0.

4. Rétrospective

En conclusion, le projet a été très intéressant et a permis d'en apprendre énormément sur les jeux de données massifs. Le problème de cardinalité rencontré sur certaines variables nominales nous a permis d'explorer des techniques de prétraitement plus pointu au problème. Les résultats obtenus sont comparables aux autres participants de la compétition ce qui montre que les choix méthodologiques de prétraitement de données sont appropriés au problème.

Si le projet était à refaire, nous prendrons encore plus de temps dans la partie de prétraitement des données afin d'avoir plus d'un pour cent des données disponibles pour l'entraînement du modèle. De plus, on aurait exploré les pistes des autres participants un peu plus en profondeur.

La personne ayant terminé 7ème à la compétition aurait utilisé entre autre du count encoding[1] ce qui est très près des choix que nous avions pris plus tôt dans la session (target encoding).

Au premier rapport nous avions dans l'impression qu'un arbre de décision serait la solution optimale. Les résultats obtenus nous ont prouvé à tort. Probablement qu'un exercice de prétraitement des données plus axé sur un arbre de décision aurait été optimal.

Malheureusement, les prévisions n'ont pas été soumises à KAggle car la compétition à terminé entre le 2 et 3 ème rapport.

Les codes sources se trouvent sur l'espace GITHUB suivant :
https://github.com/longbeachmike/MicrosoftMalwarePredictions/

[1] https://www.kaggle.com/c/microsoft-malware-prediction/discussion/84136#latest-504909