Classifications de programmes malicieux et non-malicieux à partir de propriétés binaires

Matthias Beaupère, Pierre Granger

Rapport DA - CHPS - 15 novembre 2018

Table des matières

1 Présentation du jeu de données

Nos données proviennent de la base de données UCI[?]. Cette base de données a été obtenue à partir de l'étude de 373 programmes informatiques malicieux et non-malicieux selon le processus expliqué dans un article de recherche en 2007 [?]. Cet article développe une méthode permettant d'extraire des caractéristiques à partir d'exécutables malins et bénins afin d'effectuer par la suite une classification de ces exécutables permettant de les distinguer. Trois types de carctéristiques sont extraites : des n-uplets binaires, des n-uplets assembleur et des appels à des fonctions appartenant à des librairies extérieures. Les caractéristiques binaires sont extraites des exécutables binaires tandis que les caractéristique assembleur sont obtenues après désassemblage de l'exécutable. Les carctéristiques liées aux appels de fonctions sont extraites depuis l'entête du programme.

2 Pré-traitement des données

Nous avons commencé notre étude des données par quelques visualisations de notre jeu de données. Nous avons tout d'abord visualisé l'histogramme du nombre de caractéristiques possédé par les programmes du jeu de données représenté sur la figure 1. On peut observer sur cet histogramme que la majorité des programmes possèdent une centaine de caractéristiques tandis qu'ils sont très peu à les

posséder toutes ou bien à n'en avoir aucune. On peut en outre remarquer que la quasi totalité des virus possèdent un nombre de caractéristique situé entre 80 et 110 tandis qu'il existe des programmes bénins avec plus ou moins de caractéristiques. Cette observation pourra nous mener à tenter de restreindre certaines analyses futures aux programmes possédant entre 80 et 110 caractéristiques en supposant que les programmes bénins ne respectant pas ce critère peuvent être considérés comme des valeurs extrèmes. On peut en outre observer sur l'histogramme représenté en figure 2 que la majorité des caractéristiques ne sont possédées que par quelques dizaines de programmes. On observe néanmoins qu'il existe un nombre non négligeable de caractéristiques partagées par un grand nombre de programmes. On pourra se proposer par la suite de ne pas considérer les caractéristiques correspondantes car elles sont moins significatives.

3 Techniques de validation

On présente ici les deux techniques utilisées pour valider les modèles de classification.

3.1 Data-splitting

La première technique de validation utilisée est la technique de data-splitting. On divise aléatoirement le jeu de données en deux, la première partie servant à entrainer le modèle. On utilise la deuxième partie des données pour valider le modèle en comparant les prédictions du modèle et les valeurs réelles. Cette technique produit des résultats très différent suivant la répartition des données dans les deux jeux. C'est pourquoi on rejouera souvent un grand nombre de fois le data-splitting et on observera la moyenne et l'écart-type pour connaitre la robustesse de la validation.

3.2 Cross-validation

La deuxième technique consiste à diviser le jeu de données en n parties égales. On itère sur chaque partie de la façon suivant : les $n-1$ autres parties servent à entrainer le modèle et la partie courante est utilisée pour le valider.

4 Différents algo

4.1 Régression logistique

La régression logistique permet facilement d'effectuer une régression binomiale sur des caractéristiques binaires afin d'obtenir une classification binaire d'où sa mise en oeuvre dans le problème qui nous occupe.

La régression logistique permet d'effectuer une pénalisation de type l1 ou l2 afin de limiter le nombre de caractéristiques significatives dans le calcul de la régression.

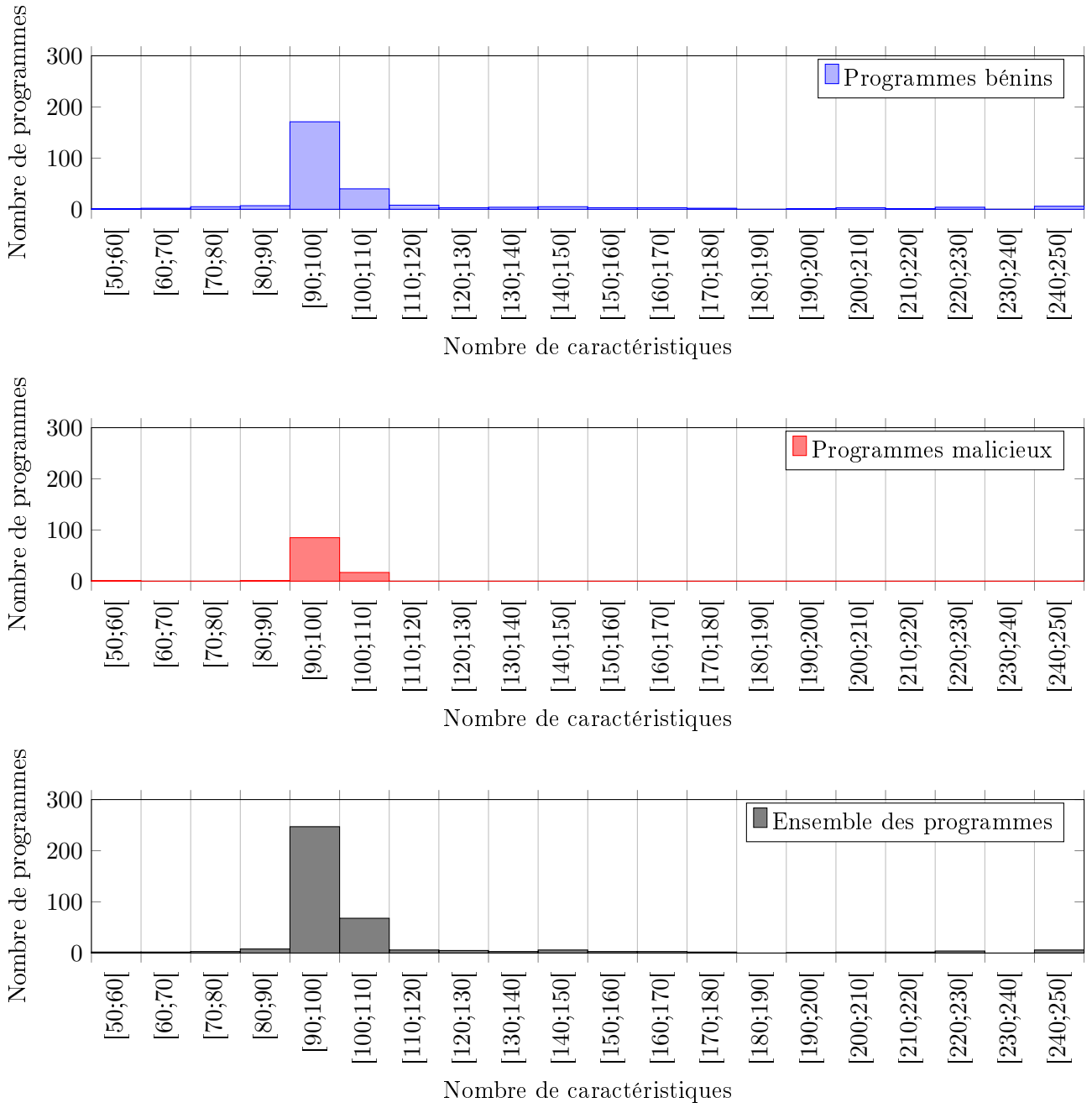


FIGURE 1 – Histogramme du nombre de caractéristiques possédé par chaque programme

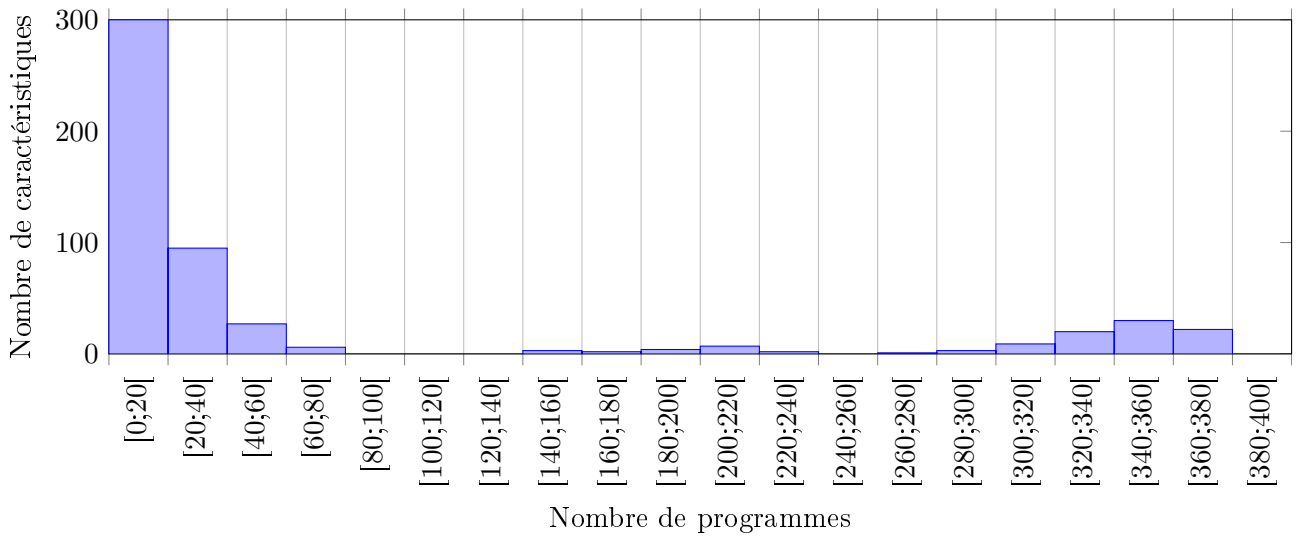


FIGURE 2 – Histogramme du nombre de programme possédant chaque caractéristique

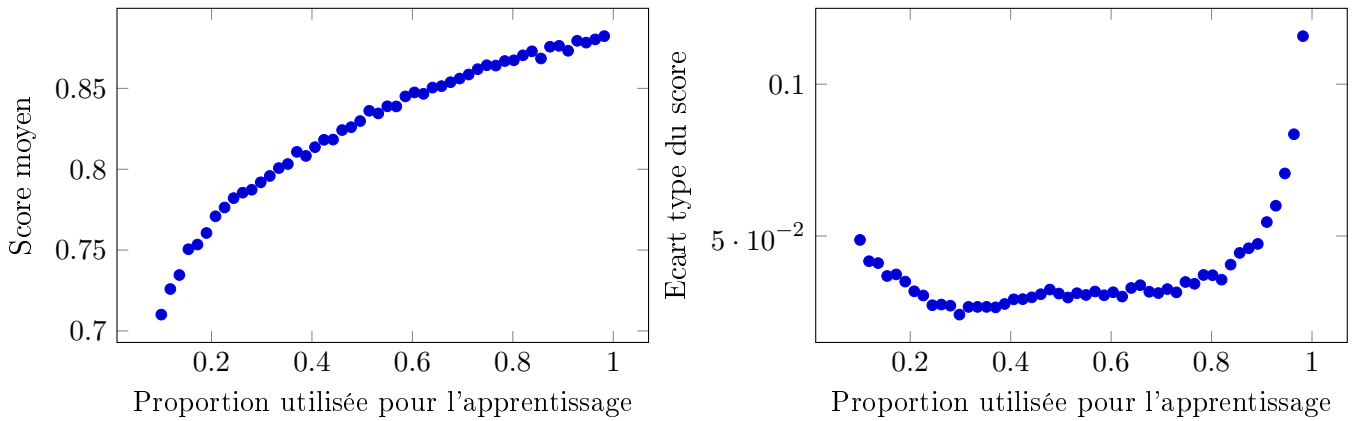


FIGURE 3 – Evolutions de la valeur moyenne et de l'écart type du score en fonction de la proportion de données utilisées pour l'apprentissage dans le cas de la régression logistique. L'évaluation est effectuée sur la partie complémentaire.

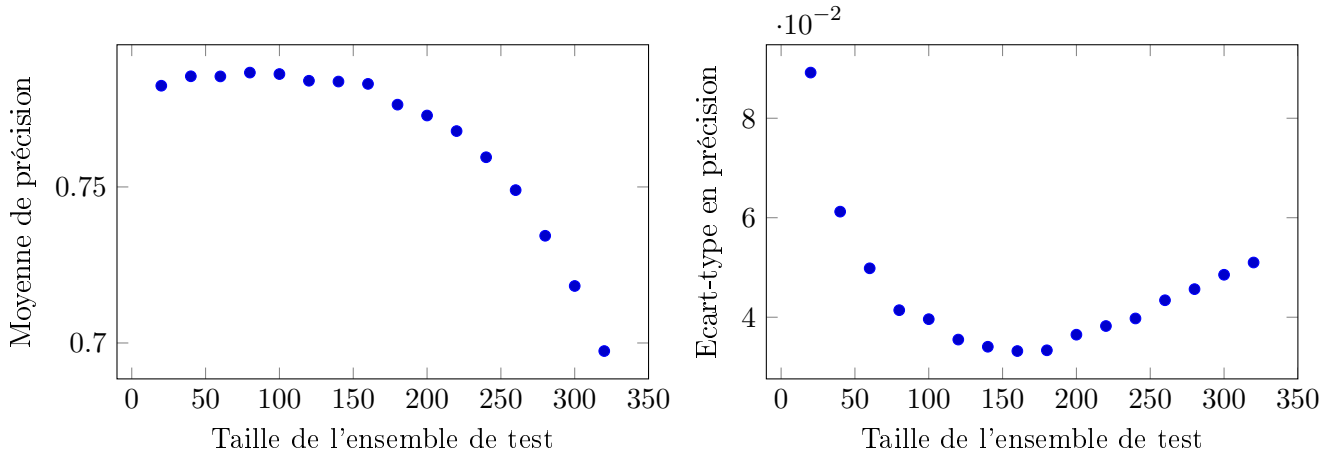
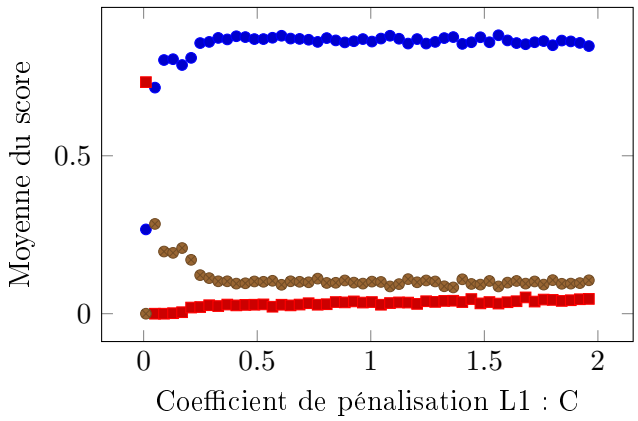


FIGURE 4 – Moyenne et Ecart-type pour 1000 data-splitting



4.2 SVM

4.3 SVC

4.4 Classification avec kmeans

4.5 Arbre de décision

4.5.1 Arbre simple

Comme premier classificateur on utilise un unique arbre de décision. On compare les deux méthodes de validation : tout d'abord le data-splitting et ensuite la validation croisée. Dans chaque cas on observe l'impact de la proportion des données d'entrainement sur la précision et la robustesse du modèle.

La figure 4 nous donne l'évolution de la précision moyenne et de l'écart-type obtenu pour un mille rejeu du data-splitting. Chaque abscisse donne un proportion des données de test. On observe un minimum d'écart-type à l'abscisse 175. Cela montre qu'on a un modèle robuste avec des données de test représentant environ 45% de la population.

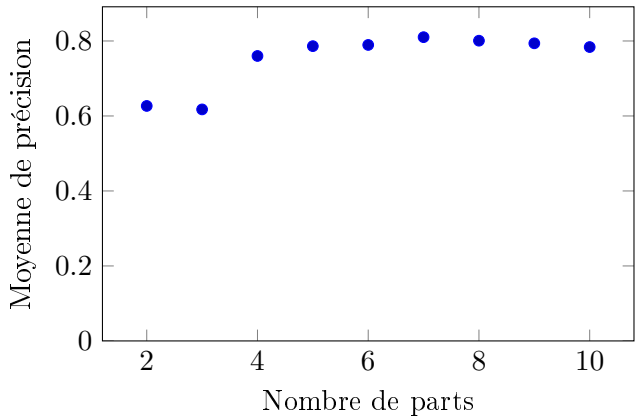
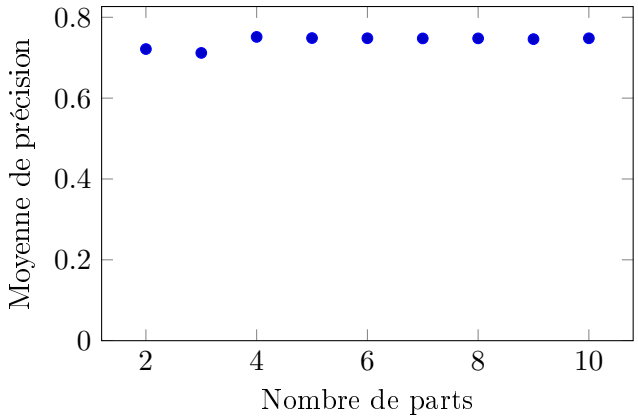


FIGURE 5 – Moyenne pour 1000 cross-validations

4.5.2 Random forest



5 Analyse des résultats

6 Pour aller plus loin...

7 Conclusion

On mérite au moins 21/20.