INF8215 - Intelligence artif.: Méthodes et algorithmes

TP3 - Phishing

Présenté à Daniel Aloise  
et Quentin Cappart  
par  
Aladji Faye - 2035541  
Louis Sérey Larose - 1846124  
Nicolas Veber -2036057

Polytechnique Montréal  
Le 13 Décembre 2020



Table des matières

[Introduction 3](#_Toc58342838)

[Présentation technique 4](#_Toc58342839)

[Prétraitement des attributs (Feature Design) 4](#_Toc58342840)

[Méthodologie 4](#_Toc58342841)

[Conclusions 5](#_Toc58342842)

[Résultats 5](#_Toc58342843)

[Discussion 5](#_Toc58342844)

[Références 6](#_Toc58342845)

# Introduction

Le travail pratique présenté ci-dessous a été effectué sur Kaggle sous le nom de l’équipe « LAN Party ». Celle-ci était composée des trois membres, soit :

* Aladji Faye – 2035541
* Louis Sérey Larose – 1846124
* Nicolas Veber -2036057

# Présentation technique

Pour arriver à nos résultats, il a été nécessaire d’appliquer plusieurs techniques vues autant en cours qu’hors cours.

## Prétraitement des attributs (Feature Design)

Décrivez et justifiez vos étapes de prétraitement des attributs et indiquez ceux que vous avez sélectionnés dans votre modèle

Nous avions 87 attributs de base, certains avec des valeurs pouvant uniquement être 0 ou 1, les autres ayant plusieurs autres valeurs (certaines étant réelles)

Le principe utilisé dans ce prétraitement est de normaliser les attributs ayant plus de deux valeurs possibles, et de laisser comme tels les attributs ayant seulement comme valeurs 0 ou 1. Pour ces valeurs, la normalisation n'est pas utile car cela revient à avoir des attributs binaires.

Pour effectuer cette différentiation, nous utilisons un dictionnaire nous permettant de compter le nombre d'éléments possibles par attribut.

Nous utilisons par ailleurs ce dictionnaire pour éliminer les attributs ayant une seule valeur possible (avec le dataset de base fourni train.csv). En effet, nous jugeons sans importance les attributs ayant une seule valeur ce qui nous permettra d'avoir des calculs plus rapides.

Finalement, 81 attributs sont retenus.

## Méthodologie

Décrivez et justifiez toutes les décisions concernant la répartition des données en ensemble d’entraînement et de validation, ainsi que les techniques utilisées pour gérer le déséquilibre entre les classes (Unbalanced data), la stratégie de régularisation, le réglage des hyperparamètres, etc. Ajoutez également toutes les informations que vous jugez nécessaires pour la compréhension de votre modèle.

Une première approche pour la séparation des données en données d'entraînement et de validation a été d'utiliser le module train\_test\_split avec une taille de 20 % pour les données de validation.

Cependant, après les résultats nous avons finalement opté pour l'utilisation de Kfold Cross validation lors des différents tests effectués pour déterminer la meilleure séparation à faire. (Avec différentes valeurs de k)

Nous avons tenté plusieurs méthodes de machine learning pour obtenir la prédiction. Les hyperparamètres ont été *tunés* avec une méthode Gridsearch

1. KNN : Nous avons utilisé KNN avec cross-validation avec k = 5.
2. Naive Bayes classifier (Gaussian) : l'hyperparamètre à tuner est la valeur de smoothing, alpha.
3. LinearSVM : Les hyperparamètres considérés sont :

* C (paramètre de régularisation)
* le nombre maximum d'itérations.

1. SVM : Les hyperparamètres considérés sont :

* C
* le kernel
* le degré du polynôme si on choisit un kernel polynomial
* gamma.

1. Decision tree : Les hyperparamètres considérés sont :

* le nombre maximal de features
* le nombre minimal de samples pour un split
* le nombre minimal de samples sur une feuille
* min\_weight\_fraction\_leaf (fraction minimale de la somme totale des poids sur une feuille)
* min\_impurity\_decrease.

1. Random Forest Classifier : Les hyperparamètres considérés sont

* la fonction de mesure de qualité ou criterion (gini ou entropy)
* le nombre d'arbres
* la profondeur maximale
* le nombre maximal de features
* le nombre minimal de samples pour un split
* le nombre minimal de samples sur une feuille
* min\_weight\_fraction\_leaf (fraction minimale de la somme totale des poids sur une feuille)
* class\_weight
* min\_impurity\_decrease

1. Deep Neural Network :

# Conclusions

## Résultats

Présentez une analyse détaillée de vos résultats à l’aide de tableaux ou de graphiques. En plus des meilleures performances obtenues, vous devez décrire et justifier l’impact de vos choix de conception (par exemple, les étapes de prétraitement, ou la régularisation) sur les performances du modèle.

## Discussion

Discutez vos résultats et indiquez quels sont les avantages et les inconvénients de votre approche et de votre méthodologie.

# Références

**Aucune source spécifiée dans le document actif.**