**INTERPRETABILITE DES CLAUSES CONTRACTUELLES**

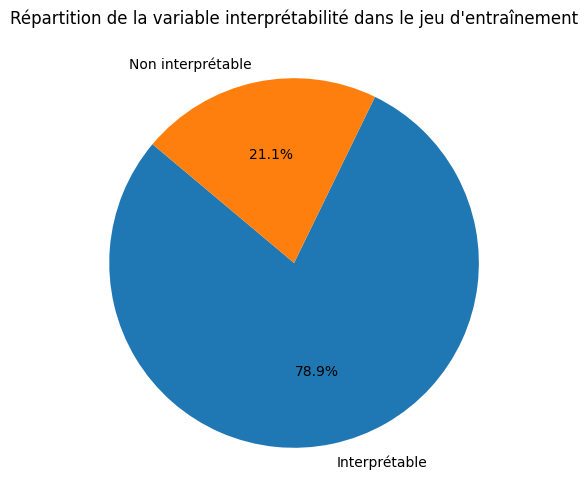
1. **Contexte et objectif**

Notre travail porte sur l’interprétabilité des clauses contractuelles dans les contrats d’assurances non-vie. L’objectif est de créer un outil par le biais du Machine Learning capable de distinguer une clause interprétable (pas clairement définie par l’assureur donc sujette à des confusions) d’une clause non-interprétable.

Cet exercice est passé par une extraction des clauses de contrat d’assurance non-vie sous format PDF grâce aux package « Fitz ». Nous avons ainsi utilisé des clauses exclues par le médiateur de l’assurance augmentées avec Copilot, des clauses des contrats d’assurance non-vie de Pacifica considérées comme non-interprétables (car Pacifica est rigoureux sur la rédaction de ses contrats d’assurance).

1. **Présentation des données**

Après construction de la base de données, on obtient un jeu de données final de **596 lignes** et **3 colonnes** *(clauses, interpretabilite, clauses\_vect)*.



* N’ayant pas suffisamment de données, nous considérons ces proportions somme un déséquilibre. sNous allons donc sur-échantillonner la classe minoritaire (interpretabilite=1) grâce à SMOTE. Nous aurions pu sous-échantillonner la classe majoritaire, mais en faisant ainsi on perd des données utiles. Il est généralement préférable d’ajouter des données synthétiques que d’en retirer.
* **Train : 417 clauses**
* **Test : 179 clauses (30 % du jeu de données)**

1. **Nettoyage des données**

Nous avons créé une fonction capable de nettoyer le texte brut et de nous ressortir « *clauses\_vect* ». Cette opération passe par :

* *Suppression des tags HTML*
* *Conversion en minuscules*
* *Tokenisation du texte en mots*
* *Suppression des stopwords*
* *Lemmatisation des mots*
* *Reconstitution du texte à partir des mots*

1. **Vectorisation des clauses**

La vectorisation a été réalisée grâce au modèle de *Word2Vec* avec les paramètres suivants :

* sentences : le modèle utilise en entrée une liste de mots
* vector\_size =100, indique la dimension des vecteurs de chaque mot.
* window = 5, il s’agit de la taille de la fenêtre contextuelle. Par exemple, dans notre cas le modèle prendra en compte 5 mots avant et 5 mots après le mot cible.
* min\_count = 1, indique le nombre d’occurrence minimum du mot pour qu’il soit pris en compte par le modèle.

1. **Modèles testés**

Nous avons commencé par tester un modèle de classification simple tel que la ***régression logistique*** qui nous sert de modèle de base.

Ensuite un modèle qui utilise de l’agrégation à savoir le ***Random Forest*** et un autre apprenant de ces erreurs à savoir ***Gradient Boosting*** ont été implémentés sur notre jeu d’entrainement.

* 1. **Hyperparamètres testés pour chaque modèle**
* Pour la régression logistique, aucune régularisation (Ridge ou Lasso) n’a été tentée car nous n’avons pas besoin dans ce cas de sélectionner les variables importantes en ramenant des coefficients à 0. Notre seul prédicteur est la variable « clauses\_vect ».
* Pour le Random Forest, nous avons testé :

**n\_estimators :** [100, 200, 300]

**max\_depth :** [0, 10, 20, 30]

**min\_samples\_split :** [2, 5, 10]

**min\_samples\_leaf :** [1, 2, 4]

* Pour le Gradient Boosting, les hyperparamètres testés sont les suivants :

**n\_estimators** : [50, 100, 200]

**learning\_rate**: [0.01, 0.1, 0.2]

**max\_depth**: [3, 5, 10]

**min\_samples\_split**: [2, 10]

**min\_samples\_leaf :** [1, 5]

**subsample :** [0.7, 1.0]

* 1. **Résultats des modèles après optimisation**

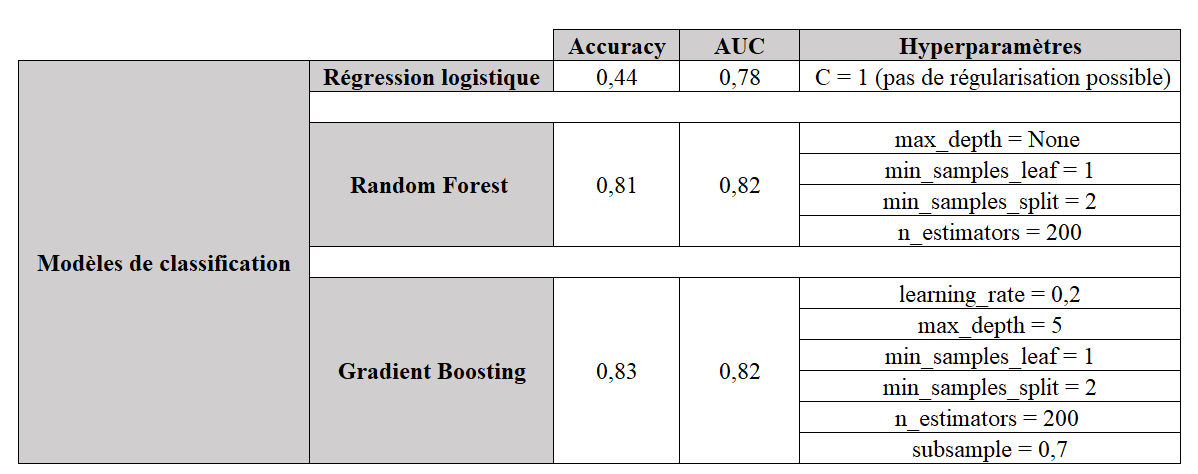
****

Tableau 1 : Métriques des différents modèles

* **Nous retenons finalement le Gradient Boosting car il donne de meilleurs métriques d’évaluation sur le jeu de test.**

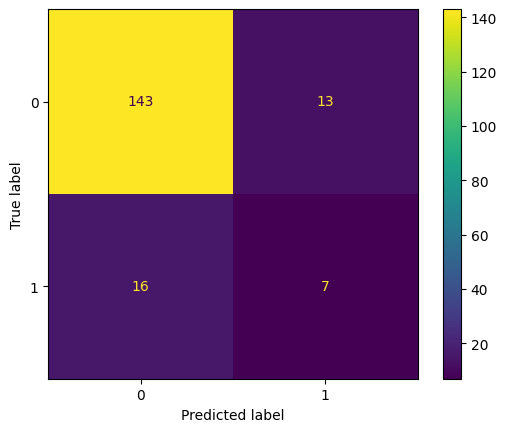
****

Figure 1 : Matrice de confusion du jeu de test avec xgboost

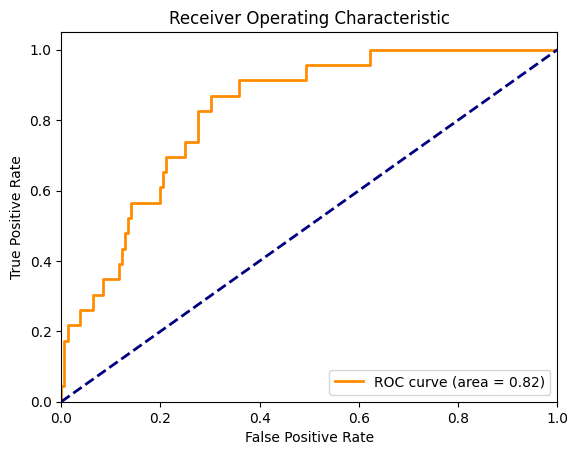
****

Figure 2 : Courbe ROC avec xgboost

L’aspect « escalier » de la courbe ROC est moins prononcé après ajout des clauses de Pacifica au jeu de données.

* 1. **Limites**
* Les différents modèles classent mieux les clauses non interprétables au détriment de celles qui sont interprétables ;
* Le modèle de vectorisation des clauses peut ne pas prendre en compte les subtilités dans les clauses contractuelles (la notion d’interprétabilité pure) 🡪 à approfondir  ;
* Les différents modèles offrent des « f1-score » relativement faibles (inférieur à 0,5) ;
* L’augmentation des données due à la faible quantité des clauses est susceptible d’introduire un biais ;