



Projet de Scoring



PRÉDIRE LA FRAUDE BANCAIRE

INTRODUCTION:

Rappel du projet




On sélectionne 22 variables qui nous paraissent les plus intéressantes.

On prépare la donnée:
outliers, rééquilibrage,
réduction de dimension...

On teste quatre modèles : le + performant sera sélectionné.

Bonus :
VotingClassifier

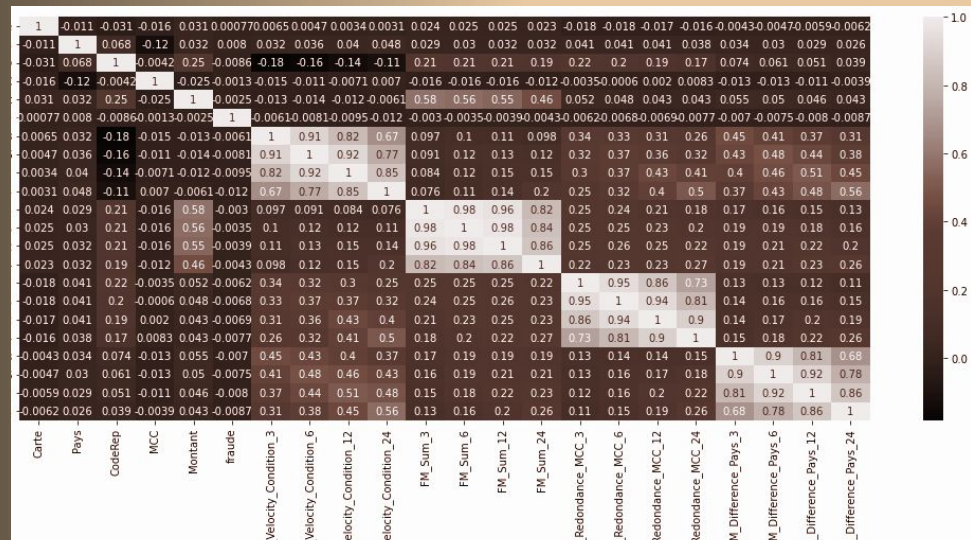
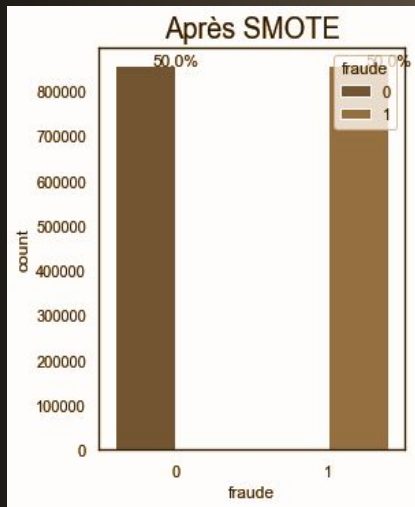
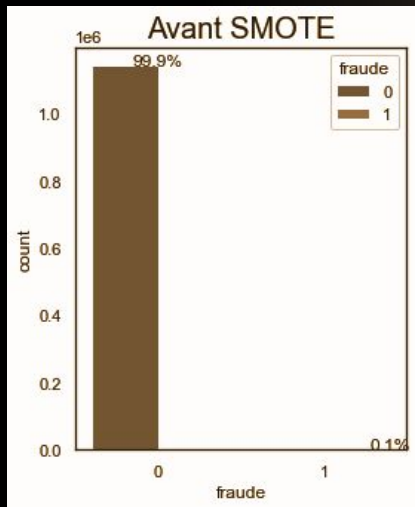
Objectif & Redéfinition des variables :

-  On souhaite se prémunir contre le risque de fraude en détectant les transactions frauduleuses : la variable *fraude* est notre Target.
-  Parmi l'ensemble des variables:
 - * nous retirons l'horodatage de la transaction.
 - * nous recodons *CodeRep* en 0/1 car on souhaite uniquement savoir si la transaction est acceptée ou non. Les codes correspondant au motif de refus ne sont pas interprétables.
-  Afin d'éviter l'overfitting et la multicolinéarité :
 - * nous utilisons la Distance de Cook pour supprimer les valeurs extrême
 - * méthode de sélection de variables

Partie 1 : SMOTE

SMOTE Synthetic Minority Oversampling Technique SÉLECTION automatique des variables

- ☞ Souhait d'une répartition équilibrée des 1 et des 0 : respectivement, les fraudes et les non-fraudes.
- ☞ On réalise un oversampling - créations d'individus synthétiques - : des individus sont ajoutés jusqu'à avoir 50% de 1.



- ☞ Réduction de dimension/Sélection variables
 - * LASSO/ACP : mauvais résultats
 - * on utilise le WoE

Partie 2 : Modélisation

☞ Vrais Positifs : on prédit que la transaction est frauduleuse & c'est bien le cas 😊

≠ Faux Positifs

☞ Vrais négatifs : on prédit que la transaction est fiable & elle l'est 😊

≠ Faux Négatifs

☞ Recall =

$$\frac{\text{Vrais positifs}}{\text{Vrais positifs} + \text{Faux Négatifs}}$$

* part des prédictions correctes

* + il est proche de 1 (100%) ; - il y a de faux négatifs, ce que l'on veut éviter le plus.

☞ Random Forest

- * méthode de bagging : vote pour chaque arbre de forêts constitués aléatoirement
- * combinaison des arbres de décision avec minimisation de la variance

☞ xgBoost

- * continuité du bagging
- * associe *Gradient Boosting* et *Random Forest*
- * *GB* : cherche à prédire les résidus

☞ Logistic Regression

- * relation entre variable expliquée et variables explicatives
- * utilise l'EMV pour estimer les coefficients
- * on veut maximiser la log-vraisemblance

☞ K-nearest Neighbors

- * classification non paramétrique
- * les points X sont classés en fonction de leurs k voisins les plus proches

Partie 3 : Résultats {1/2}

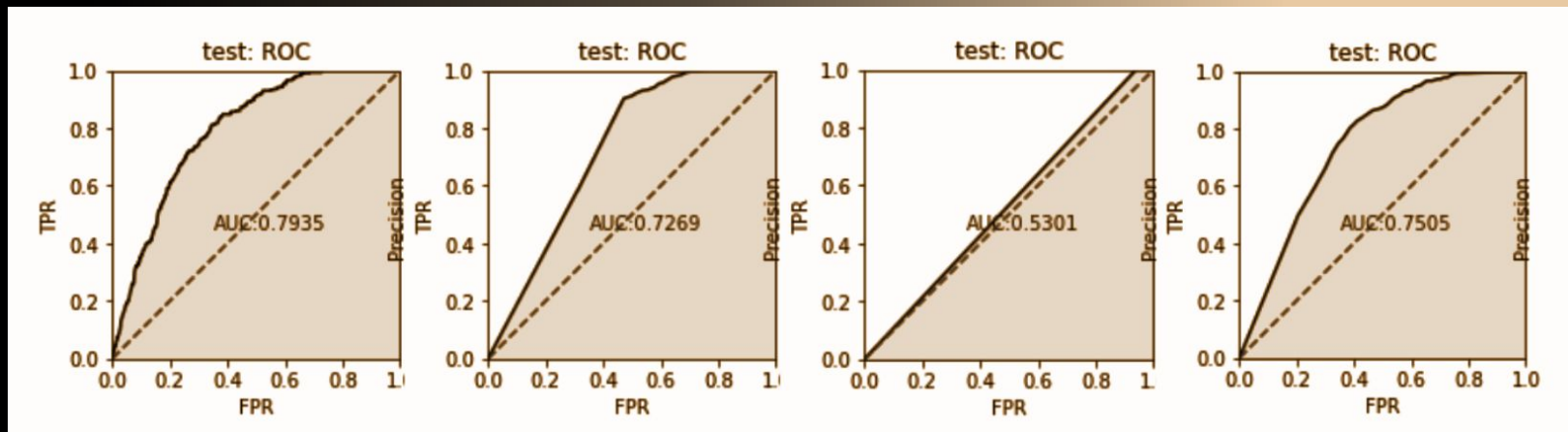


Métriques de performances

Modèle	Accuracy	Recall
Logistic Regression	0.912*	0.516*
KNN	0.859	0.469
xgBoost	0.904	0.511
Random Forest	0.903	0.510



Courbes de performances



Logistic Regression

RandomForest

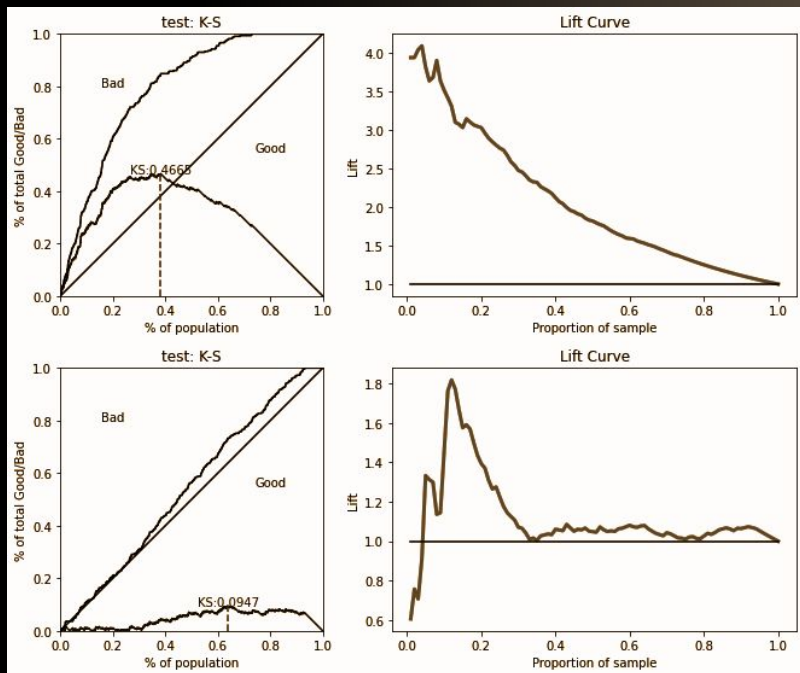
xgBoost

KNN

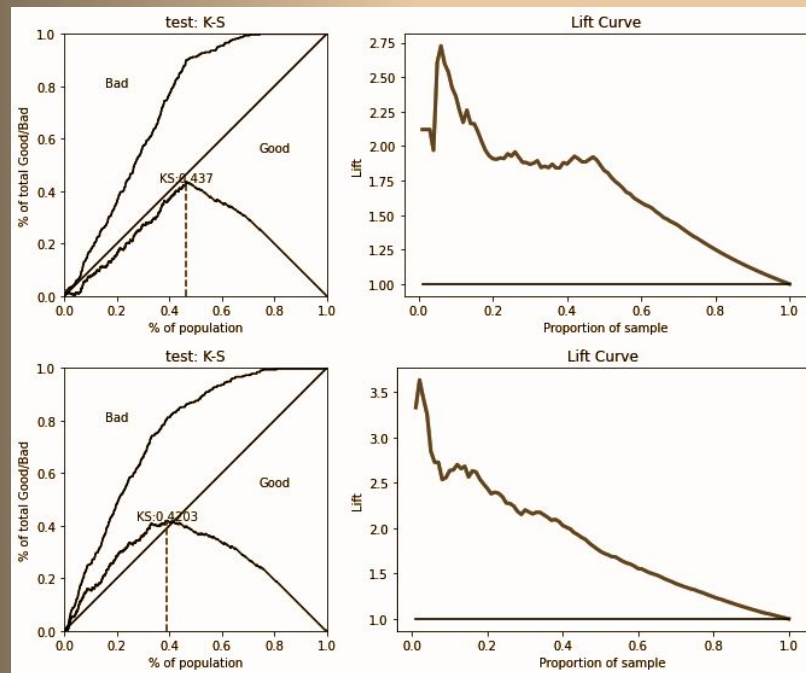
Partie 3 : Résultats {2/2}

Test de Kolmogorov-Smirnov et Courbes de Lift

Logistic Regression (haut) et RandomForest (bas)



xgBoost (haut) et KNN (bas)



Conclusion : la Logistic Regression donne de meilleurs résultats, tant au niveau des métriques (Accuracy, Recall, AUC), que pour le test de KS et la courbe de Lift.