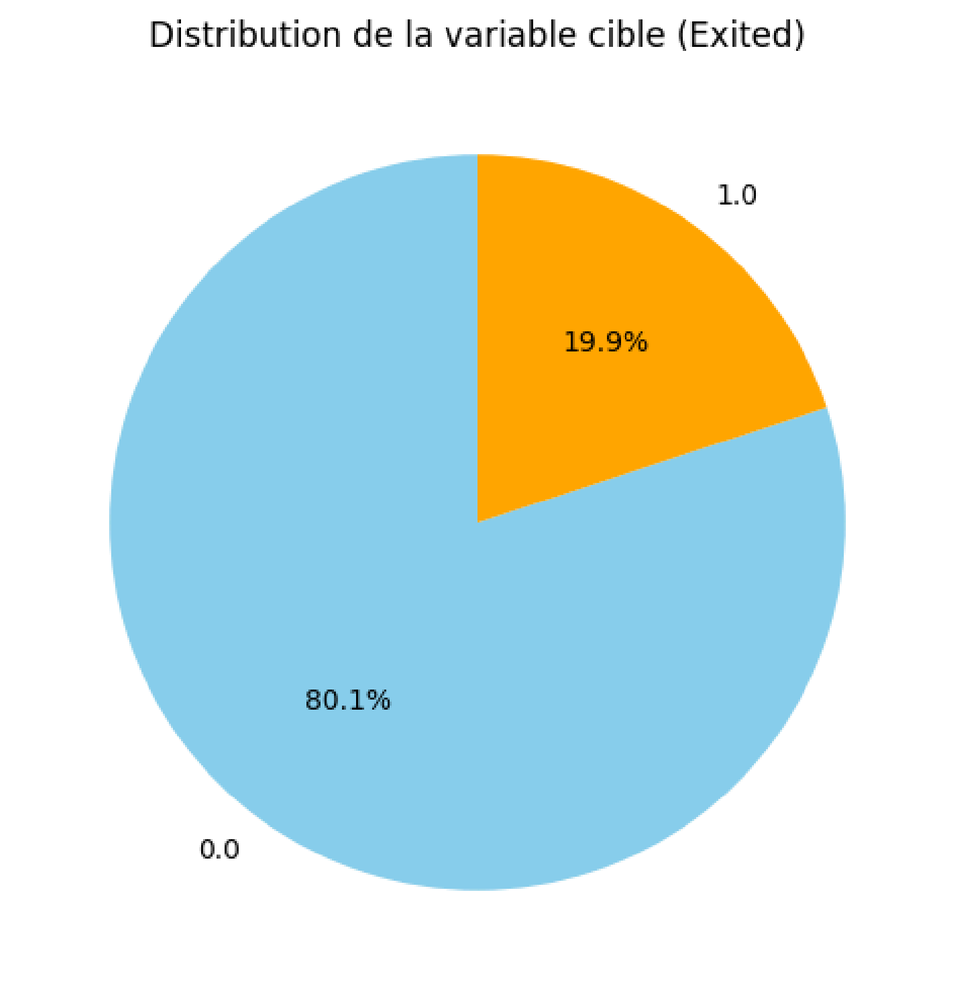
CHALLENGE DATA MINING

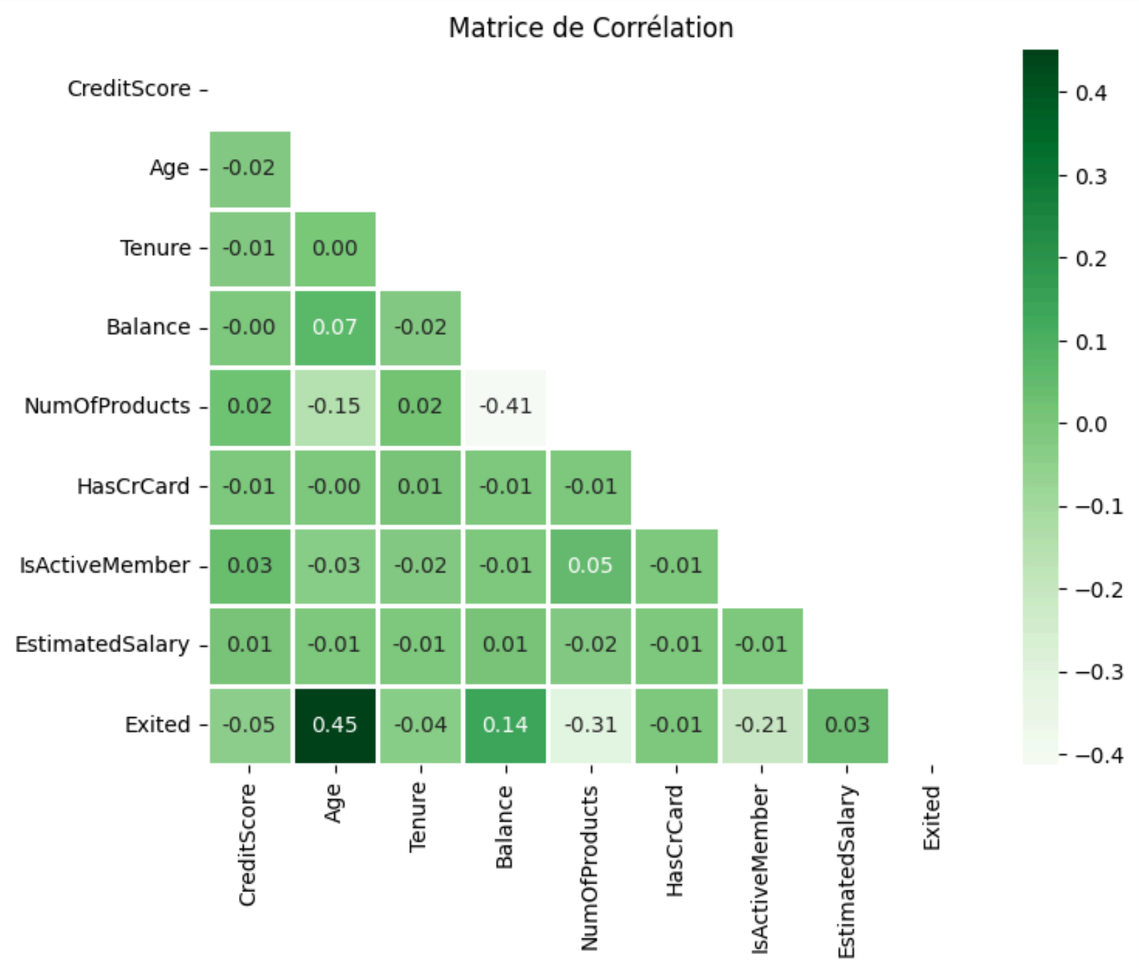
VO Nguyen Thao Nhi
LAMOUR Samanta

DISTRIBUTION DE “EXITED”



DÉSÉQUILIBRE DE CLASSE

CORRELATION

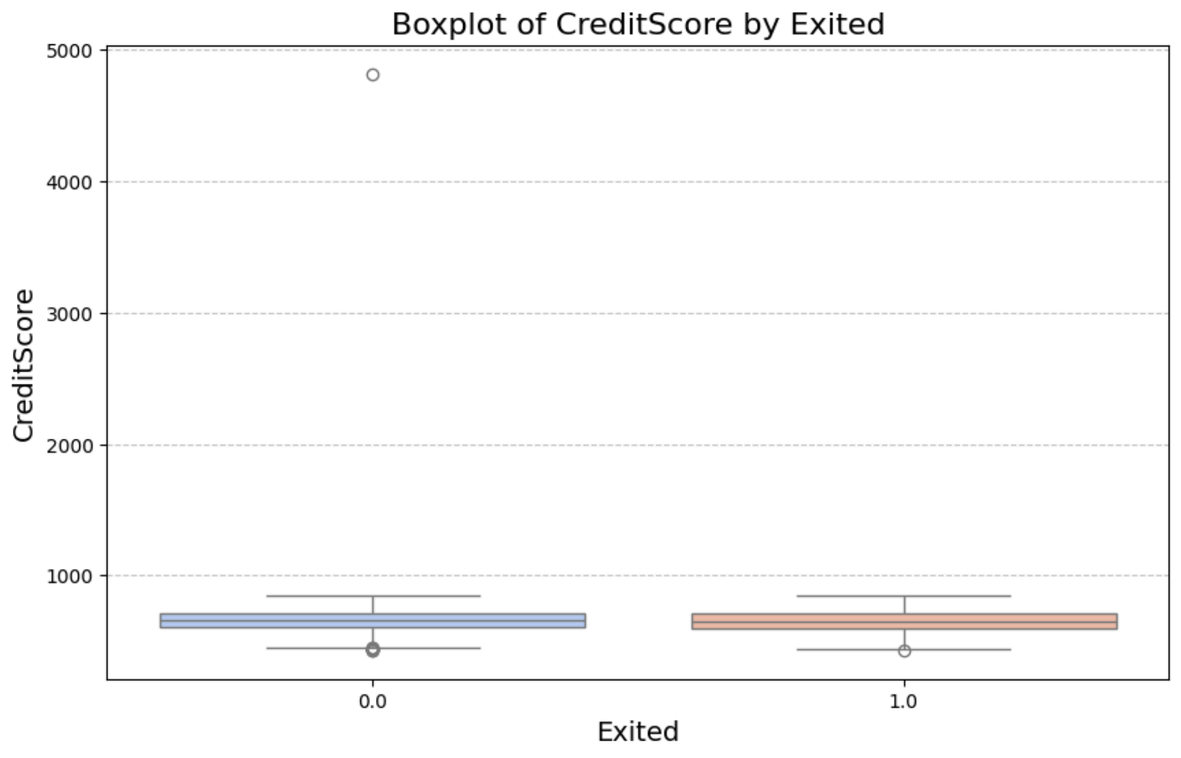


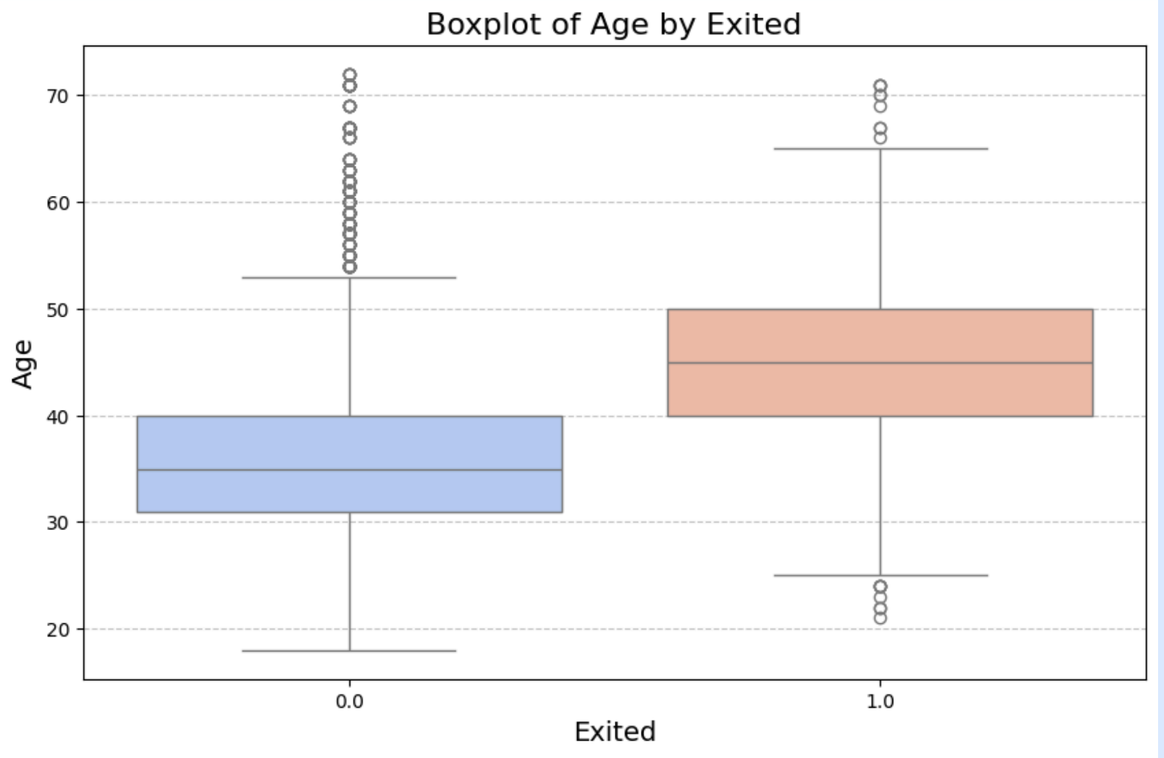
PAS DE FORTE CORRÉLATION

PRE PROCESSING

- Division des données pour l'entraînement (70%) et la validation (30%) en conservant les proportions des classes dans les deux sous-ensembles.
- Détection des outliers avec la méthode IQR: Age et CreditScore

OUTLIERS

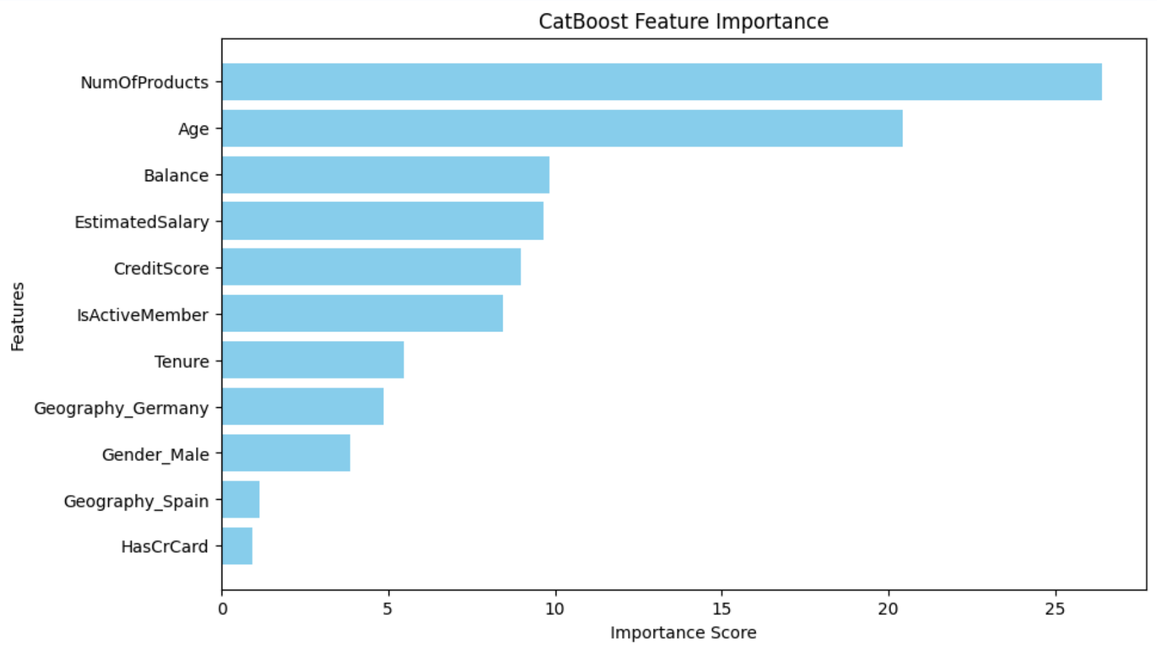

Boxplot of CreditScore by Exited
CreditScore
5000
4000
3000
2000
1000
0.0
1.0
Exited


Boxplot of Age by Exited
Age
70
60
50
40
30
20
0.0
1.0
Exited

PRE PROCESSING

- Division des données pour l'entraînement (70%) et la validation (30%) en conservant les proportions des classes dans les deux sous-ensembles.
- Détection des outliers avec la méthode IQR: Age et CreditScore
- Feature Engineering: one hot encoding pour les variables Geography et Gender
- Suppresion des variables non-pertinentes: Surname, CustomerId, id
- Feature Importance: Analyse de l'importance des variables avec CatBoost et SHAP

FEATURE IMPORTANCE



MODELES

MODÈLE 1

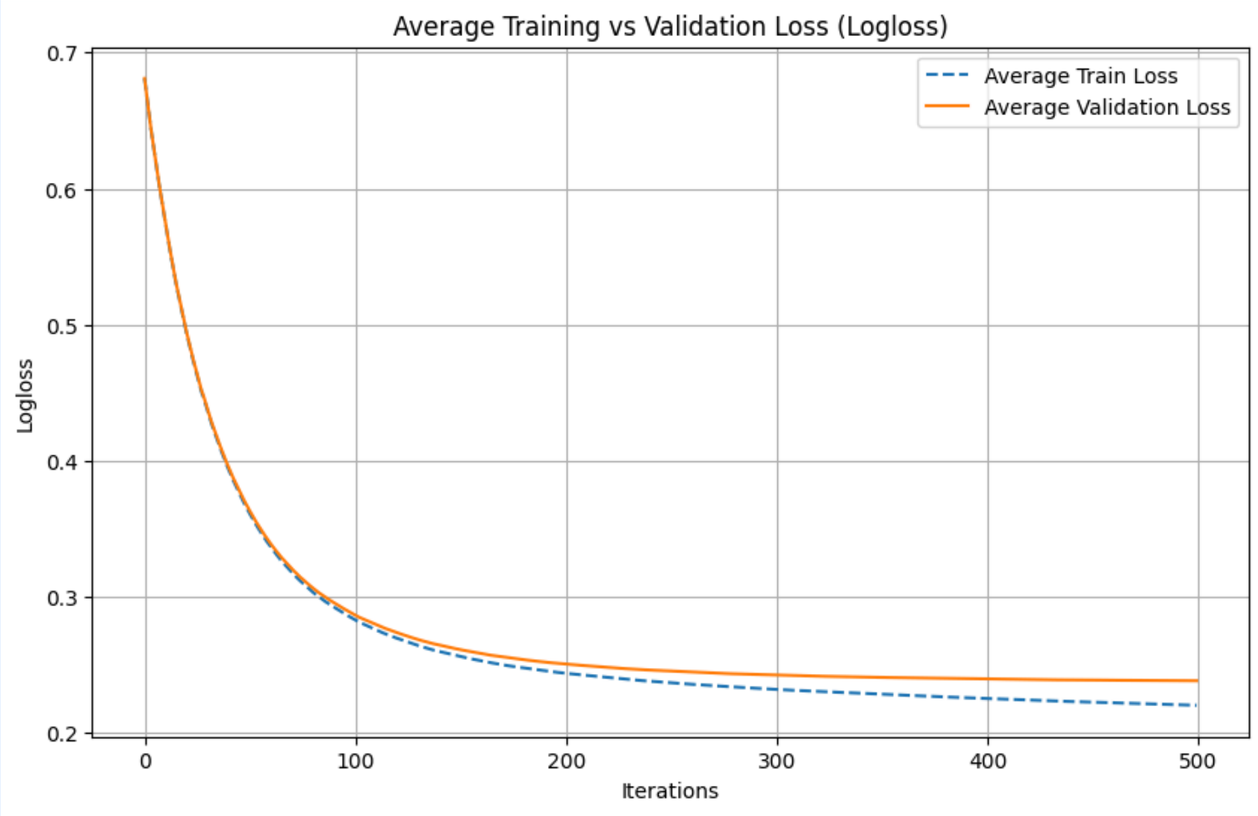
- Modèle: Catboost
- Fine tuner les hyperparamètres: GridSearchCV, StratifiedKFold (5 splits)
- Les hyperparamètres finaux:

```
params = {
    'depth': 6, 'iterations': 500, 'learning_rate': 0.01,
    'verbose': 0,
    "loss_function": "Logloss",
    'cat_features': categorical_features,
    'random_state':42
}
```

MODÈLE 1

EVALUATION

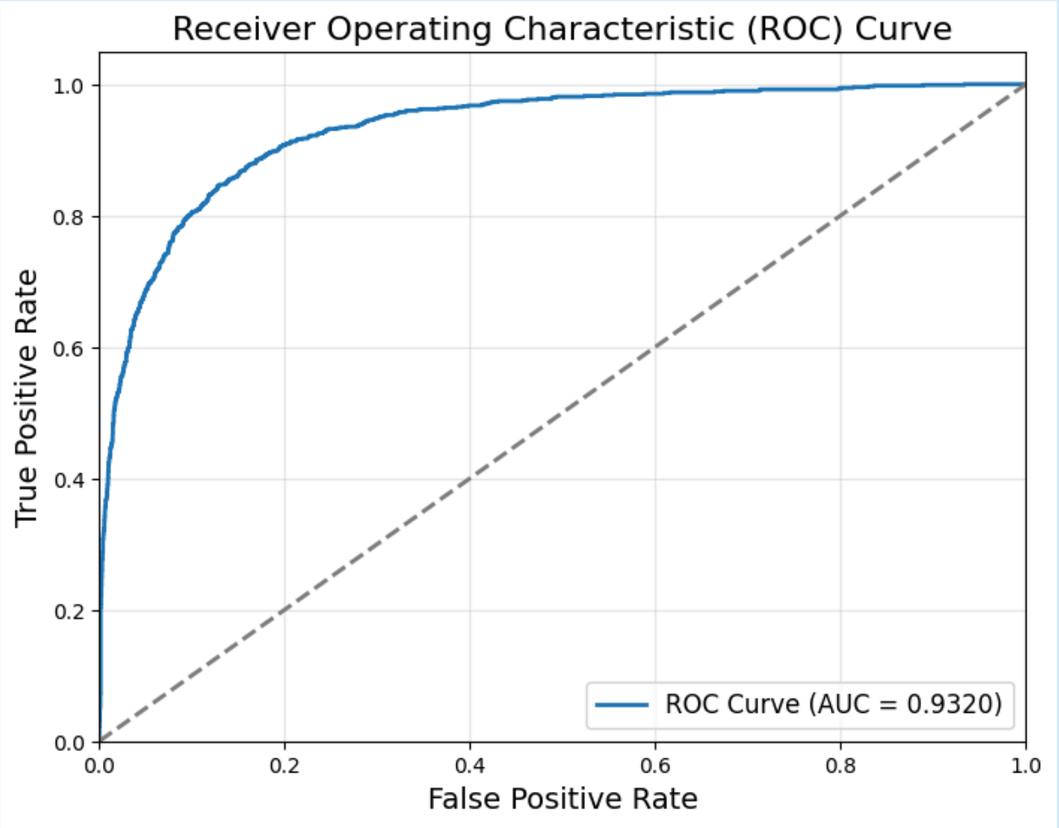
- Fold 1: Train AUC = 0.9446, Validation AUC = 0.9356
- Fold 2: Train AUC = 0.9438, Validation AUC = 0.9396
- Fold 3: Train AUC = 0.9429, Validation AUC = 0.9415
- Fold 4: Train AUC = 0.9439, Validation AUC = 0.9370
- Fold 5: Train AUC = 0.9425, Validation AUC = 0.9405
-
- Mean Train AUC: 0.9435
- Mean Validation AUC: 0.9388
- Validation AUC Standard Deviation: 0.0022



MODELE 1

EVALUATION

- AUC Score pour l'ensemble de validation: 0.9320
- AUC Score pour les données publiques sur Kaggle: 0.9326



MODÈLE 2

- Modèle: Catboost
- Fine tuner les hyperparamètres: OPTUNA
- Les hyperparamètres finaux:

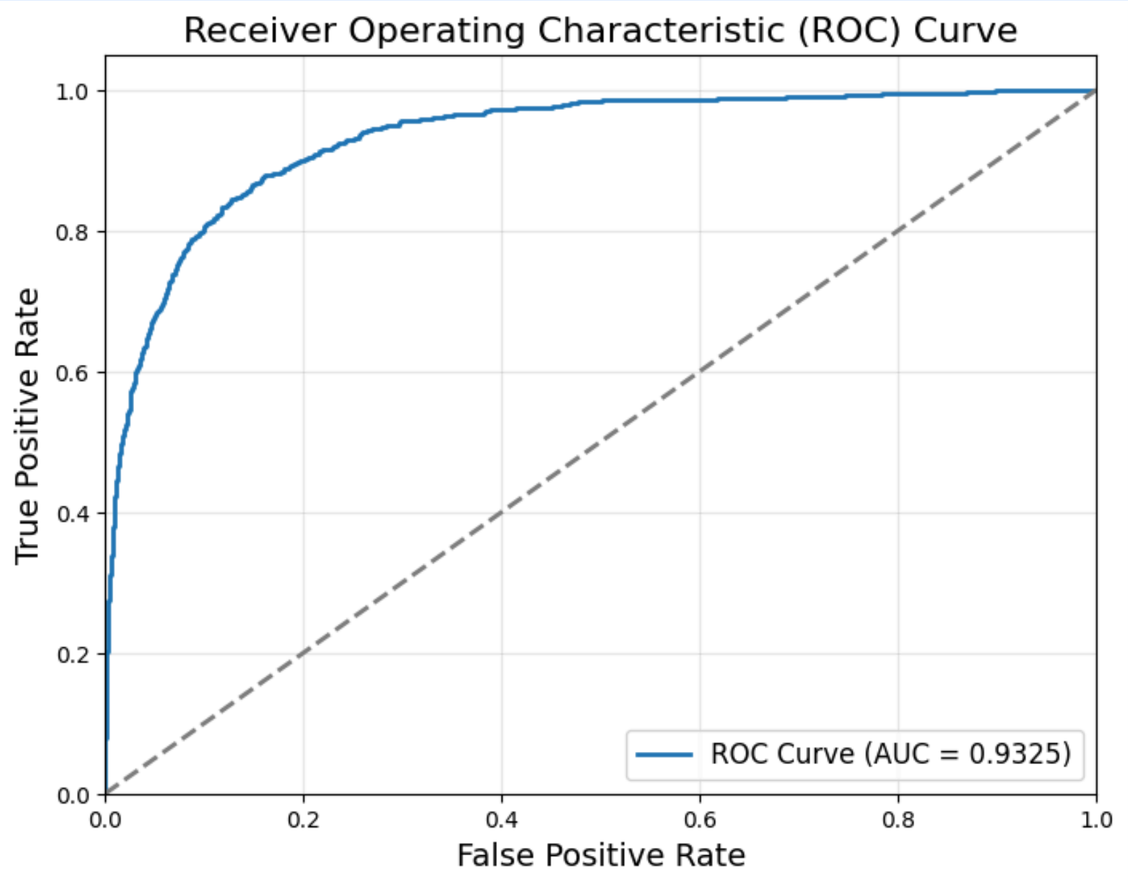
Best parameters: {'depth': 5, 'iterations': 538, 'learning_rate': 0.08318027730044576, 'l2_leaf_reg': 1.585980787407356}

- Cependant, il n’y a pas d’amélioration
- Il y a de l’overfitting (résultats assez différents sur le jeu de données de validation et de test)

MODÈLE 2

EVALUATION

- AUC Score pour l'ensemble de validation: 0.9325
- AUC Score pour les données publiques sur Kaggle: 0.9294



MODÈLE 3

- Modèle: HistGradientBoostingClassifier
- Fine tuner les hyperparamètres: OPTUNA

LE MODÈLE HISTGRADIENTBOOSTINGCLASSIFIER FAIT PARTIE DES ALGORITHMES DE BOOSTING. IL EST PARTICULIÈREMENT OPTIMISÉ POUR DES DONNÉES DE GRANDE DIMENSION GRÂCE À SON IMPLÉMENTATION HISTOGRAMME.

MODÈLE 3

Le modèle HistGradientBoostingClassifier présente des limites potentielles, notamment un risque d'overfitting dû à une mauvaise calibration des hyperparamètres ou à la présence de bruit et de redondances dans les données, ce qui peut entraîner des performances élevées en validation croisée mais une dégradation sur les données de test. Il est également sensible aux biais dans les données, notamment en cas de sur- ou sous-représentation de certaines classes ou d'une gestion inadéquate des variables catégoriques.

- AUC SCORE POUR L'ENSEMBLE DE VALIDATION: 0.9321
- AUC SCORE POUR LES DONNÉES PUBLIQUES SUR KAGGLE: 0.9288

AUTRES MODÈLES

Tableau Résumé : Autres Modèles Testés

Modèle	Description
XGBoost Simple	XGBoost avec paramètres par défaut ou légèrement optimisés.
XGBoost avec SMOTE	XGBoost appliqué sur un dataset rééquilibré via SMOTE.
LightGBM Simple	LightGBM avec configuration par défaut ou légèrement optimisée.
LightGBM avec SMOTE	LightGBM appliqué sur des données rééquilibrées.
Random Forest Simple	Modèle de forêt aléatoire classique avec hyperparamètres de base.

AUTRES MODÈLES

Random Forest avec SMOTE	Forêt aléatoire appliquée sur un dataset rééquilibré.
Stacking avec Régression Logistique	Combinaison des prédictions de tous les modèles testés, pondérées via une régression logistique.
Modèle de Deep Learning	Réseau de neurones profond optimisé pour la classification.

CHOIX FINAL


CatBoost Algorithm Features
Robust Nature of Algorithm
Algorithm Accuracy
Categorical Features Support
Easy Implementation
Faster Training & Predictions
Supporting Community Of Users
dataaspirant.com

FIN DU CHALLENGE