Machine Learning

Prédiction de fraude dans les transactions bancaire

RIAD EL HAJJAJI

Introduction

Le problème est un exemple de classification binaire supervisée, où l'objectif est de prédire si une transaction est frauduleuse ou non en utilisant les données d'entrée disponibles dans le dataset.

Les techniques de classification machine learning et deep learning suivantes peuvent être utilisées pour résoudre ce problème :

1- Régression logistique : cette technique est une méthode de classification binaire qui est efficace pour des ensembles de données relativement petits.

2- K plus proches voisins (K-NN) : cette technique est efficace pour les ensembles de données avec un grand nombre de variables d'entrée et un faible nombre de données d'entraînement.

3- SVM (Support Vector Machine) : cette technique est efficace pour les ensembles de données avec des variables d'entrée non linéaires et une grande dimensionnalité.

4- Réseaux de neurones artificiels (ANN) : cette technique est efficace pour les ensembles de données complexes avec un grand nombre de variables d'entrée et une grande quantité de données d'entraînement.

5- Réseaux de neurones récurrents (RNN) : cette technique est efficace pour les ensembles de données avec des séquences de données en entrée, comme les transactions bancaires.

Métriques de mesures de performance

- **La courbe ROC** (Receiver Operating Characteristic) : Est une courbe qui représente la performance d'un modèle de classification binaire en fonction de son seuil de classification. Elle trace le taux de vrais positifs (sensibilité) en fonction du taux de faux positifs (1 - spécificité) pour différents seuils de classification.

La courbe ROC est souvent utilisée pour évaluer la performance d'un modèle de classification binaire en comparant plusieurs modèles ou en sélectionnant le seuil de classification optimal pour un modèle donné. Plus la courbe ROC se rapproche du coin supérieur gauche du graphique, plus le modèle est considéré comme performant.

La surface sous la courbe ROC (Aire sous la courbe ROC ou AUC en anglais) est un autre métrique de performance couramment utilisée. L'AUC mesure la capacité du modèle à distinguer les classes positives des classes négatives. Une AUC de 1.0 indique que le modèle est parfaitement capable de distinguer les classes, tandis qu'une AUC de 0.5 indique que le modèle est incapable de faire mieux que le hasard.

- **Matrice de confusion** : Tableau de contingence. Elle mettra non seulement en valeur les prédictions correctes et incorrectes mais nous donnera surtout un indice sur le type d’erreurs commises

Métriques de mesures de performance

- Précision : mesure la proportion de vrais positifs par rapport à l'ensemble des prédictions positives.
- Rappel (ou sensibilité) : mesure la proportion de vrais positifs par rapport à l'ensemble des valeurs réelles positives.
- Spécificité : mesure la proportion de vrais négatifs par rapport à l'ensemble des valeurs réelles négatives.
- F1-score : une mesure de la précision et du rappel qui est utile lorsque les classes sont déséquilibrées.
- AUC-ROC : mesure la performance globale d'un modèle de classification binaire en traçant la courbe ROC et en calculant l'aire sous la courbe.

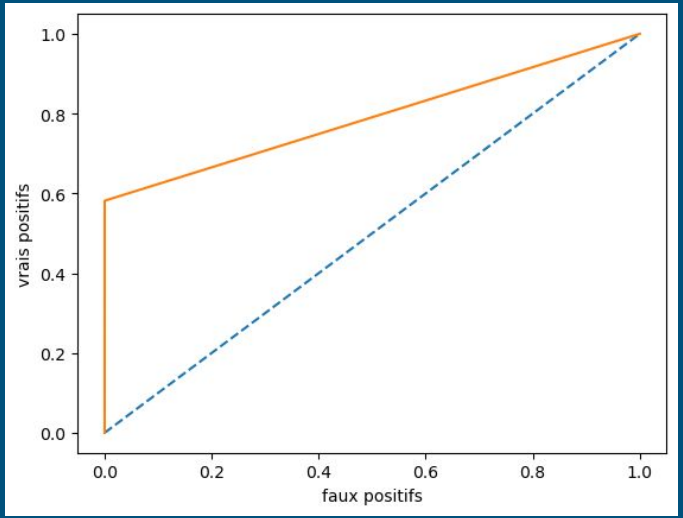
Dataset

Pour tous les algorithmes ,on va diviser le dataset en 0.2 de test et 0.8 pour l'apprentissage

```
# Diviser les données en ensembles d'entraînement et de test
X_train, X_test, y_train, y_test = train_test_split(data.drop('Class', axis=1), data['Class'], test_size=0.2, random_state=42)

# Scale the data
```

1- Régression logistique




```
Precision :  0.9991222218320986
Rappel :  0.5816326530612245
Specificite :  0.9998417276308385
F1 :  0.6951219512195121
Auc_roc :  0.7907371903460314
```

```
array([[56855,     9],
       [   41,    57]])
```

2- K plus proches voisins (K-NN) :

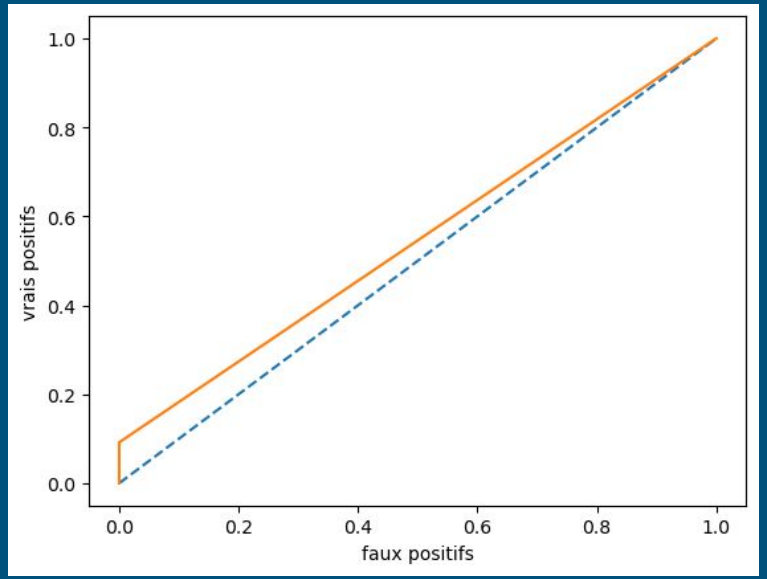
- Le choix de la valeur de k dépend de la taille de données et de la complexité du problème. on commence par des valeurs de k plus petites, et on augmente progressivement la valeur de k jusqu'à ce qu'on trouve la meilleure valeur pour votre ensemble de données.
Dans notre cas, avec plus de 284000 transactions bancaires, on peur commencer par des valeurs de k plus grandes, telles que k = 25 et ajuster en fonction de la précision et des performances de votre modèle.
- On utilisé la méthode recherche de grille (GridSearchCV) pour déterminer la meilleure valeur de k .

GridSearchCV indique que 3 est la meilleure valeur de k (J'ai testé juste des valeur sur l'intervalle [0,9] parceque mon ordinateur se plante avec des valeurs grandes

```
Best k value: 3
Accuracy: 0.9984375548611355
```

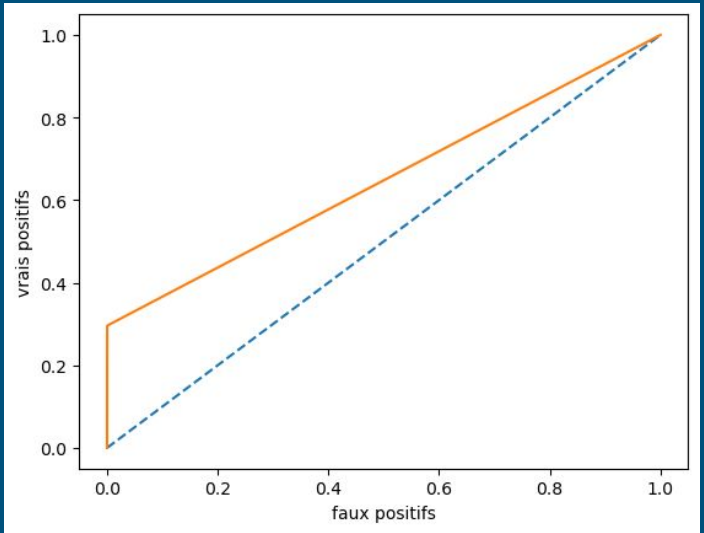
2- K plus proches voisins (K-NN) :

Avec k=3 l’algorithme knn donne




```
Precision :  0.9984375548611355
Rappel :  0.09183673469387756
Specificite :  1.0
F1 :  0.16822429906542058
Auc_roc :  0.5459183673469388
```

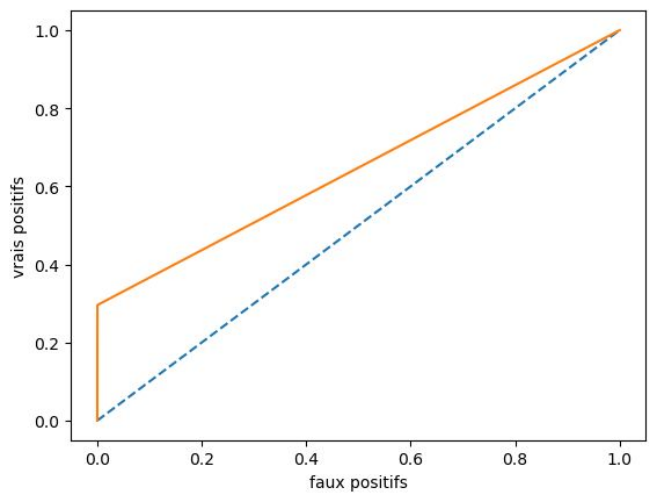
3- SVM (Support Vector Machine)




```
Precision :  0.9984551104244935
Rappel :  0.29591836734693877
Specificite :  0.9996658694428813
F1 :  0.39726027397260274
Auc_roc :  0.64779211839491
```

4- Réseaux de neurones artificiels (ANN)

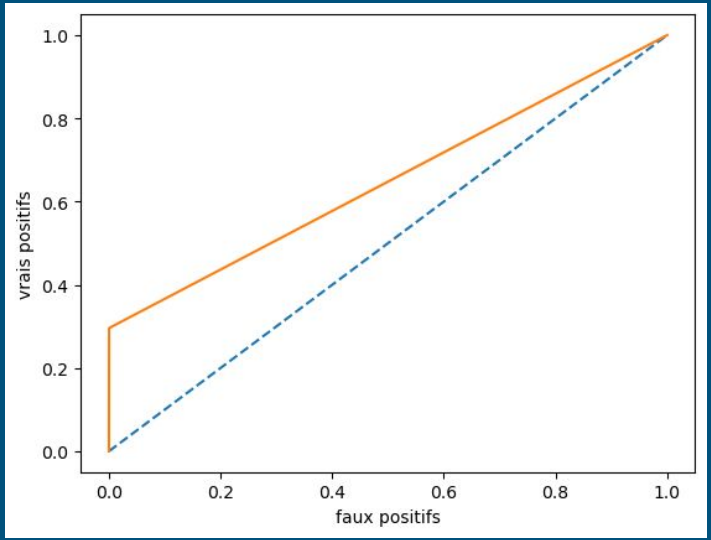
Cet algorithme utilise TensorFlow pour construire un réseau de neurones à une couche cachée avec 64 neurones, une fonction d'activation ReLU et une couche de sortie de 2 neurones avec une fonction d'activation softmax pour la classification. La fonction de perte utilisée est la "sparse categorical crossentropy" qui est appropriée pour la classification avec deux classes. L'optimiseur utilisé est l'optimiseur Adam. Le modèle est entraîné sur l'ensemble d'entraînement avec 10 époques et une taille de lot de 32. Enfin, les performances du modèle sont évaluées sur l'ensemble de test et le score d'exactitude est affiché.




```
Precision :  0.9984551104244935
Rappel :  0.29591836734693877
Specificite :  0.9996658694428813
F1 :  0.3972602739726027
Auc_roc :  0.64779211839491
```

5- Réseaux de neurones récurrents (RNN)

L'utilisation d'un réseau de neurones récurrents (RNN) peut être appropriée pour certains types de données séquentielles, tels que les séquences temporelles. Dans le cas des transactions bancaires, l'ordre temporel des transactions peut être important pour détecter la fraude.




```
Precision :  0.9984551104244935
Rappel :  0.29591836734693877
Specificite :  0.9996658694428813
F1 :  0.3972602739726027
Auc_roc :  0.64779211839491
```

5- Réseaux de neurones récurrents (RNN)

- On remarque que la régression logistique et SVM donne le meilleure score de précision que les réseaux de neurones et le k-plus proche voisin
- SVM prend plus de ressource et de temps donc on peut dire que la régression linéaire est la plus pertinent pour ce problème de classification

```
regression logistique accuracy :  0.99912222218320986
k-plus proche voisin accuracy :  0.9984375548611355
svm accuracy :  0.9991222218320986
reseaux de neurones accuracy :  0.9984551104244935
reseaux de neurones recurrents accuracy :  0.9984551104244935
```