M2ISE M1Info

Université Paris 8 - Vincennes à Saint-Denis

Master 2 Informatique des Systèmes Embarqués

Corpus, fouille de données et apprentissage

Detection de Fraude

Hammou SABBAR
Hacene YOUNSI

Lisa MEDROUK & Anna PAPPA – Université Paris VIII

Detection de Fraude avec Keras

La fraude par carte de crédit a entraîné une perte de 3 milliards de dollars pour les institutions financières nord-américaines en 2017. La montée en puissance des systèmes de paiement numériques tels qu'Apple Pay, Android Pay et Venmo a entraîné une augmentation des pertes dues à des activités frauduleuses. Le Deep Learning présente une solution prometteuse au problème de la détection de la fraude par carte de crédit en permettant aux institutions d'utiliser de manière optimale leurs données clients historiques ainsi que les détails de transaction en temps réel qui sont enregistrés au moment de la transaction. En 2017, une étude a révélé qu'une approche d'apprentissage en profondeur a fourni des résultats comparables aux méthodes de détection de fraude existantes telles que les arbres à gradient amélioré et la régression logistique. Cependant, le Deep Learning englobe un certain nombre de topologies. De plus, les différents paramètres utilisés pour construire le modèle (par ex. le nombre de neurones dans la couche cachée d'un réseau neuronal) influencent également ses résultats.

nous évaluons deux model d'apprentissage profond du réseau neuronal artificiel un Sequential et un encodeur automatique ; Nous utilisons un environnement de Google Colab Open In Colab pour la haute performances de calcul en GPU, le problème de détection de fraude courant les donnés sont déséquilibré des classes.

0.1 Dataset

L'ensemble des données obtenu sur Kaggle, contiennent des transactions effectuées par carte de crédit en septembre 2013 par des titulaires de carte européens. Cet ensemble de données présente les transactions qui ont eu lieu en deux jours, où nous avons 492 fraudes sur 284 807 transactions. L'ensemble de données est très déséquilibré, la classe positive (fraudes) représente 0,172% de toutes les transactions. Il ne contient que des variables d'entrée nu-

mériques qui sont le résultat d'une transformation PCA. Malheureusement, en raison de problèmes de confidentialité, nous ne pouvons pas fournir les fonctionnalités d'origine et plus d'informations sur les données. Les fonctionnalités V1, V2,...V28 sont les principaux composants obtenus avec PCA, les seules fonctionnalités qui n'ont pas été transformées avec PCA sont 'Time' et 'Amount'.

0.1.1 Normaliser les données

En utilisant StandardScaler, nous normalisons l'ensemble de données afin de n'avoir aucun biais. En appliquant StandardScaler et RobustScaler, nous transformerons simplement les données de sorte que leur distribution aura une valeur moyenne entre 0 et un écart type de 1.

0.1.2 Diviser les données en training, validation et test set

Pour l'ensemble de données de formation, nous excluons les transactions frauduleuses. Comme notre objectif est de détecter une transaction frauduleuse, nous formerons spécifiquement des données sur la transaction normale afin que lorsqu'une transaction frauduleuse se produise, notre encodeur automatique puisse facilement la prédire avec une erreur de reconstruction élevée.

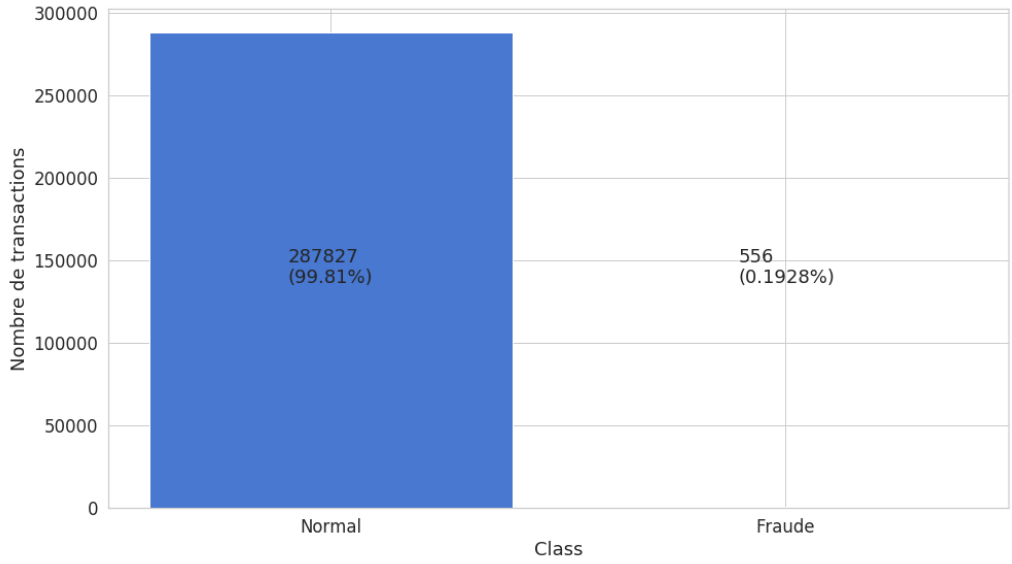

FIGURE 1 – Visualisation globale de jeu de données

Deep Learning

0.2 Algorithmes - Models

0.2.1 Autoencoders

Pour faire simple, il produit ou prédit la sortie avec la même entrée. Idéalement, le travail d'un encodeur automatique consiste à produire une sortie qui est presque similaire (sinon exacte) à l'entrée donnée. En d'autres termes, «Autoencoding» est un algorithme de compression de données où la compression et la décompression des données ont lieu. L'encodeur automatique peut être divisé en parties ci-dessous

- Encodeur : cette partie du réseau comprime ou sous-échantillonne l'entrée en moins de bits. L'espace représenté par ce nombre réduit de bits est souvent appelé espace latent ou goulot d'étranglement . Le goulot d'étranglement est également appelé «point de compression maximal» car à ce stade, l'entrée est compressée au maximum. Ces bits compressés qui représentent l'entrée d'origine sont appelés ensemble un «codage» de l'entrée.
- Décodeur : cette partie du réseau essaie de reconstruire l'entrée en utilisant uniquement le codage de l'entrée. Lorsque le décodeur est capable de reconstruire l'entrée exactement comme elle a été envoyée à l'encodeur, vous pouvez dire que l'encodeur est capable de produire les meilleurs encodages pour l'entrée avec laquelle le décodeur est capable de bien reconstruire

0.2.2 Model Sequential

Les modèles dans Keras sont définis comme une séquence de couches. Nous définissons d'abord que la couche d'entrée au bon nombre d'entrées, ce qui se fait en définissant, dans input_dim notre cas, elle est de 30 colonnes. Nous pouvons spécifier le nombre de neurones dans la couche comme premier argument, la méthode d'initialisation comme deuxième argument comme int

et spécifier la fonction d'activation en utilisant l'argument d'activation. Nous utiliserons la fonction d'activation du redresseur ('relu'). Nous avons utilisé la fonction Dropout(rate) ce qui permet d'éviter le surapprentissage.

0.2.3 Algos utilisé

Voici la comparaison des algorithmes utilisé :

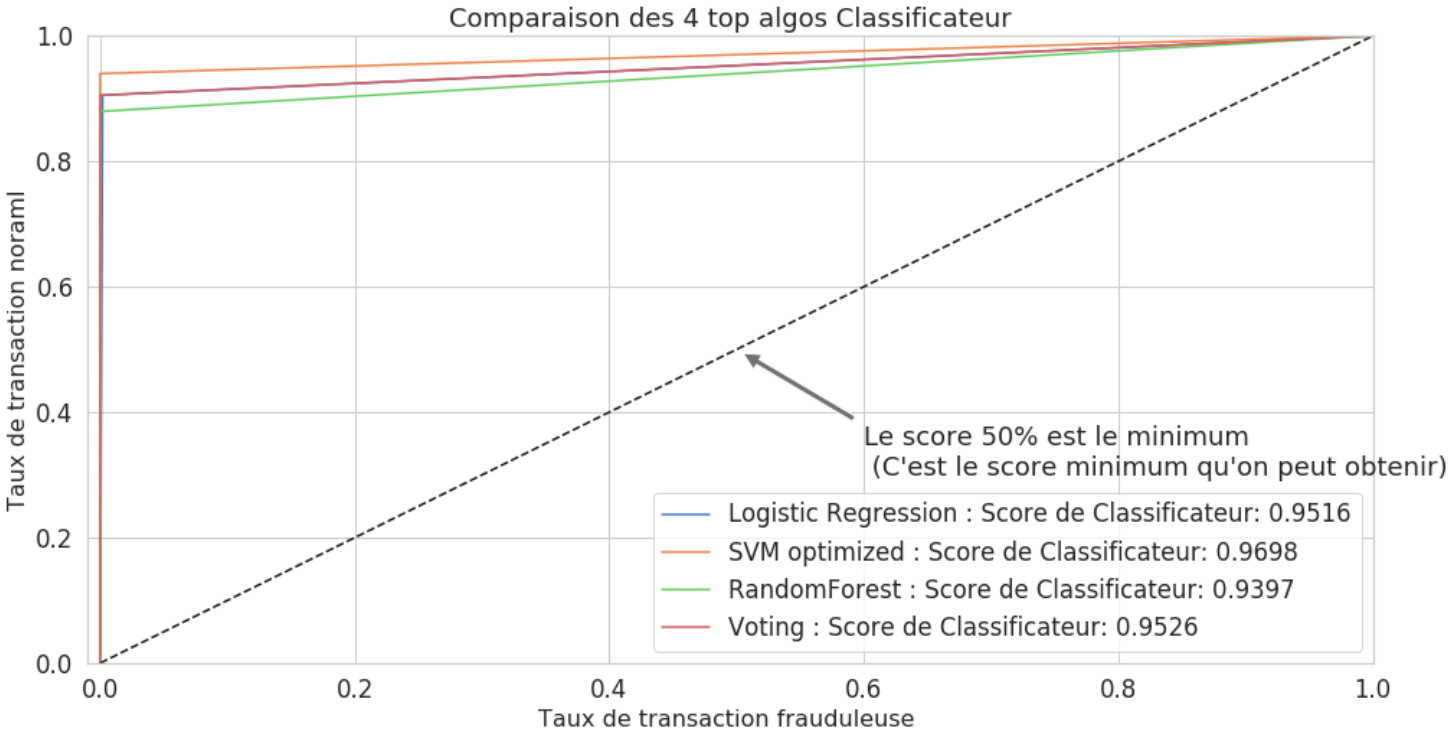

FIGURE 2 – Comparaison des algorithmes utilisé

Conclusion et Perspectives

Nous avons créé plusieurs modèles d'apprentissage avec Keras qui peut reconnaitre les transactions frauduleuses, cela nous semble très important de trouver des anomalies, cependant. nous ne connaissons pas les fonctionnalités d'origine des informations sur les données en raison de confidentialité, cela nous pose un problème pour aller loin dans l'analyse de ses variables qui peut-être très important (lieu, ip ... ect) Quant à l'erreurs est logique, compte tenu de l'ensemble des données déséquilibrées (287 827 transactions non frauduleuses contre 556 transactions frauduleuses) et de la façon dont nous avons élargi l'ensemble de données. Afin d'améliorer notre précision, nous devons échantillonner plus de données classées comme frauduleuses et utiliser moins de données classées comme non frauduleuses. De plus, en ajustant avec précision les paramètres et la structure du réseau neurones, la précision pourrait augmenter.

Bibliographie

[Cho16] Francois Chollet. *Building Autoencoders in Keras*. 14 May 2016. https://blog.keras.io/building-autoencoders-in-keras.html.

[dnnm] XCeption deep neural network model. *keras*. https://keras.io/.

[Ell] David Ellison. Fraud detection using autoencoders in keras with a tensorflow backend. https://blogs.oracle.com/datascience/fraud-detection-using-autoencoders-in-keras-with-a-tensorflow-backend.

[Ell19] David Ellison. Credit card fraud detection by neural network in keras. 16 January 2019. https://blogs.oracle.com/datascience/fraud-detection-using-autoencoders-in-keras-with-a-tensorflow-backend.

[kam] kamathhrishi. *Detecting-Fraudulent-Credit-Card-Transactions*. PhD thesis. https://github.com/kamathhrishi/Detecting-Fraudulent-Credit-Card-Transactions/blob/master/notebook.ipynb.

[Mac19] Randy Macaraeg. Credit card fraud detection. Sep 5, 2019. https://towardsdatascience.com/credit-card-fraud-detection-a1c7e1b75f59.