💵 Rapport d'Analyse et de Modélisation pour la Détection de Fraudes Financières

**Introduction :**

Dans le cadre de la prévention des fraudes financières, l'utilisation de modèles d'apprentissage automatique peut jouer un rôle crucial. Ce rapport présente une approche complète utilisant PySpark pour traiter, explorer, et modéliser un ensemble de données de transactions financières en vue de détecter les fraudes.

Les transactions numériques font désormais partie de la vie quotidienne, comme l'achat d'un produit en ligne, l'envoi d'argent à des amis, le dépôt d'argent sur un compte bancaire, à des fins d'investissement, etc. Elles présentent de nombreux avantages et ouvrent la voie à des activités frauduleuses. Les gens ont commencé à utiliser des moyens de transactions monétaires numériques pour blanchir de l’argent et donner l’impression que l’argent provient d’une source légale.

**Objectif :**

L'objectif de ce projet est de trouver les modèles de transactions effectuées et d'aider les algorithmes à apprendre ces modèles pour identifier les transactions frauduleuses et les signaler.

Analyse exploratoire des données pour extraire le modèle d'activités frauduleuses

Créer un modèle d'apprentissage automatique pour classer les transactions frauduleuses et non frauduleuses

Réduisez les faux négatifs en ajustant le modèle.

**Ensemble de données :**

Ici, nous importons simplement les 50 000 premières lignes car l'ensemble de données est énorme et prendrait beaucoup de temps à traiter. Voilà le lien de base de données :  
 <https://www.kaggle.com/code/ananthu19/online-payments-fraud-detection/input>

Voyons l'aperçu des données :

Une image contenant texte, capture d’écran, Police, document

Description générée automatiquement

**Description de l'ensemble de données**

* Step : cartographie une unité de temps dans le monde réel. Dans ce cas, 1 pas équivaut à 1 heure. Total d'étapes 744 (simulation de 30 jours).
* Type : CASH-IN, CASH-OUT, DÉBIT, PAIEMENT et TRANSFERT.
* Montant - montant de la transaction en devise locale.
* nameOrig - client qui a commencé la transaction
* oldbalanceOrg - solde initial avant la transaction
* newbalanceOrig - nouveau solde après la transaction
* nameDest - client qui est le destinataire de la transaction
* oldbalanceDest - destinataire du solde initial avant la transaction. Notez qu'il n'y a pas d'informations pour les clients commençant par M (marchands).
* newbalanceDest - destinataire du nouveau solde après la transaction. Notez qu'il n'y a pas d'informations pour les clients commençant par M (marchands).
* isFraud - Il s'agit des transactions effectuées par les agents frauduleux dans la simulation. Dans cet ensemble de données spécifique, le comportement frauduleux des agents vise à tirer profit en prenant le contrôle des comptes des clients et en essayant de vider les fonds en les transférant vers un autre compte, puis en les encaissant du système.
* isFlaggedFraud - Le modèle économique vise à contrôler les transferts massifs d'un compte à un autre et signale les tentatives illégales. Une tentative illégale dans cet ensemble de données est une tentative de transférer plus de 200 000 en une seule transaction.

Inférence 1 :

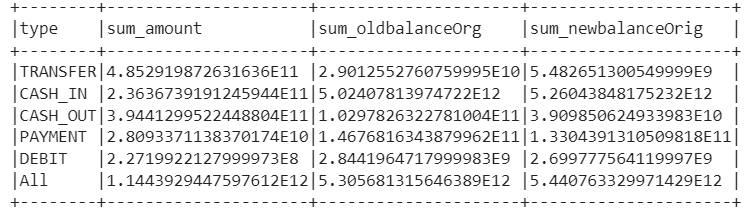
Conformément à l'algorithme actuel basé sur des règles, il n'y a eu aucun signalement lors de transactions frauduleuses en cas de retrait d'argent, ce qui constitue une préoccupation majeure pour le système de lutte contre le blanchiment d'argent. De plus, seules 16 transactions sont signalées comme frauduleuses, alors qu'environ 4 000 transactions sont en réalité des fraudes. Notre mission est maintenant de construire un algorithme efficace pour atténuer ce risque de laisser les transactions frauduleuses se débloquer.

Une image contenant texte, reçu, Police, blanc

Description générée automatiquement

Inférence 2 :

D'après le tableau, nous pouvons comprendre que la plupart des clients utilisent le système pour transférer de l'argent et que nous disposons de relativement moins de données sur les paiements effectués. Il est également très intéressant de remarquer la différence entre le nouvel et l’ancien équilibre, car cela nous raconte des histoires. Ici nous n'avons que les visuels du compte d’origine et le solde de trésorerie a été réduit dans tous les cas sauf cash\_in. Même lors du transfert, le solde a diminué, ce qui montre que nous avons plus d'informations sur l'expéditeur dans le compte d'origine.



Inférence 3 :

Dans ce tableau, nous avons les informations du compte de destination, à partir des informations de transfert, nous pouvons voir l'augmentation du nouveau solde, ce sont donc les informations du destinataire. Aucun montant de paiement n'est disponible pour les informations Dest.

Une image contenant texte, Police, capture d’écran, blanc

Description générée automatiquement

Répartition du Montant

Il est important de comprendre la distribution de nos données, car elle peut jouer un rôle majeur dans la construction de modèles et également dans la compréhension de nos données. À l'avenir, nous n'utiliserons que 50 000 lignes car le traitement de tous les enregistrements pour la création de visualisations et de modèles prend beaucoup de temps. Ici, nous vérifions la répartition du montant transactionné à l'aide de l'application

Une image contenant texte, capture d’écran, ligne, Tracé

Description générée automatiquement

À partir du graphique à barres, nous pouvons comprendre que nous avons un ensemble de données très asymétrique, il y a beaucoup de valeurs aberrantes qui vont jusqu'à 10 M avec une médiane de 33 000. La tranche supérieure (75e percentile) compte jusqu'à 450 000

**Feature engineering :**

Il est temps de se salir les mains avec l’ingénierie des fonctionnalités. Avec les informations disponibles, il est difficile de former le modèle et d'obtenir de meilleurs résultats. Nous passons donc à la création de nouvelles caractéristiques en modifiant les caractéristiques existantes. En cela, nous créons trois fonctions qui créent une fonctionnalité très pertinente pour le domaine

Différence de solde : C'est une vérité universelle que le montant débité du compte de l'expéditeur est crédité sur le compte du destinataire sans aucun écart en centimes. Mais que se passe-t-il s'il y a un écart dans le montant débité et crédité. Certains pourraient être dus aux frais perçus par les prestataires de services, mais nous devons signaler ces cas inhabituels.

Indicateur de surtension : nous devons également déclencher un indicateur lorsqu'un montant important est impliqué dans la transaction. De la répartition des montants, nous avons compris que nous avons beaucoup de valeurs aberrantes avec des montants élevés dans les transactions. Par conséquent, nous considérons le 75e centile (450 000) comme notre seuil et le montant supérieur à 450 000 sera déclenché comme indicateur.

Indicateur de fréquence : nous signalons ici l'utilisateur et non la transaction. Lorsqu’un destinataire reçoit de l’argent de nombreuses personnes, cela peut être un déclencheur, comme cela peut être le cas pour certains jeux de hasard ou de chance illégaux. Par conséquent, il est signalé lorsqu’un destinataire reçoit de l’argent plus de 20 fois.

Indicateur marchand : les identifiants des clients dans le destinataire commencent par « M », ce qui signifie qu'il s'agit de commerçants et qu'ils auront évidemment beaucoup de transactions de réception. Nous signalons donc également chaque fois qu'il y a un récepteur marchand

​  
 **Prétraitement des données :**

Le processus débute par l'initialisation d'une session Spark et la lecture d'un fichier CSV compressé contenant les données de transactions financières. Après extraction, l'ensemble de données est chargé dans un DataFrame PySpark nommé paysim.

Une exploration initiale des données révèle qu'il y a des colonnes contenant des valeurs nulles. Ces valeurs sont analysées et traitées pour garantir la qualité des données. De plus, une visualisation des types de transactions est effectuée pour mieux comprendre la distribution des données.

Dans le diagramme circulaire ci-dessous, nous pouvons clairement voir que l'étiquette cible est fortement déséquilibrée, car nous ne disposons que de 0,2 % de données frauduleuses, ce qui est insuffisant pour que la machine puisse apprendre et signaler les transactions frauduleuses.

Une image contenant capture d’écran, cercle, diagramme, Bleu électrique

Description générée automatiquement

**Analyse Exploratoire des Données**

La première partie de l'analyse exploratoire des données consiste en la création de tables pivotantes pour évaluer les sommes et les écarts-types des montants impliqués dans les transactions, en fonction du type de transaction. Cette étape aide à identifier des tendances potentielles et à repérer des schémas dans les données.

La deuxième partie de l'analyse explore la distribution des montants et utilise des techniques de fenêtrage pour calculer des indicateurs tels que les variations de solde et les indicateurs de montants exceptionnellement élevés.

**Préparation des Données pour l'Apprentissage Automatique**

La préparation des données est cruciale pour appliquer des modèles d'apprentissage automatique. Deux approches distinctes sont présentées dans ce rapport.

**Première Méthode**

La première méthode se concentre sur la sélection des caractéristiques pertinentes pour l'entraînement du modèle. Les colonnes inutiles sont éliminées, et les données sont divisées aléatoirement en ensembles d'entraînement et de test. Une conversion des variables catégorielles en valeurs numériques est effectuée, et un pipeline est créé pour orchestrer ces transformations.

**Deuxième Méthode**

La deuxième méthode se centre sur l'utilisation du modèle Naive Bayes pour la détection de fraudes. Elle commence par la création d'une colonne "label" pour représenter les étiquettes à prédire. Les caractéristiques pertinentes sont sélectionnées, et les données sont divisées en ensembles d'entraînement et de test. Le modèle Naive Bayes est entraîné sur l'ensemble d'entraînement, et ses performances sont évaluées sur l'ensemble de test.

**Modélisation avec Naive Bayes**

La modélisation est effectuée en utilisant l'algorithme Naive Bayes, adapté à un contexte multinomial. Le modèle est entraîné sur les données d'entraînement et évalué sur les données de test à l'aide de la métrique de l'aire sous la courbe ROC (Receiver Operating Characteristic). Une matrice de confusion est également générée pour visualiser les performances du modèle en termes de prédictions correctes et incorrectes.

**Conclusion**

Ce rapport offre une approche complète de la détection de fraudes financières en utilisant PySpark et des techniques d'apprentissage automatique. Les deux méthodes présentées permettent de préparer et d'analyser les données en vue de la modélisation. L'utilisation du modèle Naive Bayes démontre son efficacité dans la détection de fraudes, comme en témoignent les résultats de l'évaluation du modèle. L'ensemble du processus offre une base solide pour le déploiement et l'optimisation continus de modèles de détection de fraudes dans des environnements financiers.