



UNIVERSITE IBN ZOHR
FACULTE DES SCIENCES

Département Informatique
Filière Sciences Mathématiques et Informatique

PFE

Présenté par :

Noura Bentaher & Fatiha Ait Aadi

Pour l'obtention de la

Licence en Sciences Mathématiques et Informatique

**Détection automatique des fraudes par
cartes bancaires sur la base du
comportement des utilisateurs en utilisant les
techniques d'apprentissage automatique**

Encadrées par : Mr. KABBADJ Younes

Soutenu le : --/--/20

Année universitaire 2019-2020

Page laissée intentionnellement vide

Dédicaces

Je dédie ce mémoire à mes parents, qui m'ont encouragé à aller de l'avant et qui m'ont donné tout leur amour et leur confiance.

A la mémoire de ma grande sœur LAYLA qui nous a quittés depuis un an, à qui j'ai prié toujours pour le salut de son âme. Puisse Dieu, le tout puissant, l'avoir en sa sainte miséricorde !

A mon cher frère MAHJOUB qui n'a pas cessé de me conseiller, encourager et me soutenir tout au long de mes études. Que Dieu le protège et lui offre la chance et le bonheur.

A mon adorable petite sœur DOUAË qui sait toujours comment procurer la joie et le bonheur pour toute la famille.

A mon grand-père, mes oncles et mes tantes. Que Dieu leur donne une longue et joyeuse vie.

A tous les cousins, les voisins et les amis que j'ai connu jusqu'à maintenant. Merci pour leurs amours et leurs encouragements.

A tous les professeurs que ce soit du primaire, du moyen, du secondaire ou de l'enseignement supérieur.

Sans oublier mon binôme Fatiha pour son soutien moral, sa patience et sa compréhension tout au long de ce projet.

Noura

Je dédie ce modeste travail à mes parents. Aucun hommage ne pourrait être à la hauteur de l'amour dont ils ne cessent de me combler.

A mes très chers frères source de vie, l'amour et de bonheur, d'espoir et de motivation

A toutes mes amies tout particulièrement Noura mon amie et mon collègue pour son patience et son soutien.

A Younes KABBADJ qui nous a encadrées tout au long de la réalisation de ce projet, pour ses précieux conseils et ses encouragements continus.

A l'ensemble de mes professeurs pour tout le savoir-faire qu'ils ont su nous procurer.

Fatiha

Remerciements

Après avoir rendu grâce à Allah le Tout Puissant et le Miséricordieux.

Nos vifs remerciements sont destinés à nos professeurs du département de l'informatique, et tout le corps administratif de la faculté du science Agadir.

Nous tenons à exprimer notre profonde gratitude à notre cher professeur et encadrant M. KABBADJ pour son suivi et pour son énorme soutien, à qui nous témoignons notre reconnaissance pour ses orientations et son professionnalisme pour élaborer ce projet.

Nous adressons aussi nos vifs remerciements aux membres des jurys pour avoir bien voulu examiner et juger ce travail.

Enfin, nous remercions nos parents et tous nos camarades de promotion qui nous ont aidées de près ou de loin.

Résumé

L'objectif de ce travail est de rechercher un ensemble de modèles et d'algorithmes mathématiques qui examinent les données d'une transaction bancaire pour la classifier comme frauduleuse ou non. Les modèles sont mis en œuvre sous la forme d'un code informatique et d'algorithmes peu coûteux en temps de calcul qui peuvent donc être exécutés en temps réel.

L'objectif principal est d'appliquer différentes méthodes d'apprentissage automatique pour trouver la plus précise, en d'autres termes, celle pour laquelle le score de validation croisée est maximal. Ainsi, le principal problème à résoudre est la création d'un modèle qui pourrait détecter et bloquer instantanément une transaction frauduleuse donnée afin d'offrir une meilleure sécurité au détenteur de la carte.

Dans un premier temps, nous introduisons la notion de Machine Learning à travers sa définition et ses différentes méthodes. Nous expliquons ensuite le problème de classification et ses modèles d'interprétations. En particulier, nous décrivons des approches telles que la régression logistique, les vecteurs de support (SVM) et l'arbre de décision.

La partie suivante est consacrée à la présentation de chaque méthode de résolution de problème de classification : quelles sont les données initiales dont nous disposons et comment les interpréter pour trouver la solution ?

À la fin, nous appliquons ces méthodes aux données fournies à l'aide du langage de programmation Python et analysons les résultats.

Mots clés : Machine Learning, régression logistique, vecteurs de support (SVM), l'arbre de décision, Python, validation croisée, classification

Abstract

The work's aim is to find a set of selected mathematical models and algorithms that examine the data of a single banking transaction to classify it as fraudulent or not. Models are implemented in the form of a computer code and costly efficient algorithms which can be therefore executed in real-time.

The main objective is to apply different methods of machine learning to find the most accurate, in other words, the one in which the cross-validation score is maximal. Thus, the main problem to resolve is the creation of a model that could instantly detect and block a given fraudulent transaction in order to provide better security and user experience.

First, we will talk about Machine Learning, its definition, its different methods. We then determine the classification problem and its interpretation models. In particular, we describe such approaches as Logistic Regression, Support Vectors (SVM), and Decision Tree.

The next part is dedicated to presenting the methods for solving the classification problem: which initial data do we have and how to interpret it in order to find a solution to our problem?

At the end, we apply these methods to the provided data using Python programming language and analyze the results.

Keywords: Machine Learning, logistic regression, support vectors (SVM), decision tree, Python, cross validation, classification

Table de matières

Dédicaces.....	ii
Remerciements.....	iii
Résumé	iv
Abstract	v
Liste des figures	viii
Liste des tableaux.....	x
Liste des abréviations	xi
Introduction générale	1
1 Chapitre 1 : Machine Learning	4
1.1 Introduction	4
1.2 Définition de Machine Learning.....	4
1.3 Les différentes méthodes du ML	4
1.3.1 L'apprentissage supervisé	5
1.3.2 L'apprentissage non supervisé.....	5
1.3.3 L'apprentissage par renforcement	5
1.4 Algorithmes de Classification	6
1.4.1 La classification dans l'apprentissage automatique	6
1.4.2 Régression Logistique.....	6
1.4.2.1 Définition	6
1.4.2.2 Le modèle de Régression logistique	7
1.4.3 Arbres de décision	8
1.4.3.1 Implémentation de l'algorithme d'arbre de décision	9
1.4.4 Support Vector Machines.....	10
1.4.4.1 Fonctionnement de SVM	11
1.5 Précision, rappel et score F1 : les mesures de classification	11
1.6 Conclusion.....	13
2 Chapitre 2 : Machine Learning avec python.....	14
2.1 Introduction	14
2.2 Importation des données.....	14
2.3 Exploration des données.....	15
2.3.1 Visualisation des données	15
2.3.2 Vérification des dimensions des données.....	15
2.3.3 Obtention du type de données de chaque attribut.....	16

2.3.4	Résumé statistique des données.....	16
2.3.5	Suppression des colonnes non utiles.....	17
2.3.6	Examiner la distribution des classes	17
2.4	Division des données	18
2.5	Modélisation	19
2.5.1	Régression logistique	19
2.5.2	Arbre de décision.....	20
2.5.3	SVM.....	22
2.6	Résultats comparatifs et conclusion.....	23
3	Chapitre 3 Etude d'environnement du travail.....	24
3.1	Python	24
3.2	Bibliothèques et packages Python pour ML.....	24
3.2.1	Scikit-learn	24
3.2.2	Pandas	24
3.3	Environnement Python pour ML	25
3.3.1	Anaconda.....	25
3.3.2	Jupyter notebook.....	26
3.4	Python pour interface graphique : Tkinter	26
3.5	SQLite3	27
3.6	Editeur de texte.....	28
3.6.1	Visual code studio.....	28
3.7	PlantUML	29
3.8	DB Browser (SQLite)	30
4	Chapitre 4 : conception et réalisation de l'application.....	31
4.1	Introduction	31
4.2	Interface graphique	31
4.2.1	Modélisation d'un ATM simplifié avec contrôle de fraude.....	32
4.2.1.1	Diagramme de séquence	32
4.2.2	La base de données utilisée	33
4.2.3	Les fonctions utilisées pour la détection de la fraude	34
4.2.4	Fonctionnement	36
4.3	Conclusion.....	39
	Conclusion générale	40
	Webographie :.....	41
	Bibliographie :	44

Liste des figures

Figure 1-1 fonction logistique	7
Figure 1-2 Exemple d'arbre de décision	9
Figure 1-3 concepts importants dans SVM	11
Figure 2-1 capture pour les 5 premières observations de nos données, montrant les 10 premières variables.	14
Figure 2-2 capture pour le code d'importation du Dataset	15
Figure 2-3 Affichage des cinq premières lignes du dataframe	15
Figure 2-4 les dimensions du dataset	15
Figure 2-5 type de chaque attribut	16
Figure 2-6 description des données	17
Figure 2-7 suppression de la colonne 'Time'	17
Figure 2-8 première vérification des dimensions	18
Figure 2-9 deuxième vérification des dimensions	18
Figure 2-10 division des données en entraînement et test	19
Figure 2-11 modèle de la régression logistique et son résultat	20
Figure 2-12 modèle d'arbre de décision et son résultat	21
Figure 2-13 arbre de décision	22
Figure 2-14 modèle SVM et son résultat	23
Figure 3-1 python	24
Figure 3-2 scikit-learn	24
Figure 3-3 pandas	24
Figure 3-4 anaconda	25
Figure 3-5 capture anaconda	25
Figure 3-6 jupyter	26
Figure 3-7 un exemple de notebook Jupyter	26
Figure 3-8 tkinter	27
Figure 3-9 exemple de code tkinter	27
Figure 3-10 sqlite3	27
Figure 3-11 vscode	28
Figure 3-12 exemple d'interface de VSCode	28
Figure 3-13 plantUML	29
Figure 3-14 interface du code plantUML dans VScode	29
Figure 3-15 DB browser (SQLite)	30
Figure 3-16 interface du BD browser pour SQLite	30
Figure 4-1 ATM	31
Figure 4-2 Diagramme de séquence	32
Figure 4-3 table utilisée	33
Figure 4-4 données du test	34
Figure 4-5 le modèle de la régression logistique	34
Figure 4-6 fonction detectionFraude()	34
Figure 4-7 fonction CtrlFraude()	35
Figure 4-8 fonction confirmationID()	35
Figure 4-9 interface principale	36
Figure 4-10 les choix	37
Figure 4-11 interface pour faire retrait	37

Figure 4-12 saisir ID.....	38
Figure 4-13 ID correct.....	38
Figure 4-14 ID erronée.....	38
Figure 4-15 interface au cas d'une fraude.....	39
Figure 4-16 données incorrectes	39

Liste des tableaux

Tableau 1-1 tableau représentant les quatre paramètres	12
Tableau 2-1 illustrations des performances des algorithmes.....	23

Liste des abréviations

CART : Classification and Regression Tree
CVV : Card Code Verification
DAB : distributeurs automatiques de billets
ML: Machine learning
MMH: hyperplan marginal maximal
RL: Reinforcement Learning
SVM : Support vector machine
FN: False negative
FP: False positive
TN: True negative
TP: True positive
GUI: graphical user interface
IDE: integrated development environment
SQL: Structured Query Language
UI : l'interface utilisateur
VSCode :visual studio code
ATM: Automated teller machine
IA: intelligence artificielle
UML: Unified Modeling Language

Introduction générale

Les cartes de crédit et les guichets automatiques ont révolutionné la façon dont les transactions financières sont effectuées. Elles ne vous donnent pas seulement une passerelle pratique pour payer nos factures ou faire des achats, mais aussi de manière significative pour réduire votre temps de transaction. Vous n'êtes plus obligé de perdre votre temps dans une longue file d'attente juste pour retirer de l'argent d'une banque ou faire un transfert.

Cependant, cette technologie a également offert des nouvelles failles pour les personnes sans scrupule pour mener des activités criminelles.

La fraude par carte de crédit est un synonyme du vol qui est aussi ancien que l'humanité elle-même et peut prendre une variété illimitée de formes différentes. De nos jours le but peut être d'acheter des marchandises, transférer ou retirer un montant à partir d'un compte non-attribué à la personne qui effectue les transactions. La fraude par carte de crédit est également une usurpation d'identité. (1)

En outre, le développement de nouvelles technologies offre des moyens supplémentaires aux criminels de commettre une fraude. En termes simples, la fraude par carte de crédit est définie comme "lorsqu'une personne utilise la carte de crédit d'une autre personne pour des raisons personnelles alors que le propriétaire de la carte et l'émetteur de la carte ne sont pas au courant du fait que la carte est utilisée".

La fraude par carte commence soit par le vol de la carte physique, ou bien par l'usurpation des données confidentielles d'une carte bancaire qui sont : (2)

- **Numéro de carte** : Le numéro de carte est son identifiant unique, mais pas seulement.
- **Nom du titulaire** : C'est le nom du titulaire de la carte qui peut être une personne morale ou une personne physique. Si c'est une personne morale, alors c'est la dénomination sociale de l'entreprise qui figurera à cette place. Et si c'est une personne physique, alors c'est le nom de la personne qui y sera marqué. Pour les entreprises, le titulaire de la carte n'est pas le porteur.
- **Date d'expiration** : La date d'expiration se présente sous la forme suivante : MM/AA. Elle est composée du mois et de l'année jusqu'à

laquelle la carte est valide. Mais correspond toujours au dernier jour du mois.

- **Cryptogramme /CVV** : les 3 derniers chiffres permettant de sécuriser le paiement à distance.

Il existe plusieurs méthodes pour commettre une fraude par carte de crédit. Les fraudeurs sont des gens très talentueux et en mouvement rapide. Parmi ces méthodes on peut citer :

Le phishing (hameçonnage ou filoutage) : est une technique utilisée par des fraudeurs pour obtenir des renseignements personnels dans le but d'usurper l'identité d'une entreprise, d'un organisme financier ou d'une administration. (3) Le plus souvent, la victime reçoit donc un email semblant provenir d'une entreprise ou institution de confiance et l'invite à se connecter à son compte via un lien présent dans le courrier électronique pour mettre à jour ses informations, payer une facture, consulter ses messages etc....

Le problème c'est que ce lien ne dirige pas l'internaute vers le site officiel mais vers un site web créé par les fraudeurs qui est souvent la copie quasi conforme de l'original et qui l'invite à saisir via un formulaire son identifiant et mot de passe, ses coordonnées bancaires, ses codes de carte bancaire ou d'autres données personnelles sensibles

Ces données personnelles sont bien sûr récupérées par les pirates qui ont ensuite le champ libre pour exploiter ces informations et en tirer profit, utilisation de la carte bancaire sur internet, usurpation de votre identité auprès de services gouvernementaux ou bancaires, de services de téléphonie etc. (4)

Skimming : c'est une pratique frauduleuse basée sur le piratage et le clonage de cartes bancaires. Elle s'exerce surtout à partir des distributeurs de billets (DAB).

Pour pirater et cloner une carte bancaire, un « skimmer » a besoin de deux types d'information : les données stockées sur la carte (coordonnées bancaires) et le code personnel à 4 chiffres. Ces informations peuvent être récupérées via les distributeurs automatiques de billets, en les modifiant légèrement :

- **Avec le remplacement du lecteur de carte par un dispositif pirate** : la fente censée accueillir votre carte bancaire n'est plus celle du distributeur, mais celle conçue par un pirate. Elle est reliée à un téléphone, qui reçoit en temps réel les coordonnées bancaires des cartes scannées.

- **Avec un dispositif de surveillance pour obtenir le code à 4 chiffres** : il s'agit généralement d'une caméra pirate, cachée dans le plafonnier du distributeur. Certains skimmers utilisent également de faux claviers numériques, posés par-dessus le clavier original, qui transmettent à distance les codes saisis par les utilisateurs. Cette technique, bien que plus efficace, et aussi plus coûteuse, donc moins courante.

Une fois les données de la carte obtenues, les skimmers les copient sur des cartes vierges (appelées White Cards). Ces clones de cartes bancaires sont ensuite envoyés et vendus dans des pays étrangers (ex : USA), dans lesquels les paiements par carte ne sollicitent que la bande magnétique, et non la puce électronique qui est plus sécurisée.

Les propriétaires des cartes clonées remarquent alors des paiements inhabituels sur leurs relevés de compte, effectués depuis des pays étrangers où ils ne se sont jamais rendus. Dans ce cas, ils doivent entamer une procédure de remboursement auprès de leur établissement bancaire. (5)

Les pertes financières dues à la fraude affectent non seulement les organisations (les remboursements), mais également les individus. Si la banque perd de l'argent à cause d'un remboursement d'une transaction frauduleuse, les clients finissent par payer les pertes de l'organisation grâce à des taux d'intérêt plus élevés, des frais d'adhésion plus élevés, etc. La fraude peut également affecter la réputation et l'image d'un commerçant, causant des pertes non financières qui, bien que difficiles à quantifier à court terme, peuvent devenir visibles à long terme. Par exemple, si un titulaire de carte est victime d'une fraude avec une certaine banque, il ne peut plus faire confiance à son entreprise et choisir un concurrent.

Les fraudes détectables et indétectables constituent l'un des facteurs de risque les plus sensibles dans le secteur bancaire. Les outils traditionnels de détection de ces fraudes demeurent encore des solutions fragmentées, aux coûts opérationnels élevés, alors qu'une expertise Machine Learning peut se révéler.

Pour ça nous souhaitons explorer l'opportunité d'utiliser les méthodes de machine Learning, appliquées aux données, pour détecter avec une plus grande précision des fraudes potentielles par la carte bancaire. Le développement de l'intégration d'un module de détection de fraude par carte bancaire pourrait aider une banque à suspendre automatiquement une carte de crédit en cas de doute mais aussi d'alerter le propriétaire de la carte.

1 Chapitre 1 : Machine Learning

1.1 Introduction

Machine Learning joue un rôle clé dans nombreuses disciplines scientifiques et ses applications qui sont une partie de notre vie quotidienne. Il est utilisé par exemple pour filtrer les spams, pour la prévision météo, dans le diagnostic médical (1), la recommandation de produit, la détection de visage et Il peut jouer un rôle important dans le système de détection de fraude basé sur les données.

Les algorithmes de Machine Learning sont particulièrement adaptés à la lutte contre la fraude, en offrant comme principal avantage d'être rapidement adaptables à d'autres types de documents ou parcours, par rapport aux approches classiques qui nécessitent de refaire tout un travail à chaque nouvelle situation. (6)

Dans ce chapitre, nous allons présenter au lecteur le problème de l'apprentissage à partir des données avant de passer au domaine d'application spécifique.

1.2 Définition de Machine Learning

Le Machine Learning est une technologie d'intelligence artificielle permettant aux ordinateurs d'apprendre sans avoir été programmés explicitement à cet effet (c'est-à-dire laissez l'ordinateur apprendre quel calcul effectuer, plutôt que de lui donner ce calcul) et pour apprendre et se développer, les ordinateurs ont toutefois besoin de données à analyser et sur lesquelles s'entraîner (5).

Machine Learning est étroitement lié aux domaines de la statistique(6). Dans le même temps, il apparaît comme un sous-domaine de l'informatique et accorde une attention particulière à la partie algorithmique du processus d'extraction des connaissances.

1.3 Les différentes méthodes du ML

Pour donner à un ordinateur la capacité d'apprendre, on utilise des méthodes d'apprentissage qui sont fortement inspirées de la façon dont nous, les êtres humains, apprenons à faire des choses. Parmi ces méthodes, on compte :

- ✓ L'apprentissage supervisé (la méthode la plus utilisée en Machine Learning).
- ✓ L'apprentissage non supervisé.
- ✓ L'apprentissage par renforcement.

1.3.1 L'apprentissage supervisé

L'apprentissage supervisé est le concept derrière plusieurs applications sympas de nos jours : reconnaissance faciale de nos photos par les smartphones, filtres anti-spam des emails, etc.

Plus formellement, étant donné un ensemble de données D , décrit par un ensemble de caractéristiques X , un algorithme d'apprentissage supervisé va trouver une fonction de mapping entre les variables prédictives en entrée X et la variable à prédire Y . la fonction de mapping décrivant la relation entre X et Y s'appelle un modèle de prédiction.

$$f(X) \rightarrow Y$$

Les caractéristiques (features en anglais) X peuvent être des valeurs numériques, alphanumériques, des images... Quant à la variable prédite Y , elle peut être de deux catégories :

- **Variable discrète** : La variable à prédire peut prendre une valeur d'un ensemble fini de valeurs (qu'on appelle des classes). Par exemple, pour prédire si un mail est SPAM ou non, la variable Y peut prendre deux valeurs possibles : $Y = \{\text{SPAM} \mid \text{NON SPAM}\}$
- **Variable continue** : La variable Y peut prendre n'importe quelle valeur. Pour illustrer cette notion, on peut penser à un algorithme qui prend en entrée des caractéristiques d'un véhicule, et tentera de prédire le prix du véhicule (la variable Y). (8)

1.3.2 L'apprentissage non supervisé

A l'inverse de l'apprentissage supervisé (Supervised Learning) qui tente de trouver un modèle depuis des données labellisées $f(X) \rightarrow Y$, l'apprentissage non supervisé prend uniquement des données sans label (pas de variable à prédire Y). Un algorithme d'Unsupervised Learning va trouver des patterns ou une structure qui décrit les données. (8)

1.3.3 L'apprentissage par renforcement

L'apprentissage par renforcement (RL pour Reinforcement Learning) fait référence à une classe de problèmes d'apprentissage automatique, dont le but est d'apprendre, à partir d'expériences successives, ce qu'il convient de faire de façon à trouver la meilleure solution.

Dans un tel problème, on dit qu'un « agent » (l'algorithme, au sens du code et des variables qu'il utilise) interagit avec « l'environnement » pour trouver la solution optimale. L'apprentissage par renforcement diffère fondamentalement des problèmes supervisés et non supervisés par ce côté interactif et itératif : l'agent essaie plusieurs solutions (on parle « d'exploration »), observe la réaction de l'environnement et adapte son comportement (les variables) pour trouver la meilleure stratégie (il « exploite » le résultat de ses explorations) (9) .

1.4 Algorithmes de Classification

1.4.1 La classification dans l'apprentissage automatique

La classification est un processus de catégorisation d'un ensemble de données en classes. Elle peut être effectuée sur des données structurées ou non structurées. Le processus commence par la prévision de la classe des points de données donnés. Les classes sont souvent appelées cible, étiquette ou catégories.

La modélisation prédictive de la classification consiste à approximer la fonction de mappage des variables d'entrée aux variables de sortie discrètes. Le but principal est d'identifier dans quelle classe / catégorie les nouvelles données tomberont. (12)

1.4.2 Régression Logistique

1.4.2.1 Définition

La régression logistique est un algorithme de classification d'apprentissage supervisé utilisé pour prédire la probabilité d'une variable cible. La nature de la variable cible est dichotomique, ce qui signifie qu'il n'y a que deux classes possibles.

En termes simples, la variable cible est de nature binaire et ne peut prendre que deux valeurs soit 1 (signifie succès / oui) ou 0 (signifie échec / non).

Mathématiquement, un modèle de régression logistique prédit $P(Y = 1)$ en fonction de X . C'est l'un des algorithmes ML les plus simples qui peuvent être utilisés pour divers problèmes de classification tels que la détection du spam, la prédiction du diabète, la détection de la fraude, etc.

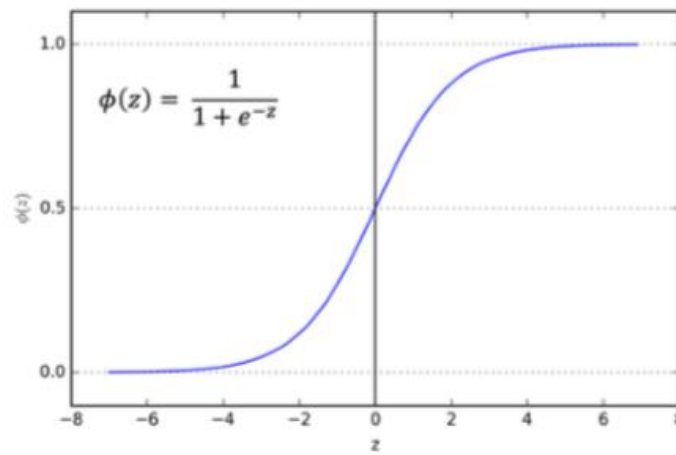


Figure 1-1 fonction logistique

1.4.2.2 Le modèle de Régression logistique

Pour les problèmes de classification binaire, un modèle linéaire $F = X \cdot \theta$ ne convient pas.

On développe alors une nouvelle fonction pour les problèmes de classification binaire, c'est la fonction logistique (aussi appelé fonction sigmoïde ou tout simplement sigma σ). Cette fonction a la particularité de donner toujours des valeurs comprises entre 0 et 1.

Donc, si la valeur de z va à l'infini positif, alors la valeur prédite de y deviendra 1 et si elle va à l'infini négatif alors la valeur prédite de y deviendra 0. Et si le résultat de la fonction sigmoïde est supérieur à 0,5, alors nous pouvons classer une donnée en classe 1 et si elle est inférieure à 0,5, nous pouvons la classer en classe 0.

Fonction Coût :

Pour écrire la Fonction Coût en une seule équation, on utilise l'astuce de séparer les cas $y = 0$ et $y = 1$ avec une annulation :

$$J(\theta) = \begin{cases} -\log(\sigma(z)) & \text{si } y = 1 \\ -\log(1 - \sigma(z)) & \text{si } y = 0 \end{cases}$$

Fonction Coût complète :

$$J(\theta) = -\frac{1}{m} \sum y \times \log(\sigma(z)) + (1 - y_i) \times \log(1 - \sigma(z))$$

Et $z = X \cdot \theta$

1.4.3 Arbres de décision

L'arbre de décision construit des modèles de classification ou de régression sous la forme d'une structure arborescente. Il décompose un ensemble de données en sous-ensembles de plus en plus petits tout en développant progressivement un arbre de décision associé. Le résultat final est un arbre avec des nœuds de décision et des nœuds de feuille. Le nœud de décision a deux branches ou plus et le nœud feuille représente une classification ou une décision. Le nœud de décision le plus haut dans un arbre qui correspond au meilleur prédicteur est appelé nœud racine. Les arbres de décision peuvent gérer à la fois des données catégorielles et numériques (11).

Dans notre graphique représentant l'arbre de décision on a les éléments suivants :

- Un nœud de décision (on pose une question)
- Un nœud terminal (on a trouvé la classe) (16)

Dans l'arbre de décision ci-dessous, les questions sont des nœuds de décision et les résultats finaux sont des feuilles.

Nous avons les deux types d'arbres de décision suivants.

- **Les arbres de classification** - Dans ce type d'arbres de décision, la variable de décision est catégorique. L'arbre de décision ci-dessous est un exemple d'arbre de décision de classification.
- **Les arbres de régression** - Dans ce type d'arbres de décision, la variable de décision est continue. (16)

1.4.3.1 Implémentation de l'algorithme d'arbre de décision

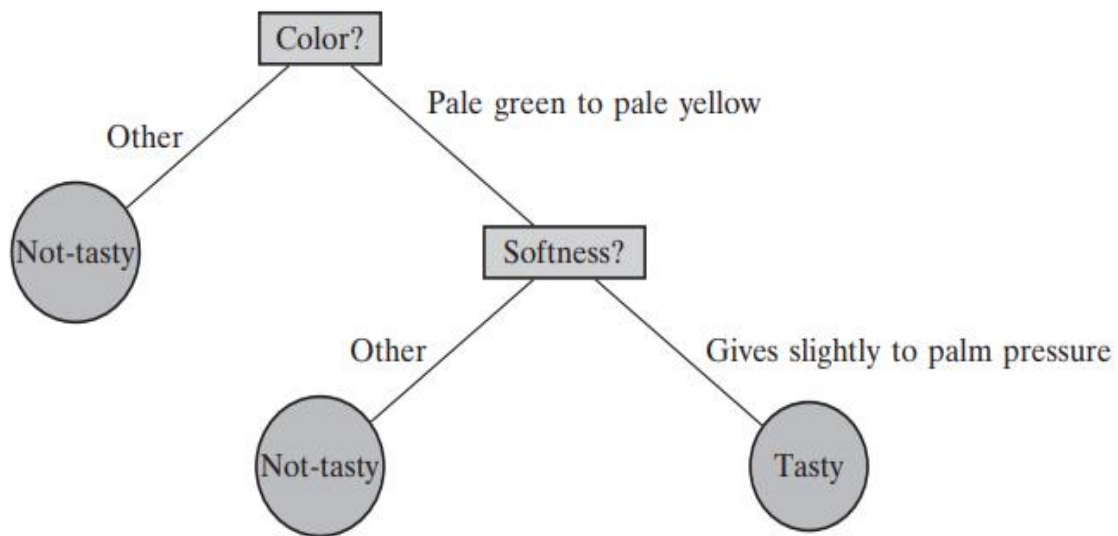


Figure 1-2 Exemple d'arbre de décision

➤ Gini Index

C'est le nom de la fonction de coût qui est utilisée pour évaluer les divisions binaires dans l'ensemble de données et fonctionne avec la variable cible catégorielle « Succès » ou « Échec ».

Plus la valeur de l'indice de Gini est élevée, plus l'homogénéité est élevée. Une valeur d'indice de Gini parfaite est 0 et la pire est 0,5 (pour un problème à 2 classes). L'index de Gini pour une division peut être calculé à l'aide des étapes suivantes :

- Tout d'abord, calculez l'indice de Gini pour les sous-nœuds en utilisant la formule $p^2 + q^2$, qui est la somme du carré de probabilité de réussite et d'échec.
- Ensuite, calculez l'indice de Gini pour la division en utilisant le score de Gini pondéré de chaque nœud de cette division.

L'algorithme CART (Classification and Regression Tree) utilise la méthode Gini pour générer des divisions binaires. (16)

➤ Construire un arbre

Comme nous savons qu'un arbre a un nœud racine et des nœuds terminaux. Après avoir créé le nœud racine, nous pouvons construire l'arborescence en suivant deux étapes :

Partie 1 : Création de nœuds terminaux

Lors de la création de nœuds terminaux d'arbre de décision, un point important est de décider quand arrêter la croissance de l'arbre ou créer d'autres nœuds terminaux. Cela peut être fait en utilisant deux critères, à savoir la profondeur maximale de l'arbre et les enregistrements de nœuds minimums, comme suit

- **Profondeur maximale de l'arbre** - Comme son nom l'indique, il s'agit du nombre maximal de nœuds dans un arbre après le nœud racine. Nous devons arrêter d'ajouter des nœuds terminaux une fois qu'un arbre a atteint la profondeur maximale, c'est-à-dire une fois qu'un arbre a obtenu le nombre maximal de nœuds terminaux.
- **Enregistrements de nœuds minimum** - Il peut être défini comme le nombre minimum de modèles d'apprentissage dont un nœud donné est responsable. Nous devons cesser d'ajouter des nœuds terminaux une fois que nous avons atteint un nombre des nœuds entre le nombre minimal et maximal. Le nœud terminal est utilisé pour faire une prédiction finale.

Partie 2 : Fractionnement récursif

Une fois que nous avons compris comment construire les nœuds, nous pouvons maintenant expliquer comment construire une arborescence. Le fractionnement récursif est une méthode pour construire l'arbre. Dans cette méthode, une fois qu'un nœud est créé, nous pouvons créer les nœuds enfants (nœuds ajoutés à un nœud existant) de manière récursive sur chaque groupe de données, généré en fractionnant l'ensemble de données, en appelant la même fonction encore et encore. (16)

1.4.4 Support Vector Machines

Les machines à vecteurs de support (SVM) sont des algorithmes d'apprentissage automatique supervisé puissants mais flexibles qui sont utilisés à la fois pour la classification et la régression. Dans les années 1960, les SVM ont été introduits pour la première fois, mais plus tard, ils ont été affinés en 1990. Les SVM ont un mode de mise en œuvre unique par rapport aux autres algorithmes d'apprentissage automatique. Ils sont devenus extrêmement populaires en raison de leur capacité à gérer plusieurs variables continues et catégorielles.

1.4.4.1 Fonctionnement de SVM

Un modèle SVM est essentiellement une représentation de différentes classes dans un hyperplan dans un espace multidimensionnel. L'hyperplan sera généré de manière itérative par SVM afin que l'erreur puisse être minimisée. Le but de SVM est de diviser les ensembles de données en classes pour trouver un hyperplan marginal maximal (MMH).

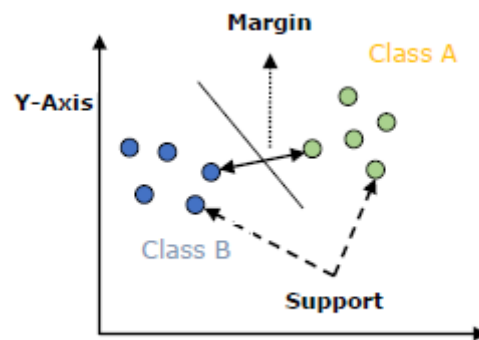


Figure 1-3 concepts importants dans SVM

Les éléments suivants sont des concepts importants dans SVM :

- **Vecteurs de support** - Les points de données les plus proches de l'hyperplan sont appelés vecteurs de support. La ligne de séparation sera définie à l'aide de ces points de données.
- **Hyperplan** - Comme nous pouvons le voir dans le diagramme ci-dessus, c'est un plan ou un hyper-plan qui définit la frontière de décision entre les différentes classes.
- **Marge** – C'est la distance entre un hyper-plan séparateur et les points de données les plus proches. Une marge maximale garantit une bonne performance lors de la généralisation puisqu'elle permet de minimiser le risque en éloignant le plan le plus possible de l'ensemble d'apprentissage.

L'objectif principal de SVM est de séparer l'ensemble de données en classes différentes en trouvant un hyper plan à marge maximale. Cela s'effectue en deux étapes :

- Premièrement, SVM générera des hyperplans qui séparent les classes de manière optimale.
- Ensuite, il choisira l'hyperplan avec la plus grande marge. (17)

1.5 Précision, rappel et score F1 : les mesures de classification

Une fois que vous avez construit votre modèle, la question la plus importante qui se pose est à quel point votre modèle est bon ? L'évaluation de votre modèle est donc la tâche la plus importante du projet qui définit la qualité de vos prévisions.

Une matrice de confusion est un tableau qui est souvent utilisé pour décrire les performances d'un modèle de classification sur un ensemble de données de test dont les vraies valeurs sont connues. Toutes les mesures peuvent être calculées en utilisant quatre paramètres. Parlons donc d'abord de ces quatre paramètres.

Tableau 1-1 tableau représentant les quatre paramètres

Actual Class	Predicted class		
		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Les vrais positifs et les vrais négatifs sont les observations qui sont correctement prédites et donc affichées en vert. Nous voulons minimiser les faux positifs et les faux négatifs afin qu'ils soient affichés en rouge. Ces termes sont un peu déroutants. Prenons donc chaque terme un par un et comprenons-le pleinement.

Vrais positifs (TP) - Ce sont les valeurs positives correctement prédites, ce qui signifie que la valeur de la classe réelle est oui et la valeur de la classe prédite est également oui. Par exemple, si la valeur réelle de la classe indique qu'il s'agit d'une fraude et que la classe prévue vous dit la même chose.

Vrais négatifs (TN) - Ce sont les valeurs négatives correctement prédites, ce qui signifie que la valeur de la classe réelle est non et que la valeur de la classe prédite est également non. Par exemple, si la classe réelle dit qu'il ne s'agit pas d'une fraude et que la classe prévue vous dit la même chose.

Faux positifs et faux négatifs, ces valeurs se produisent lorsque votre classe réelle est en contradiction avec la classe prédite.

Faux positifs (FP) - Lorsque la classe réelle est non et que la classe prédite est oui. Par exemple, si la classe réelle indique qu'il ne s'agit pas d'une fraude, mais que la classe prévue vous indique qu'il s'agit d'une fraude.

Faux négatifs (FN) - Lorsque la classe réelle est oui mais la classe prédite en non. Par exemple, si la valeur réelle de la classe indique qu'il s'agit d'une fraude et que la classe prévue vous indique qu'il ne s'agit pas d'une fraude.

Une fois que vous avez compris ces quatre paramètres, nous pouvons calculer la précision, le rappel et le score F1.

Précision - La précision est le rapport des observations positives correctement prédites au total des observations positives prévues.

$$\text{Précision} = \frac{TP}{TP+FP}$$

Rappel - Le rappel est le rapport des observations positives correctement prédites à toutes les observations dans la classe réelle.

$$\text{Rappel} = \frac{TP}{TP+FN}$$

Score F1 - Le score F1 est la moyenne pondérée de la précision et du rappel.
(15)

$$\text{Score F1} = \frac{2 * (\text{Rappel} * \text{Précision})}{(\text{Rappel} + \text{Précision})}$$

1.6 Conclusion

En conclusion, Machine Learning est considéré comme une arme très puissante contre la fraude, en utilisant ses algorithmes qui s'appuie sur des données.

Ces données sont des informations tiré du comportement des utilisateurs (la somme d'argent retirer, virement, nombre de retrait...) qui se présente comme des *features* pour le modèle choisi. Et on se basant sur ces comportement le modèle va être capable de faire des prédictions et pourquoi pas des détections efficaces du fraude.

2 Chapitre 2 : Machine Learning avec python

2.1 Introduction

Dans le chapitre précédent nous avons parlé d'apprentissage automatique et surtout comment ses techniques sont efficaces dans la détection des fraudes et que chaque technique a une façon d'être interpréter en se basant sur des données du problème, cependant il est nécessaire pour un modèle de classification de disposer de données nécessaires.

Nous avons opté pour l'utilisation d'un dataset qui décrit une liste des transactions bancaire (frauduleux ou non) avec un certain nombre de caractéristique (time, amount, ...).

A partir de ces données et un modèle, on va prédire si une transaction bancaire s'agit d'une fraude ou non.

2.2 Importation des données

Le format de données CSV est le format le plus courant pour les données ML, il est utilisé pour stocker des données tabulaires telles qu'une feuille de calcul. Dans les fichiers de données CSV, l'en-tête contient les informations de chaque champ.

Notre Dataset contient les 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.

Il est important de noter que pour des raisons de confidentialité, les données ont été anonymisées, les noms des variables ont été renommés V1, V2, V3 jusqu'à V28.

	A	B	C	D	E	F	G	H	I	J	K
1	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
2	0	-1.35980713	-0.07278117	2.53634674	1.37815522	-0.33832077	0.46238778	0.23959855	0.0986979	0.36378697	0.09079417
3	0	1.19185711	0.26615071	0.16648011	0.44815408	0.06001765	-0.08236081	-0.07880298	0.08510165	-0.25542513	-0.16697441
4	1	-1.35835406	-1.34016307	1.77320934	0.37977959	-0.50319813	1.80049938	0.79146096	0.24767579	-1.51465432	0.20764287
5	1	-0.96627171	-0.18522601	1.79299334	-0.86329128	-0.01030888	1.24720317	0.23760894	0.37743587	-1.38702406	-0.05495192
6	2	-1.15823309	0.87773675	1.54871785	0.40303393	-0.40719338	0.09592146	0.59294075	-0.27053268	0.81773931	0.75307443
7	2	-0.42596588	0.96052304	1.14110934	-0.16825208	0.42098688	-0.02972755	0.47620095	0.26031433	-0.56867138	-0.3714072

Figure 2-1 capture pour les 5 premières observations de nos données, montrant les 10 premières variables.

Pour charger le fichier de données CSV on utilise la fonction `pandas.read_csv()` de la bibliothèque Pandas. (13)

```

import pandas as pd
dataset = pd.read_csv('creditcard.csv')

```

Figure 2-2 capture pour le code d'importation du Dataset

2.3 Exploration des données

Avant d'utiliser des données par un algorithme de classification, il est nécessaire d'explorer ces données afin de découvrir leur distribution, leur format, la complétude ou incomplétude.

Ceci d'avère crucial pour la préparation des données, le choix de l'algorithme d'apprentissage ainsi que le choix de la métrique d'évaluation.

2.3.1 Visualisation des données

```

In [13]: dataset.head()

```

Out[13]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739

Figure 2-3 Affichage des cinq premières lignes du dataframe

2.3.2 Vérification des dimensions des données

C'est toujours une bonne pratique de savoir combien on a de données (nombres de lignes et de colonnes), car si nous avons trop de lignes et de colonnes, cela prendrait beaucoup de temps pour exécuter l'algorithme et former le modèle.

Et si nous avons trop moins de lignes et de colonnes, nous n'aurions pas assez de données pour bien former le modèle. (13)

```

dataset.shape
(284807, 31)

```

Figure 2-4 les dimensions du dataset

Le bloc de données(dataframe) est de forme (284807,31), ce qui implique 284807 lignes et 31 colonnes.

2.3.3 Obtention du type de données de chaque attribut

Savoir le type de données de chaque attribut est important, nous pouvons parfois avoir besoin de convertir un type de données en un autre. Par exemple, nous pouvons avoir besoin de convertir une chaîne de caractère en une valeur numérique pour un algorithme qui ne prend en charge que des valeurs numériques tel que SVM ou la régression logistique.

```
In [16]: dataset.dtypes

Out[16]: Time      float64
V1      float64
V2      float64
V3      float64
V4      float64
V5      float64
V6      float64
V7      float64
V8      float64
V9      float64
V10     float64
V11     float64
V12     float64
V13     float64
V14     float64
V15     float64
V16     float64
V17     float64
V18     float64
V19     float64
V20     float64
V21     float64
V22     float64
V23     float64
V24     float64
V25     float64
V26     float64
V27     float64
V28     float64
Amount  float64
Class   int64
dtype: object
```

Figure 2-5 type de chaque attribut

2.3.4 Résumé statistique des données

Cela peut être fait à l'aide de la fonction `describe()` de Pandas DataFrame qui fournit des statistiques pour chacune de nos colonnes. (13)

- Count
- Mean
- Standard Deviation

- Minimum Value
- Maximum value
- 25%
- Median i.e. 50%
- 75%

```
dataset.describe()
```

	Time	V1	V2	V3	V4	V5
count	284807.000000	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05
mean	94813.859575	3.919560e-15	5.688174e-16	-8.769071e-15	2.782312e-15	-1.552563e-15
std	47488.145955	1.958696e+00	1.651309e+00	1.516255e+00	1.415869e+00	1.380247e+00
min	0.000000	-5.640751e+01	-7.271573e+01	-4.832559e+01	-5.683171e+00	-1.137433e+02
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-01	-6.915971e-01
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-02	-5.433583e-02
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-01	6.119264e-01
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+01	3.480167e+01

Figure 2-6 description des données

2.3.5 Suppression des colonnes non utiles

Dans le dataset il peut exister des données qui n'ont aucune importance, dans ce cas on le supprime.

```
dataset = dataset.drop(['Time'], axis=1)
```

Figure 2-7 suppression de la colonnes 'Time'

2.3.6 Examiner la distribution des classes

Il est important de connaître la distribution des valeurs de classe dans les problèmes de classification car si nous avons une distribution de classe très déséquilibrée, c'est-à-dire qu'une classe a beaucoup plus d'observations que l'autre classe, alors elle peut nécessiter un traitement spécial au stade de la préparation des données de notre projet ML. Nous pouvons facilement obtenir la distribution des classes en Python avec l'aide de Pandas DataFrame. (13)

Avant :

```
dataset['Class'].value_counts()

0    284315
1      492
Name: Class, dtype: int64
```

Figure 2-8 première vérification des dimensions

value_counts : Renvoie une série contenant des nombres de valeurs uniques de colonne 'class'. L'objet résultant sera dans l'ordre décroissant de sorte que le premier élément est l'élément le plus fréquent.

En comparant le nombre des colonnes de classe 1 avec le nombre des colonnes de classe 0. Nous constatons que le dataset est déséquilibré.

Une technique largement adoptée pour traiter les ensembles de données très déséquilibrés est appelée resampling. Il consiste à retirer des échantillons de la classe majoritaire (Under-sampling) et / ou à ajouter d'autres exemples de la classe minoritaire (over-sampling). (15)

Après :

```
from sklearn.utils import resample
dataset_majority = dataset[dataset.Class==0]
dataset_minority = dataset[dataset.Class==1]
dataset_majority_downsampled = resample(dataset_majority, replace=False, n_samples=492, random_state=24)
dataset = pd.concat([dataset_majority_downsampled, dataset_minority])
dataset['Class'].value_counts()

1    492
0    492
Name: Class, dtype: int64
```

Figure 2-9 deuxième vérification des dimensions

utils.resample : accepte un paramètre de stratification pour l'échantillonnage selon les distributions de classe

Nous avons mis les colonnes de class 1 à dataframe nommée dataset_minority et les colonnes de class 0 à l'autre dataframe nommée dataset_majority. Nous avons pris 492 colonnes a dataset_majority Et nous l'avons mis en dataset_majority_downsampled. Enfin, nous avons concaténé les deux dataframe dataset_majority_downsampled et dataset_minority au dataset. Avec ça on obtient des données équilibrées.

2.4 Division des données

Généralement, les données seront divisées en trois segments différents : entraînement, tests et validation croisée. L'algorithme sera entraîné sur un ensemble partiel de données et de paramètres modifiés sur un ensemble de tests. La performance des données est mesurée à l'aide d'un ensemble de validation croisée. Les modèles très performants seront ensuite testés pour diverses répartitions aléatoires des données afin d'assurer la cohérence des résultats. (14)

Nous pouvons utiliser la fonction `train_test_split()` du paquet `sklearn` python pour diviser les données en ensembles.

```
y =dataset['Class'] # is output
X =dataset.drop('Class',axis=1) #is input

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(X,y,test_size=0.19)
```

Figure 2-10 division des données en entraînement et test

Dans l'exemple on a mis le 81% des données pour l'entraînement et 19% pour le test.

2.5 Modélisation

La création de modèles est une étape essentielle pour prévoir la fraude

2.5.1 Régression logistique

Sklearn.linear_model : Cette classe implémente une régression logistique régularisée

Tout d'abord, on importe le module de régression logistique et on crée un objet classificateur de régression logistique à l'aide de la fonction `LogisticRegression()`.

Ensuite, on ajuste le modèle sur le train à l'aide de `fit()` et on effectue la prédiction sur l'ensemble de test à l'aide de `Predict()` (17)

```

from sklearn.linear_model import LogisticRegression
logmodel = LogisticRegression().fit(x_train, y_train)
Predictions = logmodel.predict(x_test)
from sklearn.metrics import classification_report
print(classification_report(y_test, Predictions))

```

	precision	recall	f1-score	support
0	0.91	0.93	0.92	89
1	0.94	0.92	0.93	98
accuracy			0.93	187
macro avg	0.92	0.93	0.93	187
weighted avg	0.93	0.93	0.93	187

Figure 2-11 model de la régression logistique et son résultat

Classification_report () : Cette fonction renvoie un rapport de classification plus complet et prend également les sorties réelles et prévues comme arguments. Il renvoie un rapport sur la classification en tant que dictionnaire si vous fournissez output_dict=True ou une chaîne dans le cas contraire. (18)

Il est important de regarder la précision, le rappel et le score f1. La précision = 0.94, le rappel = 0.92 et le score f1 = 0.93 sont très haut. Cela suggère qu'il y a 92% de chances de prédire réellement une transaction frauduleuse et qu'il y a 94% de chances qu'une transaction frauduleuse prédite soit réellement vraie.

2.5.2 Arbre de décision

Dans scikit-learn, la classe sklearn.tree.DecisionTreeClassifier permet de réaliser une classification multi-classe à l'aide d'un arbre de décision.

On commence par importer le module tree et construire l'objet arbre
Le paramètre **max_depth** est un seuil sur la profondeur maximale de l'arbre.

Le paramètre **criterion** est une fonction pour mesurer la qualité d'un split. Les critères pris en charge sont « gini » pour l'impureté Gini

```
from sklearn import tree
modelTree=tree.DecisionTreeClassifier(random_state=0,criterion='gini',max_depth=9)
y_predict = modelTree.predict(x_test)
from sklearn.metrics import classification_report
print(classification_report(y_test, y_predict))
```

	precision	recall	f1-score	support
0	0.87	0.94	0.90	79
1	0.95	0.90	0.92	108
accuracy			0.91	187
macro avg	0.91	0.92	0.91	187
weighted avg	0.92	0.91	0.91	187

Figure 2-12 model d'arbre de décision et son résultat

La précision = 0.95, le rappel = 0.90 et le score f1 = 0,92 sont très bon. Cela suggère qu'il y a 90% de chances de prédire réellement une transaction frauduleuse et qu'il y a 95% de chances qu'une transaction frauduleuse prédite soit réellement vraie.

Génération de l'arbre de décision

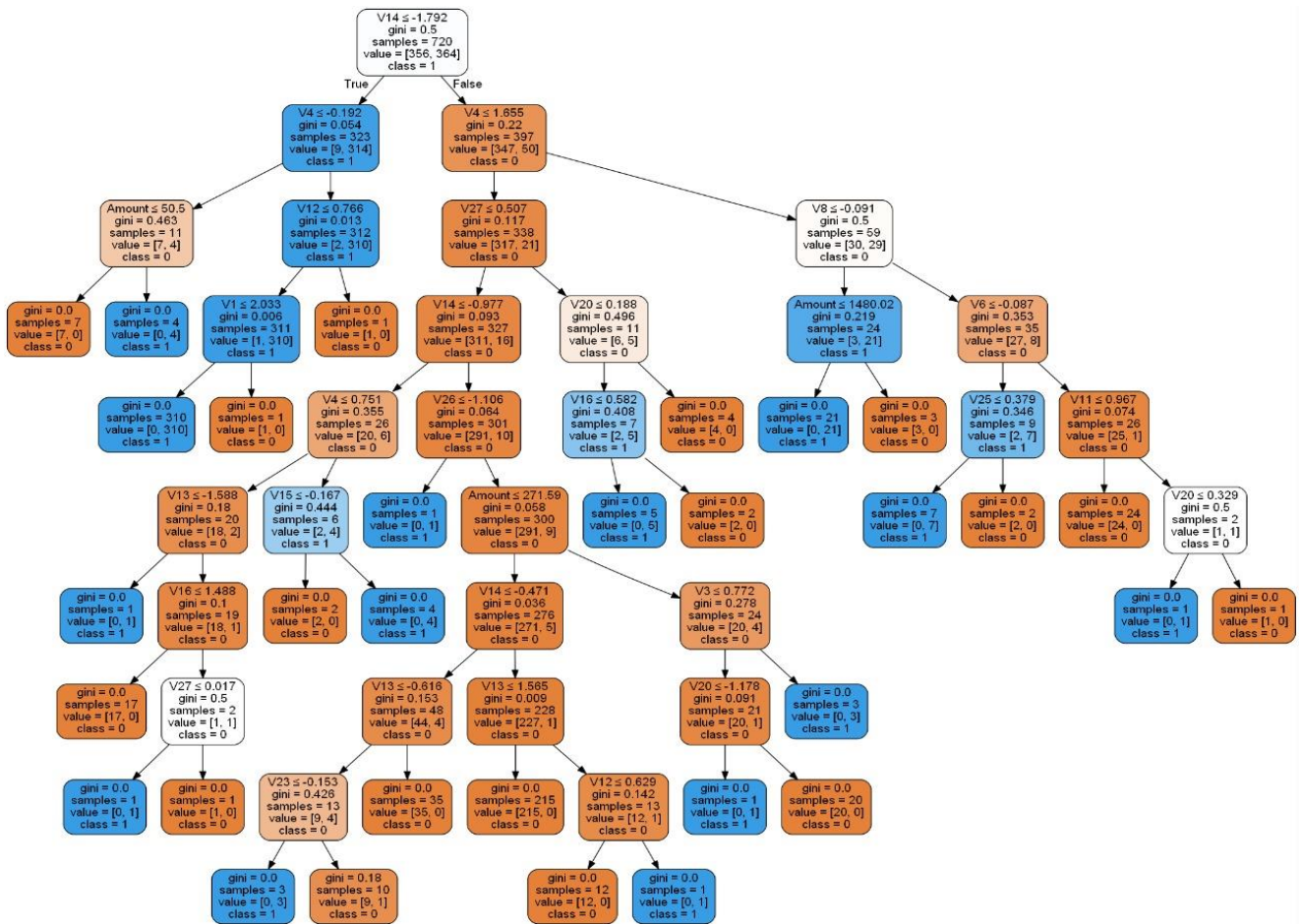


Figure 2-13 arbre de decision

2.5.3 SVM

SVM : Cette classe implémente Support Vector Machine.

Nous exploitons la classe SVC de scikit-learn pour réaliser modélisation. Ensuite nous utilisons kernel = 'linear', car ici nous créons SVM pour des données linéairement séparables. Cependant, nous pouvons le changer pour les données non linéaires. Et puis nous avons ajusté le classificateur à l'ensemble de données d'apprentissage fit (x_train, y_train) (19) et effectuez la prédiction sur l'ensemble de test à l'aide de Predict ().

Enfin, nous demandons au modèle de prédire les étiquettes pour l'ensemble de test.

```

from sklearn import svm
classifieur=svm.SVC(kernel='linear', gamma='auto', C=2)
classifieur.fit(x_train,y_train)
y_predict = classifieur.predict(x_test)
from sklearn.metrics import classification_report
print(classification_report(y_test, y_predict))

```

	precision	recall	f1-score	support
0	0.83	0.95	0.89	94
1	0.94	0.81	0.87	93
accuracy			0.88	187
macro avg	0.88	0.88	0.88	187
weighted avg	0.88	0.88	0.88	187

Figure 2-14 model SVM et son résultat

La précision = 0.94, le rappel = 0.81 et le score f1 = 0,87 sont très bon. Cela suggère qu'il y a 81% de chances de prédire réellement une transaction frauduleuse et qu'il y a 94% de chances qu'une transaction frauduleuse prédite soit réellement vraie.

2.6 Résultats comparatifs et conclusion

Le tableau suivant illustre les performances comparatives des algorithmes utilisés. D'après les résultats existants, Support Vector Machine a donné une précision de 0,94. En utilisant la régression logistique, les résultats semblent être similaires à ceux de SVM. En utilisant l'algorithme de l'arbre de décision, les résultats obtenus sont de 0,95 ce qui est supérieur aux performances des deux autres méthodes.

Le tableau suivant illustre les résultats comparatifs des trois algorithmes :

Tableau 2-1 illustrations des performances des algorithmes

	Algorithmes	Précisions
1	Régression logistique	94%
2	Arbre de décision	95%
3	SVM	94%

3 Chapitre 3 Etude d'environnement du travail

3.1 Python

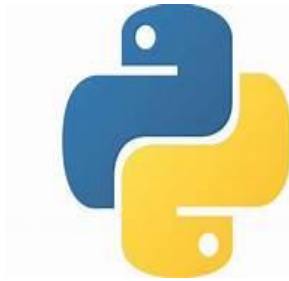


Figure 3-1 python

Python est une plateforme complète et généraliste pour le développement logiciel, très facile d'accès et capable de se spécialiser de manière très pointue dans la quasi-totalité des domaines informatiques. Python est utilisé par un public très large, des développeurs web professionnels, des chercheurs en intelligence artificielle ou en bio-informatique, des administrateurs systèmes, ou même des programmeurs occasionnels. C'est le mélange de polyvalence et de facilité qui fait la force de Python. Avec un bref apprentissage et un minimum

d'efforts, vous serez capable d'envisager n'importe quel type d'application de manière extrêmement efficace et de la terminer (ou de la faire terminer) en temps voulu. (20)

Python est le langage de programmation le plus utilisé dans le domaine du Machine Learning.

3.2 Bibliothèques et packages Python pour ML

3.2.1 Scikit-learn



Figure 3-2 scikit-learn

Scikit-learn est la librairie qui contient toutes les fonctions de l'état de l'art du Machine Learning. (14)

Scikit-learn est très utile pour les algorithmes de classification ou de régression. (15)

3.2.2 Pandas

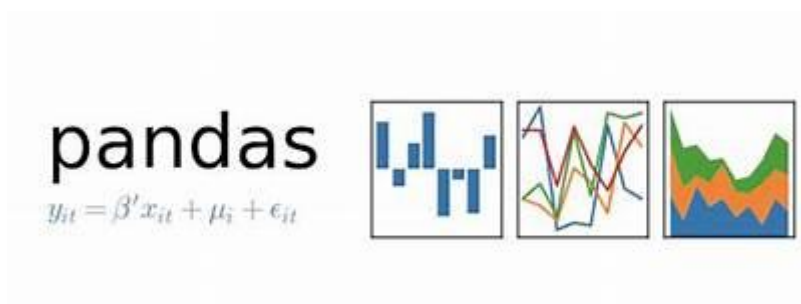


Figure 3-3 pandas

Pandas est l'une des bibliothèques le plus utilisée en ML. Elle offre de nombreuses fonctionnalités natives très utiles. Il est notamment possible de lire des données en provenance de nombreuses sources, de créer de larges dataframes à partir de ces sources, et d'effectuer des analyses agrégées basées sur les questions auxquelles on souhaite obtenir des réponses.

Des fonctionnalités de visualisation permettent également de générer des graphiques à partir des résultats des analyses, ou de les exporter au format Excel. On peut aussi s'en servir pour la manipulation de tableaux numériques et de séries temporelles. (15)

3.3 Environnement Python pour ML

3.3.1 Anaconda



Figure 3-4anaconda

Anaconda est une distribution libre et open source des langages de programmation Python et R appliqué au développement d'applications dédiées à l'apprentissage automatique (traitement de données à grande échelle, analyse prédictive, calcul scientifique), qui vise à simplifier la gestion des paquets et de déploiement. Les versions de paquetages sont gérées par le système de gestion de paquets conda.

La distribution Anaconda est utilisée par plus de 6 millions d'utilisateurs et comprend plus de 250 paquets populaires en science des données adaptés pour Windows, Linux et MacOS. (23)

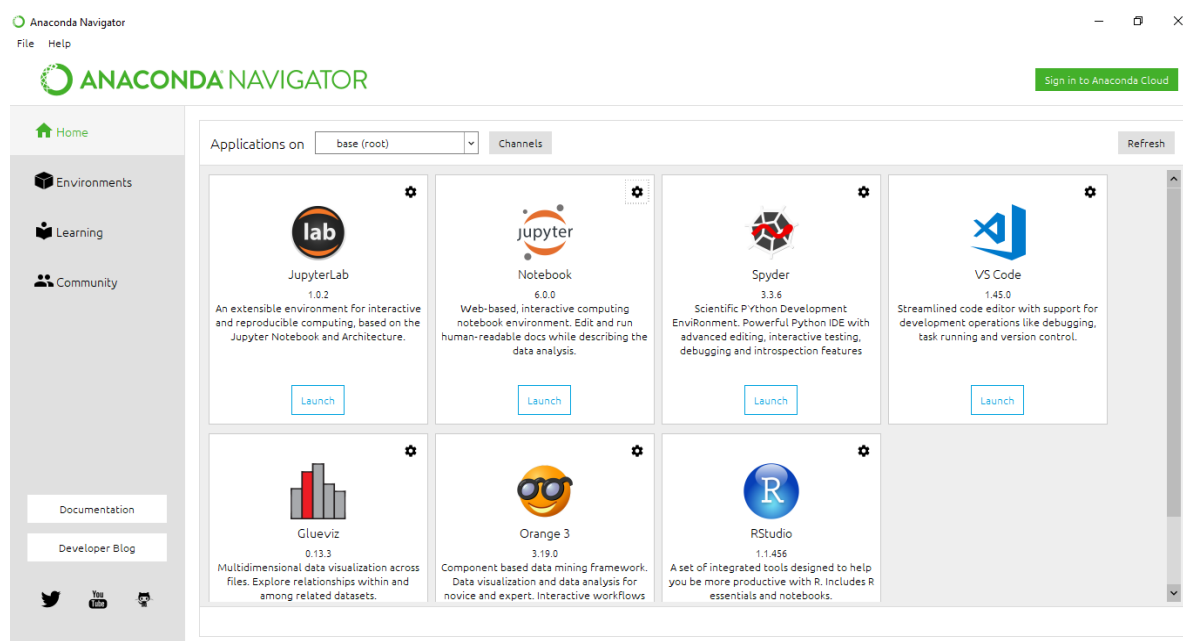


Figure 3-5 capture anaconda

3.3.2 Jupyter notebook



Figure 3-6 jupyter

Les notebooks Jupyter sont des cahiers électroniques qui, dans le même document, peuvent rassembler du texte, des images, des formules mathématiques et du code informatique exécutable. Ils sont manipulables interactivement dans un navigateur web.

Initialement développés pour les langages de programmation Julia, Python et R (d'où le nom Jupyter), les notebooks Jupyter supportent près de 40 langages différents.

La cellule est l'élément de base d'un notebook Jupyter. Elle peut contenir du texte formaté au format Markdown ou du code informatique qui pourra être exécuté. (24)

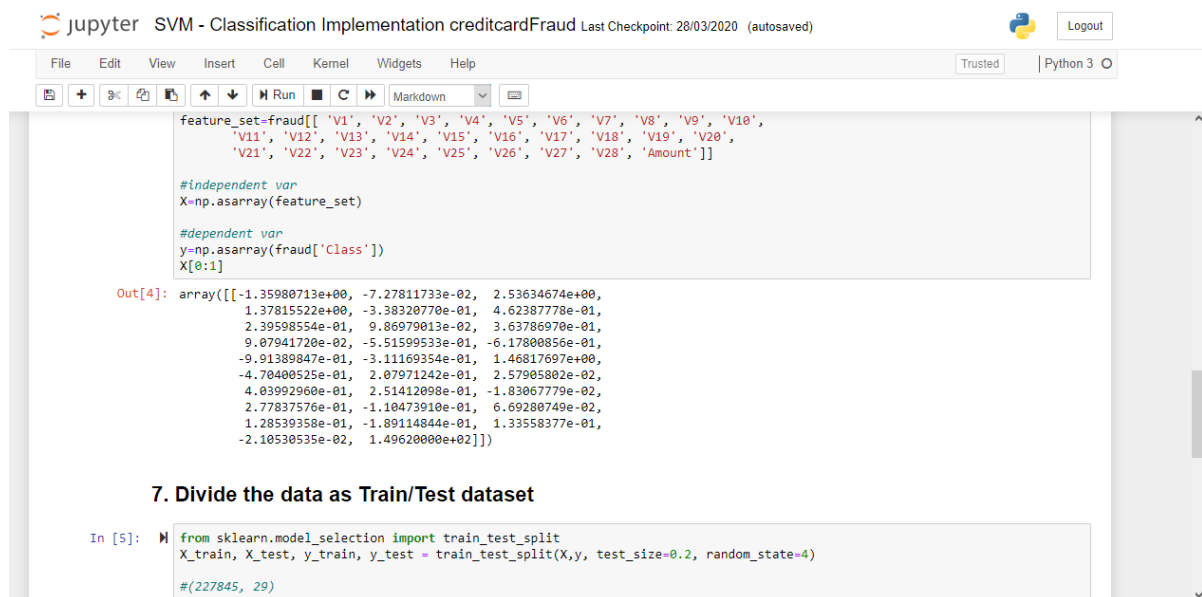


Figure 3-7 un exemple de notebook Jupyter

3.4 Python pour interface graphique : Tkinter

Tkinter (Tk interface) est un module intégré à la bibliothèque standard de Python, permettant de créer des interfaces graphiques :

- Des fenêtres,
- Des widgets (boutons, zones de texte, cases à cocher, ...),



Figure 3-8 tkinter

- Des événements (clavier, souris, ...).

Tkinter est disponible sur Windows et la plupart des systèmes Unix : les interfaces créées avec Tkinter sont donc portables. (25)

Voici un exemple de code pour faire une fenêtre :

```
# Import des noms du module
from tkinter import *

# Création d'un objet "fenêtre"
fenetre = Tk()

# Titre (Label)
titre = Label(fenetre, text = "L'informatique, c'est fantastique !")

# Affichage du titre
titre.pack()

# Ajout des autres widgets
# .....

# Démarrage de la boucle Tkinter (à placer à la fin !!!)
fenetre.mainloop()
```

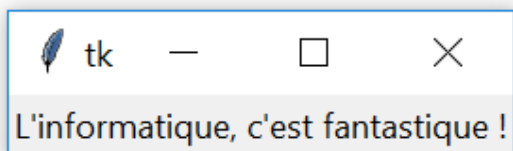


Figure 3-9 exemple de code tkinter

3.5 SQLite3



Figure 3-10 sqlite3

SQLite est une bibliothèque qui fournit une base de données légère sur disque ne nécessitant pas de processus serveur distinct et permet d'accéder à la base de données à l'aide d'une variante du langage de requête SQL. Certaines applications peuvent utiliser SQLite pour le stockage de données interne. Il est également possible de prototyper une application utilisant SQLite, puis de transférer le code dans une base de données plus grande telle qu'Oracle. (26)

3.6 Editeur de texte

3.6.1 Visual code studio

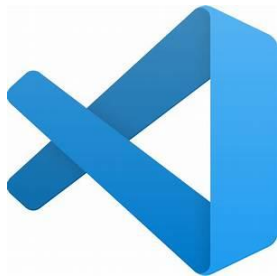


Figure 3-11 vscode

Visual Studio Code a l'avantage de combiner la facilité d'utilisation d'un éditeur de texte classique avec des fonctionnalités plus puissantes qui le rapproche d'un IDE et ne nécessite qu'une configuration minimale.

L'interface graphique (GUI) est bien pensée, chaque fonctionnalité semble apparaître au moment où vous en avez besoin, inutile de vous rappeler de tous les raccourcis clavier pour devenir un « power-user ».

VSCode est léger et rapide. Beaucoup plus qu'Atom alors que tous les deux sont développés en Node.js et utilisent le framework Electron. La différence réside peut-être dans le fait que l'UI Editeur a été développé à partir de Monaco (Visual Studio Online).

VSCode offre de nombreuses extensions, dont certaines sont installés par défaut. Il vous faudra sans doute un peu de temps pour choisir les plugins qui vous conviennent le mieux, il en existe plusieurs milliers ! (27)

Voici un exemple d'interface de VSCode :

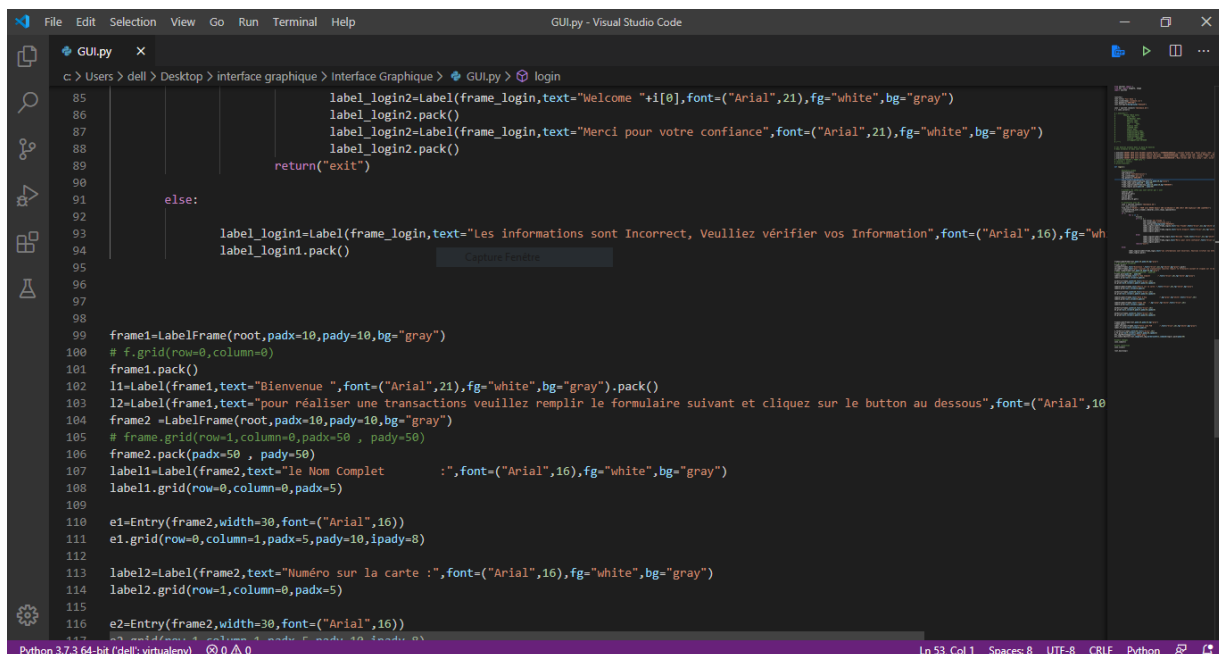


Figure 3-12 exemple d'interface de VSCode

3.7 PlantUML



Figure 3-13 plantUML

PlantUML est un outil open source permettant aux utilisateurs de créer des diagrammes UML à partir d'un langage de texte brut. Le langage de PlantUML est un exemple de langage spécifique au domaine. Il utilise le logiciel Graphviz pour disposer ses diagrammes. Il a été utilisé pour permettre aux étudiants malvoyants de travailler avec UML. PlantUML aide également les ingénieurs logiciels malvoyants à concevoir et lire des diagrammes UML.

PlantUML utilise un code bien formé et lisible par l'homme pour rendre les diagrammes.

Il existe d'autres formats de texte pour la modélisation UML, mais PlantUML prend en charge de nombreux types de diagrammes et n'a pas besoin d'une mise en page explicite, bien qu'il soit possible de modifier les diagrammes si nécessaire. (31)

Voici un exemple de code :

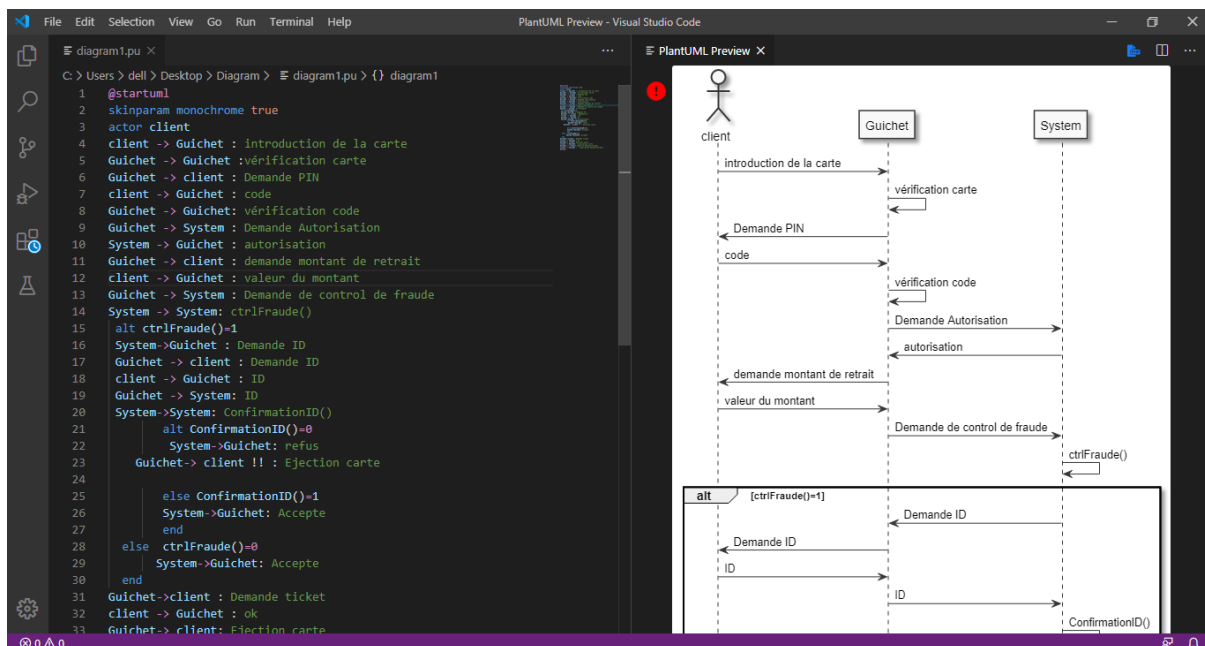


Figure 3-14 interface du code plantUML dans VScode

3.8 DB Browser (SQLite)



Figure 3-15 DB browser (SQLite)

DB Browser pour SQLite est un outil visuel et open source de haute qualité pour créer, concevoir et éditer des fichiers de base de données compatibles avec SQLite.

Il est destiné aux utilisateurs et aux développeurs souhaitant créer des bases de données, rechercher et modifier des données. Il utilise une interface similaire à une feuille de calcul et vous n'avez pas besoin d'apprendre des commandes SQL compliquées. (31)

voici un exemple de donnees dans une base de donnees :

Database Structure Browse Data Edit Pragma Execute SQL									
Table: client									
	ID	Name	CardNumber	CVV	ExpVisa	codePIN	amount	oldbalanceOrg	newbala
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	CC25034	Jayden Miller	43800689160...	323	10/20	8434	9839.64	170136.0	160296.1
2	CC59625	Gabriel Martinez	48420395966...	952	05/24	2943	1864.28	21249.0	19384.7
3	CC69314	Landon Collins	41028828525	020	01/25	5353	11668.14	41554.0	20885.8

Figure 3-16 interface du BD browser pour SQLite

4 Chapitre 4 : conception et réalisation de l'application

4.1 Introduction

Le nombre de fraudes contre les comptes et cartes bancaires ont explosé ces dernières années. Dans la grande majorité des cas, ce sont des fraudes par carte bancaire dont la plus répandue se fait via les ATM. (32)

Les organisations bancaires devraient alors bloquer les transactions frauduleuses avant d'être effectives et contrôler les actes de fraude dans le secteur bancaire, notamment par le ATM pour accorder l'accès aux vrais titulaires des comptes seulement. (33)

Un guichet automatique (ATM) est un guichet bancaire électronique qui permet aux clients d'effectuer des transactions de base sans l'aide d'un représentant de la succursale ou caissier. Toute personne possédant une carte bancaire peut accéder à la plupart des distributeurs automatiques de billets, permettant aux consommateurs d'effectuer rapidement des transactions libre-service, des opérations bancaires courantes comme les dépôts et les retraits mais aussi des d'autres transactions telles que le paiement des factures et les transferts. (34)



Figure 4-1 ATM

Ces machines sont des caisses en charpente de fer stationnées dans les halls bancaires, les supermarchés, les sites de loisirs et des emplacements vitaux pour un accès facile aux comptes d'utilisateurs pendant et après les horaires de travail des banques.

4.2 Interface graphique

Les guichets automatiques sont de plus en plus utilisés récemment. L'analyse des comportements des utilisateurs permet de mieux comprendre leur utilisation. Nous avons choisi de créer une interface graphique, pour simuler la détection de fraude sur les guichets automatiques en temps réel. Cette interface graphique implémente les fonctionnalités de base d'un guichet automatique tels que la vérification de l'identité, le retrait et la consultation de solde. L'utilisation de cette

interface graphique a été possible en créant une mini-bdd des utilisateurs, ainsi que l'implémentation du contrôle de fraude.

4.2.1 Modélisation d'un ATM simplifié avec contrôle de fraude

4.2.1.1 Diagramme de séquence

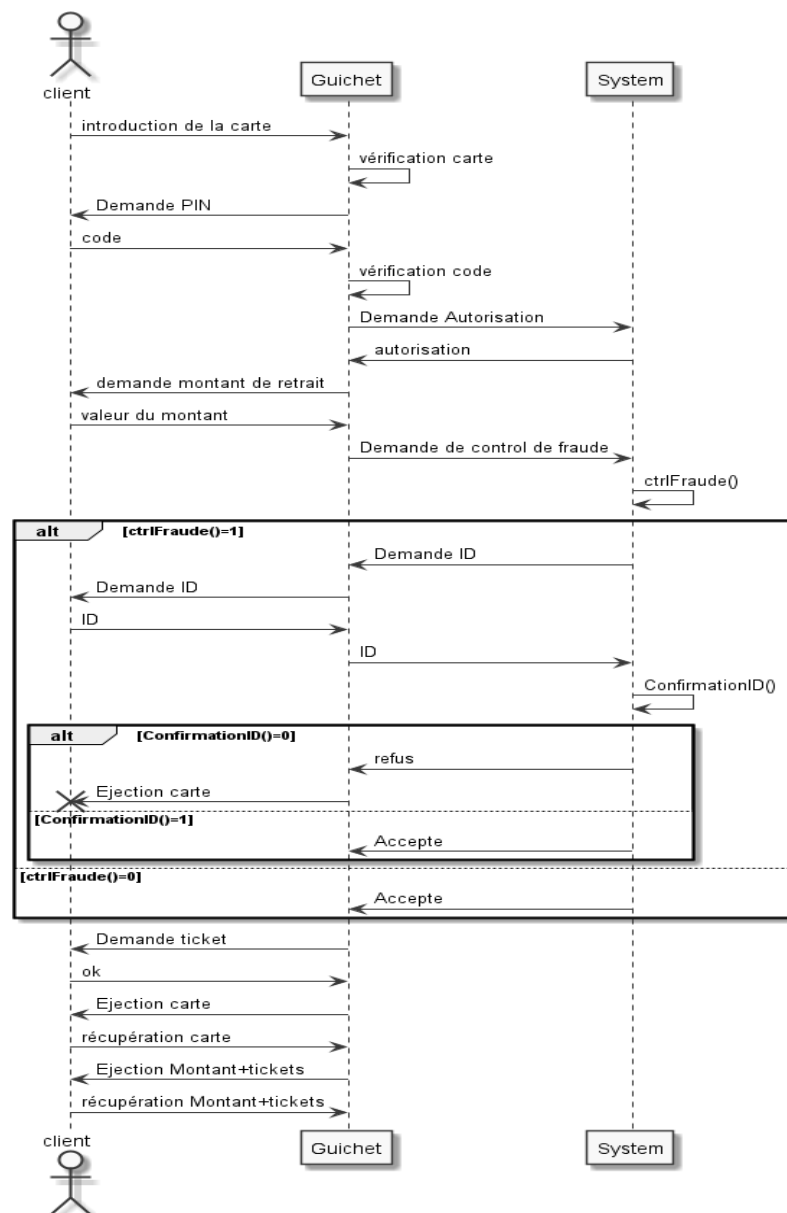


Figure 4-2 Diagramme de séquence

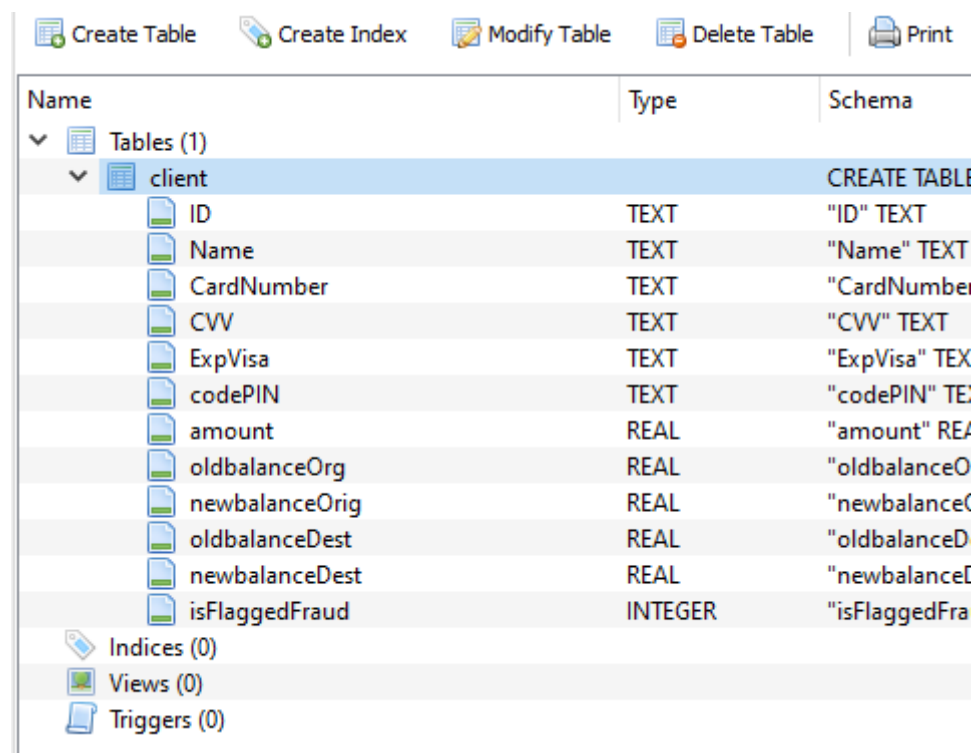
Le client remplit le formulaire de connexion (introduit sa carte bancaire), le guichet vérifie alors la validité des données (de la carte) et demande le code au client. Si le code est correct, il envoie une demande d'autorisation de prélèvement au groupement de banques(système). Ce dernier renvoie l'autorisation afin de prélever le montant demander.

Le client saisit le montant à retirer et enfin le guichet demande au système de détecter les fraudes.

4.2.2 La base de données utilisée

Nous avons construit notre base de données en se basant sur un dataset qui a des champs non anonymisés. Notre base de données contient une seule table qui contient des enregistrements, chaque enregistrement constitue une transaction.

Pour la constitution de la base de données, nous avons combiné des features du DataSet avec des informations bancaires de clients imaginaires (Name, CardNumber CVV, ExpVisa, codePin) puis on a ajouté l'identifiant du client qui va être utilisé pour le contrôle de fraude.



Name	Type	Schema
Tables (1)		
client		CREATE TABLE
ID	TEXT	"ID" TEXT
Name	TEXT	"Name" TEXT
CardNumber	TEXT	"CardNumber
CVV	TEXT	"CVV" TEXT
ExpVisa	TEXT	"ExpVisa" TEX
codePIN	TEXT	"codePIN" TE
amount	REAL	"amount" RE
oldbalanceOrg	REAL	"oldbalanceO
newbalanceOrg	REAL	"newbalanceC
oldbalanceDest	REAL	"oldbalanceD
newbalanceDest	REAL	"newbalanceI
isFlaggedFraud	INTEGER	"isFlaggedFra
Indices (0)		
Views (0)		
Triggers (0)		

Figure 4-3 table utilisée

Les données dans le dataset ont été divisées en deux: les données d'entraînement et les données de test qui sont stockées dans la base de données.

Database Structure Browse Data Edit Pragmas Execute SQL									
Table: client									
	ID	Name	CardNumber	CVV	ExpVisa	codePIN	amount	oldbalanceOrg	newbala
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	CC25034	Jayden Miller	43800689160...	323	10/20	8434	9839.64	170136.0	160296.1
2	CC59625	Gabriel Martinez	48420395966...	952	05/24	2943	1864.28	21249.0	19384.72
3	CC69314	Landon Collins	41028828525...	929	01/25	5353	11668.14	41554.0	29885.86
4	CC67036	Jes Johnson	45551893782...	367	02/24	6462	7817.71	53860.0	46042.29
5	CC25597	Matthew Wilson	40097266088...	439	07/22	8741	7107.77	183195.0	176087.2
6	CC57469	Jackson Young	44786059560...	172	06/21	5441	7861.64	176087.23	168225.9
7	CC15926	Hunter Martinez	44864734765...	185	05/24	6884	4024.36	2671.0	0.0
8	CC31951	Julia Lewis	45653660578...	239	04/21	0096	5337.77	41720.0	36382.22
9	CC32388	Emma Scott	41984244577...	972	11/20	3738	9644.94	4465.0	0.0
10	CC48802	Anthony Jones	45905934722...	305	01/23	9227	3099.97	20771.0	17671.00
11	CC88738	Matthew King	49758543073...	580	04/25	6784	363378.75	363378.75	0.0
12	CC56494	Jayden Lewis	40095366100...	428	11/23	2938	363378.75	363378.75	0.0
13	CC55858	Sophia Rodrig...	49040287975...	267	04/25	5439	655676.97	655676.97	0.0
14	CC64296	Gabriel Edwards	45117814974...	417	12/21	4064	655676.97	655676.97	0.0
15	CC96605	Anna Phillips	49793599631...	712	08/21	0219	56745.14	56745.14	0.0
16	CC46367	Jes Yound	47446374795...	230	11/23	6501	56745.14	56745.14	0.0

Figure 4-4 données du test

4.2.3 Les fonctions utilisées pour la détection de la fraude

Pour détecter les fraudes, nous nous sommes basés sur les techniques d'apprentissage automatique. Nous avons choisi la méthode de régression logistique pour cette application.

```
from sklearn.linear_model import LogisticRegression
logmodel = LogisticRegression().fit(X_train, y_train)
```

Figure 4-5 le modèle du régression logistique

En utilisant ce modèle, nous construisons une fonction `detectionFraude()` qui prend comme paramètres les *features* que nous avons utilisé pour l'implémentation et aussi la phase d'entraînement.

```
import numpy as np
def detectionFraude(a,b,c,d,e,f):
    x=np.array([a,b,c,d,e,f]).reshape(1,6)
    return int(logmodel.predict(x))
```

Figure 4-6 fonction `detectionFraude()`

La fonction **detectionFraude()** est utilisée donc dans le control de fraudes avec la fonctions **CtrlFraude()** :

```

def CtrlFraude():

    conn = sqlite3.connect('Database.db')
    c = conn.cursor()
    find_user=("SELECT * FROM client WHERE Name=? AND CardNumber=? AND CVV=? AND ExpVisa=? AND codePIN=?")
    c.execute(find_user,[(name),(nbcard),(cvv),(exp),(password)])
    r=c.fetchall()
    if r:
        for i in r:
            f=detectionFraude(i[6],i[7],i[8],i[9],i[10],i[11])
            if f==0:
                top1.configure(background="#2d2a29")
                frame_login2=LabelFrame(top1,padx=10,pady=10,bg="gray")
                frame_login2.pack(padx=50 , pady=10)
                label_login1=Label(frame_login2,text="Demande accepter,transaction complète,veuillez prendre votre ticket ",width=
                label_login1.pack(pady=30)
            else:
                label_login1=Label(frame_login2,text="Saisissez votre ID : ",width=20,font=("Arial",21),fg="white",bg="#E94040")
                label_login1.pack(pady=30)
                e_loginid=Entry(frame_login2,width=20,font=("Arial",21))
                e_loginid.pack(pady=30,ipady=8)
                buttonOK=Button(frame_login2,text="Confirmer",width=15,command=ConfirmationID)
                buttonOK.pack()

```

Figure 4-7 fonction CtrlFraude()

Alors si la fonction **CtrlFraude()** détecte une fraude elle fait une demande d'identifiant pour confirmer l'identité au client.

La confirmation de l'ID se fait via la fonction **ConfirmationID()** :

```

def ConfirmationID():
    if entryid.get()==i[0]:
        top1.configure(background="#2d2a29")
        frame_login2=LabelFrame(top1,padx=10,pady=10,bg="gray")
        frame_login2.pack(padx=50 , pady=10)
        label_login1=Label(frame_login2,text="Demande accepter,transaction complète,veuillez prendre votre ticket ",width=
        label_login1.pack(pady=30)
    else:
        label_login1=Label(frame_login2,text="le système rencontre des problèmes!",width=40,font=("Arial",16),fg="white",bg=
        label_login1.pack(pady=10)
        label_login1=Label(frame_login2,text="transaction bloquée",width=40,font=("Arial",16),fg="white",bg="#E94040")
        label_login1.pack(pady=10)

```

Figure 4-8 fonction confirmationID()

Si l'ID n'est pas le même que celui dans la base de données alors La machine n'éjecte pas la carte et arrête la transaction

Sinon le distributeur demande au client s'il désire un ticket. Après la réponse du client, la carte est éjectée et récupérée par le client, l'argent est alors délivré. (Ainsi que le ticket), Le client récupère enfin l'argent et son ticket.

Et si la fonction **CtrlFraude()** ne détecte pas une fraude alors Le client complète la transaction bancaire comme d'habitude.

4.2.4 Fonctionnement

Interface principale de guichet

Dans cette interface, le client saisit les données nécessaires pour une transaction bancaire.



The screenshot displays a banking interface with a dark background. At the top, a grey header box contains the text "Bienvenue" and a smaller instruction: "pour réaliser une transactions veuillez remplir le formulaire suivant et cliquez sur le button au dessous". Below this is a form with several input fields. The first field is labeled "le Nom Complet" and contains the text "Jayden Lewis". The second field is labeled "Numéro sur la carte" and contains the number "4009536610056530". The third field is labeled "Date d'Exp" and contains "11/23". The fourth field is labeled "Champ CVV" and contains "428". Below these fields is a field labeled "Votre code PIN" containing the number "2938". At the bottom of the form is a button with an icon of a hand holding a card. A callout box with an orange border points to this button, containing the text "Bouton pour se connecter".

Figure 4-9 interface principale

Lorsque vous saisissez toutes les informations demandées et vous cliquez sur le bouton, le système vérifie que les données sont correctes.

➤ **Les données sont correctes :**

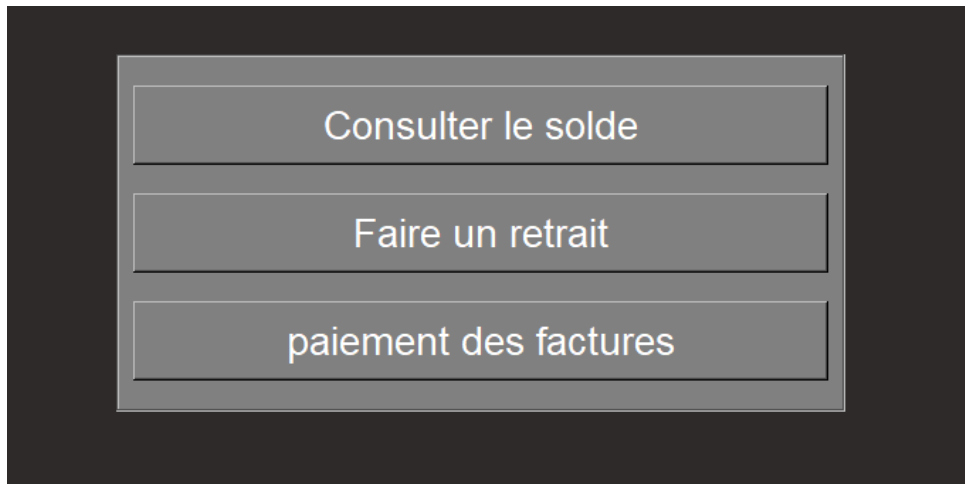


Figure 4-10 les choix

Cette interface vous permet de choisir la transaction bancaire que vous souhaitez. Si vous voulez voir votre solde, retirer d'Argent ou bien payer vos factures.

Pour le retrait :

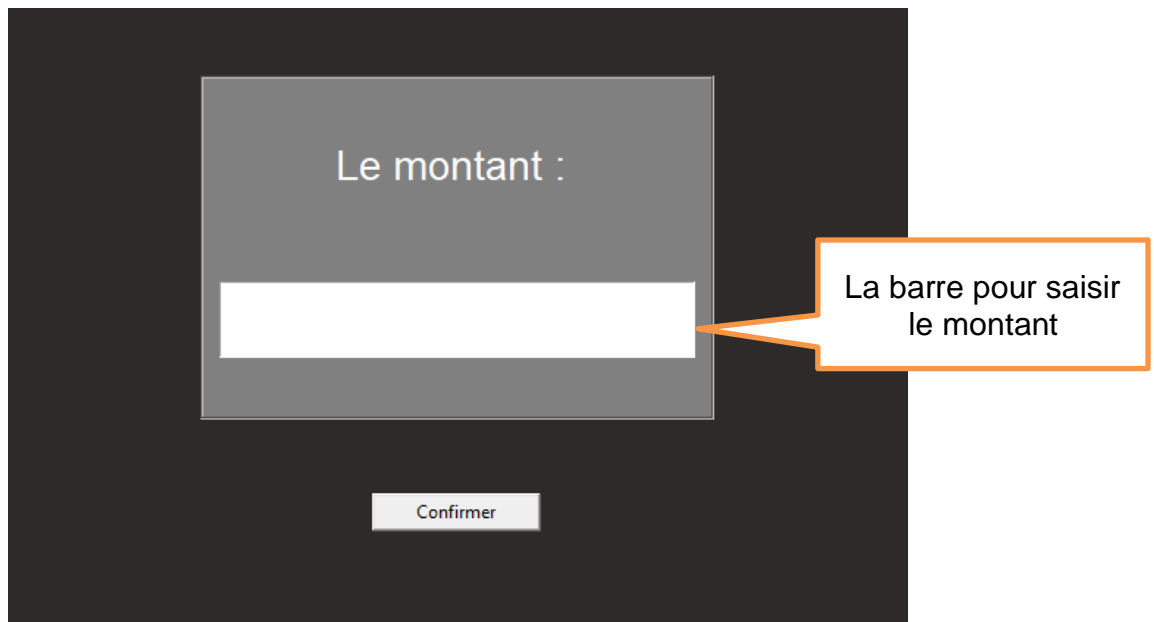


Figure 4-11 interface pour faire retrait

Après avoir saisi le montant, le système appelle la fonction **ctrlFraude()** qui détecte les fraudes.

La fonction récupère les données des transactions précédentes pour l'entraînement et les nouvelles données pour tester si la transaction est une fraude ou non.

Au cas ou il ne s'agit pas d'une fraude, le système affiche un message indiquant qu'il n'y a pas de problème de transaction.

Au cas où il est une fraude le système demande au client de confirmer son identité via la saisissons du ID à l'espace « Saisissez votre ID »

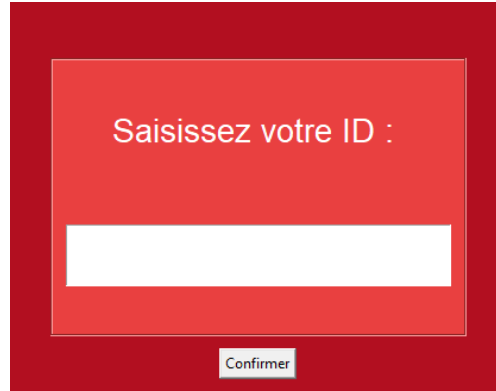


Figure 4-12 saisir ID

Après avoir inséré le code, la fonction **confirmationID()** vient pour s'assurer que le id est correct.

S'il et correct l'utilisateur termine sa transaction.

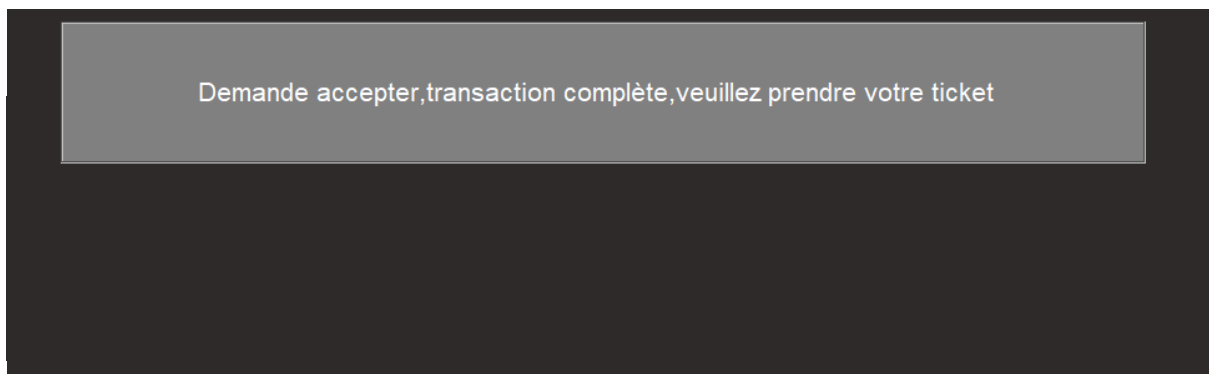


Figure 4-13 ID correct

sinon un message qui indique que la transaction a été bloquée en raison d'un problème d'analyse des données s'affiche.

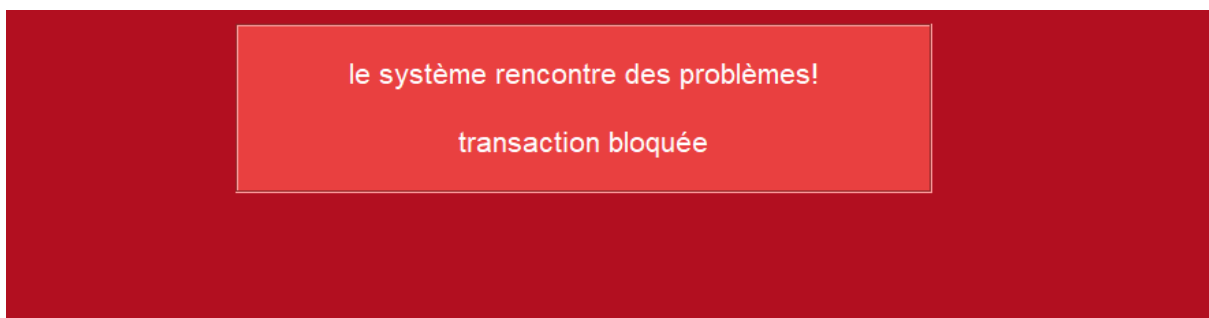


Figure 4-14 ID erronée

➤ **Les données sont incorrectes :**

Si les données saisies sont incorrectes le système affiche le message suivant :

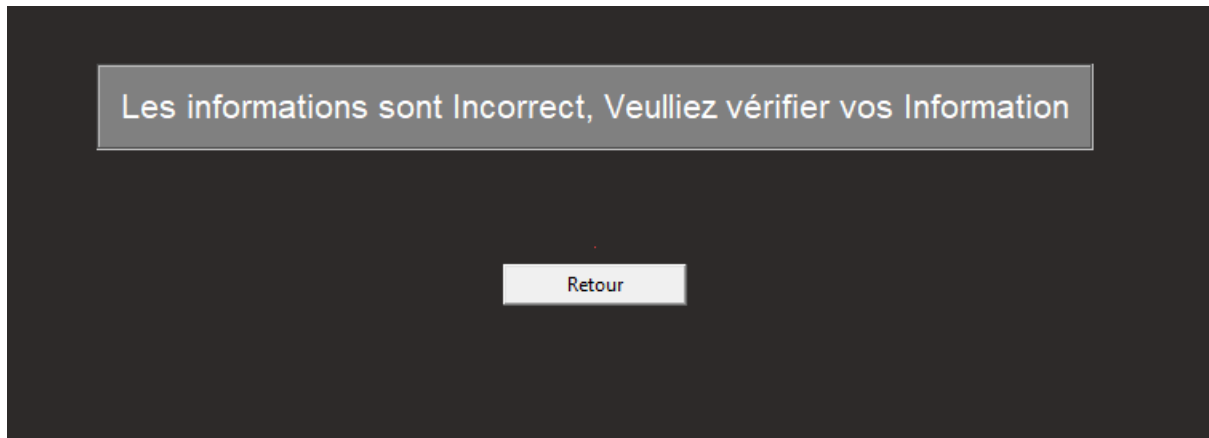


Figure 4-16 données incorrectes

4.3 Conclusion

Les guichets automatiques bancaires (ATM) devraient être conçus avec plus de sécurité et permettre de bloquer les transactions frauduleuses le plus possible. Dans notre interface graphique nous avons essayé de simuler le comportement d'un guichet automatique muni d'un contrôle de fraude qui utilise comme technique de classification la régression logistique.

Conclusion générale

La détection des fraudes est un défi de taille. Pourtant, les transactions frauduleuses sont rares et ne représentent qu'une très petite fraction de l'activité au sein d'une organisation. Néanmoins, un faible pourcentage de l'activité peut rapidement se transformer en des pertes financières importantes sans disposer d'outils pour y faire face. La bonne nouvelle, c'est qu'avec les progrès du Machine Learning, les systèmes peuvent apprendre, s'adapter et découvrir de nouvelles façons de prévenir la fraude. Les spécialistes des données ont réussi à résoudre ce problème grâce au ML et à l'analyse prédictive. (31)

Afin de détecter une transaction frauduleuse nous nous sommes basé sur plusieurs algorithmes tel que la régression logistique, SVM et les arbres de décision en faisant des implémentations avec des données que nous avons explorées, visualisées, modifiées et utilisées.

Les résultats obtenus pour chaque algorithme montrent que les arbres de décision sont le modèle le plus performant pour la détection de la fraude avec une précision de 95%.

Au sein de ce projet nous avons aussi essayé de créer une application qui fonctionne comme un guichet automatique avec un système de détection de fraude basé sur les modèles avec lesquels nous avons travaillé pour vérifier si des transactions correspondent ou non à des fraudes.

Webographie :

- [3]. Phishing (hameçonnage ou filoutage). *Phishing (hameçonnage ou filoutage)*. [Online] 09 18, 2018. [Cited: 05 11, 2020.] <https://www.economie.gouv.fr/dgccrf/Publications/Vie-pratique/Fiches-pratiques/Phishing-hameconnage-ou-filoutage>.
- [4]. Chevillard, Steve. QU'EST CE QUE LE PHISHING ? *ASTUCES & AIDE INFORMATIQUE*. [Online] 03 24, 2016. <https://www.astuces-aide-informatique.info/215/definition-phishing>.
- [5]. C'est quoi le skimming ? *panoptinet*. [Online] 2014. <https://www.panoptinet.com/cybersecurite-pratique/cest-quoi-le-skimming.html>.
- [6]. D'halluin, Florent. MACHINE LEARNING ET LUTTE CONTRE LA FRAUDE. *netheosleblog*. [Online] 2019. <https://www.netheos.com/blog/machine-learning-et-lutte-contre-la-fraude>.
- [7]. L, Bastien. Machine Learning et Big Data : définition et explications. *le big data*. [Online] 7 6, 2018. <https://www.lebigdata.fr/machine-learning-et-big-data>.
- [9]. benzaki, younes. L'apprentissage supervisé – Machine Learning. *Mr mint*. [Online] 2017. <https://mrmint.fr/apprentissage-supervise-machine-learning>.
- [10]. BENZAKI, Younes. L'apprentissage non supervisé – Machine Learning. *https://mrmint.fr*. [Online] 2016-2017 . <https://mrmint.fr/lapprentissage-non-supervise-machine-learning>.
- [11]. Éditorial, Le DAP Comité. *data analytics post*. [Online] <https://dataanalyticspost.com/Lexique/apprentissage-par-renforcement/>.
- [12]. Waseem, Mohammad. Comment implémenter la classification dans l'apprentissage automatique? *edureka*. [Online] 12 4, 2019. <https://www.edureka.co/blog/classification-in-machine-learning/#classification>.
- [13]. Sidana, Mandy. Intro to types of classification algorithms in Machine Learning. *medium*. [Online] 2 28, 2017. <https://medium.com/@Mandysidana/machine-learning-types-of-classification-9497bd4f2e14>.
- [15]. *tutorialspoint*. [Online] https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_classification_algorithms_decision_tree.htm.
- [16]. joshi, renuka. Exactitude, précision, rappel et score F1: interprétation des mesures de rendement. *blog*. [Online] 2016. <https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/>.
- [17]. Seguro, Porto. Resampling strategies for imbalanced datasets. *kaggle*. [Online] 2018. <https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets>.

- [18]. How Machine Learning Facilitates Fraud Detection? *maruti techlabs*. [Online] 2020. <https://marutitech.com/machine-learning-fraud-detection/>.
- [19]. navlani, avinash. understanding logistic regression python. *datacamp*. [Online] 2019. <https://www.datacamp.com/community/tutorials/understanding-logistic-regression-python>.
- [20]. stojiljkovic, mirko. Logistic Regression in Python. *real python*. [Online] 2020. <https://realpython.com/logistic-regression-python/>.
- [21]. Support Vector Machine Algorithm. *javatpoint*. [Online] <https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm>.
- [24]. L, Bastien. Python : tout savoir sur le principal langage Big Data et Machine Learning. *le big data*. [Online] 7 11, 2019. <https://www.lebigdata.fr/python-langage-definition>.
- [25]. Anaconda (Python distribution). *wikipedia*. [Online] 2019. [https://fr.wikipedia.org/wiki/Anaconda_\(Python_distribution\)](https://fr.wikipedia.org/wiki/Anaconda_(Python_distribution)).
- [26]. Poulain, Patrick Fuchs & Pierre. 18 Jupyter et ses notebooks. *Cours de Python*. [Online] https://python.sdv.univ-paris-diderot.fr/18_jupyter/.
- [27]. CFAURY. Tkinter. *info*. [Online] 2019. <https://info.blaisepascal.fr/tkinter>.
- [28]. Derfoufi, Younes. Python et les bases de données SQLite3. *python tres facile*. [Online] 2019. <https://www.tresfacile.net/python-et-les-bases-de-donnees-sqlite3/>.
- [29]. Favier, Sandrine. TOP 5 DES MEILLEURS ÉDITEURS DE TEXTE. *pixelsandbytes*. [Online] 2019. <https://pixelsandbytes.fr/5-meilleurs-editeurs-code/>.
- [30]. plantUML. *wikipedia*. [Online] <https://en.wikipedia.org/wiki/PlantUML>.
- [31]. db browser. *info.blaisepascal*. [Online] <https://info.blaisepascal.fr/cpge-db-browser-for-sqlite>.
- [32]. fraude bancaire. *que choisir*. [Online] <https://www.quechoisir.org/actualite-piratage-infographie-vous-et-la-fraude-bancaire-n72243/>.
- [33]. , KOSSIVI KOBAYILI KOSSI. L'intelligence artificielle face à la fraude bancaire. *economie numerique*. [Online] 2019. <http://blog.economie-numerique.net/2019/03/28/lintelligence-artificielle-face-a-la-fraude-bancaire/>.
- [34]. Guichet automatique bancaire (ATM). *le financier*. [Online] 2020. <https://www.lefinancier.fr/votre-argent/services-bancaires/guichet-automatique-bancaire-atm-2>.
- [35]. OBIANG-NDONG, Fred NTOUTOUME. *memoire online*. [Online] <https://www.memoireonline.com/08/11/4751/Scoring-credit-une-application-comparative-de-la-regression-logistique-et-des-reseaux-de-neurone.html>.

- [36]. Brownlee, Jason. Supervised and Unsupervised Machine Learning Algorithms. *machine learning mastery*. [Online] 3 16, 2016. <https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/>.
- [37]. —. Logistic Regression for Machine Learning. *machine learning mastery*. [Online] 4 1, 2016. <https://machinelearningmastery.com/logistic-regression-for-machine-learning/>.
- [38]. pant, Ayush. towards data science. *introduction to logistic regression*. [Online] 1 22, 2019. <https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148>.
- [39]. mishra, avinash. Introduction to Logistic Regression - Sigmoid Function, Code Explanation. *analytics steps*. [Online] 8 21, 2019. <https://www.analyticssteps.com/blogs/introduction-logistic-regression-sigmoid-function-code-explanation>.
- [40]. Andile, Maximilien. Machine Learning : classification à l'aide des arbres de décisions : fonctionnement et application en NodeJS. *M@XCode*. [Online] 2019. <https://maximilienandile.github.io/2016/10/17/Machine-Learning-classification-a-l-aide-des-arbres-de-decisions-fonctionnement-et-application-en-NodeJS/>.
- [41]. ATM écrémage: il est sur le point d'obtenir pire 2020. *Routestofinance*. [Online] 2020. <https://fr.routestofinance.com/atm-skimming-it-s-about-to-get-worse>.
- [42]. Office Word. *cours informatiques gratuit*. [Online] <https://cours-informatique-gratuit.fr/dictionnaire/microsoft-office-word/>.
- [43]. Office Excel. *cours inforamtique gratuit*. [Online] 2020. <https://cours-informatique-gratuit.fr/dictionnaire/office-excel/>.
- [44]. Ailenz. La vraie différence entre Data Science, Machine Learning et Data Mining. *Loïc Moncany*. [Online] <https://loicmoncany.com/la-vraie-difference-entre-data-science-machine-learning-et-data-mining/>.
- [45]. Savy, Raphaël. Comment le Machine Learning peut permettre de lutter contre la fraude. [Online] 30 Avril 2019. https://www.decideo.fr/Comment-le-Machine-Learning-peut-permettre-de-lutter-contre-la-fraude_a11039.html.

Bibliographie :

[1]. Pozzolo, Andrea Dal. Adaptive Machine Learning for Credit Card Fraud Detection. Bruxelles : s.n., 2015.

[2]. Kavila, Lakshmi Selvani Deppthi. Machine learning for credit card fraud detection system. anil nerrukonda : Research india publications, 2018. 09734562.

[8]. Bishop, Christopher M. Pattern Recognition and Machine Learning. Cambridge : s.n., 2006. 978-0387-31073-2.

[14]. Shai Shalev-Shwartz, Shai Ben-David. Understanding Machine Learning. cambridge : s.n., 2014. 978-1-107-05713-5.

[22]. PETTIER, Christophe COMBELLES Gabriel. PYTHON : le développement autrement Tour d'horizon de la plateforme. s.l. : Creative Commons Attribution-Share, mai 2011.

[23]. Saint-Cirgue, Guillaume. Apprendre le ml en une semaine. 2019.