



ECOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE ET D'ANALYSE DES
SYSTÈMES - RABAT

RAPPORT DE STAGE DE FIN DE DEUXIÈME ANNÉE
Filière Génie Logiciel

Création d'une Application Web pour la Détection des Transactions Frauduleuses

Réalisé par :

Miloud ECHARKI

Encadré par :

Pr :Reda MASTOURI

Membres de Jury :

• ****

Année universitaire : 2021/2022



Remerciements :

Je voudrai tout d'abord adresser toutes mes gratitudees au professeur Reda MASTOURI, pour sa confiance, sa disponibilit   et surtout cette opportunit   pour bien ma  triser comment traiter un projet de puis le recensement des besoins jusqu'   l'impl  mentation de la solution.

Je d  sirai aussi remercier tous ceux qui ont contribu      la r  ussite de ce projet, par leurs conseils et leurs connaissances qu'ils m'ont partag  es.



Résumé

Le présent document est le fruit de mon travail dans le cadre de projet de stage de fin de 2ème année. Ce projet avait comme but la Création d'une application web pour détecter les transactions frauduleuses. L'idée derrière ce projet est de fournir aux entreprises une plate-forme d'échange, de conseil et de détection des transactions frauduleuses basée sur un modèle d'apprentissage automatique (ML). Donner une probabilité précise de l'exactitude de la transaction est de savoir si elle est frauduleuse ou non. Je vous présenterai donc tout au long de ce rapport, les étapes que j'ai suivies ainsi que les outils que j'ai utilisés pour réaliser ce projet de stage de fin d'année.

Mots clés :

ML, Transaction, Probabilité, Api ,Detection , ML life cycle, Training, Model ,Dataset, Documentation.

Table des matières

Remerciements	2
Résumé	3
Liste des abréviations	3
Introduction générale	1
1 Cadre général du projet	3
1.1 Qu'est-ce que RIDE et sa mission	3
1.2 Qu'est-ce que Imagine	4
1.3 Le but de Imagine	6
1.4 Staffing	7
1.5 Problématique et objectifs	8
1.6 Machine Learning	8
1.7 Planification et démarche de travail	10
1.7.1 Les acteurs du projet	11
1.7.2 Les sprints du projet	11
1.7.3 Mode de travail	12
1.8 Conclusion	12
2 State of the Art	13
2.1 Types de techniques de détection de fraude	13
2.1.1 Techniques d'analyse de données statistiques	14
2.1.2 AI-based techniques	15
2.2 ML et la détection des fraudes	16
2.3 Le ML et les systèmes traditionnels de détection des fraudes	17
2.4 Conclusion	18
3 Dataset & Model	19
3.1 Model	20
3.2 Entraîner le modèle	21
3.2.1 Documentation a l'aide de Swagger UI	23
3.3 MLflow Tracking	24
3.4 Conclusion	26
4 Mise en oeuvre et réalisation	27
4.1 Architecture de l'application	27
4.2 Outils de travail	28
4.2.1 Framwork FastAPI	28
4.2.2 Swagger UI	29
4.2.3 Uvicorn server	30
4.2.4 Docker	31
4.2.5 Jupyter	31
4.2.6 MLflow	32
4.2.7 ReactJS framework	32
4.2.8 Le langage HTML	33
4.2.9 Storybook	33
4.2.10 Le langage CSS	34
4.2.11 Le langage javascript	34

4.3	Réalisation de l'application	35
4.4	Conclusion	37
Conclusion et perspective		38
Bibliographie		38

Table des figures

1.1	Logo Ride	4
1.2	Imagine Logo	5
1.3	Imagine platform	5
1.4	Staffing	7
1.5	Category of ML tasks	9
1.6	Life cycle d'un projet ML	10
1.7	Plan de travail prévu du projet	11
2.1	Types de techniques de détection.	13
2.2	Traitement d'une transaction ML.	17
3.1	Kaggle.	19
3.2	Gradient Boosted Trees for Regression	20
3.3	Random layout.	23
3.4	Endpoints de notre backend.	23
3.5	Entraîner le Modèle.	24
3.6	Efficacité de notre Modèle	24
3.7	Resultat d'une prediction.	24
3.8	Expériences sur MLflow	26
3.9	Détails d'une Expérience	26
4.1	Architecture de l'application	28
4.2	Logo FastAPI	29
4.3	Logo Swagger UI	30
4.4	Logo Uvicorn server	30
4.5	Logo Docker	31
4.6	Logo Jupyter	32
4.7	Logo MLflow	32
4.8	Logo ReactJS	33
4.9	Html	33
4.10	Logo Storybook	34
4.11	Css	34
4.12	Javascript	35
4.13	Page D'accueil.	35
4.14	Footer.	36
4.15	Charger un fichier csv	36
4.16	Résultat de prédiction.	37

Liste des tableaux

1.1 Les Sprints du projet 12

Liste des abréviations

- **ML** :Machine Learning
- **CDC** : Cahier des charges.
- **DCU** : Diagramme des cas d'utilisation.
- **ENSIAS** : École nationale supérieure d'informatique et d'analyse des systèmes.
- **Imagine** : Intelligent Moroccan Assembly for Governance and Ingenious Novelty in Entrepreneurship
- **API** :Application Programming Interface
- **MLOps** : Machine Learning Operations
- **Ride** : Research and innovation for development and entrepreneurship

Introduction générale

Open banking, crypto-monnaies, intelligence artificielle... toutes ces technologies changent la façon dont les gens effectuent leurs opérations bancaires. Mais ces progrès offrent aussi des opportunités aux criminels. Les banques et les grandes entreprises les plus prospères et les plus rentables de demain seront celles qui déploient le plus d'efforts en matière de prévention du crime financier et de défense aujourd'hui. L'ampleur du défi n'est peut-être pas bien connue, mais le fait qu'il constitue une dépense majeure pour les banques est une évidence. L'intelligence artificielle (IA) recèle un énorme potentiel pour les banques, non seulement dans la lutte pour fidéliser leur clientèle, mais aussi en tant qu'arme contre la criminalité financière. L'objectif de ce stage est de chercher là où les technologies permettant aux entreprises de recueillir des informations de plus en plus précises sur les désirs et les comportements des clients. La technologie aidera également les banques et les grandes entreprises à établir des profils de fraude et à suivre les criminels par le biais de réseaux financiers. Dans ce rapport, vous trouverez une explication détaillée du travail effectué pour la réalisation de cette application

- Le premier chapitre est consacré à la présentation de la problématique et le cadre général du projet.
- Le deuxième chapitre se focalise sur l'analyse des besoins et la conception du projet en expliquant la méthode de étapes que nous avons adoptées. et la concep-

tion du projet en expliquant la méthode de conception utilisée.

- Le dernier chapitre aborde la phase du développement et mise en oeuvre, avec une présentation de l'application.

Chapitre 1

Cadre général du projet

Dans ce chapitre j'entame le projet dans son cadre général : présentation de l'organisme d'accueil Research and innovation for development and entrepreneurship (Ride) , présentation de l'idée du projet, présentation du contexte du projet, de la problématique et des objectifs fixés, et de la démarche de la réalisation adoptée.

1.1 Qu'est-ce que RIDE et sa mission

Ride est synonyme de (research and innovation for development and entrepreneurship) recherche et d'innovation pour le développement et l'entrepreneuriat. Depuis 2013, les dirigeants de RIDE ont travaillé sans relâche pour promouvoir sa recherche appliquée de manière agile et efficace, en adhérant à une chaîne de valeur pré-mondiale basée sur :

- • Une analyse approfondie des besoins territoriaux, accompagnée d'un plan bien défini et d'une évaluation juridique et technico-commerciale préliminaire.
- • Le développement de brevets, de marques et de licences pour protéger nos idées innovantes, ainsi que la création d'un prototype qui répond aux besoins du territoire.
- • Proposer notre solution sur le marché de deux manières :

- o Le transfert et la vente du prototype ou de l'idée (brevets, marques et licences) à une industrie qui le financera et le fabriquera.
- o La création de notre propre entreprise (Startup)

Afin de mener à bien ses missions majeures et de se concentrer sur ses objectifs, nos centres CREDIT RIDE disposent :

- • Un capital humain qualifié et de haut niveau, qui comprend des experts, des chercheurs et des professionnels travaillant dans divers domaines innovants.
- • Partenaires académiques et industriels nationaux et internationaux
- • Une gestion budgétaire et financière bien maîtrisée de tous nos services à différents stades de production ;
- • Une production scientifique de haut niveau qui répond à diverses exigences internationales



FIGURE 1.1 – Logo Ride

1.2 Qu'est-ce que Imagine

une plate-forme de gestion intelligente et de gouvernance administrative pour la recherche et l'innovation entrepreneuriale. Il facilite les partenariats et les ras-



FIGURE 1.2 – Imagine Logo

sement public-privé, ainsi que les chercheurs, les organisations non gouvernementales (ONG), les fondations et les entreprises.

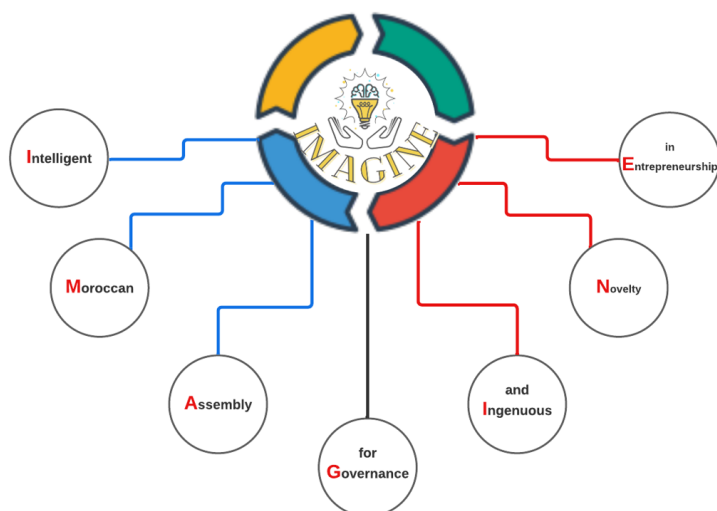


FIGURE 1.3 – Imagine platform

La plateforme ajoute plusieurs nouvelles fonctionnalités :

- • Permettre la communication et la collaboration entre les PPP, les chercheurs, les organisations non gouvernementales (ONG), les fondations et les entreprises territoriales.
- • Rassembler les personnes qui s'intéressent aux questions territoriales liées à un thème spécifique.

- • Viser la sécurité de l'information et le professionnalisme des services car il est limité au territoire, ce qui permet de mieux gérer les données et de mieux contrôler les services, de les accélérer et de les rendre plus efficaces.
- • S'assurer que les demandes, les services et les idées publiés sur cette plateforme font l'objet d'un suivi.
- • Utiliser la même base de données et les mêmes informations que celles qui sont alimentées et utilisées par différentes parties du territoire.
- • Créer un espace de recrutement intelligent qui encourage les promoteurs de projets et les prestataires de services à soumettre des appels d'offres.
- • Créer des contrats signés numériquement.

1.3 Le but de Imagine

Voici les principales fonctionnalités qui seront incluses dans notre plateforme IMAGINE :

- • Fournir une solution technique qui assure un recrutement intelligent des talents locaux en fonction de leurs expériences et de leurs qualités.
- • Permettre la soumission de problèmes ou de solutions à un thème.
- • Renseigner (approvisionnement, collaboration, formation, etc.)
- • S'assurer que les différents utilisateurs communiquent entre eux
- • Administration des paiements sécurisée (dons, abonnements, etc.)
- • Intégration de services (digitalisation, etc.)
- • L'architecture du système de notification (par la plateforme, les emails, etc.)
- • Créer un espace où les gens peuvent partager des articles et des publications (recherche scientifique, etc.)

- • Les contrats et les factures doivent être gérés (intégration des signatures électroniques)
- • Les organisations et les entreprises ont eu accès à un tableau de bord de gestion et de planification.
- • La prise de rendez-vous en ligne est disponible.

1.4 Staffing

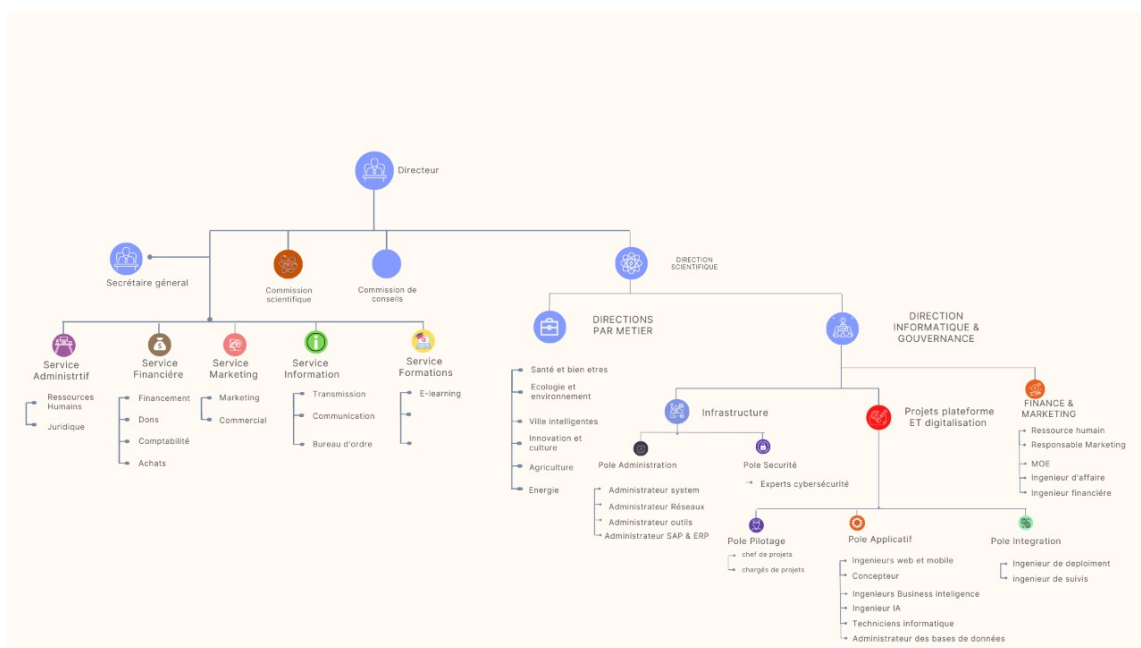


FIGURE 1.4 – Staffing

1.5 Problématique et objectifs

Les services financiers numériques et le e-commerce simplifient la vie des gens, leur permettant de commander des biens ou de gérer leurs finances depuis n'importe où. De plus, avec les mesures de confinement dues au coronavirus, l'importance des services en ligne s'est encore accrue. Open banking, crypto-monnaies, intelligence artificielle... changent la façon dont les gens effectuent leurs opérations bancaires. Mais ces progrès offrent aussi des opportunités aux criminels. pour effectuer des transactions frauduleuse L'objectif est de chercher là où les technologies permettant aux entreprises de recueillir des informations de plus en plus précises sur les désirs et les comportements des clients. La technologie aidera également les banques et les grandes entreprises à établir des profils de fraude et à suivre les criminels par le biais de réseaux financiers.

1.6 Machine Learning

L'apprentissage automatique (ML) est un ensemble d'algorithmes informatiques et de méthodes statistiques qui peut s'améliorer progressivement par l'expérience et par l'utilisation des données. C'est vrai considéré comme faisant partie de l'intelligence artificielle. Les algorithmes d'apprentissage automatique construisent un modèle basé sur des exemples de données, appelées données d'entraînement, afin d'apprendre des modèles à partir de ces données et faire des prédictions ou des décisions en fonction des modèles qu'il a appris.

Les approches d'apprentissage automatique s'inscrivent principalement dans trois catégories :

— — L'apprentissage supervisé est l'endroit où vous avez des variables d'entrée

(x) et une variable de sortie (Y) et vous utilisez un algorithme pour apprendre le mappage fonction de l'entrée à la sortie, l'objectif est d'approximer la fonction de mappage si bien que lorsque vous avez de nouvelles données d'entrée (x) que vous pouvez prédire les variables de sortie (Y) pour ces données.

- L'apprentissage non supervisé est l'endroit où vous n'avez que des données d'entrée (X) et non variables de sortie correspondantes, l'objectif de l'apprentissage non supervisé est de modéliser la structure ou la distribution sous-jacente dans les données afin de en savoir plus sur les données.
- L'apprentissage par renforcement consiste à prendre des mesures appropriées pour maximiser récompense dans une situation particulière. C'est surtout pour trouver le meilleur possible comportement ou chemin qu'il devrait prendre dans une situation spécifique. le renforcement l'agent décide quoi faire pour effectuer la tâche donnée. En l'absence d'un ensemble de données de formation, il est tenu d'apprendre de son expérience.

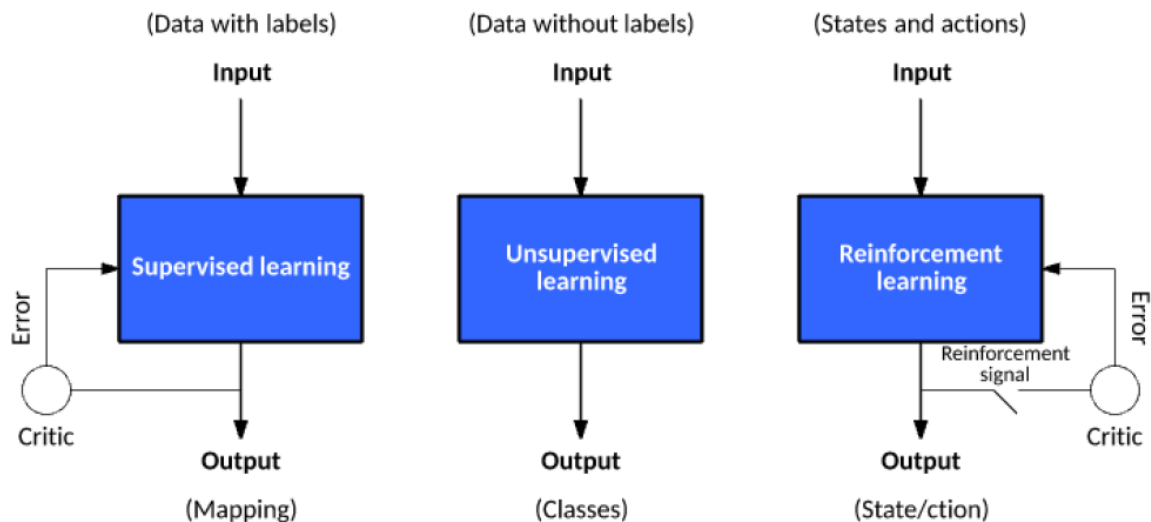


FIGURE 1.5 – Category of ML tasks

1.7 Planification et démarche de travail

Un projet d'apprentissage automatique typique passe par un cycle continu à partir de la collecte de données, le prétraitement et le nettoyage, la formation et le déploiement de modèles, ce le cycle de vie peut être représenté sous la forme d'un flux multi-composants, où chaque flux consécutif affecte le reste du flux.

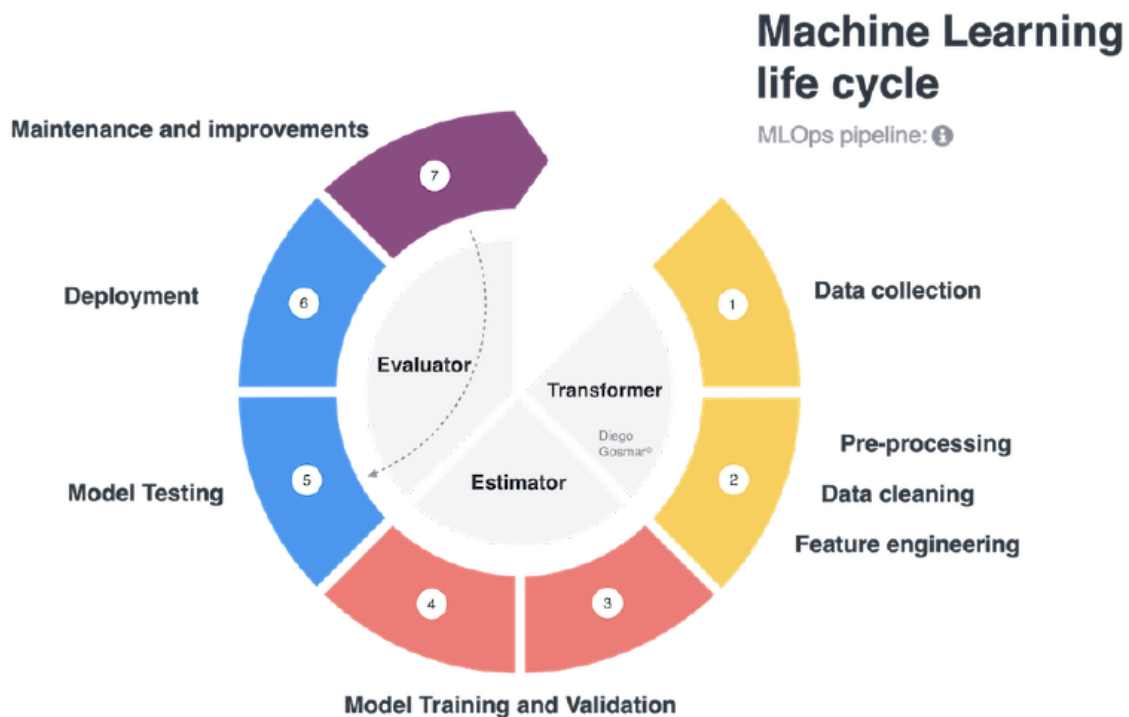


FIGURE 1.6 – Life cycle d'un projet ML

Pour planifier notre projet et suivre l'avancement, nous avons utilisé le diagramme de GANTT. Ceci La méthode visuelle est efficace puisqu'elle représente graphiquement l'avancement du projet. Le diagramme de GANTT a deux objectifs : planifier de manière optimale et communiquer selon l'horaire établi. Il nous permet également de lister les différentes tâches à effectuer comme date de début et de fin de cette tâche.

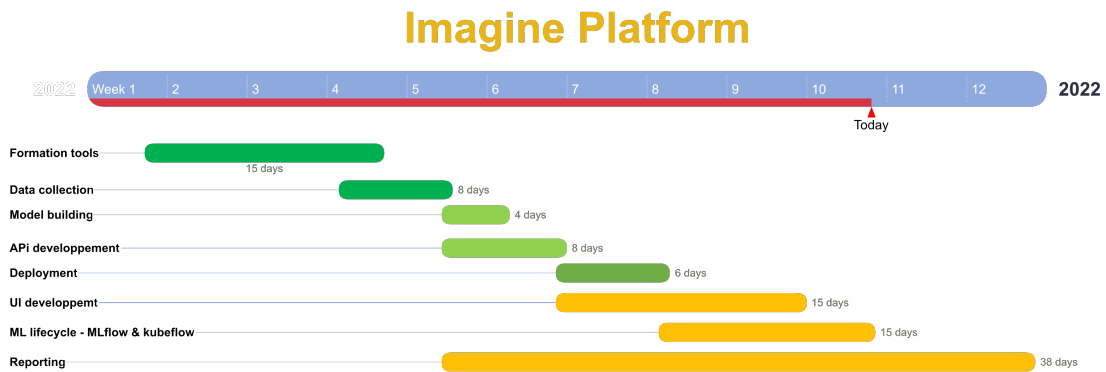


FIGURE 1.7 – Plan de travail prévu du projet

1.7.1 Les acteurs du projet

- **L'équipe du travail** : La tâche de la conception et de développement de l'application Back-end et front-end est assigné à moi.
- **Le product owner** : Qui est ici **RIDE** représenté par M. **HABBANI** Ahmed. Il tient le rôle de définir les fonctionnalités et de s'assurer du bon fonctionnement du projet.
- **Le SCRUM master** : Mon encadrant, MR.**MASTOURI** Reda a tenu ce rôle, en veillant sur la réalisation du projet.

1.7.2 Les sprints du projet

Pour le bon déroulement du projet, le travail sera découpé en sprint. À la fin de chaque sprint, on aura un livrable qui sera examiné par l'encadrant afin de planifier les modifications et les évolutions à effectuer dans le sprint suivant. La planification du projet selon les sprints et comme suit :

Numéro	Durée	Description
1	2 semaines	Formation sur les outils et les technologies de travaux.
2	2 semaines	La recherche du modèle convenables a nos besoins
3	3 semaines	Développement de L api déploiement du modèle
4	2 semaines	Création de UI reactJS

TABLE 1.1 – Les Sprints du projet

1.7.3 Mode de travail

Afin de réaliser le projet, on doit définir un mode de travail convenable. Le mode de travail que je propose est le mode agile en utilisant SCRUM. Je cherche à avoir une communication efficace avec le client pour le satisfaire en lui livrant fréquemment un état d'avancement tangible. D'ailleurs, le temps est un facteur important dans ce projet. De ce fait, il faut sentir l'avancement quotidien du projet. En addition, il faut faire une attention continue à l'excellence technique, à une meilleure conception et produire uniquement ce qui est nécessaire dans les meilleurs délais.

1.8 Conclusion

Dans ce chapitre introductif, j'ai pu décrire le contexte général du projet, et déterminer son objectif principal. En addition, la présentation de l'organisation RIDE. et la place de ML dans le problème

Chapitre 2

State of the Art

2.1 Types de techniques de détection de fraude

La détection des fraudes implique généralement des techniques basées sur l'analyse des données. Ces techniques sont généralement classées comme des techniques d'analyse de données statistiques et des techniques basées sur l'intelligence artificielle ou l'IA.

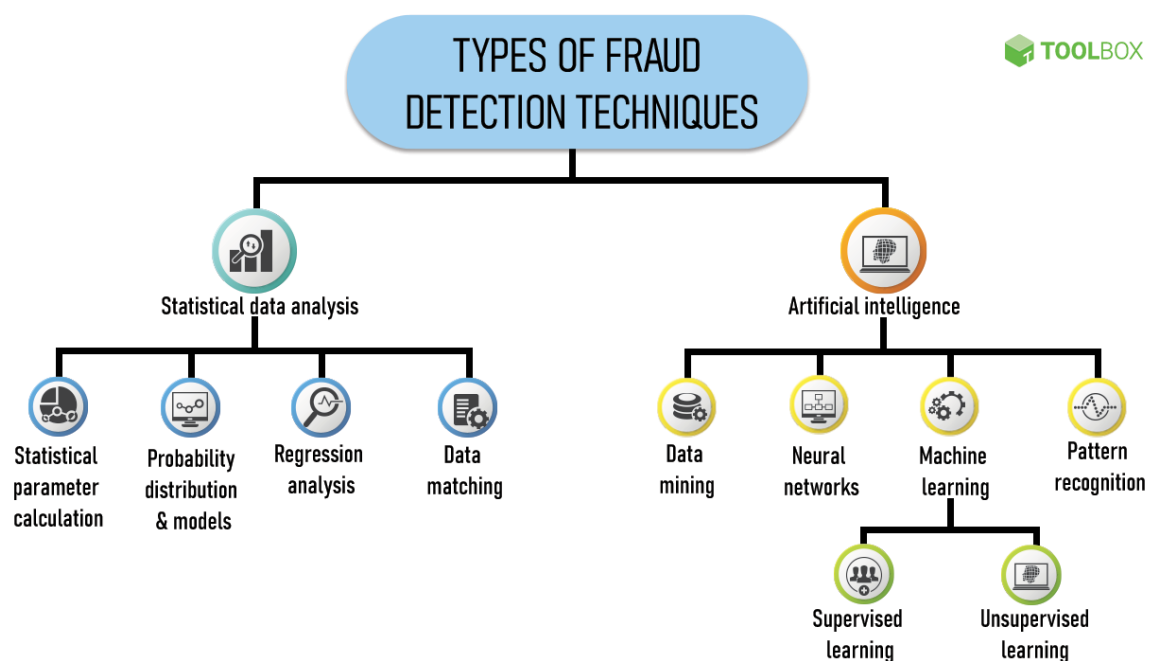


FIGURE 2.1 – Types de techniques de détection.

2.1.1 Techniques d'analyse de données statistiques

L'analyse des données statistiques pour la détection de la fraude effectue diverses opérations statistiques telles que la collecte de données sur la fraude, la détection de la fraude et la validation de la fraude en menant des enquêtes détaillées. Ces techniques sont subdivisées en les types suivants :

1. Statistical parameter calculation

statistiques Le calcul des paramètres statistiques fait référence au calcul de divers paramètres statistiques tels que les moyennes, les quantiles, les mesures de performance et les distributions de probabilité pour les données liées à la fraude collectées au cours du processus de saisie des données.

2. Regression analysis

L'analyse de régression vous permet d'examiner la relation entre deux ou plusieurs variables d'intérêt. Il estime également la relation entre les variables indépendantes et dépendantes. Cela permet de comprendre et d'identifier les relations entre plusieurs variables de fraude, ce qui aide davantage à prédire les activités frauduleuses futures. Ces prédictions sont basées sur les modèles d'utilisation des variables de fraude dans un cas d'utilisation potentiellement frauduleux.

3. Probability distributions and models

Dans cette technique, les modèles et les distributions de probabilité de diverses activités frauduleuses commerciales sont cartographiés, soit en termes de différents paramètres, soit en termes de distributions de probabilité.

4. Data matching

L'appariement des données est utilisé pour comparer deux ensembles de données recueillies (c.-à-d. les données sur la fraude). Le processus peut être effectué soit sur la base d'algorithmes, soit sur des boucles programmées. En outre, la correspondance des données est utilisée pour supprimer les enregistrements en double et identifier les liens entre deux ensembles de données à des fins de marketing, de sécurité ou autres.

2.1.2 AI-based techniques

Le déploiement de l'IA pour la prévention de la fraude a aidé les entreprises à améliorer leur sécurité interne et à rationaliser leurs processus métier. Grâce à l'amélioration de l'efficacité, l'IA est devenue une technologie essentielle pour prévenir la fraude dans les institutions financières. Les techniques de détection de la fraude basées sur l'IA comprennent les méthodes suivantes :

1. Data mining

L'exploration de données pour la détection et la prévention de la fraude classe, regroupe et segmente les données et trouve automatiquement des associations et des règles dans les données qui peuvent signifier des modèles intéressants, y compris ceux liés à la fraude.

2. Neural Networks

Les réseaux neuronaux sous détection de fraude effectuent la classification, le regroupement, la généralisation et la prévision des données liées à la fraude qui peuvent être comparées aux conclusions soulevées dans les audits internes ou les documents financiers formels.

3. ML

La détection de la fraude avec l'apprentissage automatique devient possible grâce à la capacité des algorithmes de ML à apprendre des modèles de fraude historiques et à les reconnaître dans les transactions futures. L'apprentissage automatique utilise des méthodes d'apprentissage supervisées ou non supervisées.

4. Pattern recognition

Les algorithmes de reconnaissance de formes détectent des classes approximatives, des clusters ou des modèles de comportement suspect, automatiquement (non supervisés) ou manuellement (supervisés).

2.2 ML et la détection des fraudes

Le concept derrière l'utilisation de ML dans la détection de la fraude est que les transactions frauduleuses ont des caractéristiques spécifiques que les transactions légitimes n'ont pas. Sur la base de cette hypothèse, les algorithmes d'apprentissage automatique détectent les modèles dans les opérations financières et décident si une transaction donnée est légitime. Les algorithmes de ML peuvent repérer des modèles qui semblent sans rapport ou qui passent inaperçus pour un humain. En explorant et en étudiant des tonnes de cas de comportement frauduleux, les algorithmes de ML déterminent les modèles frauduleux les plus furtifs et s'en souviennent pour toujours.



FIGURE 2.2 – Traitement d’une transaction ML.

2.3 Le ML et les systèmes traditionnels de détection des fraudes

Le modèle traditionnel de détection de la fraude est basé sur un système statique basé sur des règles, également appelé système de production ou système expert. Bien que ces systèmes soient efficaces depuis longtemps, certains de leurs inconvénients majeurs les rendent inadaptés aux environnements numériques modernes. Un système statique basé sur des règles dépend fortement du travail humain. Mais naturellement, les meilleurs analystes coûtent cher. Et leur travail prend du temps. De plus, même les meilleurs experts créent des règles basées sur leurs connaissances, leurs compétences et leur expérience, qui sont toujours limitées. De telles règles peuvent atteindre des tailles énormes et devenir si complexes qu’il est presque impossible pour un étranger de les comprendre en cas de besoin. En outre, la création d’une nouvelle règle et sa mise en œuvre prennent un certain temps lorsqu’elles sont effectuées à la main.

La détection des fraudes à l’aide de l’apprentissage automatique peut résoudre tous ces problèmes. Il peut battre les systèmes traditionnels de détection de la fraude en termes de rapidité, de qualité et de rentabilité. Un système d’apprentissage automatique non supervisé peut traiter de nouvelles données de manière autonome tout le temps et mettre à jour ses modèles et modèles immédiatement

2.4 Conclusion

Dans ce chapitre , j'ai pu présentai les différents techniques de détection de fraude, avec une description de chaque technique. En addition, j'ai fait une comparaison avec les systèmes de détection traditionnel et les systèmes basé sur ML.

Chapitre 3

Dataset & Model

Dataset est une collection de données (stockant sur un fichier ou une base de données) que nous avons besoin pour faire notre analyse et développer des modèles de machine learning. Les datasets peuvent être représentés sous différents types, que ce soient des tableaux, des graphes, des arbres ou autres. On travaille souvent avec des structures de tableaux dans les algorithmes de machine Learning.

Chaque valeur présente dans un dataset est associée à un attribut et à une observation. Dans notre cas, on va prendre un dataset de KAGGLE qui est sous format .csv. Le dataset contient plus de 20 000 enregistrements de transactions décrits par 112 entités numériques.



FIGURE 3.1 – Kaggle.

3.1 Model

Gradient Boosting est un algorithme de boost populaire. Dans l'amplification de gradient, chaque prédicteur corrige l'erreur de son prédécesseur. Contrairement à Adaboost, les poids des instances d'entraînement ne sont pas modifiés, mais chaque prédicteur est entraîné en utilisant les erreurs résiduelles du prédécesseur comme étiquettes. Il existe une technique appelée Gradient Boosted Trees dont l'apprenant de base est CART (Classification and Regression Trees). Le diagramme ci-dessous explique comment les arbres boostés par gradient sont entraînés pour les problèmes de régression. L'ensemble se compose de N arbres.

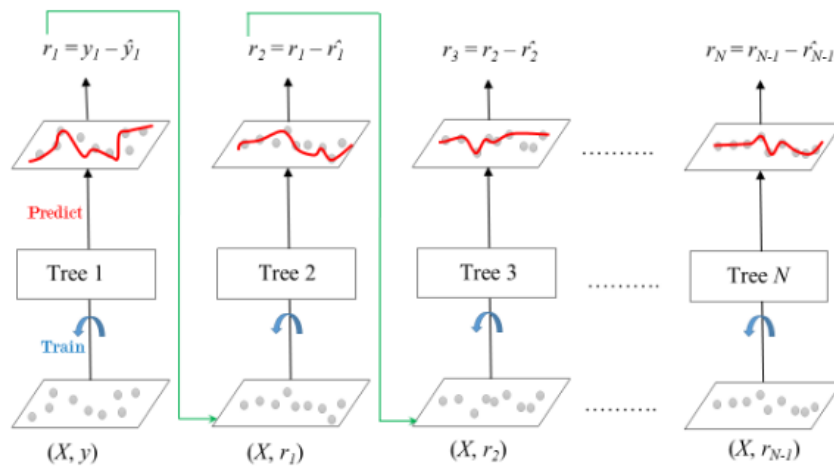


FIGURE 3.2 – Gradient Boosted Trees for Regression

Tree1 est formé à l'aide de la matrice de fonctionnalités X et des étiquettes y. Les prédictions étiquetées \hat{y}_1 sont utilisées pour déterminer les erreurs résiduelles de l'ensemble d'entraînement r_1 . Tree2 est ensuite entraîné à l'aide de la matrice de fonctions X et des erreurs résiduelles r_1 de Tree1 en tant qu'étiquettes. Les résultats prédits \hat{r}_1 sont ensuite utilisés pour déterminer le r_2 résiduel. Le processus est répété jusqu'à ce que tous les arbres N formant l'ensemble soient entraînés.

Il existe un paramètre important utilisé dans cette technique connu sous le nom de shrinkage.

Shrinkage fait référence au fait que la prédiction de chaque arbre de l'ensemble est réduite après avoir été multipliée par le taux d'apprentissage (η) qui varie entre 0 et 1. Il existe un compromis entre l' η et le nombre d'estimateurs, la diminution du taux d'apprentissage doit être compensée par une augmentation des estimateurs afin d'atteindre certaines performances du modèle. Puisque tous les arbres sont entraînés maintenant, des prédictions peuvent être faites.

Chaque arbre prédit une étiquette et la prédiction finale est donnée par la formule,
$$y(\text{pred}) = y_1 + (\eta * r_1) + (\eta * r_2) + \dots + (\eta * r_N)$$

3.2 Entraîner le modèle

La recherche aléatoire est également une méthode de recherche non informée qui traite les itérations indépendamment.

Toutefois, au lieu de rechercher tous les ensembles d'hyperparamètres dans l'espace de recherche, il évalue un nombre spécifique d'ensembles d'hyperparamètres au hasard. Ce nombre est déterminé par l'utilisateur.

Comme elle effectue moins d'essais dans le réglage des hyperparamètres, la méthode nécessite moins de calcul et de temps d'exécution que la recherche en grille.

Mais, étant donné que la recherche aléatoire teste les ensembles d'hyperparamètres au hasard, elle risque de manquer l'ensemble idéal d'hyperparamètres et de renoncer aux performances maximales du modèle.

La recherche aléatoire est également appelée optimisation aléatoire ou échan-

tillonnage aléatoire.

La recherche aléatoire est une méthode dans laquelle des combinaisons aléatoires d'hyperparamètres sont sélectionnées et utilisées pour entraîner un modèle. Les meilleures combinaisons d'hyperparamètres aléatoires sont utilisées. La recherche aléatoire présente une certaine similitude avec la recherche en grille.

Cependant, une distinction clé est que nous ne spécifions pas un ensemble de valeurs possibles pour chaque hyperparamètre. Au lieu de cela, nous échantillonnons les valeurs à partir d'une distribution statistique pour chaque hyperparamètre. Une distribution d'échantillonnage est définie pour chaque hyperparamètre afin d'effectuer une recherche aléatoire.

Cette technique nous permet de contrôler le nombre de tentatives de combinaisons d'hyperparamètres. Contrairement à la recherche en grille, où toutes les combinaisons possibles sont tentées, la recherche aléatoire nous permet de spécifier le nombre de modèles à former. Nous pouvons baser nos itérations de recherche sur nos ressources de calcul ou sur le temps pris par itération. L'image ci-dessous montre une mise en page aléatoire.

Nous avons utilisé la bibliothèque `sci-kit learn` (`sklearn`) lors de la mise en œuvre de la recherche en grille, en particulier `GridSearchCV`. À partir de la même bibliothèque, nous utiliserons `RandomizedSearchCV`. Semblable à `GridSearchCV`, il est destiné à trouver les meilleurs paramètres pour améliorer un modèle donné.

Une différence clé est qu'il ne teste pas tous les paramètres. Au lieu de cela, la recherche est effectuée au hasard. Comme nous le noterons plus loin, une différence pratique entre les deux est que `RandomizedSearchCV` nous permet de spécifier le nombre de valeurs de paramètres que nous cherchons à tester. C'est grâce à l'utilisation de `n_iter`.

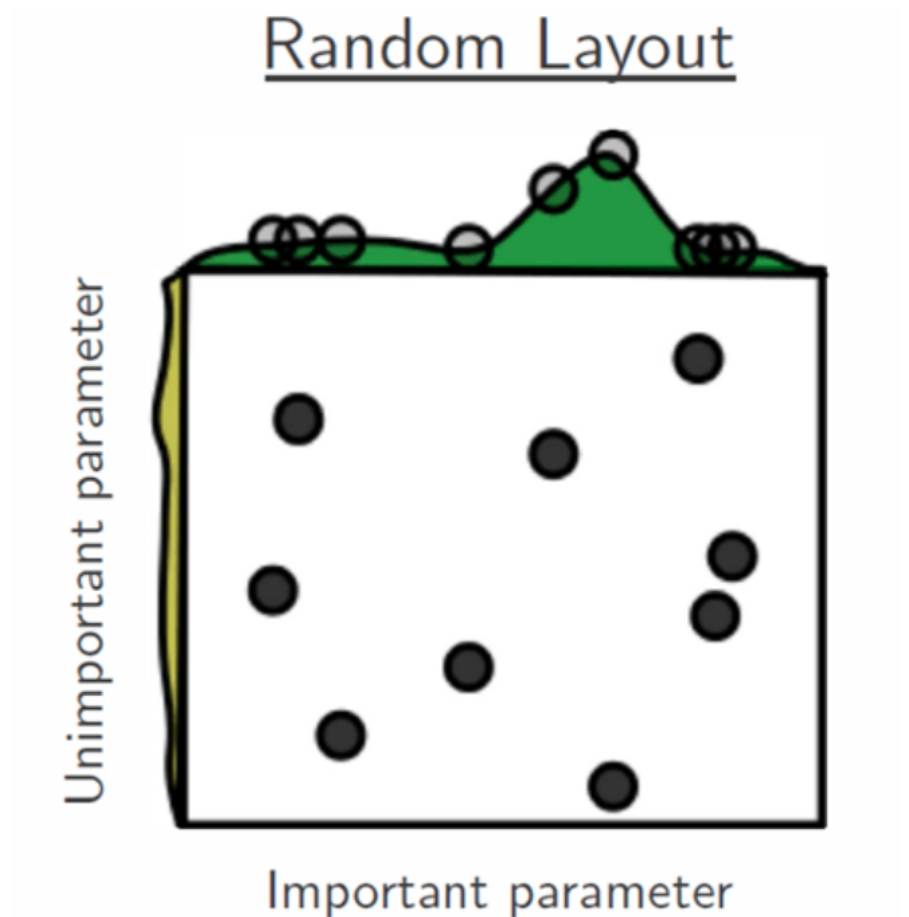


FIGURE 3.3 – Random layout.

3.2.1 Documentation a l'aide de Swagger UI

Swagger UI est un outils de document de back-end qui visualise tout les end-points de notre backend et on peut faire les tests

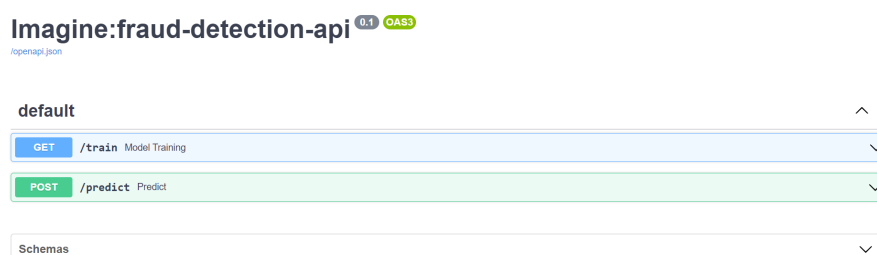


FIGURE 3.4 – Endpoints de notre backend.

/**train** pour trainer notre modèle. les champs demander seront automatiquement rempli par swagger ui ou on peut les remplir

Name	Description
n_splits integer (query) minimum: 1	3
n_iter integer (query) minimum: 1	10

FIGURE 3.5 – Entraîner le Modèle.

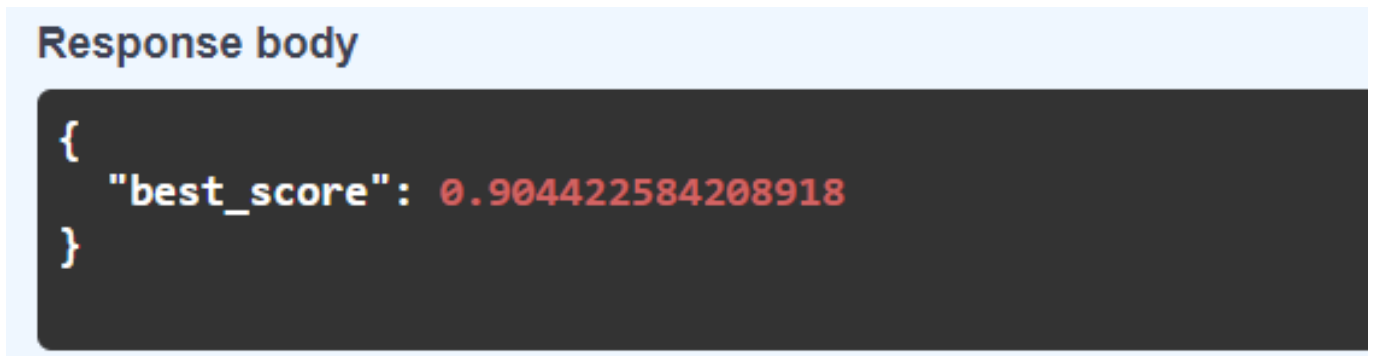


FIGURE 3.6 – Efficacité de notre Modèle

/predict pour faire la prédiction qui demande l'utilisateur d'entrer un échantillon soit ils seront remplie automatiquement

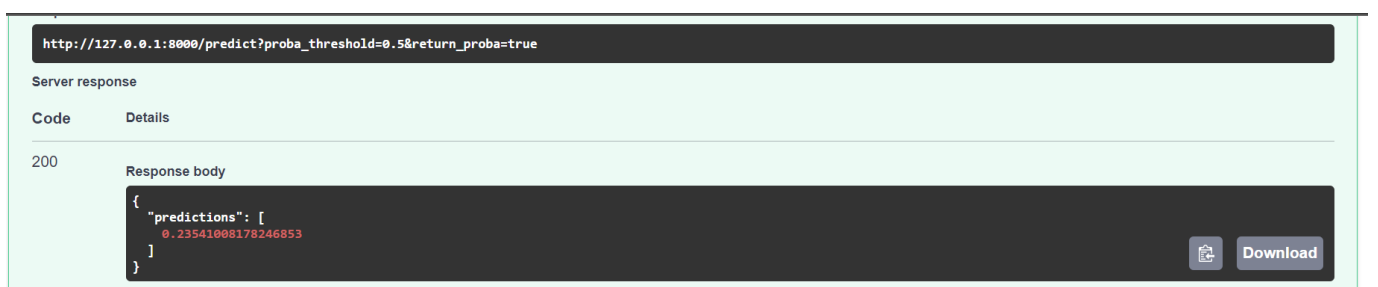


FIGURE 3.7 – Resultat d'une prediction.

3.3 MLflow Tracking

MLflow Tracking est un composant d'API et d'interface utilisateur qui enregistre des données sur les expériences d'apprentissage automatique et vous permet

de les interroger.

On peut utiliser ce composant pour consigner plusieurs aspects de nos exécutions.

Voici les principaux composants qu'on peut enregistrer :

1. **Source** : peut être le nom du fichier qui lance l'exécution. Sinon, si vous utilisez un projet MLflow, il peut s'agir du nom du projet et du point d'entrée de l'exécution.
2. **Parameters** : peut être n'importe quel paramètre d'entrée clé-valeur que vous choisissiez, tant que les valeurs et les clés sont toutes deux des chaînes.
3. **Artifacts** : sont des fichiers de sortie (dans tous les formats). Les artefacts vous permettent d'enregistrer des images, des PNG par exemple, des modèles (tels que des modèles scikit-learn décapés) et des fichiers de données tels que des fichiers Parquet.
4. **RunTime** : vous permet d'enregistrer l'heure de début et de fin
5. **Metrics** : Il permettent d'enregistrer des mesures clé-valeur contenant des valeurs numériques. Il est possible de mettre à jour chaque métrique pendant toute la durée d'une exécution. Cela nous permet, par exemple, de suivre la convergence de la fonction de perte du modèle. De plus, MLflow nous permet de visualiser l'historique complet de chaque métrique.

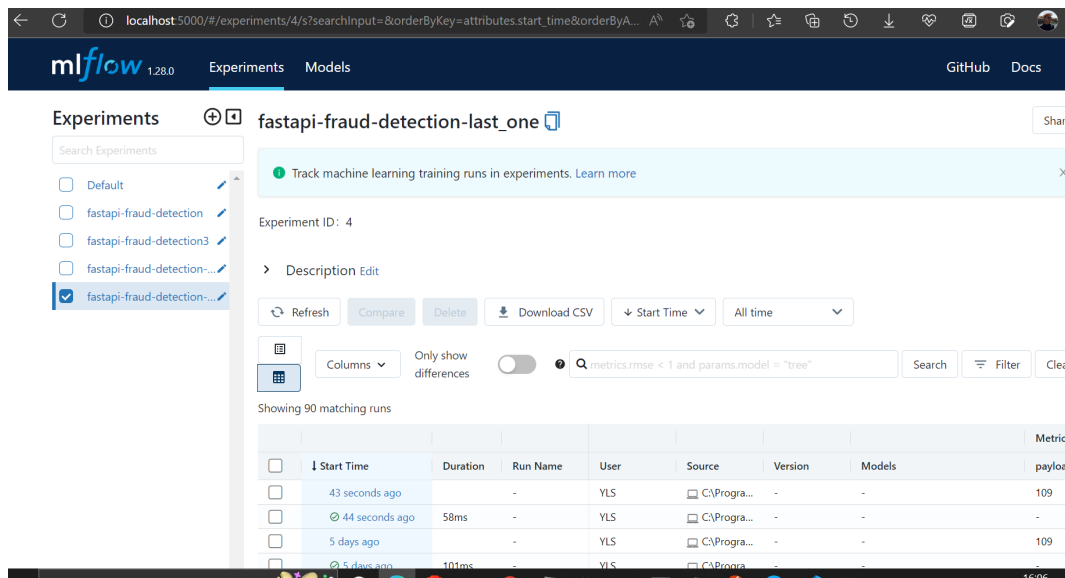


FIGURE 3.8 – Expériences sur MLflow

[fastapi-fraud-detection-last_one](#) > Run 1fdb6f17a308452d8063c598196e697f

Run 1fdb6f17a308452d8063c598196e697f

Run ID: 1fdb6f17a308452d8063c598196e697f Date: 2022-09-23 16:06:03

User: YLS Duration: 58ms

Lifecycle Stage: active

> Description [Edit](#)

▼ Parameters (3)

Name	Value
Result	0.2742875037336301
proba_threshold	0.5
return_proba	True

FIGURE 3.9 – Détails d'une Expérience

3.4 Conclusion

Dans ce chapitre , j'ai pu présentai la source de données et le model dont nous avons utilisé et aussi l'algorithme utiliser pour entrainer notre model. En addition, nous avons fait le cycle de vie du modèle avec MLflow

Chapitre 4

Mise en oeuvre et réalisation

Dans ce chapitre, je présentai la partie réalisation et mise en œuvre de mon travail. Pour cela, je présentai, en premier lieu, et une présentation de l'application (les outils utilisés et les écrans de l'application). En second lieu, j'élaborai les difficultés que j'ai rencontré.

4.1 Architecture de l'application

Cette application est un exemple de modèle d'apprentissage automatique entièrement packagé qui a :

- Gradient boosted trees comme modèle de base.
- Formation et ajustement des modèles à l'aide de l'optimisation de la recherche aléatoire(random search optimization).
- exposition de modèle en tant qu'API REST utilisant le framework FastAPI.
- Docker pour emballer le tout dans un environnement d'exécution conteneurisé.

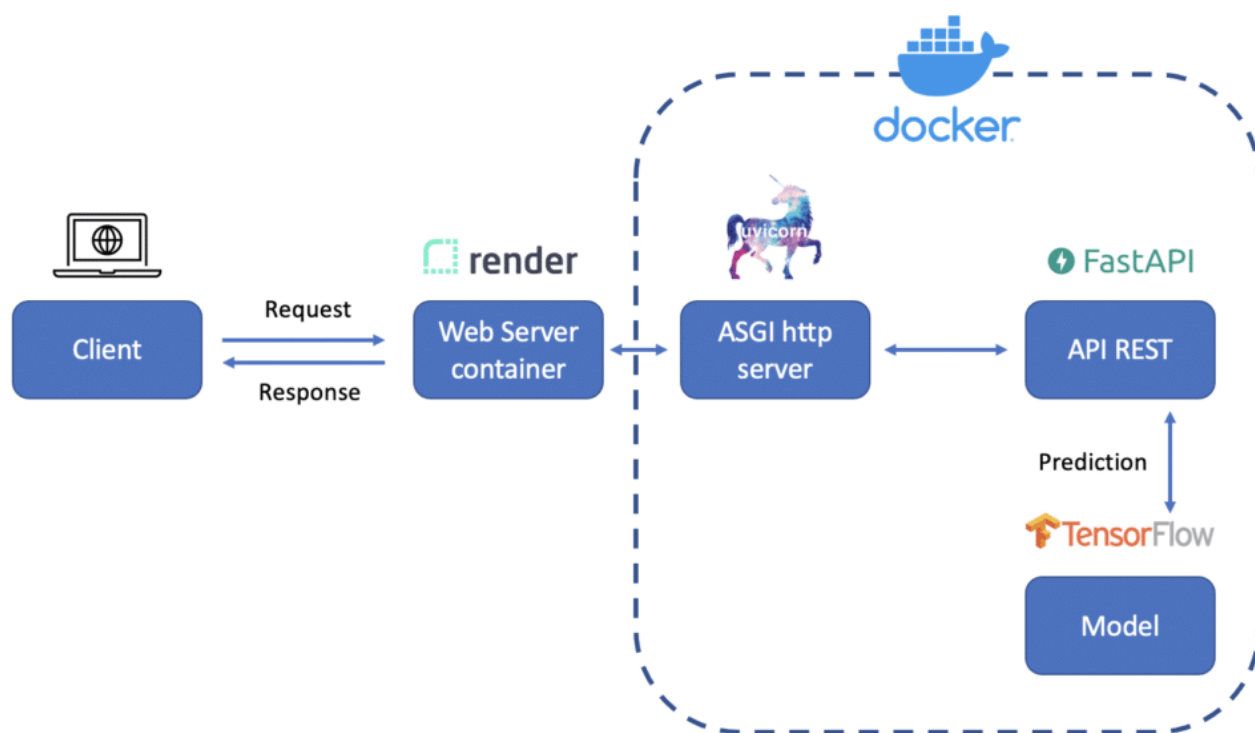


FIGURE 4.1 – Architecture de l'application

4.2 Outils de travail

4.2.1 Framework FastAPI

FastAPI est un framework web haute performance, open source, permettant de créer des APIs avec Python à partir de la version 3.6. Aujourd'hui, de nombreuses entreprises de renommée mondiale comme Uber, Netflix et Microsoft utilisent FastAPI pour créer leurs applications.

Son premier point fort est un haut niveau de performances, comparables avec NodeJS et Go grâce à Starlette et pydantic. Ce framework est également très rapide à coder, offrant une accélération du développement.

Le nombre de bugs et de possibilités d'erreurs humaines est fortement réduit. En outre, FastAPI est très intuitif à l'utilisation, avec des fonctionnalités de complétion et de débogage.

Conçu pour être simple à apprendre et à utiliser, cet outil réduit le temps de

lecture de documentation. La duplication de code est également minimisée. En outre, FastAPI délivre un code prêt pour la production avec une documentation interactive automatique. Enfin, il est basé sur les standards ouverts OpenAI et JSON Schema.

Ce framework est conçu pour optimiser l'expérience du développeur, afin qu'il puisse écrire un code simple pour construire des APIs prêtes pour la production en usant des meilleures pratiques.



FIGURE 4.2 – Logo FastAPI

4.2.2 Swagger UI

Swagger est un ensemble d'outils pour aider les développeurs dans la conception, le build, la documentation et la consommation d'API.

En 2010, Swagger n'était qu'une spécification open source pour construire des API REST. A cela est venu se greffer divers outils, également open source, pour aider à implémenter et visualiser les API; on parle là de Swagger UI, Swagger Editor et Swagger Codegen.

Swagger UI est l'interface graphique permettant de visualiser et d'interagir avec ses API.

Elle offre ainsi la possibilité de les tester facilement et de les rendre plus accessible pour le client.



FIGURE 4.3 – Logo Swagger UI

4.2.3 Uvicorn server

Uvicorn est une implémentation de serveur Web ASGI pour Python.

Jusqu'à récemment, Python manquait d'une interface serveur/application minimale de bas niveau pour les frameworks asynchrones. La spécification ASGI comble cette lacune et signifie que nous sommes maintenant en mesure de commencer à créer un ensemble commun d'outils utilisables dans tous les frameworks asynchrones.

Uvicorn prend actuellement en charge HTTP/1.1 et WebSockets



The lightning-fast ASGI server.

FIGURE 4.4 – Logo Uvicorn server

4.2.4 Docker

Docker est un ensemble de produits de plate-forme en tant que service qui utilisent la virtualisation au niveau du système d'exploitation pour fournir des logiciels dans des packages appelés conteneurs. Les conteneurs sont isolés les uns des autres et regroupent leurs propres logiciels, bibliothèques et fichiers de configuration ; ils peuvent communiquer entre eux par des canaux bien définis.

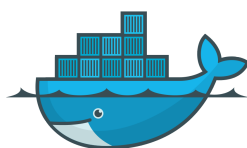


FIGURE 4.5 – Logo Docker

4.2.5 Jupyter

Le Jupyter Notebook est une application web open-source qui vous permet pour créer et partager des documents contenant du code en direct, des équations, des visualisations et le texte narratif. Les utilisations comprennent : nettoyage et transformation des données, numérique simulation, modélisation statistique, visualisation de données, apprentissage automatique, etc. plus. Jupyter Notebooks est un projet dérivé du projet IPython, qui avait l'habitude d'avoir un projet IPython Notebook lui-même. Le nom, Jupyter, vient des principaux langages de programmation pris en charge qu'il prend en charge : Julia, Python et R. Jupyter est livré avec le noyau IPython, qui vous permet d'écrire vos programmes en Python, mais il existe actuellement plus de 100 autres noyaux que vous pouvez.



FIGURE 4.6 – Logo Jupyter

4.2.6 MLflow

MLflow est un outil qui permet de répondre à ces questions. Il permet aux Data Scientists de suivre facilement les progrès réalisés lors du développement et de la mise au point des modèles. Il s'occupe du packaging et du déploiement des modèles, quel que soit le cadre ou le langage de programmation utilisé pour les créer. En outre, MLflow fournit un registre, où les modèles que nous voulons conserver ou partager peuvent être stockés en toute sécurité et restent facilement accessibles.



FIGURE 4.7 – Logo MLflow

4.2.7 ReactJS framework

React (aussi appelé React.js ou ReactJS) est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page (ou portion) HTML à chaque changement d'état.

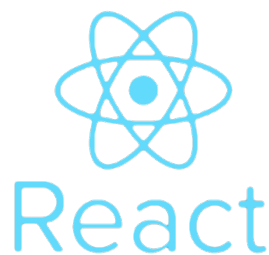


FIGURE 4.8 – Logo ReactJS

4.2.8 Le langage HTML

Hypertext Markup Language, généralement abrégé HTML, est le format de données conçu pour représenter les pages web. C'est un langage de balisage qui permet d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images et des formulaires de saisie.



FIGURE 4.9 – Html

4.2.9 Storybook

Storybook c'est une librairie open source créée pour le développement de composant UI qui se concentre sur l'isolation des composants, permettant une création plus organisée et donc plus efficace.

C'est un environnement de développement permettant de naviguer dans un catalogue de composant (book), de voir les différents états d'un composant par use-case (story) tout en apportant des outils de test, de prévisualisation et de

documentation du composant.

Storybook fonctionne avec la plupart des frameworks UI, que ce soit le sacro-saint triptyque Vue-React-Angular autant que Svelte, React Native ou même les Web Components .



FIGURE 4.10 – Logo Storybook

4.2.10 Le langage CSS

Les CSS, Cascading Style Sheets (feuilles de styles en cascade), servent à mettre en forme des documents web, type page HTML ou XML. Par l'intermédiaire de propriétés d'apparence (couleurs, bordures, polices, etc.) et de placement (largeur, hauteur, côte à côte, dessus-dessous, etc.), le rendu d'une page web peut être intégralement modifié sans aucun code supplémentaire dans la page web.



FIGURE 4.11 – Css

4.2.11 Le langage javascript

JavaScript (souvent abrégé en « JS ») est un langage de script léger, orienté objet, principalement connu comme le langage de script des pages web. Mais il est aussi utilisé dans de nombreux environnements extérieurs aux navigateurs web .. C'est un langage à objets utilisant le concept de prototype, disposant d'un typage

faible et dynamique qui permet de programmer suivant plusieurs paradigmes de programmation : fonctionnelle, impérative et orientée objet.



FIGURE 4.12 – Javascript

4.3 Réalisation de l'application

La page d'accueil est la première interaction entre l'utilisateur et l'application. Ainsi on a essayé de la rendre plus attractive et expliquant simplement l'intérêt et les fonctionnalités réalisées par l'application.

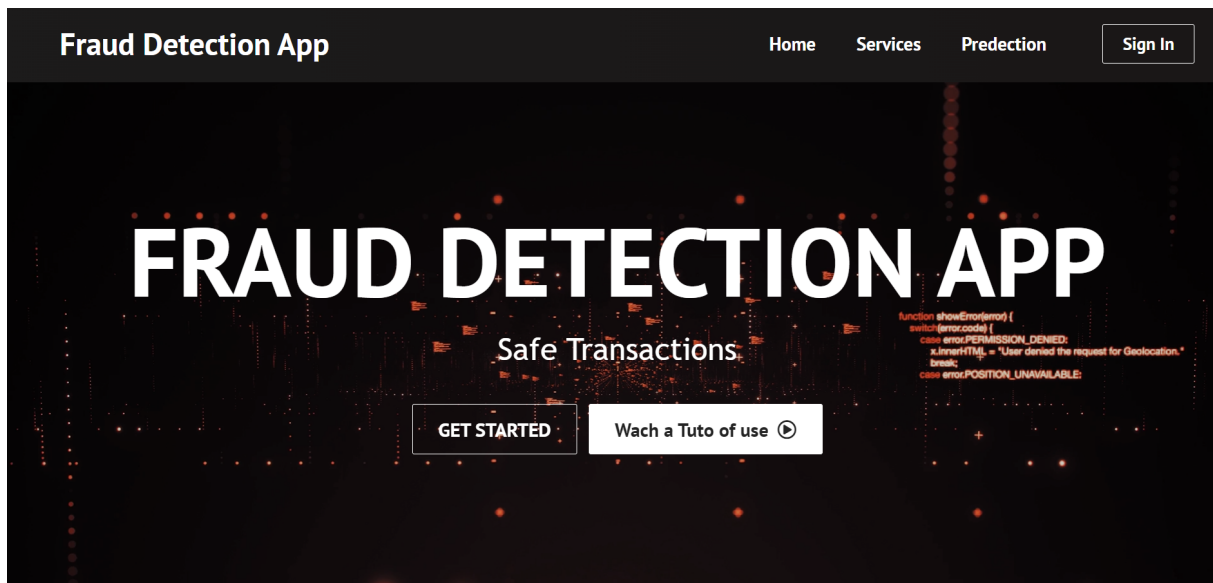


FIGURE 4.13 – Page D'accueil.

Page services cette page procure l'utilisateur par des informations et des conseils pour éviter ce genre de crimes

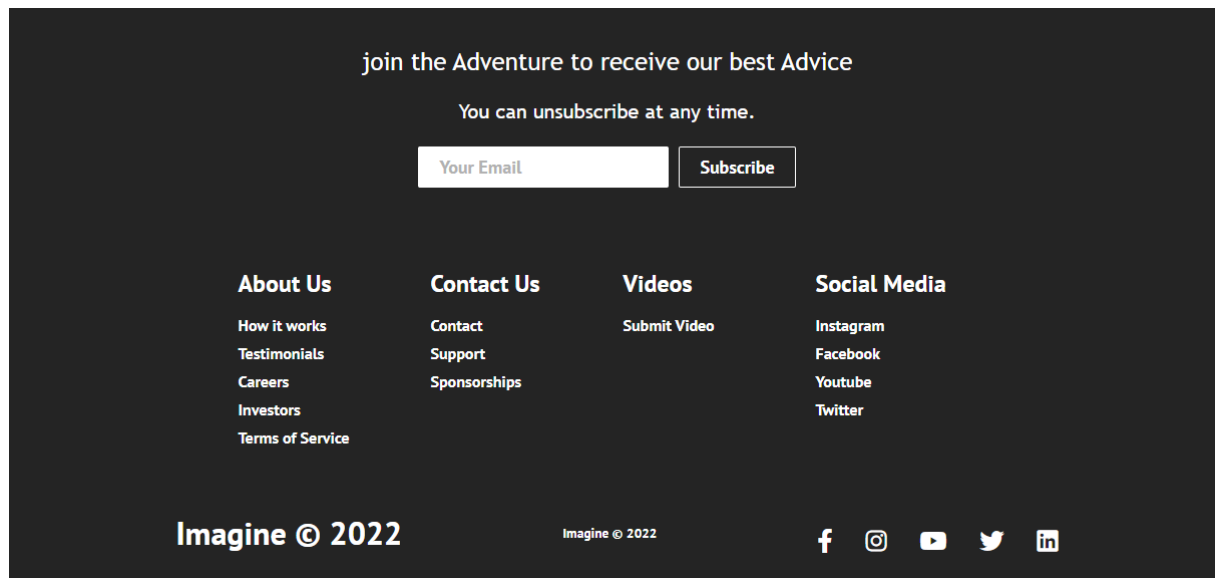


FIGURE 4.14 – Footer.

Page prediction

Prediction est la page principale de cette application qui expose le service primordiale offerte . Elle demande a l'utilisateur de charger un fichier .csv qui contient une transaction .

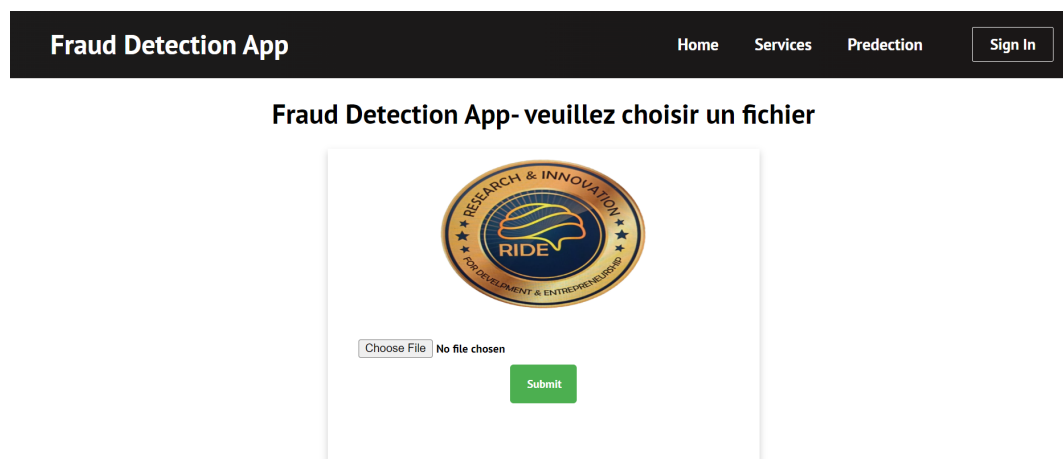


FIGURE 4.15 – Charger un fichier csv

Une fois cliquant sur Submit utilisateur va obtenir le result :

Le résultat sous format d'une probabilité accompagnée par un commentaire expliquant le résultat de cette transaction

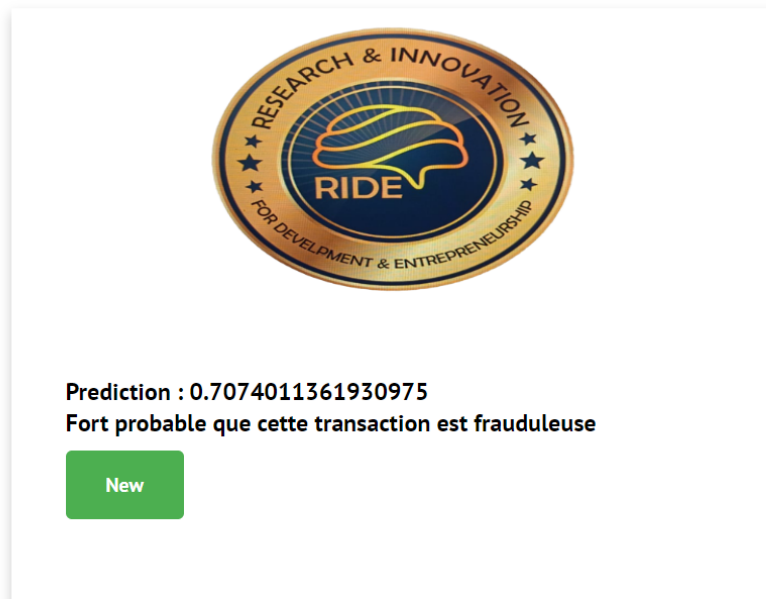


FIGURE 4.16 – Résultat de prédiction.

4.4 Conclusion

Dans ce chapitre, j'ai présenté les interfaces réalisées dans cette application, Après avoir spécifié et s'adapter avec les outils de travail, j'ai commencé l'implémentation de l'application et j'ai pu présenter les captures d'écran précédentes.

Conclusion et perspectives

Le travail effectué dans ce stage vise à la réalisation de notre application Fraud Detection App, nous avons analysé le problème et est venu avec une application que nous espérons être une solution efficace et bénéfique.

En premier lieu, nous avons commencé le premier chapitre avec la représentation de l'organisation d'accueil et le projet Imagine , nous avons décrit les sources de données qu'on va utiliser et déterminer le modèle convenable pour résoudre notre problème ensuite l'explication de la partie d'entraînement entraînement de notre modèle

Nous sommes ensuite passés à la description du type de données que nous allons utiliser. pour former notre modèle d'apprentissage automatique et comment nos choix ont affecté le développement de l'application web, avant de passer à la description du modèle que nous avons utilisé et comment il utilise les données que nous avons fournies pour faire ses prédictions après que nous examinons comment exactement nous mettons notre modèle en production afin qu'il puisse être utilisé facilement sur le Web back-end de l'application.

Enfin, nous avons pris le temps de réaliser notre application tout en précisant les outils de développement ainsi que les langages de programmation utilisés.

Ce projet nous a permis d'acquérir une expérience personnelle et professionnelle. Il a été très bénéfique pour nous parce que nous avons eu la chance d'améliorer nos connaissances dans le domaine de la science des données et de l'apprentissage automatique, mais c'était aussi un bon début pour nous pour commencer à créer une plate-forme à partir de zéro.

Bibliographie

- [1] Edrawmax. creation of architecture. <[>](https://www.edrawmax.com/online/app.html?VcyxDgIhDAbgp2E1cpiLziqTGgdvJvWo5gxQAlyUt7ck5+DU9v/Sn+4voQ7vvJFqp6Tst+Z4qnoQnb5iyhTA8TqEAB6t6NZ6crhCaz+i61vGv/I+YOHYTSnsTH6W0oihQ8HBU2psVmg25Iw2gmeCfzP9hRGjIVrzxBbayJ/mZ1jkl8=> 1.09/2022.>[2] FastAPI. Dockerfile creation. <<a href=)
- [3] geeksforgeeks. gradient-boosting. <[>](https://www.geeksforgeeks.org/ml-gradient-boosting/> .08/2022.>[4] intellias. machine learning in fraud detection. <<a href=)
- [5] kaggle. source de dataset <[>](https://www.kaggle.com/datasets/volodymyrgavrysh/fraud-detection-bank-dataset-20k-records-binary> .8/2022,.>[6] kdnuggets. ML lifecycle. <<a href=)
- [7] ML lifecycle. Manage lifecycle.

- [8] Lightgbm. lightgbm.lgbmclassifier. <<https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMClassifier.html>> .2022.
- [9] Overleaf. rediger le rapport. <<https://www.overleaf.com/project>> .09/2022.
- [10] Pickle. save the model. <<https://www.analyticsvidhya.com/blog/2021/08/quick-hacks-to-save-machine-learning-model-using-pickle-and-joblib/>> 1.09/2022.
- [11] spicework. fraud-detection. <<https://www.spiceworks.com/it-security/vulnerability-management/articles/what-is-fraud-detection/>> 12.08/2022.