République Algérienne Démocratique et Populaire
الجمهورية الجزائرية الديمقراطية الشعبية
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
وزارة التعليم العالي و البحث العلمي

ESI ECOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE

المدرسة الوطنية العليا للإعلام الآلي
(المعهد الوطني للتكوين في الإعلام الآلي سابقا)
École nationale Supérieure d'Informatique
ex. INI (Institut National de formation en Informatique)

Rapport de stage

Option : Systèmes Informatiques (SQ)

Option : Systèmes d'Informations et Technologies (ST)

Analyse et prédiction du churn des clients grace à un modèle de Machine Learning

Réalisé par :

Mr. ASSELAH Wahid
Mlle. LAGGOUN Amina

Encadré par :

Mr. BENTALEB Abdelfettah
(Data scientist, Djezzy)

Année universitaire : 2023/2024

Résumé

Le « churn » des clients est un indicateur clé de la qualité du service et de la performance de l'opérateur, ce qui a un impact sur les revenus et les opérations.

Pour y remédier, il est essentiel de mettre en place une gestion proactive de l'attrition. Il s'agit de surveiller les comportements et les préférences des clients en temps réel afin d'identifier les signes précurseurs d'une désaffection potentielle. En repérant ces indicateurs, les opérateurs peuvent élaborer et mettre en œuvre des stratégies visant à améliorer la fidélisation des clients.

Au cours des dernières années, l'apprentissage automatique a joué un rôle essentiel dans l'amélioration des stratégies de contrôle préventif du taux de désaffiliation. En exploitant des algorithmes avancés et des modèles prédictifs, les opérateurs peuvent analyser de grands volumes de données clients pour découvrir des schémas et des évolutions complexes susceptibles d'indiquer un potentiel de départ.

Dans cette étude, nous proposons une approche fondée sur l'apprentissage automatique, où nous appliquons une variété d'algorithmes de pointe sur un ensemble de données minutieusement restreint, fourni par l'établissement d'accueil. Notre objectif est de sélectionner la solution optimale à la suite d'une évaluation approfondie des résultats obtenus.

Mots clés : Churn des clients, Gestion proactive de l'attrition, Apprentissage automatique (Machine Learning), Analyse de données clients

---

Table des matières

Table des figures

Liste des tableaux

Liste des sigles et acronymes

SUT Services à Utilité Technique

FNI Fonds National d'Investissement

DWH Data WareHouse

ETL Extract, Transfer, Load

DBSS DataBase Summary Sheet

BTEQ Basic TeraData Query

BI Business Intelligence

HDFS Hadoop Distributed File System

TelCo Telecommunications' Company

SIM Subscription Identity Module

SMOTE Synthetic Minority Oversampling Technique

TP True Positives

FN False Negatives

CBC categorical boost classifier

CatBoostClassfier categorical boost classifier

ML Machine Learning

Remerciements

Nous adressons nos remerciements à M. Amrani, le chef du département de Data Science au sein de DJEZZY, pour nous avoir accueillis au sein de son équipe. Nous exprimons également notre gratitude à notre mentor, M. Taleb Abdelfettah. Ses conseils constants, son expertise et son dévouement absolu ont été essentiels à l'aboutissement de notre projet. Aussi, nos remerciements vont à l'encontre de M. Walid Boukhalfa pour son esprit de collaboration et son soutien tout au long de notre stage. Nos échanges ont été très bénéfiques et constructifs.

Enfin, nous remercions l'École nationale d'informatique (ESI) de nous avoir donné l'occasion de nous lancer dans cette aventure professionnelle par le biais de ce stage. Nous apprécions grandement notre institution pour la qualité de son enseignement, qui nous a permis d'acquérir les compétences essentielles pour réussir dans ce projet.

Introduction

Les clients représentent le potentiel de toutes les entreprises. Ainsi, celles-ci doivent adopter des stratégies de fidélisation afin de retenir cette clientèle et d'éviter qu'elle ne se tourne vers des concurrents, notamment dans le domaine de la télécommunication où la concurrence est rude. Cela implique la nécessité de prédire en amont les clients susceptibles de se désengager, afin de réagir de manière pro-active avant qu'il ne soit trop tard. En effet, il est plus coûteux d'acquérir de nouveaux clients que de fidéliser les clients existants.

Dans le cadre de notre stage pratique de 2èmê année de cycle supérieur (2CS) effectué au sein de l'entreprise Djezzy, l'un des leaders de la télécommunication en Algérie. Notre travail repose principalement sur l'analyse approfondie des données relatives au comportement des clients de Djezzy, ainsi que sur le développement d'un modèle de prédiction et de classification basé sur l'apprentissage automatique pour l'identification des clients susceptibles de rompre leur contrat d'abonnement.

Chapitre 1

Présentation de l'organisme d'accueil

1.1 L'entreprise Djezzy

Djezzy est un opérateur de télécommunications algérien créé en 2001. Il est un acteur majeur dans le domaine de la téléphonie mobile, comptant plus de 14 millions d'abonnés. L'entreprise propose une large gamme de services, comprenant notamment des forfaits prépayés et post-payés, des services de Data, ainsi que des options de services à valeur ajoutée et de SUT (Services à Utilité Technique).

Depuis Juillet 2022, date à laquelle Veon avait signé l'acte de cession de la totalité de ses actions dans l'entreprise au profit du Fonds National d'Investissement, Djezzy devient une entreprise nationale. Elle est désormais la propriété du Fonds National d'Investissement (FNI) à hauteur de 96,57% et de Cevital avec 3,43%. De ce fait, Djezzy est actuellement contrôlée intégralement par deux actionnaires Algériens.

Figure 1.1: Logo de Djezzy

1.2 L'organigramme de Djezzy

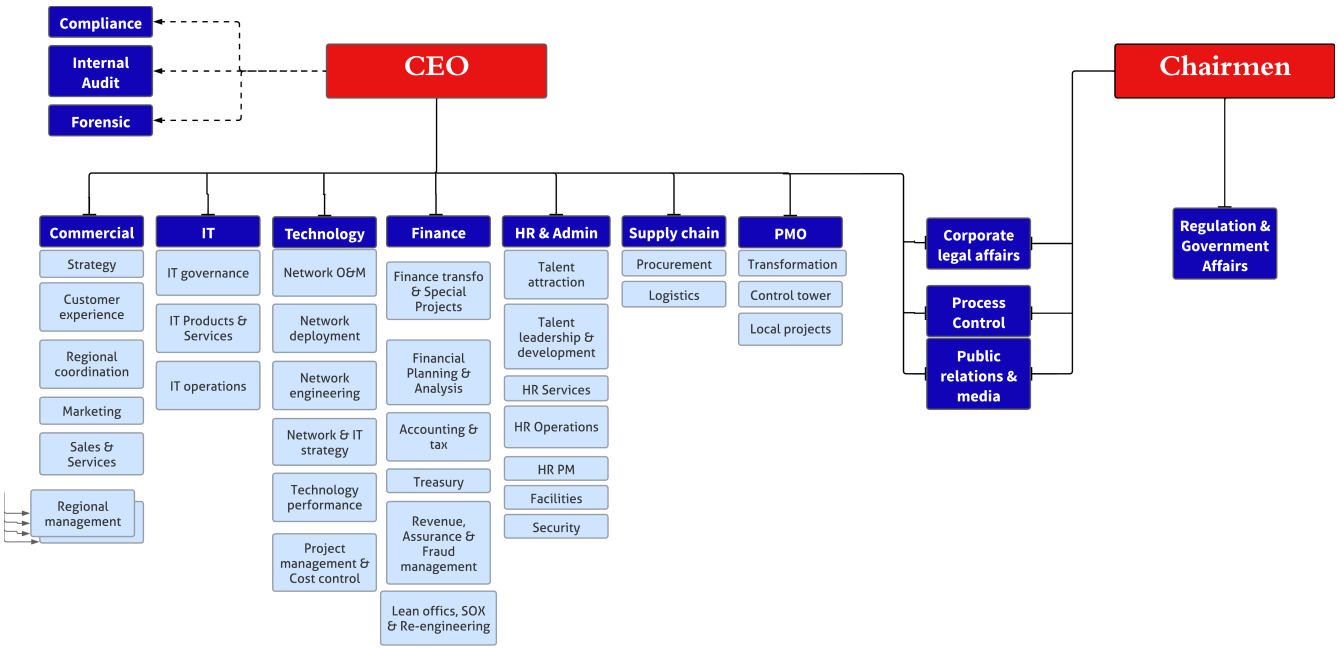


Figure 1.2: Organigramme de Djezzy

1.3 Les données Djezzy

Comme toute grande entreprise, Djezzy possède un large volume de données, ce dernier est structuré suivant une architecture qui obéit à des standards dans la matière et qui désigne la manière dont sont organisées les données.
Dans ce qui suit , nous allons explorer l'architecture de données de Djezzy en analysant les éléments qui la composent et en exposant les motifs de leur utilisation.

1.3.1 L'architecture globale des données Djezzy

En 2017, dans le cadre de sa transformation digitale, Djezzy a mis en place une plateforme Big Data en parallèle à son Data Warehouse (DWH) (entrepôt de données). Cette initiative visait à permettre à Djezzy de traiter les grandes masses de données qu'elle possède, que ce soit en batch ou en temps réel, tout en assurant un stockage efficace et à moindre coût.

Les deux plateformes (DWH et Big Data) travaillent en parallèle et répondent à des besoins bien spécifiques.

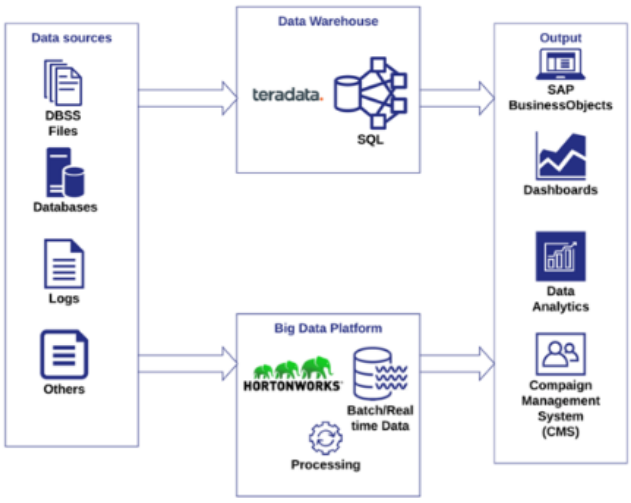


Figure 1.3: Schéma de l'architecture globale des données Djezzy

1.3.2 Data WareHouse

Le Data Warehouse est basé sur la solution Teradata Database et a pour mission de gérer divers types de données, notamment les données transactionnelles, les informations sur les clients, les données d'appels, etc. Il est principalement utilisé pour générer un grand nombre de rapports qui aident à la prise de décisions.

Au niveau du DWH, l'ingestion des données est réalisée grâce à l'outil ETL Informatica Powercenter. Cet outil est alimenté par diverses sources de données provenant de DBSS1 et d'autres bases de données. Les données sont agrégées à l'aide de Teradata BTEQ et de scripts shell.

Pour l'analyse des données du DWH, l'entreprise utilise les outils BI : SAP BusinessObjects et Qlik Sense.
En ce qui concerne les analyses avancées (ex: prédiction du churn), Djezzy exploite l'outil d'analyse prédictive Kxen.

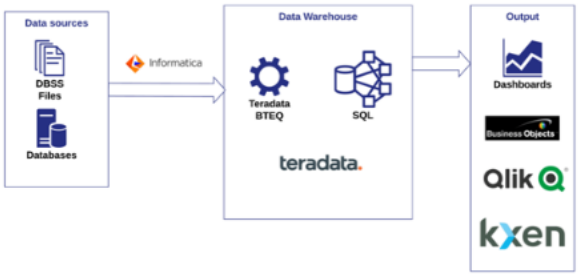


Figure 1.4: Schéma Data WareHouse Djezzy

1.3.3 Big Data

La plateforme Big Data de Djezzy, basée sur Hortonworks, est principalement utilisée pour gérer les flux en temps réel afin d'orienter les campagnes clients vers les segments appropriés au moment opportun, ainsi que pour d'autres traitements en lot.

La plateforme Big Data comprend deux pipelines distincts : l'un pour les traitements

en lot (batch) et l'autre pour les traitements en temps réel. Dans la couche batch, les données sont ingérées à l'aide de l'outil de gestion de flux de données Nifi et sont stockées dans HDFS pour être soumises aux requêtes du logiciel d'entreposage de données Hive. Ces données sont également traitées par le moteur de calcul distribué Spark. Les résultats de cette couche sont utilisés en entrée pour les modèles de Churn et d'Affinité client, ainsi que pour d'autres analyses et rapports.

Dans la couche temps réel, les données peuvent être ingérées directement via la plateforme de streaming distribuée Kafka ou par le processus Kafka dans Nifi. Ensuite, ces données sont transférées vers la plateforme de calcul et de stockage en mémoire Ignite pour alimenter le système de gestion de campagne avec les segments clients appropriés en temps réel. Pour des raisons de persistance des données, celles stockées dans Ignite sont transférées vers la base de données distribuée Cassandra, étant donné que la première plateforme ne permet pas la persistance des données.

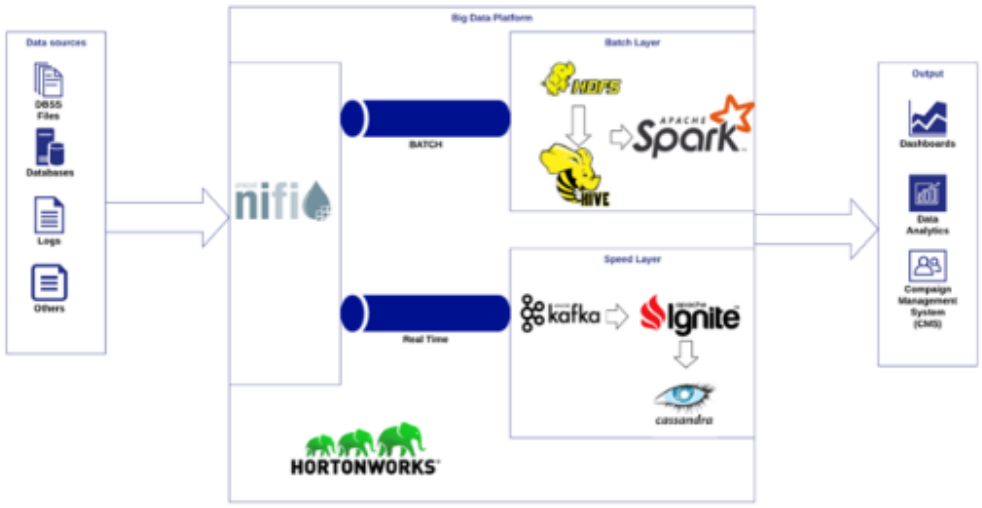


Figure 1.5: Schéma Big Data Djezzy

Une description plus détaillée des outils mentionnés précédemment est donnée en Annexe A

Chapitre 2

État de l'art

2.1 Méthodologie de travail

Pour une productivité optimale, notre équipe s'est appuyée sur un système d'objectifs hebdomadaires afin de progresser. Chaque étape de travail était discutée et validée par notre superviseur au cours des visites régulières au siège de Djezzy situé à Dar El Beida.

Etant donné que dans le domaine de la science des données (Data Science) est un domaine nouveau pour nous, notre première semaine a servi de phase de documentation. Nous avons effectué des recherches approfondies sur les pratiques courantes de traitement des données et nous nous sommes familiarisés avec des concepts en rapport avec le Machine Learning et les divers modèles de classification.

Après maintes discussions et concertation avec notre superviseur concernant les outils à utiliser, nous opté pour ce qui suit. *Python* (voirAnnexe B.) s'est imposé comme le langage de prédilection, avec la bibliothèque Pandas pour une manipulation transparente des données. Notre espace de travail organisé comprenait *Visual Studio Code* (voir Annexe B.) et *Jupyter Notebook* (voir Annexe B.). Pour le travail collaboratif, nous avons adopté GitHub.

Dès la deuxième semaine, notre ensemble d'outils était configuré et opérationnel, ouvrant la voie à des progrès substantiels. Nous avons élaboré un plan de travail s'étalant sur la période d'un mois.

2.2 Problématique

Avec la diversité des offres présentées par les entreprises de télécommunications, les clients s'orientent souvent vers l'opérateur dont les forfaits correspondent le mieux à leurs préférences et à leurs besoins.

L'évolution rapide du marché TelCo a pour conséquence les opérateurs tentent de s'adapter à ce marché en multipliant les offres pour répondre aux attentes des différentes tranches du public cible.

Dans certains cas, si ce n'est dans la plupart des cas, les clients sont contraints de s'adapter également. Ils ont tendance à changer de carte SIM pour celle qui garantit la meilleure qualité tout en préservant leur porte-monnaie, quittant ainsi l'opérateur d'origine : les clients, alors, **"churn"**.

Qu'est-ce que le "churn" des clients ?

On parle du churn des clients (ou l'attrition de la clientèle) lorsque des clients ou des abonnés cessent de faire affaire avec une entreprise ou un service.

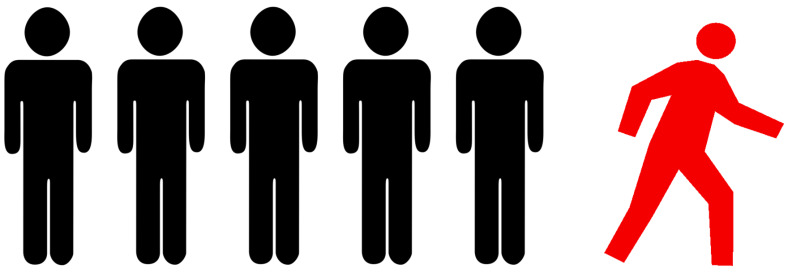


Figure 2.1: Attrition des clients

Dans le secteur des télécommunications, les clients peuvent choisir parmi un grand nombre de fournisseurs de services et passer activement de l'un à l'autre. En effet, le taux

d'attrition dans l'année 2022 dans le secteur des télécommunications était de 22% sur ce marché hautement concurrentiel. [1]

En s'attaquant au problème du désabonnement, ces entreprises peuvent non seulement préserver leur position sur le marché, mais aussi croître et prospérer. Plus il y a de clients dans leur réseau et plus le chiffre d'affaires de ces entreprises augmente.

2.3 Étude de l'existant

Djezzy a mis en place un modèle basé sur l'apprentissage automatique (Machine Learning) qui a la capacité de prédire le "churn" des clients. Ce modèle exploite les données historiques des clients pour analyser et identifier les tendances qui indiquent quand un client est susceptible de résilier son contrat ou de se désabonner des services de l'entreprise.

Cependant, avec l'émergence du Big Data, l'entreprise dispose désormais d'une quantité considérable d'informations sur ses clients, telles que : type d'offres auxquelles le client est abonné, type de réseau mobile auquel le client est abonné ,type d'appareil où la puce est déployée.
Cette abondance de données souligne la nécessité impérieuse de développer un modèle de churn plus précis, capable de tirer pleinement parti de ces nouvelles sources d'informations pour anticiper efficacement le churn.

[1] Source : https://techsee.me/resources/reports/state-of-customer-churn-telecom-survey-report/

Chapitre 3

La solution proposée

Dans cette section, nous aborderons en détails notre approche pour résoudre le problème de la perte de clientèle. Nous détaillerons les stratégies et mécanismes spécifiques que nous avons développés pour mettre en œuvre une approche dynamique de lutte contre l'attrition en exploitant les techniques de l'apprentissage automatique.

3.1 Analyse et traitement des données

3.1.1 Description des données

Les informations utilisées pour développer cette solution ont été recueillies auprès du département de data science de Djezzy. Pour des raisons de confidentialité, les données ne sont pas récentes. Elles comprennent 25 970 enregistrements de clients accompagnés de l'information booléenne de désabonnement, collectées dans un laps de temps spécifique.

L'attribut "churn" est la principale variable cible (target) de l'ensemble de données, puisque notre objectif premier consiste à analyser les taux d'attrition, les autres colonnes servent de variables caractéristiques (features) dont la description est donnée ci-dessous:

- **Subs_id:** Identifiant du client
- **sex:** Genre du client
- **Global_Profile:** Type d'offres auxquelles le client est abonné
- **LINE_TYPE:** Génération de réseau mobile auquel le client est abonné
- **Wilaya:** Wilaya de résidence du client
- **Age_Years:** Age du client

- **Devicetype:** Type d'appareil où la puce est déployée
- **yr:** Durée écoulée (en années) depuis l'activation initiale de la SIM.
- **Mr:** calculée en soustrayant le nombre d'années *yr* (multiplié par 12 mois chacune) du nombre total de mois depuis l'activation initiale de la carte SIM.
- **Value_Segment:** Degrés de valeur qu'apporte le client à Djezzy
- **Behavior_Segments:** Comportement du client
- **number_subscription:** Nombre de souscriptions du client
- **nb_supended:** Compteur de suspension du compte du client

3.1.2 Exploration des données

Pour obtenir une analyse approfondie des données fournies, il nous était essentiel de prendre connaissance de certaines informations telles que :

– Le nombre de valeurs uniques par attribut.

sex	2
Global_Profile	73
LINE_TYPE	3
Wilaya	61
Age_Years	82
Devicetype	10
yr	22
Mr	12
Value_Segment	8
Behavior_Segments	14
number_subscription	238
nb_supended	8
Churn	2

Figure 3.1: Nombre de valeurs uniques par attribut

– Le nombre de valeurs nulles par attribut.

```
sex                      0
Global_Profile           0
LINE_TYPE                0
Wilaya                   0
Age_Years                0
Devicetype             180
yr                       0
Mr                       0
Value_Segment           23
Behavior_Segments        0
number_subscription      0
nb_supended              0
Churn                    0
```

Figure 3.2: Nombre de valeurs nulles par attribut

– Le pourcentage de lignes où l'attribut a une valeur nulle.

```
Le pourcentage de lignes où la valeur Devicetype est nulle est de :
0.693107431651906 %
--------------------------------------------------------------------------------
Le pourcentage de lignes où la valeur Value_Segment est nulle est de :
0.08856372737774355 %
```

Figure 3.3: pourcentage de valeurs nulles par attribut

3.1.3 Prétraitement des données

Avant de commencer le prétraitement des données, une étape préliminaire a consisté à supprimer les colonnes jugées peu significatives :

- **Colonnes vides**

  Nous avons omis les colonnes entièrement vides.

- **Subs_id**

  La colonne "Subs_ id", qui indique les identifiants des clients, a été exclue de l'analyse. L'unicité des valeurs de chaque ligne a démontré son manque d'efficacité dans la prédiction du taux de désabonnement.

Par la suite, les noms de colonnes ont été toutes converties en lettres minuscules.

Traitement des valeurs manquantes (nulles)

La gestion des valeurs manquantes, un aspect incontournable de tout projet de Machine Learning, requiert des choix adaptés en fonction de la nature du problème. Ces choix incluent le remplacement par des valeurs (telles que la moyenne, la valeur précédente ou suivante) ou la suppression pure et simple des données.

Dans notre cas, deux champs présentent des valeurs manquantes :

- **value_segment**

  Conformément à l'analyse préalablement réalisée (où le pourcentage de valeurs manquantes pour chaque attribut a été calculé), il est ressorti que le champ "value_segment" possède un pourcentage extrêmement faible de valeurs manquantes.
  Par conséquent, nous avons _presque_ choisi de supprimer les lignes où le champ "value_segment" est nul.
  Toutefois, après une seconde étude approfondie des données, nous avons découvert que le champ "behavior_segment" n'est qu'une description du champ "value_segment". De ce fait, une dépendence a été remarquée. Elle est comme suit :

Behaviour_Segment	**Value_Segment**
New Customer →	NEW

Table 3.1: Correspondance entre 'behaviour_segment' et 'value_segment'

  En outre, chaque cas de valeur nulle dans le champ "value_segment" correspondait à la configuration du "behavior_segment" en tant que "new_customer" (nouveau client).
  En conséquence, nous avons décidé de remplir toutes les valeurs manquantes de ce champ par la valeur "NEW".

- **devicetype**

  Néanmoins, l'attribut en question présentait un pourcentage relativement élevé de valeurs manquantes, ce qui rendait la situation plus complexe que dans le cas précédent.
  Afin de faciliter la manipulation des données, nous avons classé les valeurs manquantes dans une catégorie appelée "Unknown". En attribuant ces valeurs à une

catégorie désignée, nous pouvons garantir qu'elles seront traitées de manière cohérente tout au long de notre chaîne de traitement des données.

Traitement des valeurs erronées

En ce qui concerne cette rubrique, nous avons développé une fonction utilisant la bibliothèque fuzzywuzzy, un outil Python de correspondance de chaînes de caractères, qui facilite l'identification des similitudes entre des valeurs distinctes au sein de chaque attribut.

L'application de cette fonction à nos attributs a fait apparaître des schémas significatifs, en particulier en ce qui concerne le champ "wilaya". Ce résultat était anticipé, à la fois en raison du dépassement du nombre de 58 (correspondant au nombre de Wilayas Algériennes) en ce qui concerne les valeurs uniques [*voir figure 3.1*], ainsi que de la tendance à la variation orthographique observée lorsqu'il s'agit de lieux géographiques.

Après avoir identifié ces valeurs semblables, nous les avons croisées avec l'orthographe "normalisée" et avons rectifié les entrées erronées.

Codage des données

Dans le processus de codification des informations fournies, nous avons procédé à la catégorisation des champs suivants :

- **wilaya**

  Pour l'attribut "Wilaya", nous avons mis en place un dictionnaire contenant les noms des wilayas comme clés et leurs identifiants numériques correspondants comme valeurs.
  Grâce à un processus de remplacement manuel, nous avons exécuté cette conversion selon le schéma qui suit : Chaque fois qu'une clé (nom de wilaya) était identifiée dans le dictionnaire, nous la remplacions par sa valeur correspondante (identifiant numérique).

- **global_profile**

  En examinant les valeurs uniques de l'attribut "global_profile" (représentant différentes offres), nous avons remarqué un suffixe commun indiquant le type d'offre. Ces suffixes varient selon trois catégories : "Postpayed", "Prepayed" et "Hybrid".

Pour simplifier le traitement, nous avons décidé de remplacer chaque offre par son type respectif.

```
New Hayla Maxi Prepaid_4G (NewHAYLAMAXI)
Play prepaid_4G (Play)
GO prepaid_2G (Go)
Line postpaid_RCLINE1200_4G (Line)
Smart postpaid_RCSMARTS_3G (SMARTPost)
Hayla postpaid_RCHAYLA1000_4G (HaylaB2CPost)
B2C HAAARBA Control_B2CNEWHYBRIDOFFER1500_4G (HAAARBAB2Ctrl)
B2C HAAARBA Control_B2CNEWHYBRIDOFFER1500_3G (HAAARBAB2Ctrl)
B2C Special Control_RCB2CSP1HYB2000_3G (B2CSP1HYB)
```

Figure 3.4: Exemples de types de profils

- **age_years**

  Nous avons classé les clients en trois catégories d'âge : "Adolescent", "Adulte" et "Senior". Ces catégories ont été attribuées en fonction de l'âge.

```python
def label_age(age):
    if 14 <= age <= 19:
        return "Ado"
    elif 20 <= age <= 59:
        return "Adult"
    else:
        return "Senior"
```

Figure 3.5: Codage d'age des clients

Choix des champs importants

Lorsque nous avons atteint le stade de la sélection des caractéristiques déterminantes, une démarche méthodique a consisté d'abord à évaluer diverses métriques par essais et erreurs. En fin de compte, nous avons déterminé que le coefficient de corrélation entre chaque champ et l'attribut cible "churn" était le plus efficace.

La visualisation des résultats de ce processus a été réalisée à l'aide d'une matrice de confusion. La coloration de la matrice de confusion à l'aide d'une carte thermique [*figure 0.14 ci-dessous*] a permis de faire une observation intéressante : le fait que la colonne du taux d'attrition soit toujours ombrée de la même couleur nous a permis de constater que toutes les caractéristiques avaient une importance équivalente. Cette observation nous

a incités à incorporer chaque caractéristique en tant que composant fondamental dans la création de notre modèle.

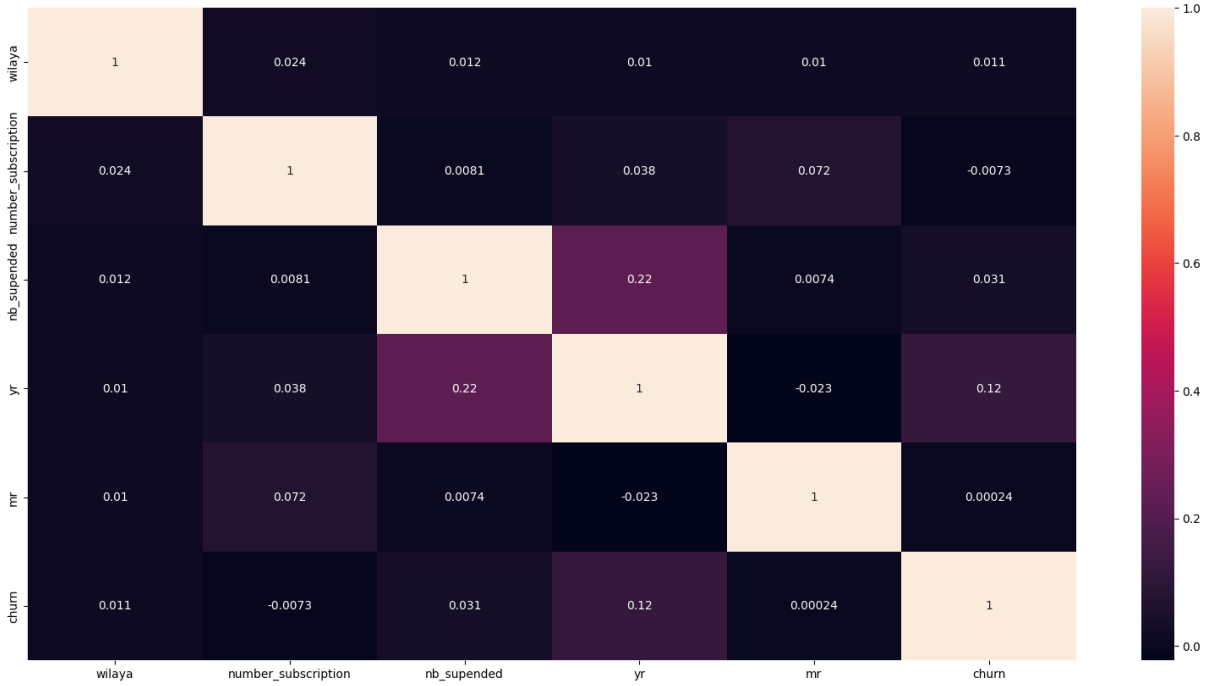


Figure 3.6: Matrice de confusion entre "Churn" et les champs à valeurs numériques

3.1.4 Problème de déséquilibre des données

La table de données fournie par l'établissement d'accueil comprend 24948 lignes (96%) avec un état de non-résiliation (churn = 0), tandis que seulement 1022(4%) lignes correspondent à des résiliations (churn = 1).

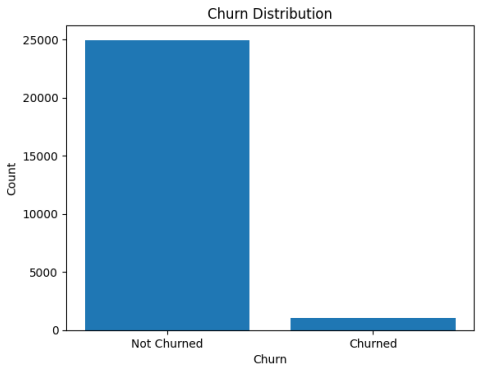


Figure 3.7: Distribution du Churn dans le DataFrame

Cette disparité de poids se traduit par un déséquilibre de 24 dans les données posant

un véritable défi pour l'efficacité du modèle que nous cherchons à construire.

Pour aborder cette question, plusieurs solutions ont été envisagées, que voici :

Choix de la mesure de performance

Étant donné que la variable cible "churn" dispose d'un ensemble de données déséquilibré, l'utilisation de la précision en tant que mesure de performance n'est pas appropriée.

L'utilisation de la précision comme mesure implique un score de précision très élevé. Cela pourrait sembler impressionnant pour un classificateur, mais en réalité, cela serait trompeur(le paradoxe de la précision).
À la place, nous avons choisi d'utiliser le taux de rappel, qui est plus adapté aux problèmes présentant un déséquilibre entre les classes.

Le rappel quantifie le nombre de prédictions positives correctes parmi toutes les prédictions positives, calculé en divisant le nombre total de vrais positifs (TP) par la somme des vrais positifs et des faux négatifs (FN).

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

Ajustement des Hyperparamètres

Certains algorithmes de machine learning permettent de donner davantage d'importance à la classe minoritaire lors de l'entraînement du modèle. Cette approche peut être réalisée en ajustant certains hyperparamètres, notamment le paramètre spécifique au modèle " scale_pos_weight" pour l'algorithme CatBoostClassifier.

En augmentant ce poids, nous accordons une pondération plus élevée aux exemples positifs par rapport aux exemples négatifs lors de l'apprentissage.

L'objectif de cette approche est d'aider le modèle à mieux capturer les caractéristiques de la classe minoritaire, améliorant ainsi sa capacité à prédire correctement ces cas relativement rares.

Le paramètre *stratify*

Au lieu d'utiliser la division par défaut fournie par la fonction train_test_split de scikit-learn, il est recommandé d'utiliser le paramètre "stratify". (consulter Référence 1)

Dans notre cas, où nous avons une forte disproportion de 96% de non churn et seulement 4% de churn dans nos données, l'utilisation de "stratify" garantit que cette même répartition sera préservée à la fois dans l'ensemble d'entrainement,de test et dans l'ensemble de validation.

Cette approche aidera le modèle à identifier les cas de churn malgré leur rareté.

La technique SMOTE

SMOTE est une technique de suréchantillonnage qui génère des échantillons synthétiques pour la classe minoritaire. Cet algorithme aide à résoudre le problème de surajustement qui peut se poser avec le suréchantillonnage aléatoire. Il se focalise sur l'espace des caractéristiques pour créer de nouvelles instances en utilisant une interpolation entre les instances positives qui sont voisines les unes des autres.

Le processus SMOTE commence par sélectionner aléatoirement une instance de la classe minoritaire (positives). Ensuite, les K voisins les plus proches de cette instance sont identifiés à l'aide d'une métrique de distance. Parmi ces K voisins, N sont choisis pour générer de nouvelles instances synthétiques, où N représente le nombre d'instances synthétiques souhaité. Ces nouvelles instances sont créées en ajustant les vecteurs de caractéristiques des voisins en fonction d'une valeur entre 0 et 1, puis en les ajoutant à l'instance d'origine.
Ce processus est répété pour chaque instance de la classe minoritaire jusqu'à ce que le nombre d'exemples synthétiques souhaité soit atteint. (consulter Référence 2)

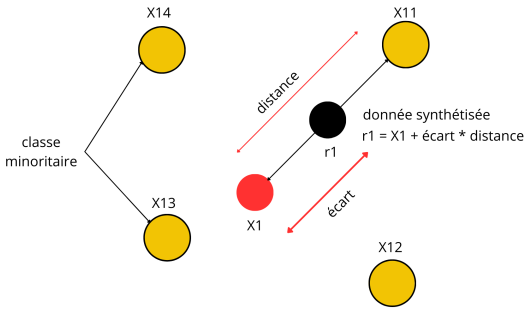


Figure 3.8: Méthode SMOTE

3.2 Implémentation du modèle

3.2.1 Fractionnement de l'ensemble des données

Avant de commencer l'implémentation du modèle,nous avons procédé à la division des données en utilisant la fonction train_test_split de scikit-learn. Cette opération nous a permis de répartir nos données de la manière suivante : 80% pour l'entraînement du modèle et 20% pour le test et la validation.
Les 20% alloués au test et à la validation ont ensuite été subdivisés en 80% pour le test et 20% pour la validation.

Comme mentionné plus haut, nous avons également pris en compte le problème de déséquilibre de classe en utilisant le paramètre stratify pour garantir une répartition équilibrée des classes dans les ensembles de données.

3.2.2 Comparaison des Algorithmes de Classification

Après avoir terminé l'analyse et le nettoyage des données, nous sommes passés à la phase de conception du modèle de prédiction.

Au cours de cette phase, nous avons conduit une série de tests afin de sélectionner l'algorithme de classification le plus approprié à notre problème. Pour rappel, afin d'équilibrer les données, nous avons utilisé la méthode SMOTE expliquée précédemment.

Les résultats des tests réalisés sur les différents modèles sont présentés dans le tableau ci-dessous.

Algorithme utilisé	**Semote 2**	**Matrice de confusion**	**Taux de rappel**
Regression Logistique	Sans	$\begin{matrix}981 & 0\\ 58 & 0\end{matrix}$	0
	Avec	$\begin{matrix}765 & 216\\ 24 & 34\end{matrix}$	0.58
Forêt d'arbres décisionnels	Sans	$\begin{matrix}981 & 0\\ 58 & 0\end{matrix}$	0
	Avec	$\begin{matrix}849 & 132\\ 31 & 27\end{matrix}$	0.46
k-moyennes	Sans	$\begin{matrix}981 & 0\\ 58 & 0\end{matrix}$	0
	Avec	$\begin{matrix}981 & 0\\ 58 & 0\end{matrix}$	0
AdaBoost Classifier	Sans	$\begin{matrix}981 & 0\\ 58 & 0\end{matrix}$	0
	Avec	$\begin{matrix}783 & 198\\ 21 & 37\end{matrix}$	0.63
XGBoost Classifier	Sans	$\begin{matrix}976 & 5\\ 57 & 1\end{matrix}$	0.017
	Avec	$\begin{matrix}965 & 16\\ 55 & 3\end{matrix}$	0.05
Analyse discriminante linéaire	Sans	$\begin{matrix}981 & 0\\ 58 & 0\end{matrix}$	0
	Avec	$\begin{matrix}756 & 225\\ 25 & 33\end{matrix}$	0.56
k plus proches voisins	Sans	$\begin{matrix}979 & 2\\ 58 & 0\end{matrix}$	0
	Avec	$\begin{matrix}826 & 155\\ 31 & 27\end{matrix}$	0.46
Gradient boosting	Sans	$\begin{matrix}981 & 0\\ 58 & 0\end{matrix}$	0
	Avec	$\begin{matrix}849 & 132\\ 32 & 26\end{matrix}$	0.44
CatBoost Classifier	Sans	$\begin{matrix}981 & 0\\ 58 & 0\end{matrix}$	0
	Avec	$\begin{matrix}783 & 198\\ 21 & 40\end{matrix}$	0.64

Table 3.2: Résultats obtenus par les différents classificateurs

3.2.3 Choix du modèle

Au vu des résultats obtenus, le taux de rappel enregistré avec le modèle **CatBoostClassifier** est le plus élevé. Nous avons alors retenu cet algorithme en raison de sa performance de **0.64** (en appliquant la méthode Smote). Cette décision découle également d'autres facteurs, notamment l'efficacité de CBC dans la gestion des caractéristiques catégoriques, ainsi que sa capacité à gérer les déséquilibres de classe.

3.2.4 CatBoostClassifier

Le CatBoostClassifier est un algorithme d'apprentissage automatique capable de traiter efficacement les caractéristiques catégorielles et numériques. Il utilise une combinaison de techniques telles que le **boosting ordonné**, les **permutations aléatoires** et l'**optimisation basée sur le gradient** pour obtenir de bonnes performances sur un large éventail d'ensembles de données, y compris ceux comportant des caractéristiques catégorielles complexes. (consulter Référence 4)

Principe de fonctionnement (consulter Référence 5)

Le CatBoost est basé sur la technique du boosting ordonné, une approche visant à résoudre le problème du décalage de prédiction, qui se produit lorsque le modèle utilise des informations qu'il a déjà apprises des étapes précédentes de l'apprentissage pour effectuer des prédictions sur de nouvelles données.

À chaque itération, un nouvel ensemble de données d'apprentissage est créé, garantissant que le modèle ne voit pas les mêmes données qu'auparavant et n'a pas accès aux étiquettes cibles de ces nouvelles données. Le modèle est formé sur ces données sans aucune connaissance préalable des étiquettes, ce qui empêche les biais indésirables.

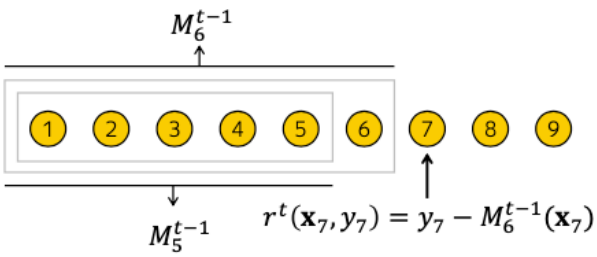


Figure 3.9: Principe de renforcement ordonné, ordonné par σ

Cette approche consiste à générer s+1 permutations aléatoires indépendantes, $\sigma_0, \sigma_1, \ldots, \sigma_s$, à partir de l'ensemble de données d'entraînement. Ces permutations sont utilisées pour construire des arbres de décision symétriques, assurant ainsi l'optimisation de l'efficacité en termes de temps de calcul.

Pour chaque échantillon i, un gradient est calculé :

$$grad_{r,j}(i) = \frac{\delta L(y_i, M_{r,j}(i))}{\delta s}$$

La valeur de la feuille pour l'arbre construit en fonction de l'i-ème échantillon est déterminée comme la moyenne des gradients calculés, ce qui peut être exprimé comme :

$$avg(grad_{r,\sigma r(i)-1})$$

Chaque arbre capture des informations spécifiques. Une fois construits, ces arbres contribuent à un modèle global : optimisation basée sur le gradient, améliorant ainsi la précision des prédictions grâce à différentes perspectives.

3.2.5 Amélioration du modèle

Étant donné que l'algorithme CatBoost Classifier offre une option pour équilibrer les données sans nécessiter l'utilisation de la méthode Smote, nous avons choisi d'exploiter cette fonctionnalité en ajustant le paramètre : scale_pos_weight.
De plus, nous avons créé une fonction grâce à *Optuna* (voir Annexe B.) pour effectuer une optimisation automatisée des hyperparamètres du modèle CatBoostClassifier. Cette fonction recherche les hyperparamètres qui maximisent la métrique "recall" en utilisant des essais itératifs. Cette approche permet d'optimiser efficacement les paramètres du modèle en améliorant sa capacité à détecter les vrais positifs.

Personnalisation des Hyperparamètres

Pour améliorer les performances d'un modèle, il est essentiel d'ajuster méticuleusement ses hyperparamètres. Ci-dessous les hyperparamètres sur lesquels nous avons travaillé.

- **colsample_bylevel** : Fraction de caractéristiques à considérer à chaque niveau de l'arbre.
- **depth**: Profondeur maximale des arbres de décision.
- **boosting_type**: Type d'algorithme de boosting utilisé (dans notre cas, "Ordered").

- **bootstrap_type**: Mécanisme d'échantillonnage (dans notre cas, "Bernoulli").
- **scale_pos_weight**: Poids relatif des exemples positifs par rapport aux négatifs. Il est calculé comme le rapport entre le nombre d'exemples négatifs (churn=0) et le nombre d'exemples positifs (churn=1), utilisé pour traiter les problèmes de déséquilibre de classe.
- **random_seed**: Graine aléatoire pour la reproductibilité des résultats.
- **thread_count**: Le nombre de threads utilisés pour l'entraînement, généralement défini sur -1 pour exploiter tous les threads disponibles.
- **iterations**: Nombre d'itérations d'apprentissage.
- **learning_rate**: Taux d'apprentissage pour la mise à jour des poids.
- **l2_leaf_reg**: Régularisation L2 appliquée aux poids, elle permet de pénaliser les poids élevés. Cela signifie que le modèle est encouragé à avoir des poids plus petits, ce qui peut aider à simplifier le modèle et à le rendre moins susceptible de surajuster les données d'entraînement.
- **subsample**: Fraction d'exemples d'entraînement utilisée à chaque itération, elle permet de contrôler la quantité de données utilisée à chaque étape de l'apprentissage.
- **loss_function**: Fonction de perte utilisée pour évaluer la performance du modèle, dans notre cas, "Logloss" signifie la perte logarithmique, souvent utilisée pour les problèmes de classification. Elle mesure la différence entre les prédictions du modèle et les vraies étiquettes de classe.

3.2.6 Résultats du test

Les résultats du test du modèle, après avoir sélectionné et ajusté les hyperparamètres, sont résumés dans le tableau ci-dessous.

Matrice de confusion	**Taux de rappel**
$\begin{smallmatrix} 665 & 333 \\ 1 & 38 \end{smallmatrix}$	0.93

Table 3.3: Résultat du test du modèle CatboostClassifier aprés réglage des hyperparamètres

L'apport de la personnalisation des hyperparamètres est remarquable au vu du score de rappel qui est passé de 0.64 à 0.93.

3.2.7 Validation du modèle

Nous avons utilisé la cross validation sur notre modèle pour valider les résultats obtenus précédemment.

Validation croisée

Il s'agit d'une approche privilégiée en raison de sa simplicité et de sa tendance à fournir une évaluation plus impartiale et plus réaliste des performances du modèle par rapport à d'autres solutions telles que la division train/test basique.

La validation croisée consiste à évaluer les modèles d'apprentissage automatique à l'aide d'un petit échantillon de données.
Elle utilise un paramètre appelé k, qui représente le nombre de groupes (plis) dans lesquels les données sont divisées.

Cette technique est principalement utilisée dans la pratique de l'apprentissage automatique (ML) pour évaluer les performances d'un modèle sur de nouvelles données non vues.

Voici le processus global :

1. Mélanger aléatoirement l'ensemble des données.
2. Diviser l'ensemble des données en k groupes distincts.
3. Pour chaque groupe :
    - L'utiliser comme ensemble de données de test.
    - Employer les autres groupes comme ensemble de données d'entraînement.
    - Former un modèle sur les données de formation et l'évaluer sur les données de test.
    - Conserver le résultat de l'évaluation et rejeter le modèle.
4. Compiler un résumé des performances du modèle à l'aide de l'ensemble des scores d'évaluation.

Résultats de la validation

En expérimentant différentes valeurs de plis, à savoir 5, 10 et 20 afin d'atténuer les risques de surajustement (overfitting) et de sous-ajustement (underfitting), il est à noter que le résultat le plus favorable, avec un taux de rappel de 0.85, a été obtenu lorsque l'ensemble des données de validation a été divisé en 20 groupes. Ce résultat est cohérent avec les performances précédemment observées (résultat du test).

3.3 Interface graphique

3.3.1 Serialisation du modele

Afin de garantir une utilisation fluide du modèle entraîné pour notre application de prédiction de churn, nous avons choisi d'utiliser la bibliothèque Python Pickle, un module intégré.

Pickle simplifie la sérialisation de l'objet modèle, nous permettant ainsi de le stocker sous forme de fichier binaire. Plus précisément, nous avons employé la fonction pickle.dump() pour enregistrer le modèle entraîné dans un fichier nommé 'churnModel.pkl'.

Lorsqu'il est récupéré ultérieurement à l'aide de la fonction pickle.load(), ce fichier binaire a la capacité de reconstituer le modèle dans son intégralité, y compris son architecture et les poids entraînés.

3.3.2 Interface graphique

Pour simplifier l'utilisation du modèle, nous avons créé une interface graphique.

Cette interface permet aux utilisateurs du modèle travaillant chez Djezzy de prédire les cas de churn en saisissant le nom du fichier CSV où se trouve le dataset ou en entrant directement les informations du client.

L'interface graphique du modèle est présentée en Annexe C.

Conclusion

Le présent travail entre dans la cadre des efforts fournis par Djezzy pour garder son portefeuille client. Nous avons proposé une solution basée sur les techniques d'apprentissage automatique (ML) où nous appliquons une variété d'algorithmes de pointe à un ensemble de données méticuleusement conservées fournies par l'institution hôte. Notre objectif est de sélectionner la solution optimale après une évaluation approfondie des résultats obtenus.

Les données fournies par Djezzy ont été l'objet d'analyse profondes et de prétraitements. Nous avons abordé le problème du déséquilibre des données et suggéré l'utilisation de techniques telles que SMOTE pour résoudre ce problème.

La mise en œuvre de notre modèle implique le partitionnement des données, la comparaison des algorithmes et la sélection du modèle le plus approprié. Nous avons choisi l'algorithme "CatBoostClassifier" au vu du meilleur taux de rappel qu'il a donné (0.64) comparé aux autres algorithmes. Néanmoins, nous avons apporté des améliorations complémentaires pour renforcer ses performances en procédant au réglage de ses hyperparamètres. Les résultats de nos tests ont démontré l'efficacité du modèle proposé pour prédire le taux de perte de clientèle. Notre modèle amélioré a atteint un taux de rappel de 0.92.

Par ailleurs, nous avons développé une interface graphique pour faciliter l'utilisation de notre modèle d'apprentissage automatique. Cette interface fournit une plateforme conviviale permettant aux opérateurs d'accéder aux prédictions du modèle et de les interpréter.

Annexes

Annexe A : Technologies utilisées au sein de Djezzy

Ces outils font partie de l'arsenal d'outils utilisés au sein de Djezzy, comme précisé dans la section "Son architecture".

Informatica PowerCenter

> "Une solution ETL permettant de répondre avec efficacité à un très large éventail de besoins : traiter les données volumineuses à partir de données en entrée appelées source vers des destinations SGBD ou fichiers (csv, txt, xml…) appelées cibles."

[1]

Teradata BTEQ

C'est un utilitaire (fonctionnant principalement en recevant des commandes ou des instructions directes de l'utilisateur : Djezzy) piloté par des commandes qui permet aux utilisateurs d'interagir avec un ou plusieurs systèmes de base de données Teradata.

SAP BusinessObjects BI

> "SAP BusinessObjects Business Intelligence est une suite centralisée pour le reporting, la visualisation et le partage des données. En tant que couche de BI sur site pour la Business Technology Platform de SAP, elle transforme les données en informations utiles, disponibles à tout moment et en tout lieu."

[2]

[1] Source : https://www.alphorm.com/tutoriel/formation-en-ligne-informatica-powercenter-niveau-debutant
[2] Source : https://www.sap.com/products/technology-platform/bi-platform.html

Qlik Sense

”Qlik Sense est une plateforme d’analyse de données. Elle applique le principe de business intelligence pour améliorer la pertinence des résultats de son moteur associatif. Le système se base sur un hébergement dans le cloud. Les utilisateurs peuvent ainsi accéder aux outils et aux données, tout en anticipant leurs contraintes de mobilité.”

[3]

Hortonworks

L’éditeur d’Hortonworks Data Platform (HDP), une plate-forme de données basée sur Hadoop qui comprend entre autres les systèmes Hadoop Distributed File System (HDFS), Hadoop MapReduce, Apache Pig, Apache Hive, Apache HBase et Apache ZooKeeper. Cette plate-forme est utilisée pour analyser, stocker et manipuler de grandes quantités de données. ”

[4]

Nifi

”Un logiciel libre de gestion de flux de données. Il permet de gérer et d’automatiser des flux de données entre plusieurs systèmes informatiques, à partir d’une interface web et dans un environnement distribué.”

[5]

Apache Hive

”Une infrastructure d’entrepôt de données intégrée sur Hadoop permettant l’analyse, le requêtage via un langage proche syntaxiquement de SQL ainsi que la synthèse de données”

[6]

[3]Source : https://www.journaldunet.fr/web-tech/guide-de-l-entreprise-digitale/1499131-qlik-sense-la-plateforme-qui-met-l-analytics-a-la-portee-de-tous/

[4]Source : https://fr.wikipedia.org/wiki/Hortonworks

[5]Source : https://fr.wikipedia.org/wiki/Apache_NiFi

[6]Source : https://fr.wikipedia.org/wiki/Apache_Hive

Apache Spark

”un framework open source de calcul distribué. Il s’agit d’un ensemble d’outils et de composants logiciels structurés selon une architecture définie. Développé à l’université de Californie à Berkeley par AMPLab3, Spark est aujourd’hui un projet de la fondation Apache. Ce produit est un cadre applicatif de traitements des mégadonnées (big data) pour effectuer des analyses complexes à grande échelle.”

[7]

Apache Kafka

”Une plateforme open-source de streaming d’événements distribués utilisée par des milliers d’entreprises pour les pipelines de données haute performance, l’analyse en continu, l’intégration de données et les applications critiques.”

[8]

Apache Ignite

”Un système de gestion de base de données distribué pour le calcul haute performance. La base de données d’Apache Ignite utilise la RAM comme niveau de stockage et de traitement par défaut, appartenant ainsi à la classe des plates-formes informatiques en mémoire”

[9]

Apache Cassandra

”Un système de gestion de base de données de type NoSQL conçu pour gérer des quantités massives de données sur un grand nombre de serveurs, assurant une haute disponibilité en éliminant les points de défaillance unique.”

[10]

[7] Source : https://fr.wikipedia.org/wiki/Apache_Spark
[8] Source : https://kafka.apache.org/
[9] Source : https://en.wikipedia.org/wiki/Apache_Ignite

Annexe B : Environnement de développement

Voici une présentation des outils de développement qui ont été mentionnés dans le rapport.

Python

> "Python est un langage de programmation interprété, multiparadigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet."

[11]

Visual Studio Code

> "Un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré"

[12]

Jupyter Notebook

> "Une application web open-source permettant de créer et de partager des documents contenant du code, des équations, des visualisations, et du texte narratif. Anciennement appelé IPython Notebooks, il s'agit d'un environnement de calcul interactif basé sur le web permettant de créer des documents notebooks. "

[13]

Optuna

> "Optuna est un outil de recherche automatisé permettant l'optimisation des hyperparamètres de vos modèles en Machine Learning."

[14]

[11] Source : https://fr.wikipedia.org/wiki/Python_(langage)
[12] Source : https://fr.wikipedia.org/wiki/Visual_Studio_Code
[13] Source : https://datascientest.com/jupyter-notebook-tout-savoir

Annexe C : Interface graphique

Pour créer l'interface utilisateur graphique (GUI), nous avons utilisé la bibliothèque tk-inter, qui est la bibliothèque GUI standard pour Python.
Tkinter offre un moyen pratique de créer plusieurs composants d'interface graphique, ce qui en fait un bon choix pour la création d'applications interactives comme celle-ci.

Les figures ci-dessous présentent l'interface graphique développée pour la prédiction du churn par notre modèle.

Prédiction manuelle

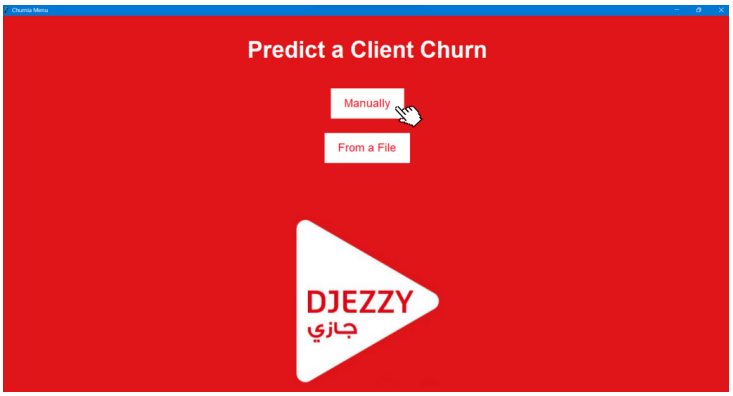


Figure 4.1: Interface : Interface d'accueil

En cliquant sur le bouton "Manually", une interface d'information apparaîtra. L'utilisateur devra alors remplir les informations du client.

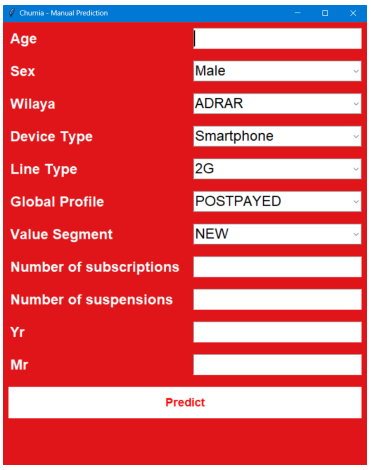


Figure 4.2: Interface : Prédiction manuelle

En cliquant sur le bouton "Predict", le résultat de la prédiction de churn de ce client

s'affichera : en rouge en cas de churn et en vert en cas de non-churn.

(a) Interface : Cas de non-churn

(b) Interface : Cas de churn

Figure 4.3: Interface : Résultats de prédiction manuelle

Prédiction par fichier CSV

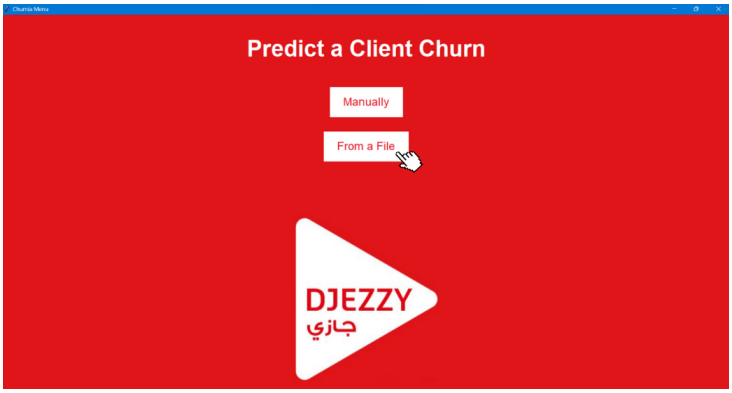


Figure 4.4: Interface : Prédiction à partir d'un fichier CSV

En cliquant sur le bouton 'From a file', une fenêtre de sélection de fichier de l'ordinateur s'ouvre, permettant de choisir un fichier CSV.

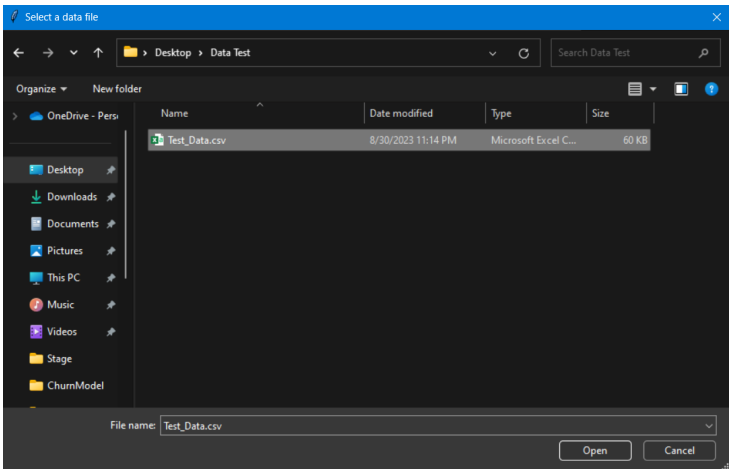
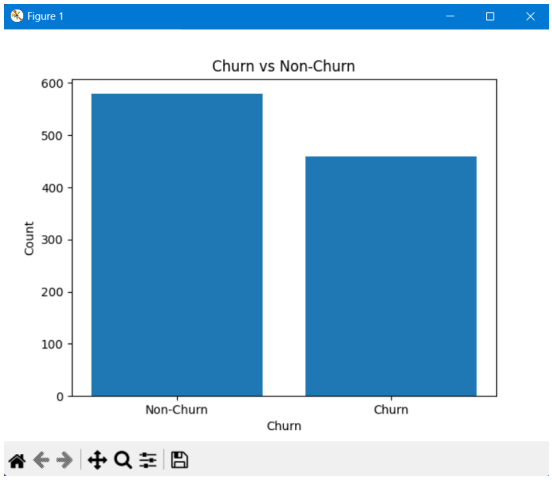


Figure 4.5: Interface : Selection d'un fichier CSV

Les résultats de la prédiction du fichier 'Test_Data.csv' sélectionné s'affichent dans les figures ci-dessous.



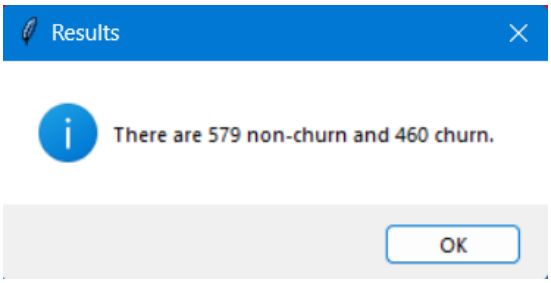


Figure 4.6: Interface : Résultats de prédiction à partir d'un fichier CSV

Références

MENON, Saaransh (**2020**). "Stratified sampling in Machine Learning." *Medium*. Disponible sur : https://medium.com/analytics-vidhya/stratified-sampling-in-machine-learning-f5112b5b9cfe. (Consulté le **18 août 2023**)

SATPATHY, Swastik (**2020**). "SMOTE for Imbalanced Classification with Python." *Analytics Vidhya*. Disponible sur : https://www.analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techniques/. (Consulté le **18 août 2023**)

AGRAWAL, Raghav (**2023**). "Building Customer Churn Prediction Model With Imbalance Dataset." *Analytics Vidhya*. Disponible sur : https://www.analyticsvidhya.com/blog/2023/02/building-customer-churn-prediction-model-with-imbalance-dataset/. (Consulté le **20 août 2023**)

OPPERMANN, Artem (**2023**). "What Is CatBoost?" *builtin*. Disponible sur : https://builtin.com/machine-learning/catboost. (Consulté le **23 août 2023**)

Jessie Man Wai Chin, Yi Lin Ooi, Yaqi Shi, Shwen Lyng Ngew (**2021**). "CatBoost: unbiased boosting with categorical features." *statwiki*. Disponible sur : https://wiki.math.uwaterloo.ca/statwiki/index.php?title=CatBoost:_unbiased_boosting_with_categorical_features#:~:text=In%20ordered%20boosting%2C%20a%20new,set%20of%20new%20training%20samples. (Consulté le **23 août 2023**)