

Rapport de Stage de fin d'études
Data scientist junior
Marketing prédictif : prédiction du choix d'un
prospect



1 avril 2021 – 31 août 2021

Tuteur de stage académique : Jean-Philippe Kotowicz

Tuteur de stage professionnel : OUANOUNOU Jacob



Mgen technologies

76 Av. de Fontainebleau, 94270 Le Kremlin-Bicêtre

Table des matières

1)Remerciements :	4
2)Introduction :	5
3) Présentations des entreprises et l'équipe :	6
3.1) HN services :	6
3.2) MGEN :	7
3.3) L'équipe :	8
4)Cadre de stage :	9
4.1) Contexte générale :	9
4.2) La méthodologie :	10
4.3) Environnement de travail :	11
5) Les travaux réalisées :	12
5.1) Description des données :	12
5.1.1) Sources des données :	12
5.1.2) Compositions des données :	13
5.2) Nettoyage des données :	13
5.3) Statistiques descriptives :	14
5.4) Pré traitement et encodage des données :	19
5.5) Modélisation et performances :	20
5.5.1) Les métriques d'évaluation :	20
5.5.2) Acceptation ou refus de la proposition :	21
5.5.3) Classification multi class :	26
5.5.4) Industrialisation du modèle :	28
7)Conclusion	31
8)Bibliographie :	32
9)Annexes :	33

1)Remerciements :

Pour commencer je tiens à remercier tout le corps enseignant et encadrant du département Génie mathématique pour la formation et les valeurs que j'ai acquises pendant ces trois années.

Je tiens à remercier l'entreprise Hn-Services pour m'avoir permis d'effectuer mon stage chez l'un de ses clients ainsi que l'ensemble de ses personnels pour l'accueil et le suivi tout au long de stage.

De même, je souhaite remercier, tout particulièrement, mon tuteur de stage professionnel, Jacob Ouanounou aux côtés duquel j'ai pris beaucoup de plaisir à travailler. Ses précieux conseils, sa disponibilité et son dynamisme m'ont été utiles au quotidien dans mon travail et pour la rédaction de ce rapport.

Je remercie toute l'équipe Data science au sein de MGEN, dirigé par François Lieu, pour son accueil chaleureux, la bonne ambiance de travail qui a rendu ce stage très enrichissant.

Je remercie l'ensemble du personnel de MGEN Technologies, pour son aide et sa sympathie.

2)Introduction :

Ce rapport est un bilan du travail effectué lors de mon stage ingénieur au sein de l'entreprise Hn-Services du 1 avril 2021 au 31 août 2021.

Ce stage se positionne comme une première expérience professionnelle sur un projet concret en data science. En effet les tâches qui m'ont été confiées m'ont permis d'être impliqué sur plusieurs cas d'usage tout en manipulant des données massives de l'entreprise et appliquer mes compétences mathématiques et informatique en parallèle.

HN services est une société de services numériques, anciennement SS2I. Le stage a été effectué chez l'un de ses clients qui est la MGEN. C'est la mutuelle générale de l'éducation nationale avec 4 millions d'adhérents. Elle est la première mutuelle de santé en France en nombre d'adhérents et en cotisations. Avec les nombreux services qu'elles proposent (santé, prévoyance...) la Mgen possède un énorme entrepôt de données alimenté chaque jour par les données des adhérents. Afin d'exploiter ces données et améliorer la productivité et l'efficacité des services proposées l'entreprise a décidé de créer une équipe data science. J'ai intégré cette équipe de sa création comme un data scientist junior. En effet cette équipe travaille sur plusieurs projets en coordination avec les équipes métiers et projets.

Pendant mon stage j'ai travaillé sur le périmètre marketing dont le but était de permettre aux équipes marketing d'anticiper le choix d'un prospect (client potentiel) à partir de ses informations personnelles et la proposition que lui sera faite. Cette mission a été faite sur plusieurs étapes :

- Compréhension des besoins et repérage des données.
- Extraction des données et nettoyage.
- Pré-traitement des données et data visualisation.
- Modélisation et entraînement des modèles.
- Evaluation des performances et choix du modèle.
- Création d'une API afin d'industrialiser le projet.

Dans ce rapport je vais présenter les entreprises et l'équipe, les tâches effectuées, les technologies utilisées ainsi que les résultats trouvés et les perspectives.

3) Présentations des entreprises et l'équipe :

3.1) HN services :

Fondé en 1983 par Michel HOCHBERG et Lionel NIQUET sous le nom de Hochberg Niquet Sarl, société anonyme à responsabilité limitée. Le statut de l'entreprise est modifié et devient en 1990 HN services SA. Aujourd'hui la société est implantée dans 5 autres pays, en Roumanie (Bucarest), aux USA (new York), Espagne (Madrid), Portugal (Lisbonne) et Luxembourg.



La société est en pleine croissance depuis 2012, en 2020 elle a atteint 90 M€ des chiffres d'affaires et l'effectif regroupe plus que 1200 collaborateurs.

HN services est une société de services numérique (ESN). L'expertise reconnue d'HN lui permet d'accompagner ses clients, principalement des grands comptes, dans les domaines de la banque/assurance, de la retraite/prévoyance, du retail, de l'industrie et des services :

- Conseil : Business Strategy, IT Transformation.
- MOE : analyse et conception, développement, gestion de projets.
- Développement et Management d'application : expertise technique, modèles contractuels variés.
- CRM : utilisation de Salesforce pour le traitement des données.

HN services propose des différents modes d'interventions dans le but d'accompagner leurs clients dans leurs transformations et leurs activités : centre de compétences ou de services, mission, projet, maintenance.

HN institut :

En 1989 HN services a créé un centre de formation appelé HN institut. Ce centre a permis à des milliers des collaborateurs, venant des divers domaines, d'avoir une formation adaptée et de commencer une carrière dans le milieu informatique. Ces formations permettent de répondre aux besoins spécifiques des clients. En 2019 HN a fêté les 30 ans de son école de formation, HN Institut. Plus de 7000 jeunes y ont depuis fait leurs premières armes.

3.2) MGEN :

Une mutuelle ou complémentaire de santé est un organisme de protection sociale à but non lucratif relevant des domaines de santé, de la prévoyance et de l'assurance. Les mutuelles sont des sociétés de personnes et non de capitaux, sans actionnaire à rémunérer. Les fonds de ce type de société proviennent des cotisations des membres et leur fonctionnement est régi par le Code de la Mutualité. Elle est indépendante de tout pouvoir politique, financier ou syndical.

La MGEN est la mutuelle générale de l'éducation nationale fondée en 1946 par les responsables syndicaux et mutualistes du syndicat national des Instituteurs de la fédération de l'éducation nationale. Le groupe est un majeur acteur dans la protection sociale avec plus de 4 millions d'adhérents, 10000 salariés et plus que 2 milliards d'euro comme chiffre d'affaires en 2020. Elle se positionne comme la première mutuelle de la fonction publique et aussi la première mutuelle santé en nombre de cotisations.

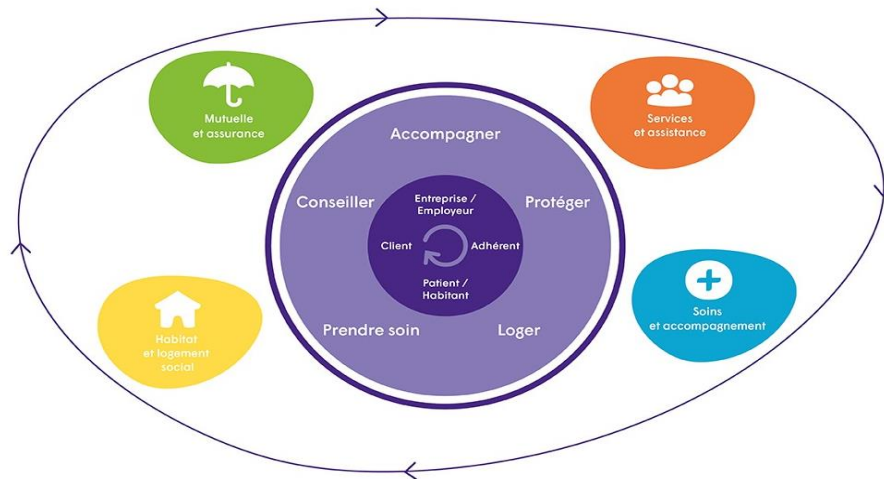
La particularité de la Mgen est le fait de gérer la sécurité sociale (régime obligatoire) de quelques personnels : l'Éducation nationale, de l'Enseignement supérieur, de la Recherche et de l'Innovation, de la Culture et de la Communication, des Sports, de la Jeunesse, de l'Éducation populaire et de la Vie associative et du Développement Durable. Elle propose de même une complémentaire santé individuelle ouverte à tous les publics depuis 2015 via MGEN Filia. L'objectif de la MGEN est de faciliter l'accès aux soins tout en couvrant le plus possible les besoins en protection sociale de ses adhérents.

Depuis le 13 septembre 2017 le groupe MGEN fait partie du Groupe VYV. Il a rejoint MNT, MGEFI, Harmonie Fonction Publique et Mutuelle Mare-Gaillard.



Le groupe VYV protège 10 millions des personnes et compte 40000 collaborateurs. IL devient le 1er acteur mutualiste De santé et de protection sociale en France.

Le Groupe VYV a été créé avec l'objectif de constituer un pôle mutualiste de santé et de protection sociale tout en s'appuyant sur les savoir-faire de ses 4 métiers :



3.3) L'équipe :

Au sein de la MGEN l'équipe Data science a été créée en mars 2021 et a pour rôle d'analyser les données de l'EDE (entrepôt des données) et les appliquer dans les différents projets métiers (risque, marketing, fraude...). L'équipe est composée de 3 stagiaires, 2 data scientist internes et un consultant. Afin d'assurer ses missions l'équipe s'appuie sur la méthodologie agile ou tous les membres communiquent énormément : une réunion chaque matin qui dure entre 15 et 30 minutes ou chacun parle des tâches effectuées la veille, les problèmes rencontrés et les tâches prévus pour la journée. Une réunion d'une 1h chaque fin de semaine afin de faire un compte rendu. Pour le suivi des projets, l'équipe organise des sprints d'une durée de 3 semaines ou le suivi est fait à l'aide d'un outil appelé JIRA.



Vu les conditions sanitaires compliquées et le fait de travailler à 80% à la maison, cette méthodologie a été très bénéfique afin de garder les contacts avec tous les membres de l'équipe et de se sentir comme en présentiel. Je suis arrivé en même temps que la majorité des membres de l'équipe ce qui m'a facilité l'intégration.

4)Cadre de stage :

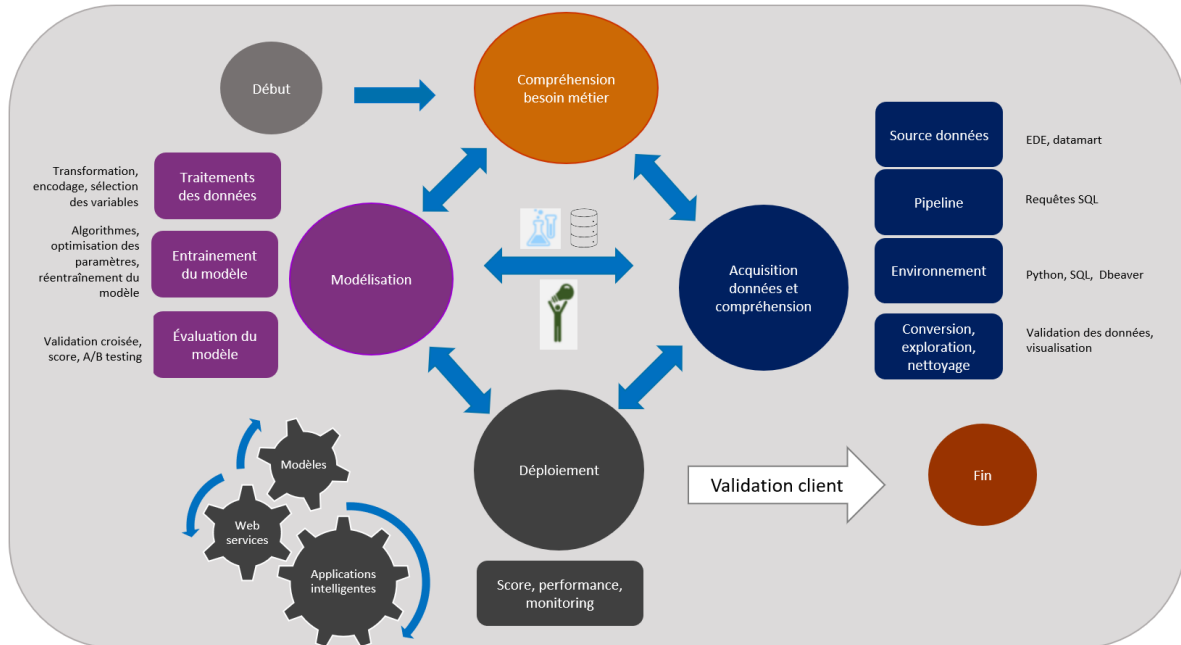
4.1) Contexte générale :

La MGEN gère le régime obligatoire des personnels de l'Education nationale et de l'Enseignement. Elle propose également des offres complémentaires individuelle. Une personne souscrite à au régime obligatoire à la MGEN n'est pas obligé à choisir la MGEN pour la complémentaire santé.



Dans le but de renforcer sa notoriété et ainsi augmenter son nombre d'adhérents au régime complémentaire, La MGEN essaie de contacter plusieurs prospects chaque mois et le proposer des offres afin de le convaincre de rejoindre la mutuelle. Pour ça elle dépense 10 euros pour la prospection de chaque personne (cout moyen) et du coup 2000000 euro pour 200000 personnes par an. Or chaque année on a qu'un tiers des propositions qui sont acceptés alors l'équipe data science a décidé de mettre en place un modèle de machine Learning permettant d'anticiper le choix d'un prospect à partir de ses informations personnelles et quelques informations sur la proposition. En effet ce modèle va permettre dans un premier temps de prédire si la personne aura plus la tendance ou pas à accepter la proposition et dans un deuxième temps de savoir les offres qui vont lui intéresser le plus. Ce projet va permettre de cibler que les personnes intéressées et du coup réduire les couts de prospection.

4.2) La méthodologie :



La méthodologie suivie tout au long du projet est comme le montre le graphe au-dessus. En effet comme chaque projet data science ça commence par la compréhension des besoins métiers en se basant sur les requêtes SQL. Puis on aborde la phase de la collecte des données ou on les extrait à partir de l'entrepôt des données. Ces données seront nettoyées et visualisées afin de les analyser. On passe à une phase de pré traitement (encodage, transformation des variables) afin d'appliquer les modèles de machine Learning appropriés et atteindre l'objectif. On évalue les performances de nos modèles en utilisant des métriques spécifiques et on essaie d'optimiser le modèle final en jouant sur les hyperparamètres.

A la fin on arrive à la phase d'industrialisation ou le modèle sera mis à disposition de tous les utilisateurs notamment l'équipe métier concerné par le projet.

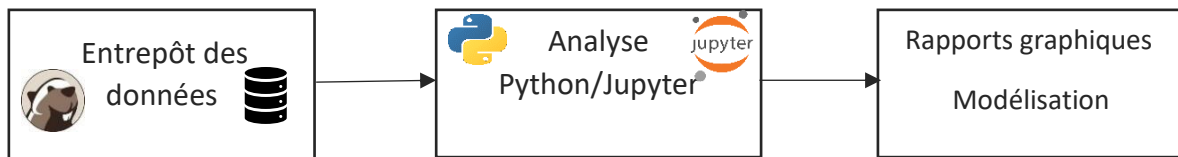
4.3) Environnement de travail :

La mise en œuvre de solutions Big Data, permettant la réalisation de projets de Data Science, n'est possible que si l'on dispose d'un environnement adapté. En effet, l'estimation et la calibration des modèles nécessitent d'importantes ressources à la fois pour le stockage des données (stockage physique) et pour leur traitement (mémoire RAM et processeurs).

Afin d'effectuer ce stage, La MGEN m'a permis d'avoir une machine puissante : 16 Go de RAM et i7 comme processeur. Cette configuration m'a beaucoup facilité le travail.

L'entrepôt des données utilisé par la MGEN est de type netezza, entrepôt des données de hautes performances, et pour requêter les données on utilise le logiciel dbeaver permettant l'administration et le requêtage de base de données avec SQL.

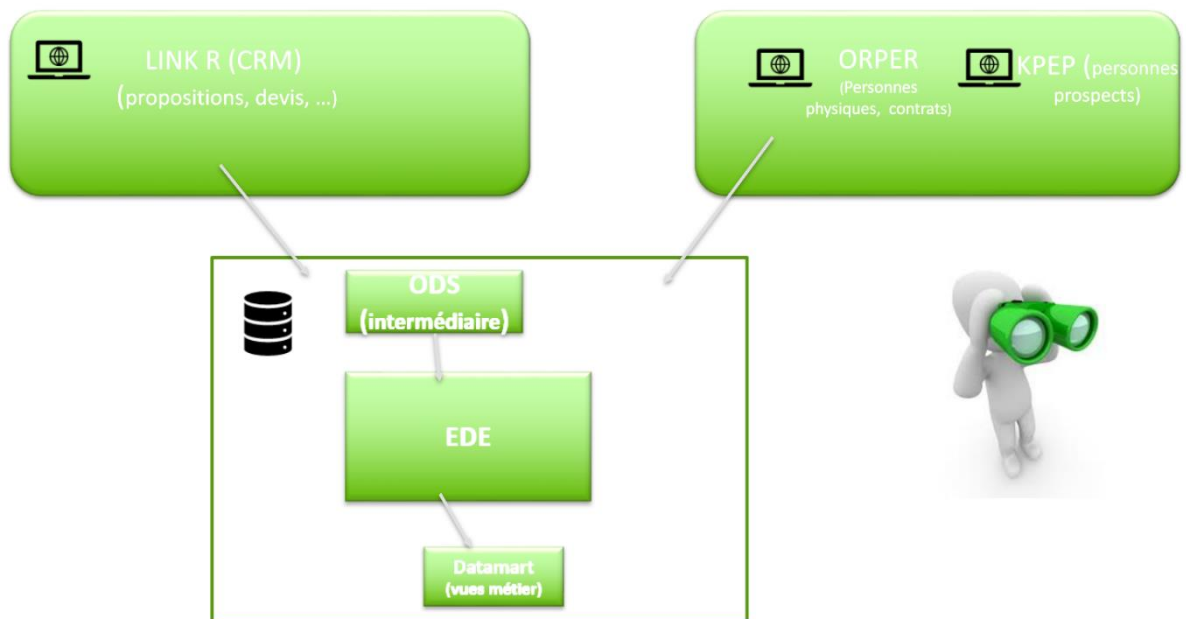
Le langage de programmation utilisé est python, un langage interprété de programmation Open Source, avec jupyter notebook. Ce dernier étant très en vogue car il est l'un des langages les plus accessibles, du fait de son apprentissage rapide et intuitif. La popularité de Python en tant que langage de programmation pour le Big Data provient également de ses différents packages et bibliothèques de science des données. Anaconda est une distribution Python mettant à disposition Conda, un outil de gestion de librairies. Celui-ci permet de les mettre à jour et d'installer plus facilement les librairies dont on a besoin.



5) Les travaux réalisés :

5.1) Description des données :

5.1.1) Sources des données :



Le schéma au-dessus présente le flux des données pour le périmètre sur lequel j'ai travaillé. En effet toutes les données à propos des propositions sont contenues dans les tables CRM. Le flux des données provient de systèmes de gestion commercial LINK R et KPEP. Les données sont fournies quotidiennement 7 jours sur 7, des systèmes de gestion vers le système Décisionnel et elles sont chargées la nuit.

Les données étudiées sont les résultats des différents extractions et jointures faites sur les tables CRM ainsi que d'autres tables référentielles permettant de récupérer les libellés des champs. Afin que les jointures fonctionnent correctement je me suis basé sur le modèle relationnel de l'EDE et trouvé les bonnes relations entre les tables. Chaque prospect possède un IDKPEP présentant son identifiant unique dans la base des données. Chaque proposition possède son numéro unique IDPAL. Ces deux champs ont permis de faire la plupart des jointures entre les tables.

5.1.2) Compositions des données :

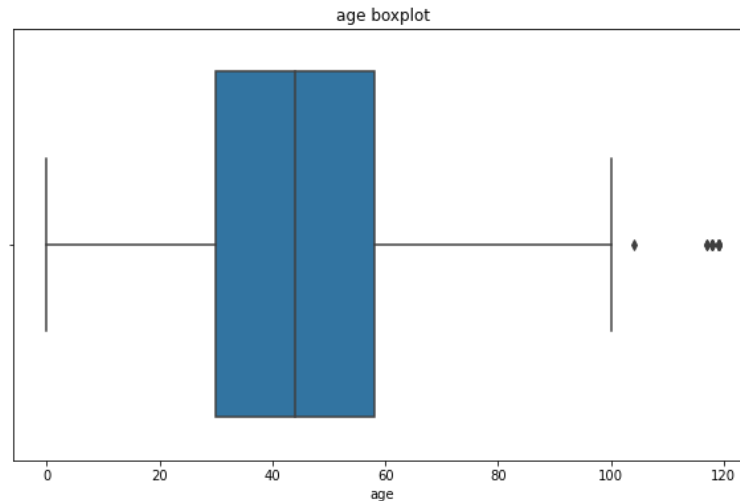
L'EDE contient les propositions que à partir de l'année 2018 donc j'ai décidé de travailler sur les années 2018, 2019, 2021 et évité l'année 2020 vu que ce n'était pas une année normale et elle ne représente pas une année significative pour faire mes analyses.

En discutant avec l'équipe métier et data science on a choisi un certain nombre des variables qui paraissent les plus importantes pour notre besoin. On a obtenu les informations suivantes :

- Caractéristiques socio-démographiques : L'âge, le sexe, la géolocalisation, la civilité, la situation familiale.
- Caractéristiques des contrats : Le statut de contrat RC et RO afin de savoir si c'est un adhérent, un ancien adhérent ou un prospect pur.
- Caractéristiques des propositions : Type d'offre, la gamme, la date de création.
- Le choix final du prospect.
- Si le prospect a été ciblé ou pas par une campagne marketing.

5.2) Nettoyage des données :

Pendant cette phase j'ai essayé d'obtenir un jeu des données propre afin d'avoir des résultats d'analyses de bonne qualité.

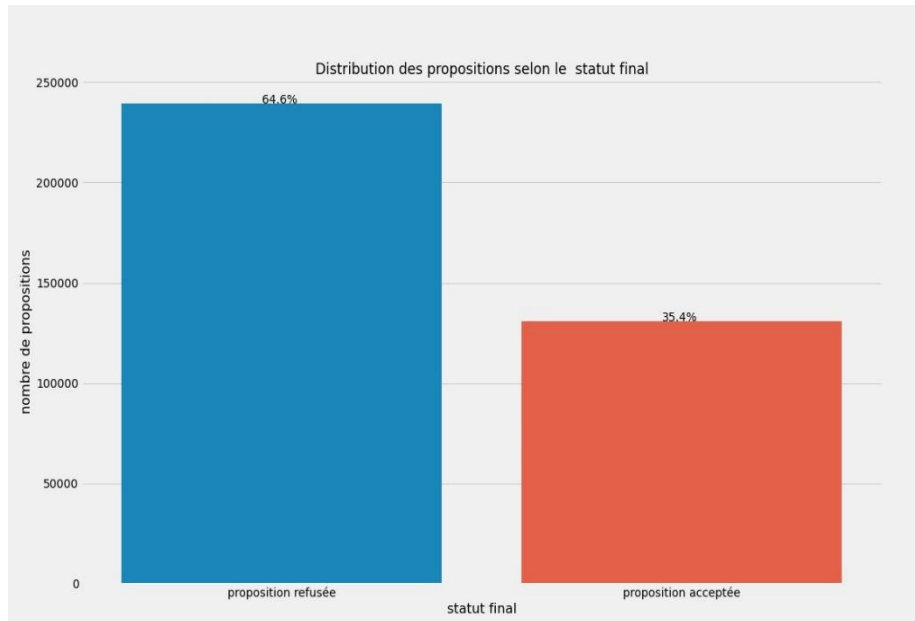


Alors j'ai supprimé les valeurs aberrantes (l'Age) comme montre la boîte à moustache au-dessus et certaines lignes qui manquaient beaucoup d'informations (sexe, situation familiale, data de naissance).

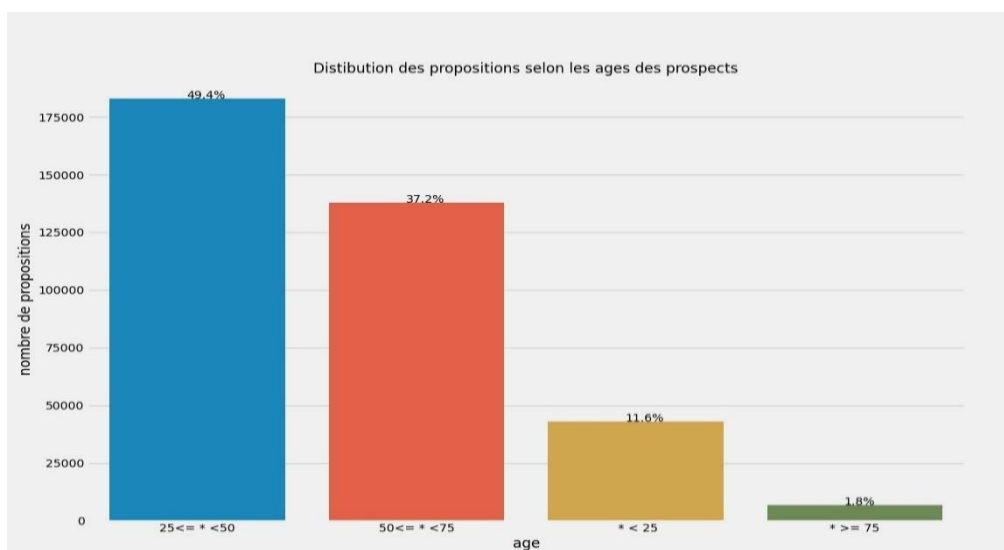
5.3) Statistiques descriptives :

Afin de bien comprendre la base des données et les relations entre les variables et la Target (choix final du prospect) je suis passé à une phase de visualisation des données. En effet cette étape m'a permis de s'assurer de cohérence générale de la base et de l'analyser. Pour ça j'ai utilisé les librairies seaborn et matplotlib de python.

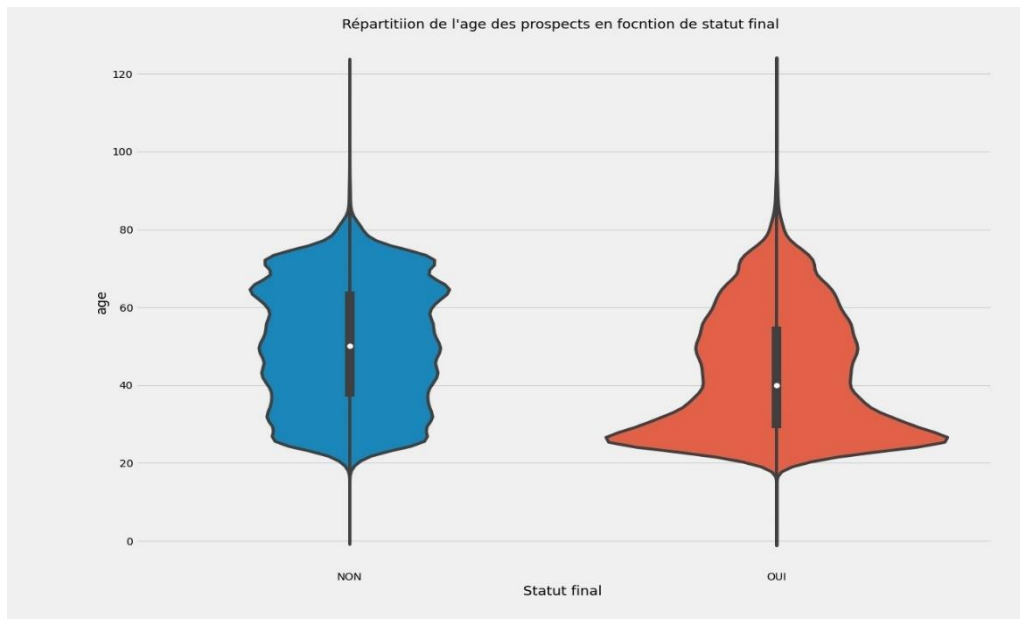
J'ai commencé par voir la distribution des propositions selon ma Target :



L'échantillon des données est composé de 64% de refus et 36% des acceptation. La base est assez équilibrée pour faire de la classification. Ensuite je me suis intéressé aux comportements de la Target en fonction de quelques variables. J'ai commencé par l'âge :

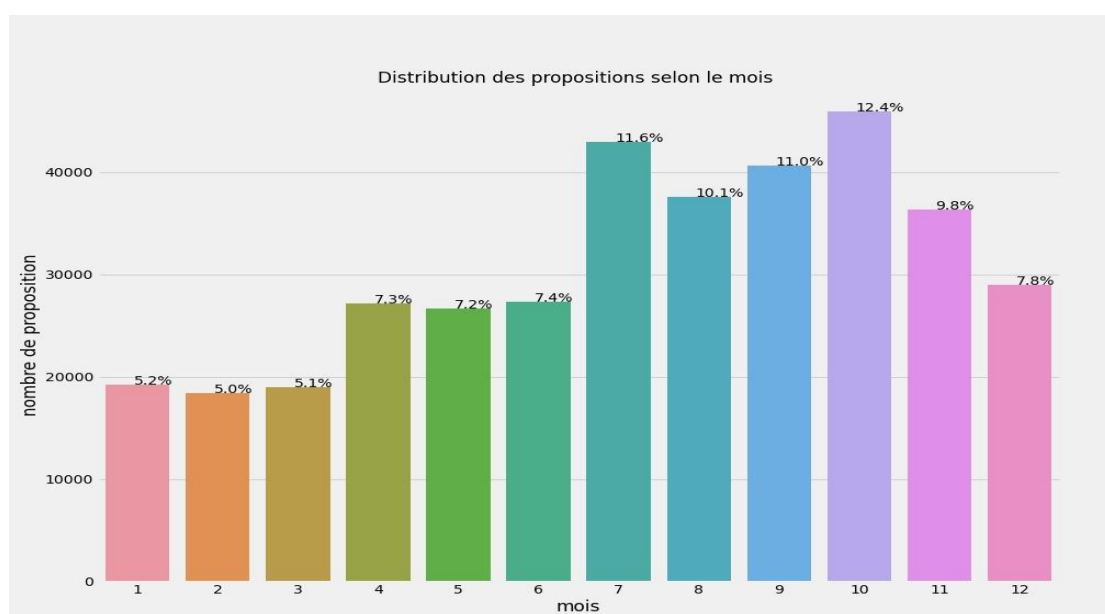


J'ai constaté que 50% des prospects sont âgés entre 25 et 50 ans, 38% entre 50 et 75 ans et une minorité de plus que 75 ans. La moyenne d'âge est de 44 ans.

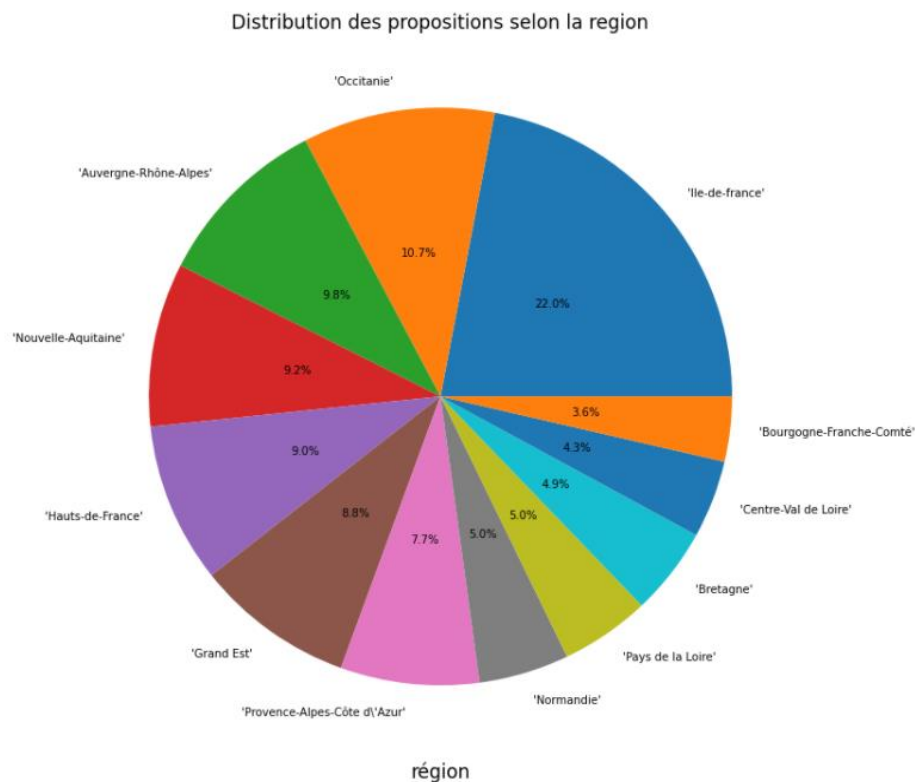


En étudiant la répartition de l'âge en fonction du choix du prospect, J'ai constaté que la moyenne d'âge des personnes acceptant les propositions est de 39 ans avec un pic autour de la trentaine. Pour les prospects refusant les propositions on a un petit pic autour de 70 ans sinon la distribution est uniforme.

Les prospects sont de plupart des femmes (66%) mais le genre n'a pas d'influence sur le choix car la proportion d'acceptation / refus est à chaque fois 0.33/0.66 pour les deux genres.

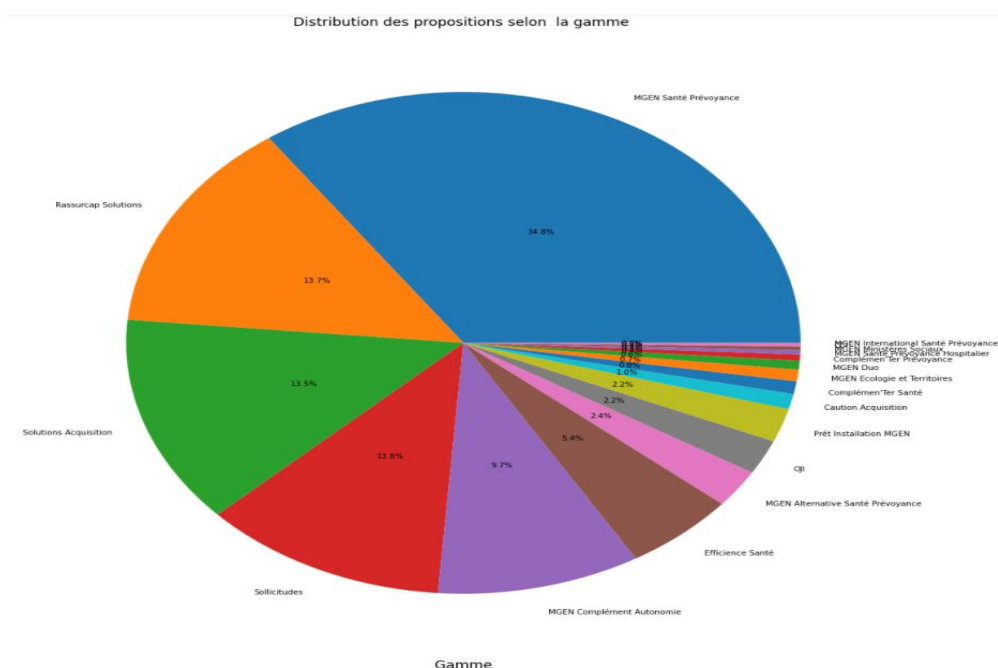


En étudiant la répartition des propositions selon les mois j'ai constaté que 50% des prospections se font avant la rentrée (juillet, août, septembre, octobre) vu que les offres sont destinées à des personnels de l'éducation nationale. J'ai constaté une stabilité de nombre des propositions pendant les 4 premiers mois de l'année.



En termes de répartition géographique, une part plus importante des prospects habitent en Ile-de-France (22%). En effet c'est dans cette région ou on a les plus des acceptations (40%). Les régions de Occitanie et Auvergne-Rhône-Alpes possèdent aussi des taux d'acceptations élevés.

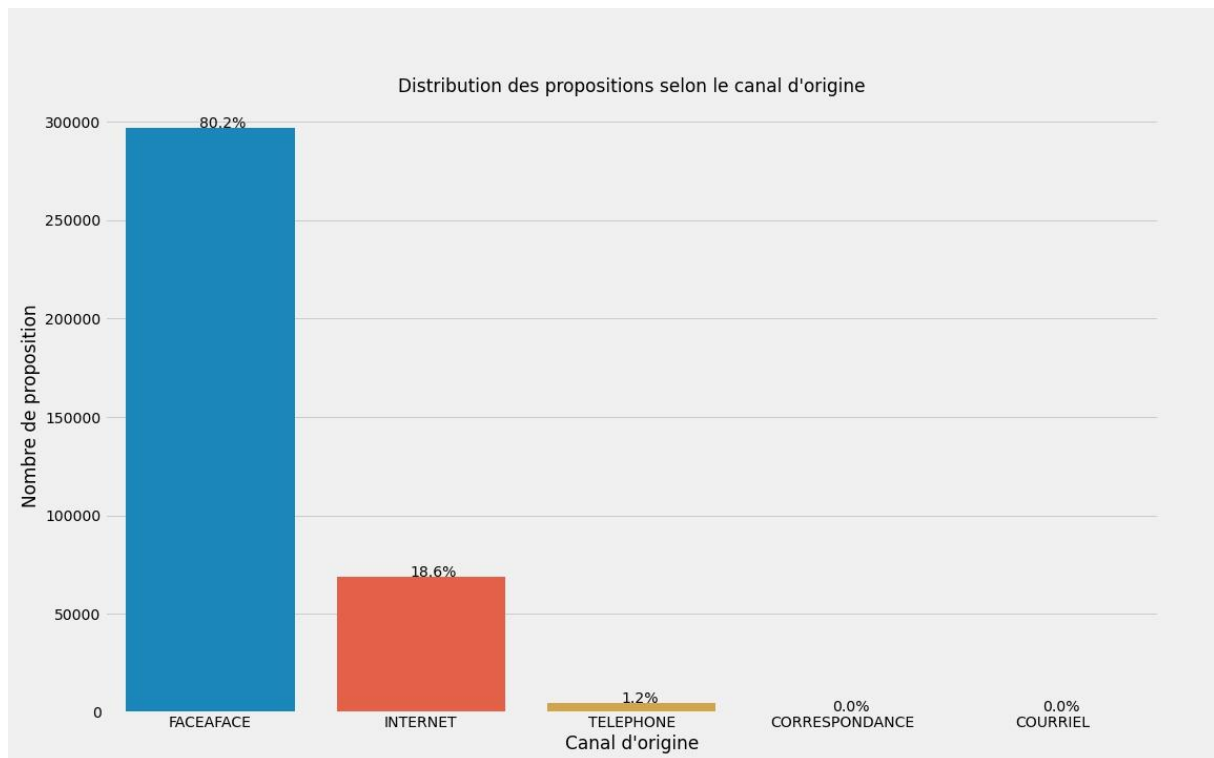
J'ai regardé aussi la répartition des propositions selon la gamme proposée :



La Mgen propose plusieurs gammes (18) possédant chacune une ou plusieurs offres (niveau au-dessous de gamme). La gamme Mgen santé prévoyance présente la gamme la plus proposée ces deux dernières années (35%). En effet les propositions contenant cette gamme sont le plus acceptées (49%). Cette gamme correspond à la gamme basique chez la Mgen et elle contient 24 offres.

L'offre la plus présente dans les propositions est la K00034 (40%) qui est contenu dans la gamme santé prévoyance. Elle est la plus prise (47% d'acceptations). En fait, cette offre correspond à la mutuelle des fonctionnaires et contractuels pour les couples avec enfants souhaitant une protection performante sur les soins peu remboursés. MGEN Référence intègre également de la prévoyance, des services et des actions sociales et solidaires en cas de difficulté de santé ou financière.

Les prospects sont contactés de différentes manières : physiquement, par téléphone ou par mail. J'ai regardé la répartition des canaux d'origines dans le jeu des données :



La majorité des prospects ont été contacté face à face (80%) donc par l'intermédiaire d'un commercial. Cette méthode permet de récupérer 40% d'acceptations. La prospection physique coute plus cher que les autres types de prospection donc une solution IA peut réduire ce cout.

La proportion des personnes contactés par téléphone et correspondance est faible mais le taux d'acceptation pour cette catégorie est plus élevé que la face à face (44%).

La base des données contient juste 10% des propositions qui appartiennent à des campagnes. La Mgen sélectionne une catégorie spécifique des personnes qui essaie de les faire les propositions à travers des campagnes. J'ai constaté que les propositions hors campagne aboutissent un peu plus à des acceptations (36% contre 28%).

5.4) Pré traitement et encodage des données :

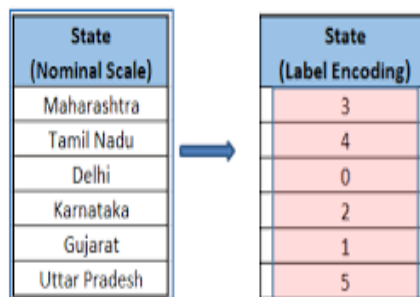
Pendant cette partie je me suis intéressé au pré traitement des données afin qu'elles soient utilisables par les modèles des prédictions. En effet j'ai changé le type de quelques variables :

- Data de naissance en valeur numérique âge.
- Data de proposition en valeur numérique mois.
- Statut d'assuré RO et RC en valeur numérique (1 ou 0)

J'ai aussi ajouté une autre variable adhérent en se basant sur 3 autres variables (statut RC, RO et le NOIDE qui signifie si la personne était déjà chez la MGEN ou pas).

J'ai encodé les autres variables selon deux méthodes :

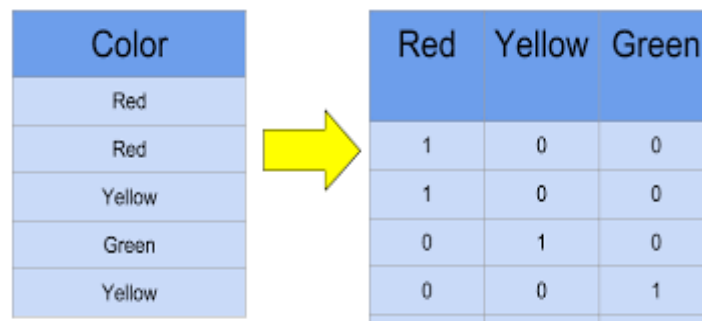
Les variables origine, statut finale et offre avec Label encoder.



The diagram illustrates the process of label encoding for categorical data. On the left, a table titled 'State (Nominal Scale)' lists five Indian states: Maharashtra, Tamil Nadu, Delhi, Karnataka, Gujarat, and Uttar Pradesh. A blue arrow points to the right, where a second table titled 'State (Label Encoding)' shows the same states mapped to numerical values: 3, 4, 0, 2, 1, and 5 respectively.

State (Nominal Scale)	State (Label Encoding)
Maharashtra	3
Tamil Nadu	4
Delhi	0
Karnataka	2
Gujarat	1
Uttar Pradesh	5

Les variables civilité, situation familiale, canal d'origine, adhérent, département, Gamme proposition avec one hot encoder :



The diagram illustrates the process of one-hot encoding for categorical data. On the left, a table titled 'Color' lists five color entries: Red, Red, Yellow, Green, and Yellow. A yellow arrow points to the right, where a second table shows the one-hot encoded representation. The table has three columns: Red, Yellow, and Green. Each row corresponds to an entry in the 'Color' table, with a '1' in the column corresponding to the color and '0' in the others.

Color	Red	Yellow	Green
Red	1	0	0
Red	1	0	0
Yellow	0	1	0
Green	0	0	1
Yellow	0	0	1

L'encodage et la normalisation des variables est fait pendant la phase d'entraînement du modèle et aussi pendant la phase de test lors de laquelle on teste les performances du modèle. Pour ça j'ai automatisé cette étape afin de l'utiliser juste avant de transférer les entrées au modèle.

5.5) Modélisation et performances :

5.5.1) Les métriques d'évaluation :

Afin d'évaluer les performances des modèles et les comparer on utilise des métriques calculées à partir de la matrice de la confusion suivante :

		Classe réelle	
		-	+
Classe prédite	-	True Negatives <i>(vrais négatifs)</i>	False Negatives <i>(faux négatifs)</i>
	+	False Positives <i>(faux positifs)</i>	True Positives <i>(vrais positifs)</i>

Avec :

- TN : classe négative et sa prédiction est négative.
- FN : classe négative et sa prédiction est positive.
- FP : classe positive et sa prédiction est négative.
- TP : classe positive et sa prédiction est positive.

A partir de cette matrice on définit nos métriques :

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}$$

Ça correspond au taux de réussite du modèle c'est le pourcentage de bonnes prédictions. Cette métrique n'est pas représentative du modèle car on peut bien classer une catégorie et pas une autre du coup on a une Accuracy élevée.

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

Le rappel, Recall, ou sensibilité est le taux de vrais positifs c'est à dire la proportion de positifs que l'on a correctement identifiés. Dans notre cas c'est la capacité du modèle à identifier toutes les propositions acceptées. C'est le critère qui va nous intéresser le plus.

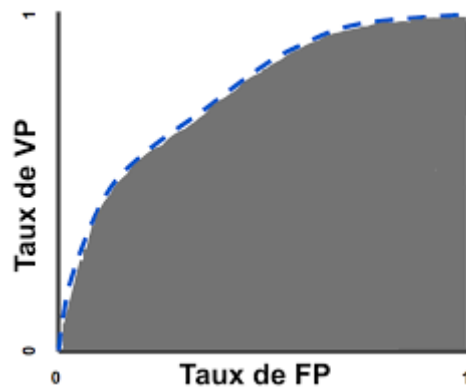
$$\text{Précision} = \frac{TP}{TP+FP}$$

La précision correspond à la proportion de prédictions correctes parmi les points que l'on a prédits positifs.

$$\text{F1 score} = \frac{2}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}}$$

Cette mesure permet d'évaluer le compromis entre le rappel et précision qui est leur moyenne harmonique.

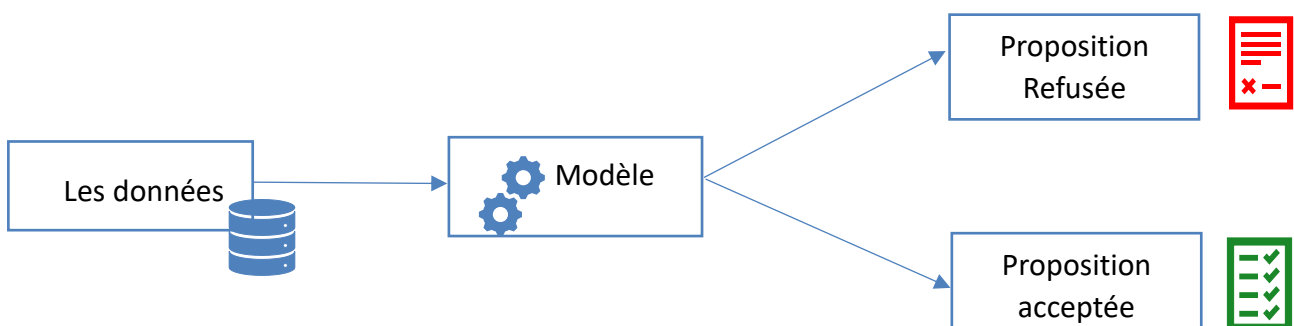
Courbe ROC : receiver operating characteristic, la courbe de sensibilité/spécificité. En effet elle présente le taux de vrais positifs en fonction de taux de faux positifs.



L'aire sous la courbe (ou Area Under the Curve – AUC) est un indice synthétique calculé pour les courbes ROC. L'AUC correspond à la probabilité pour qu'un événement positif soit classé comme positif par le test sur l'étendue des valeurs seuil possibles.

5.5.2) Classification binaire : acceptation ou refus de la proposition :

La première approche à laquelle on a pensé est de classer les propositions selon le choix de prospect et du coup d'arriver à un modèle capable d'anticiper le choix d'une personne en se basant sur quelques informations :



J'ai commencé par la phase d'apprentissage qui permet à la machine de comprendre la logique du modèle en se basant sur les exemples des cas. Puis une phase test permettant d'évaluer les modèles et l'améliorer. Pour ça j'ai travaillé avec les données de années 2018 et 2019 comme des données d'entrainements et les données de 2021 comme pour tester.

Ce problème correspond à un cas de classification donc j'ai choisi quelques algorithmes de classification afin de le résoudre. J'ai créé un script permettant de d'entrainer plusieurs modèles et de le tester puis retourner le meilleur selon un critère d'évaluation qu'on choisit parmi les métriques citées au-dessus.

Dans un premier temps j'ai travaillé sur le dataset entier et j'ai trouvé les résultats suivants :

Modèle	Accuracy	Recall	Précision	F1 score	AUC
Random Forest	0.7118	0.5984	0.5367	0.5659	0.6811
Régression logistique	0.7208	0.7132	0.5419	0.6158	0.7187
SVM	0.7207	0.7182	0.5416	0.6175	0.7200
Arbre de décision	0.6557	0.5184	0.4572	0.4859	0.6185
Gradient boosting	0.7300	0.6604	0.5592	0.6056	0.7111
Xgboost	0.7339	0.6179	0.5702	0.5931	0.7024

Ces résultats sont ceux sur l'échantillon test des données. J'ai gardé la régression logistique et svm qui donnent les meilleures performances entre Accuracy et Recall.

L'une des étapes de la modélisation consiste à optimiser les hyperparamètres du modèle. Les hyperparamètres sont toutes les variables de calibration d'un modèle qui sont fixées arbitrairement en première approche. L'objectif de l'exploration des hyperparamètres est de rechercher parmi les différentes configurations des hyperparamètres celle qui offre les meilleures performances. La fonction Grid Search permet d'effectuer une recherche brute sur l'ensemble des valeurs possibles pour les hyperparamètres spécifiés et ainsi sélectionner la combinaison permettant d'optimiser le modèle.

J'ai appliqué la fonction Grid Search sur les deux modèles et j'ai trouvé que la régression logistique est arrivée à des meilleurs performances :

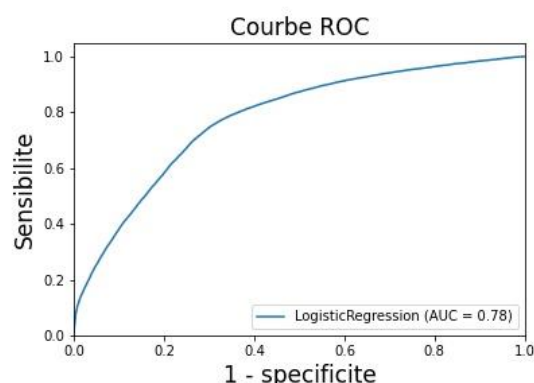
Accuracy : 0.7252

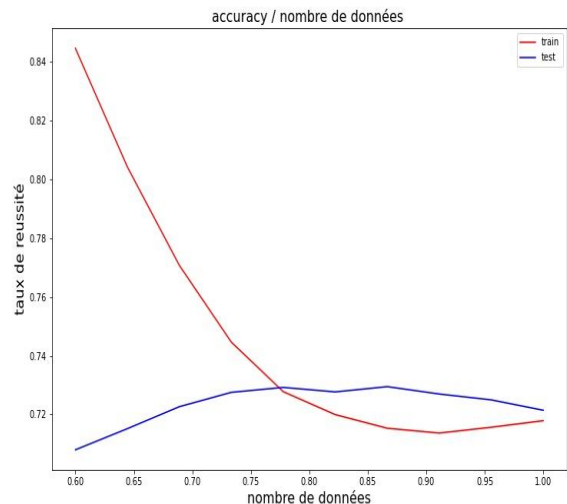
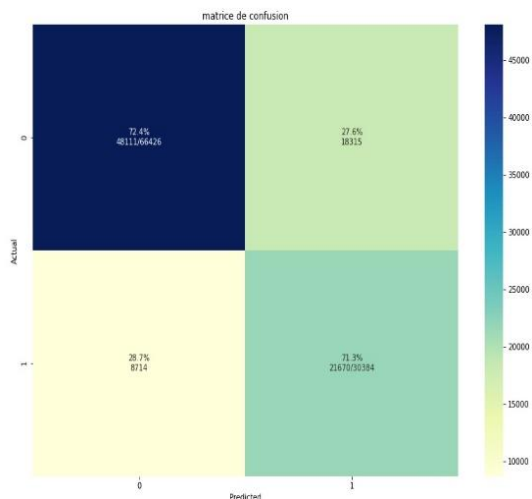
f1 score : 0.6158

Recall score : 0.7185

précision score : 0.5432

AUC : 0.7192





J'ai tracé la courbe d'apprentissage ou j'ai entraîné le modèle sur différentes parties des données et testé ses performances. J'ai trouvé que ce dernier s'est généralisé sur des nouvelles données.

Donc on a trouvé que ce premier modèle était performant et permettait de détecter 71% des propositions acceptées. Ensuite on a voulu améliorer le Recall. Pour ça on s'est basé sur la data viz faite au début ou on a remarqué l'influence d'âge sur le choix de prospect. Donc on a décidé de partager le jeu des données en deux et de créer deux modèles différents. En se basant sur le graphe des âges et les quantiles on a choisi de travailler sur les moins de 30 ans et le plus de 30 ans. Donc j'ai refait le même travail précédemment avec le test des plusieurs modèles de machine Learning en commençant par le moins de 30 ans :

Modèle	Accuracy	Recall	Précision	F1 score	AUC
Random Forest	0.6774	0.5713	0.6969	0.6279	0.6726
Régression logistique	0.7001	0.6512	0.6988	0.6742	0.6979
SVM	0.7137	0.7141	0.6938	0.7038	0.7137
Arbre de décision	0.6167	0.5002	0.6215	0.5543	0.6115
Gradient boosting	0.7147	0.6139	0.7426	0.6722	0.7102
Xgboost	0.6950	0.6211	0.7038	0.6599	0.6916

Ces résultats sont ceux sur l'échantillon test des données. J'ai gardé Xgboost et svm qui donnent les meilleures performances entre Accuracy et Recall. J'ai appliqué la fonction Grid Search sur les deux modèles et j'ai trouvé que Xgboost est arrivée à des meilleurs performances :

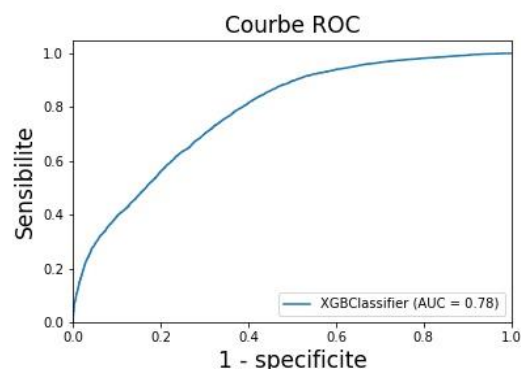
Accuracy : 0.7002

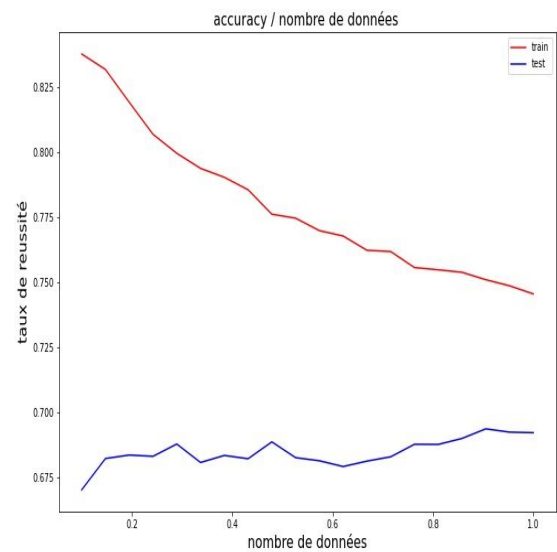
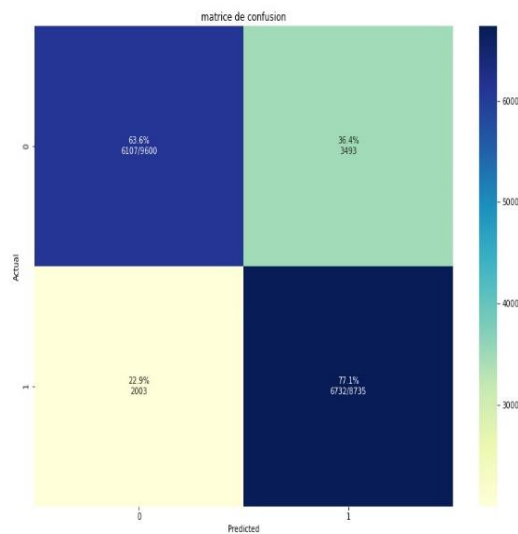
f1 score : 0.7101

Recall score : 0.7706

précision score : 0.6583

AUC : 0.7034





J'ai constaté une amélioration remarquable au niveau du Recall ou on est passé de 71% à 77% et le modèle s'est généralisé bien d'après la courbe d'apprentissage.

Puis j'ai passé sur l'autre partie du dataset (le plus de 30 ans) ou j'ai testé aussi des modèles :

Modèle	Accuracy	Recall	Précision	F1 score	AUC
Random Forest	0.6987	0.6437	0.4722	0.5448	0.6819
Régression logistique	0.7199	0.6984	0.4999	0.5827	0.7133
SVM	0.7199	0.6996	0.5000	0.5832	0.7137
Arbre de décision	0.6349	0.4922	0.3822	0.4302	0.5913
Gradient boosting	0.7167	0.7418	0.4961	0.5946	0.7244
Xgboost	0.7203	0.6461	0.5005	0.5641	0.6976

J'ai gardé le gradient boosting qui donne les meilleures performances entre Accuracy et Recall. J'ai appliqué la fonction Grid Search et j'ai trouvé les performances suivantes :

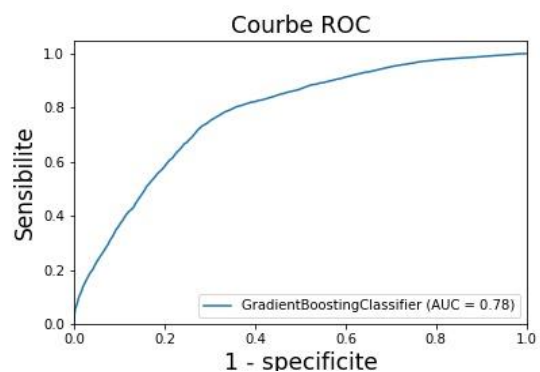
Accuracy : 0.7167

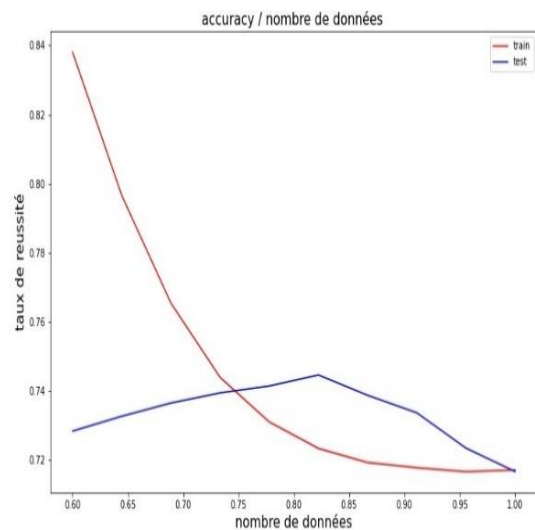
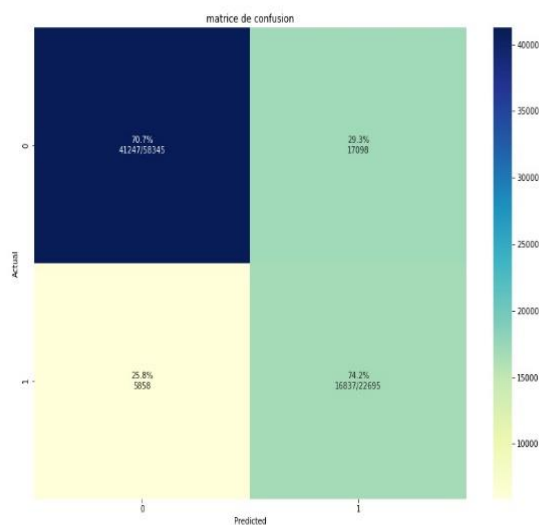
f1 score : 0.5946

Recall score : 0.7455

précision score : 0.4981

AUC : 0.7249





J'ai remarqué une amélioration du Recall ou on est passé du 71% à 74%. Donc l'approche de partager le dataset a bien donné des meilleurs résultats et on a décidé de garder cette méthode en travaillant avec deux modèles.

ROI : retour sur investissement :

Afin de montrer le retour sur investissement de nos travaux aux métiers marketing, j'ai fait de calcul sur le gain possible en appliquant note solution aux propositions de l'année 2021 :

Cette année contient 96815 propositions : - Proposition refusée : 66429,
- Proposition acceptée : 30386

Donc un taux de conversion de 31%.

* Le premier modèle sur les moins de 30 ans :

En simulant 8110 propositions on a un taux de conversion de 83% (seuil de 50%).

En simulant 2627 propositions on a un taux de conversion de 85,5% (seuil de 80%).

* Le deuxième modèle sur le plus de 30 ans :

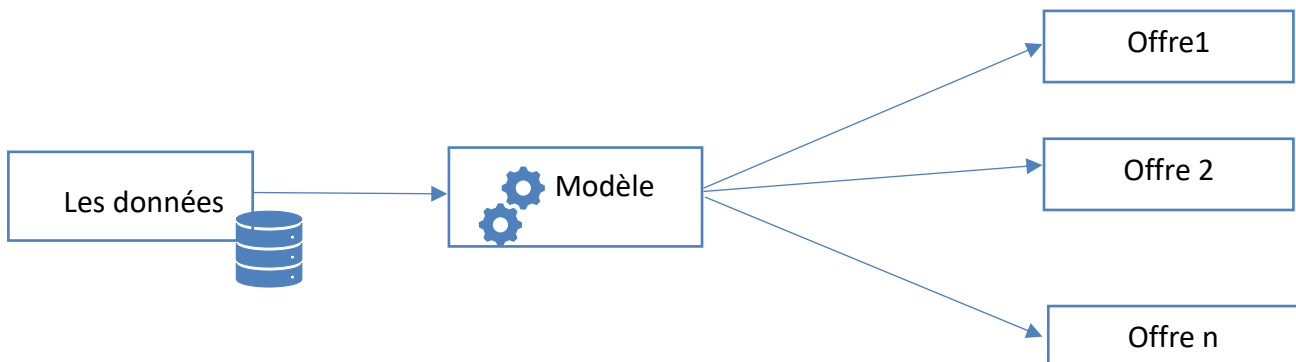
En simulant 32083 propositions on a un taux de conversion de 50% (seuil de 50%).

En simulant 1109 propositions on a un taux de conversion 87,7% (seuil de 80%).

➡ Alors si on simule juste 40193 on a un taux de conversion de 66.5% (seuil 50%) donc on prédit 26728 de propositions acceptées. On a atteint presque la totalité des acceptations en simulant 46000 propositions en moins donc en dépensant 460000 mille euros en moins.

5.5.3) Classification multi class :

Dans cette partie on a décidé de passer à une classification multi class ou à partir des mêmes données on essaie de prédire l'offre, le niveau au-dessous de gamme, choisi par le prospect. En effet les deux premiers modèles vont nous dire si la personne sera intéressée ou pas par la proposition et si c'est le cas on passe à la classification multi class comme ça on exclut toutes les propositions qui n'ont pas abouti.



La base des données contient une centaine des offres mais leur répartition est déséquilibrée : on a des offres apparus 20000 et d'autres à 5 fois. Donc j'ai décidé de créer une classe divers contenant les offres avec moins de 100 exemples dans l'échantillon. Donc j'ai obtenu un nouveau jeu des données avec une quarantaine des offres.

J'ai refait ma phase de modélisation et j'ai essayé plusieurs modèles de machine Learning et deep Learning aussi :

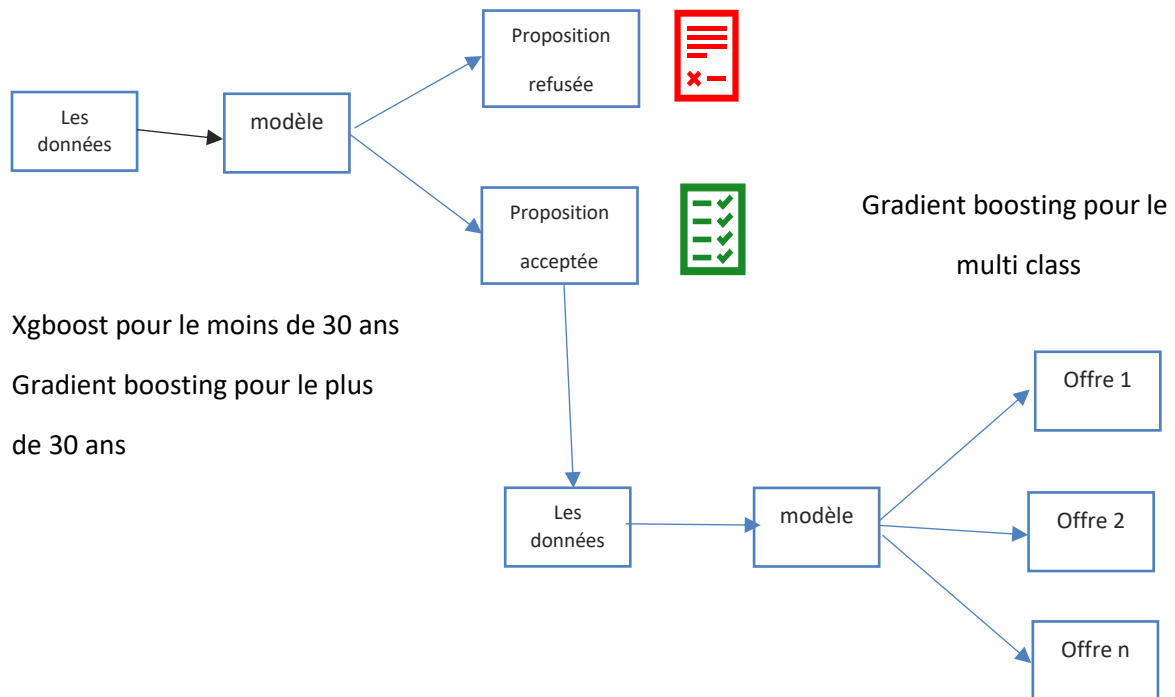
Modèle	Accuracy
Random Forest	0.4504
Régression logistique	0.4518
SVM	0.527
KNN	0.4856
Gradient boosting	0.5617
Xgboost classifieur	0.4515
Réseaux des neurones	0.57

Les performances ne sont pas très performantes pour la plupart des modèles. Même en optimisant les meilleurs modèles (gradient boosting et réseau des neurones) les performances ne sont pas améliorées. Alors on a décidé de prédire les 3 ou les 4 offres qui peuvent intéresser un prospect ou

lieu d'une seule. On a recalculé notre Accuracy en supposant que la classe est bien prédite si elle est parmi les 4 ayant les plus grandes probabilités.

En utilisant cette méthode j'ai réussi à atteindre 87% d'Accuracy avec le gradient boosting.

Du coup j'ai associé les deux approches dans un script final sous la forme suivante :



5.5.4) Industrialisation du modèle :

Après avoir validé les deux modèles de prédiction, classification binaire et multi class, j'ai créé un script final associant les deux précédents. En sortie on a la probabilité si la personne va accepter ou non la proposition. Dans le cas d'acceptation on donne aussi les 3 offres qui vont intéresser les prospects avec les probabilités correspondantes. Afin que ces travaux soient exploitables par les métiers (les commerciaux) on a décidé de créer une API en utilisant la Framework Flask.

Une API web permet à de l'information et à des fonctionnalités d'être manipulées par des programmes informatiques via internet. Flask est un cadre de travail (Framework) Web pour Python. Ainsi, il fournit des fonctionnalités permettant de construire des applications Web, ce qui inclut la gestion des requêtes HTTP et des canevas de présentation.

J'ai aussi créé une interface graphique en utilisant du HTML et CSS afin de rentrer les entrées du modèle d'une manière plus simple.

Pour utiliser l'api on se met dans le dossier contenant le script et les modèles et on lance le script API. On obtient dans la console l'affichage suivant :

```
Anaconda Prompt (Anaconda3) - python Api_Marketing.py

(base) C:\Users\AKHALBOUS\Desktop\API_marketing>python Api_Marketing.py
* Serving Flask app "Api_Marketing" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with windowsapi reloader
* Debugger is active!
* Debugger PIN: 996-084-758
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [26/Aug/2021 16:26:14] "[37mPOST /predict HTTP/1.1[0m" 200 -
127.0.0.1 - - [26/Aug/2021 16:26:14] "[33mGET /favicon.ico HTTP/1.1[0m" 404 -
```

Puis on saisit dans le navigateur l'adresse <http://127.0.0.1:5000/> :

la probabilité de acceptation est de 80.47705292701721 % le prospect aura tendance à accepter cette proposition les offres les plus probables sont : ['K00034' 'K00059' 'K00044'] avec les probabilités suivantes : [59 11 11]

On saisit les données pour le modèle et on clique sur predict et on a comme résultat la probabilité d'acceptation avec les 3 offres les plus probables.

Cette interface est graphique est juste un prototype afin que ça sera plus parlant pour les commerciaux lors de présentation et ça sera améliorée dans un deuxième temps.

6)Perceptives :

Afin d'améliorer le modèle d'autres variables peuvent être ajouté comme les ressources ou la consommation des adhérents. En effet d'autres modèles seront mis en place afin de cibler que les adhérents RO ou RC ou les deux afin de les proposer des offres complémentaires.

Au niveau de l'industrialisation la MGEN compte passer à l'utilisation de dataiku dans quelques mois.

Dataiku est une plateforme de développement intégré, destinée aux professionnels de la donnée, pour réaliser ses études Big Data. C'est un outil permettant de créer un projet data science de préparation jusqu'à déploiement. Il permet aux différents profils (data scientist ou analyste) de collaborer et travailler ensemble sur plusieurs projets au même temps. Les spécialistes peuvent l'utiliser pour explorer, développer et produire leurs propres data plus efficacement.

Le logiciel de Dataiku permet d'améliorer une infrastructure existante qu'il s'agisse d'une Data Warehouse SQL ou d'un cluster Spark. C'est un environnement permettant la coexistence entre tous les standards de technologies Big Data et les différents langages.

7) Conclusion

Ce stage m'a permis de travailler sur un projet concret en data science et d'assister à la création d'une nouvelle équipe dans ce domaine. Pendant ces 5 mois, je suis passé par toutes les étapes d'un projet professionnel dans une grande entreprise. J'ai pu surmonter la difficulté de travailler à distance grâce à l'aide de toute l'équipe, la cohésion et les échanges réguliers qu'on avait.

La mission de mon stage était de créer un outil de marketing prédictif pour le prospect. J'ai donc commencé par découvrir l'entrepôt des données de la MGEN et extraire la data. Cette partie était la plus dure pour moi vu mon manque des connaissances dans le domaine de mutuelle. On a eu beaucoup d'aide de la part des autres équipes qui nous ont expliqué le contenu de l'EDE.

Puis je suis passé à la compréhension des données et leur nettoyage ainsi que la visualisation que m'a aidé énormément dans la suite. J'ai ensuite passé à la phase de modélisation et performances où j'ai appliqué les techniques de machine Learning et trouvé le meilleur modèle. J'ai créé une interface graphique et une API permettant d'utiliser le modèle directement via le navigateur.

A la fin je peux dire que ce stage a renforcé mon envie de continuer à travailler sur les données et les exploiter dans différents domaines ainsi monter en compétences car le data science est en train d'évoluer remarquablement ces dernières années.

8)Bibliographie :

1)<https://scikit-learn.org/stable/>

2)<https://sql.sh/cours>

3)[Cours gm4/Introduction machine learning/Bruno portier](#)

4)<https://flask-restful.readthedocs.io/en/latest/>

5) <https://openclassrooms.com/fr/courses/1603881-apprenez-a-creer-votre-site-web-avec-html5-et-css3>

6) <https://datascientest.com/seaborn>

7) <https://www.mgen.fr/>

8) <https://www.hn-services.com/fr/>

9) Annexes :

```
from datetime import datetime
from flask import Flask, request, jsonify, render_template
from flask_json import FlaskJSON, JsonError, json_response, as_json

##import C:\Users\AKHALBOUS\Desktop\code_python\encoder_one_hot.npy as
encoder_one_hot

import pandas as pd
import numpy as np
import seaborn as sns
from matplotlib import pyplot as plt
import pickle
import re
from numpy import linalg as la

app = Flask(__name__)
FlaskJSON(app)

def encodage(vect, encoder1, encoder2, encoder3):
    aux = encoder2.transform(vect[one_hot_columns])

    vect = vect.drop(columns =one_hot_columns, axis=1)
    vect["origine"] = encoder1.transform(vect["origine"])
    vect[['mois', "age"]] = encoder3.transform(vect[['mois', "age"]])
    vect[['mois', "age"]] = encoder3.transform(vect[['mois', "age"]])

    res= pd.concat([vect.reset_index(drop = True),
pd.DataFrame(aux.toarray()).reset_index(drop = True)], axis=1)
    return res

def multi(n,model,x_test,encoder):
    y_pred = model.predict_proba(x_test)
    x = np.arange(y_pred.shape[0]*n).reshape(y_pred.shape[0],n)
    z = []
    p = np.arange(y_pred.shape[0]*n).reshape(y_pred.shape[0],n)
    for i in range(y_pred.shape[0]):
        x[i] = y_pred[i].argsort()[n:][::-1]
        z.append(encoder.inverse_transform(x[i]))
        p[i] = (np.sort(y_pred[i])[n:][::-1]) *100
    p = ''.join(map(str, p))
    z = ''.join(map(str, z))
    res = '\n les offres les plus probables sont : {} \n \n avec les
probabilités suivantes : {} \n'.format(z,p)

    return res

def offre(res,prob,vect,n,encoder):
    if res == 1:
        filename = 'model_multiclass_XGB.sav'
        model = pickle.load(open(filename, 'rb'))
        a1 = ' \n la probabilité de acceptation est de {} % \n \n le
prospect aura tendance à accepter cette proposition
```

```

\n'.format(prob[0][1]*100)
    a2 = multi(n,model,vect,encoder)
    a3 = a1 + a2
else :
    a3 = "\n la probabilité de refus est de {} % \n \n le prospect aura
tendance à refuser cette proposition \n ".format(prob[0][0]*100)
    return a3

def prospection(vect,encoder):
    if (vect["age"] < 30).any():
        filename = 'model_inferieur_q1_xgb.sav'
        model = pickle.load(open(filename, 'rb'))
        pred = model.predict(vect)
        prob = model.predict_proba(vect)
        res = int(pred)
        b =offre(res,prob,vect,3,encoder)

    else :
        filename = 'model_superieur_q1_xgb.sav'
        model = pickle.load(open(filename, 'rb'))
        pred = model.predict(vect)
        prob = model._predict_proba_lr(vect)
        res = int(pred)
        b = offre(res,prob,vect,3,encoder)
    return b

one_hot_columns =
['SEXE','CIVILITÉ','SITUATION_FAMILIALE','CANAL_ORIGINE','adherent','depart
ement',"GAMME_PROPOSITION"]
encoder_hot = np.load('encoder_one_hot.npy', allow_pickle=True).tolist()
encoder_label = np.load('encoder_label.npy', allow_pickle=True).tolist()
encoder_multi = np.load('encoder_multi.npy', allow_pickle=True).tolist()
normalizer = np.load('normalizer.npy', allow_pickle=True).tolist()

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict',methods=['POST'])
def predict():

    d = request.form.to_dict()
    vect = pd.DataFrame([d.values()], columns=d.keys())
    vect["departement"] = int( vect["departement"])
    vect["age"] = int( vect["age"])
    vect["mois"] = int( vect["mois"])
    vect["STATUT_ASSURÉ_RC"] = int( vect["STATUT_ASSURÉ_RC"])
    vect["STATUT_ASSURÉ_RO"] = int( vect["STATUT_ASSURÉ_RO"])
    output = prospection(encodage(vect,encoder_label, encoder_hot,
normalizer),encoder_multi)

    return render_template('index.html', prediction_text= output)

@app.route('/Marketing', methods=['POST'])
def Marketing():
    data = request.get_json(force=True)
    try:

```

```

        vect = pd.DataFrame.from_dict(pd.json_normalize(data),
orient="columns")
        res = prospection(encodage(vect,encoder_label, encoder_hot,
normalizer),encoder_multi)
        print(res)
        return('')

    except (KeyError, TypeError, ValueError):
        raise JsonError(description='Invalid value.')

if __name__ == '__main__':
    app.run(debug=True)

```