Rapport Prédiction sur les résiliations clients

Rakotondrabe Philippe

Mekkiou Adam

STS1 DEV

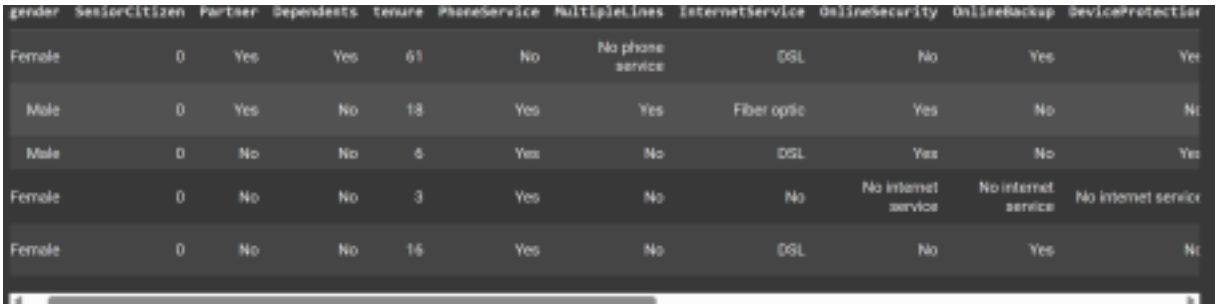
L'objectif de ce projet est de s'intéresser à une société et de prédire pour ses clients pourquoi ils résilient leur abonnement et comment y remédier.

Il nous est donné tout au départ 3 fichiers csv : x_train et y_train qui sont les données d'entraînement, et x_test les données de test.

Y_train est un ensemble de données d'entrainement qui affiche deux valeurs binaires non numériques (No ou Yes), qui correspond à l'acceptation de la résiliation du client ou non. C'est la valeur que nous devons prédire.

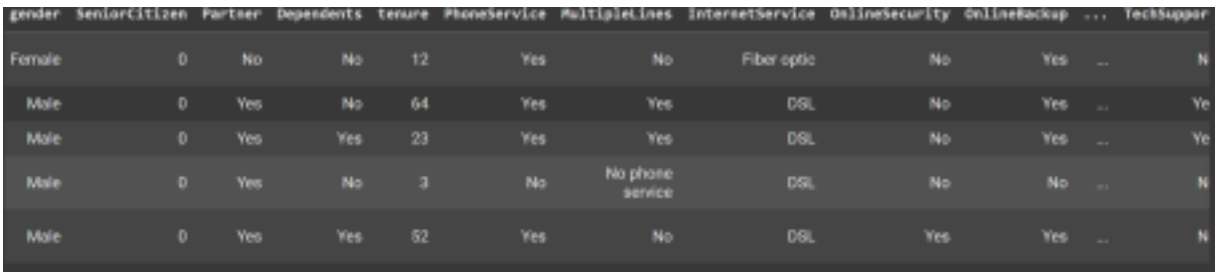
0	No
1	Yes
2	No
3	No
4	Yes

X_train est un ensemble de données d'entrainement, alphanumériques, il ya de flotants et des strings et aussi des entiers. Tout ces données donnent plus d'informations sur les clients comme les services auxquels ils sont abonnés (PhoneService), les informations de compte, leur type de contrat et méthode de paiement par exemple (Contact, PaymentMethod). Et leurs données un peu plus personnelles, comme les données démographiques, leur genre et leur personnes pris en charge.



gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtectio
Female	0	Yes	Yes	61	No	No phone service	DSL	No	Yes	Ye
Male	0	Yes	No	18	Yes	Yes	Fiber optic	Yes	No	N
Male	0	No	No	6	Yes	No	DSL	Yes	No	Ye
Female	0	No	No	3	Yes	No	No	No internet service	No internet service	No internet service
Female	0	No	No	16	Yes	No	DSL	No	Yes	N

X_test est un ensemble de données de Test. Ce jeu de données est similaire à x_train excepté son objectif qui est ici de que la machine puisse "tester" ce qu'elle a appris de son entrainement. Puis, le nombre de lignes et colonnes qui est légèrement différent de X_train. Et il y a une colonne en plus dans X_Test qui Row_ID permettant d'identifier de manière unique chaque ligne.



gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	...	TechSuppor
Female	0	No	No	12	Yes	No	Fiber optic	No	Yes	...	N
Male	0	Yes	No	64	Yes	Yes	DSL	No	Yes	...	Ye
Male	0	Yes	Yes	23	Yes	Yes	DSL	No	Yes	...	Ye
Male	0	Yes	No	3	No	No phone service	DSL	No	No	...	N
Male	0	Yes	Yes	52	Yes	No	DSL	Yes	Yes	...	N

Ces trois jeux de données vont nous être utile pour réaliser des prédictions.

L'objectif ici est de prédire si les prochains clients vont résilier ou non.

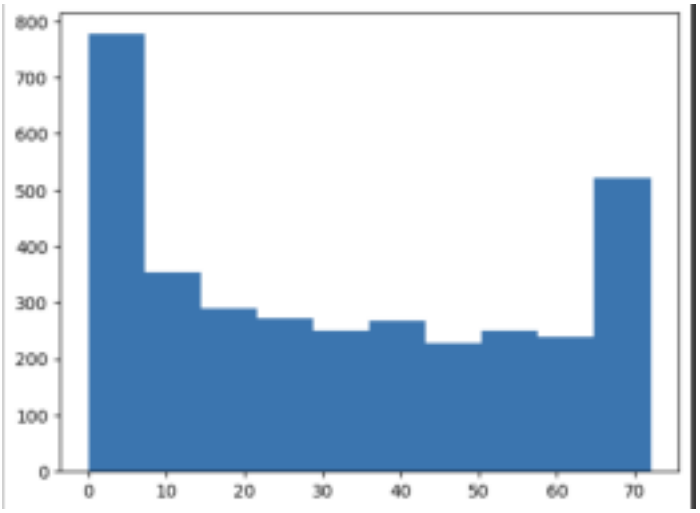
Nous allons dans un premier temps sélectionner une variable afin de commencer une analyse,

Nous avons choisi la colonne 'tenure' dans X_Train affichant le nombre de mois auquel il est resté abonné avec la société. Nous verrons la répartition de chaque client plus en précision.

	tenure
0	61
1	18
2	6
3	3
4	16
...	...
3446	49
3447	23
3448	1
3449	15
3450	69

3451 rows × 1 columns

dtype: int64



Cet histogramme nous montre la distribution de la variable 'tenure', on observe une variation plutôt homogène, cependant des valeurs extrêmes sont observées.

Ici sur l'axe des abscisses on retrouve la durée de l'abonnement et sur l'axe des ordonnées le nombre de client, soit ici le nombre de client par durée d'abonnement dans la société

Cette évolution évolue de 0 à 72 mois.

On peut voir qu'il y a quasiment 800 clients ayant un abonnement de 0 à 8 mois et 520 clients environ ayant un abonnement de 65 à 72 mois. Excepté cela le graphique évolue de manière homogène vers les 300 clients abonnés de 8 à 65 mois.

Il y a donc beaucoup de nouveaux clients et d'anciens.

La moyenne ici s'élève à 32.40 environ, cela veut dire que la période moyenne d'abonnement s'élève à 32 mois.

On obtient une variance de 588.7 ce qui est élevée, cela signifie donc que les valeurs de la variable contract, sont très dispersés. Il y a beaucoup de valeurs élevées et beaucoup de valeurs faibles.

On obtient comme médiane 29, ce qui se rapproche de la moyenne qui est de 32. Ainsi, 29 mois représente la valeur centrale de la liste en durée d'abonnement.

Nous allons ensuite construire la matrice de corrélation afin d'identifier et illustrer les coefficients de corrélation entre les variables du jeu de données en question.

Nous allons au passage créer une première amélioration de prédiction : transformer les valeurs alphanumériques en valeurs numériques, cela permettra d'utiliser le jeux de données dans des outils de calculs tels que la matrice de corrélation qui affichera des coefficients, Un simple « rechercher et remplacer » sur excel a suffit, voici la traduction en valeur numérique des colonnes

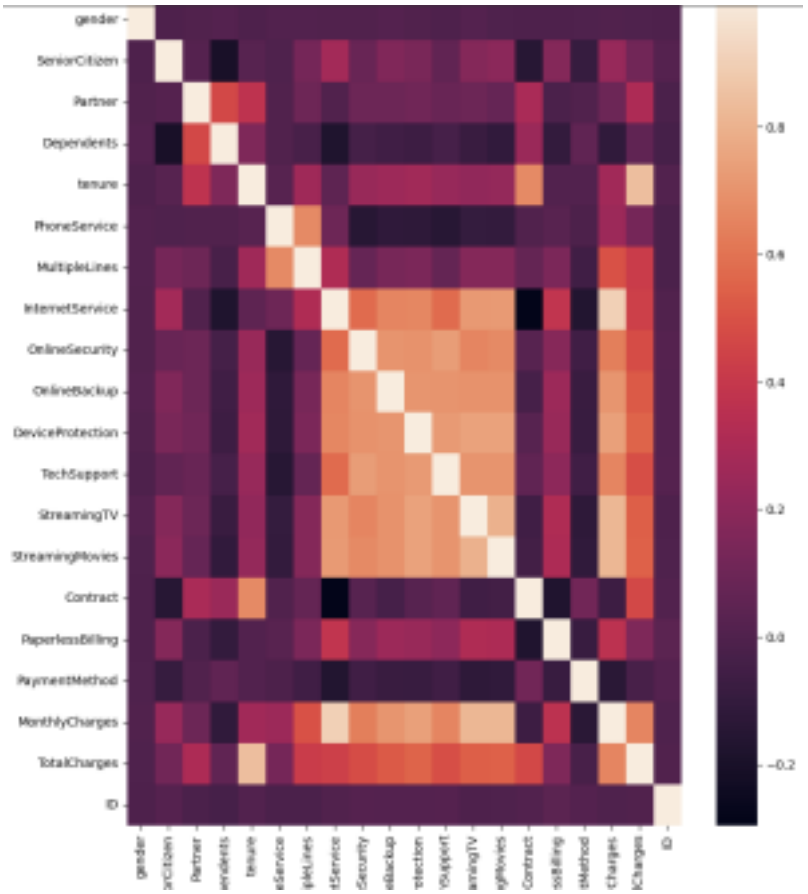
Gender : 0 = female, 1 = male | Partner : 0 = No, 1 = yes | Dependents : 0 = No, 1 = yes | Phone service : 0 = No, 1 = yes | Multiples lines : 0 = no phone service, 1 pour no, 2 pour yes | Internet services : 0 = no, 1 = DSL, 2 = fiber optic | Onlinesecurity, onlinebackup, deviceprotection, techsupport streaming tv, streaming movie : 0 pour no internet service, 1 pour no, 2 pour yes| Contract = 0 = Month to month, 1 = one year, 2 = two year | Paperlessbiling : 0 = No, 1 = yes | Paymentmethod : 0 = bank stransfer, 1 = Electronic check, 2 = Mailed check, 3 = credit card

Ce qui donne ceci :

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DevicePr
0	0	0	1	1	61	0	0	1	1	2	
1	1	0	1	0	18	1	2	2	2	1	
2	1	0	0	0	6	1	1	1	2	1	
3	0	0	0	0	3	1	1	0	0	0	
4	0	0	0	0	16	1	1	1	1	2	
...	...	...	...	...	...	...	...	...	...	...	...
3446	0	0	1	1	49	1	2	2	1	1	
3447	0	0	0	0	23	1	1	0	0	0	
3448	0	0	0	0	1	1	1	0	0	0	
3449	1	0	0	1	15	1	1	1	1	1	
3450	0	0	1	0	69	1	2	0	0	0	

3451 rows × 20 columns

Nous pouvons dresser ainsi la matrice de corrélation en heatmap



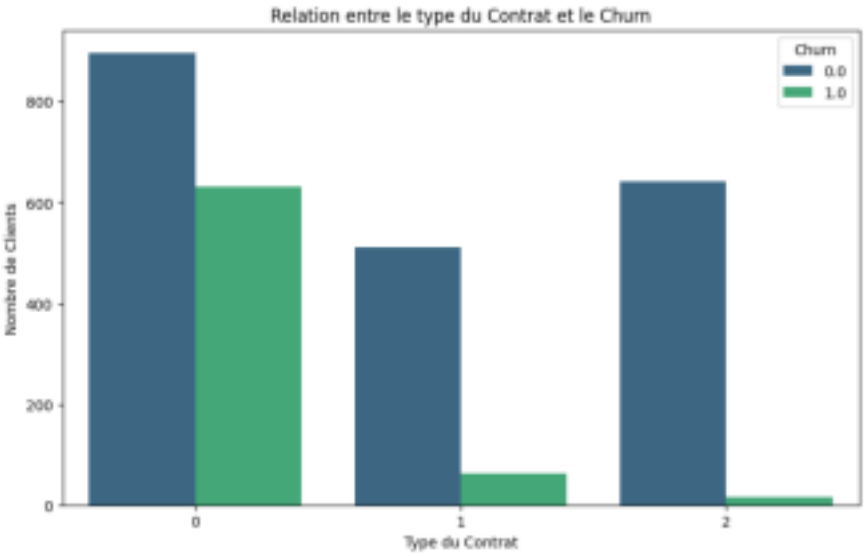
On retrouve les mêmes variables sur l'axe des abscisses et des ordonnées, les coordonnées d'une case [x ;y} est un coeficient allant de 0 à 1, plus la valeur se rapproche de 1 et plus et plus les deux categories x et y sont en influence, c'est-à-dire que si une variable augmente, l'autre augmente aussi de manière linéaire. Ceux-ci sont représentés par des couleurs (chaude se rapprochant de 0 et froide se rapprochant de 1), on appelle cette matrice une heatmap.

Par lecture visuelle, le genre est en corrélation faible avec l'ensemble des données, il n'influence donc pas trop la prédiction. A l'inverse les variables tenure et totalcharges sont en très grande corrélation montrant que plus le client reste longtemps dans la société et plus ses charges sont grandes, ce qui est logique. En revanche, les services (internet services jusqu'à streaming movie) sont en corrélation élevée mais cela ne traduit pas une relation de cause à effet car ce sont des services distincts, le client peut en effet choisir ce service dans son abonnement ou un autre en fonction de ses gouts, ce n'est pas parce qu'un client choisi tel service qu'il en choisira un autre également.

Une covariance élevée entre deux variables influence directement les coefficients de la matrice de corrélation. En effet, si la covariance est élevée, elle indique une forte relation linéaire entre les deux variables. Cependant, des valeurs extrêmes ou des variances élevées peuvent fausser la perception de cette relation, car elles peuvent amplifier ou atténuer les coefficients de corrélation, ce qui peut mener à des erreurs de prédiction

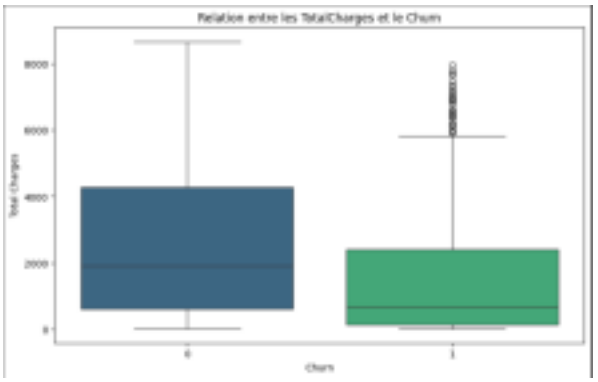
Nous allons maintenant comparer la variable dépendante (résiliation donc churn) avec le jeux de données avec deux graphiques afin d'avoir un début de prédictions :

4



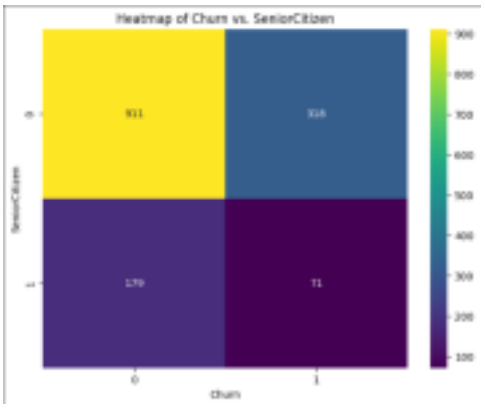
Ici le graphique en barres présentant le nombre de clients par type de contrat dont ceux ayant résiliés en vert. On s'apperçoit qu'il y a plus de 600 clients parmis les 900 ayant un contrat = 0 soit de type « month-to-month » ayant résilié (churn = 1), ainsi les clients ayant la durée de contrat le plus court sont ceux susceptibles de résilier, il s'agit pour eux comme une période d'essai. A l'inverse ceux ayant une durée de contrat très élevée (2 mois) soit contract = 2, sont ceux qui ont le moins resilié . Il faut donc porter une attention particulière à ceux s'engageant dans un contrat mensuel, ils sont les plus susceptibles de partir.

Un autre graphique (bloxsplot) ou on compare la resiliation par rapport aux charges totales



De même ici on observe d'une part un nombre de client qui résilient ayant un total de charge faible pour la majorité des cas (c'est logique car ils ne restent pas longtemps), nous observons tout de même en minorité ayant des charges totales élevées et ayant résilié. Ce sont des outliers, on peut admettre que cette minorité sont ceux n'arrivant plus à tenir face aux charges qu'ils sont obligés de résilier.

5



Autre exemple : Commentaire Graphique Relation Churn avec Variable indépendante choisi "SeniorCitizen" Dans ce Graphique, celui-ci nous montre une heatMap de la relation de la variable dépendante Churn et de la variable indépendante SeniorCitizen.

On observe une forte relation pour les non seniors (0) avec 911 qui ont le plus tendance à ne pas résilier (variable churn 0). Cela nous en dit plus sur le genre des personnes ne pas résilier le plus, ce sont les jeunes.

Alors que nous observons par ailleurs une faible relation chezlesseniors(personnes âgés), avec 179 pour lesseniors qui n'ont pas résiliés et 71 qui ont résiliés.

Les seniors qui ont résiliés possèdent un taux plus important que les non seniors résiliant. 26% pour les non séniors et 28% pour les seniors

On comprend que les seniors sont les plus sensibles, à résilier leur abonnement, ce qui est dû probablement à leur vulnérabilité financière ou à leur besoin d'accompagnement dû à certains

besoins spécifiques comme le changement technologique. Nous pouvons en déduire que l'Entreprise doit œuvrer pour proposer des offres plus adaptéEs aux seniors.

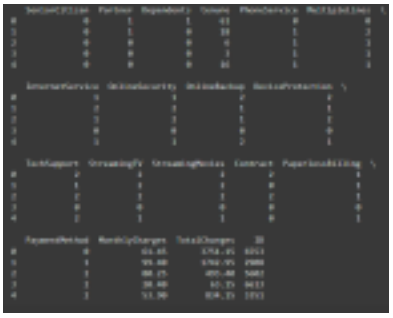
Pour le taux de non résilitation (Churn 0), les non séniors ont plus de probabilité de rester des clients par rapport aux séniors. (74% pour les non seniors contre 71%) 74% = 911/1229 * 100

Cela s'explique que les non séniors donc les jeunes soient attiré par les prix et les offres attractifs qui sont adaptés à leur attentes. Donc ils sont plus fidèles à rester.

Nettoyage des données

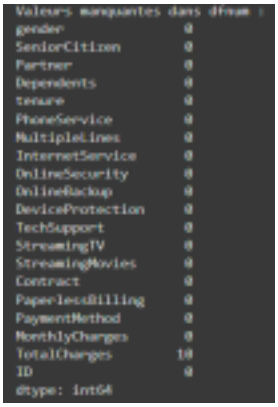
Nous allons optimiser les données que nous avons :

1er nettoyage : supprimer la variable ‘genre’ du jeu car elle n’influence que peu le jeu de données d’après la matrice de corrélation ce qui donne ceci pour x_train :



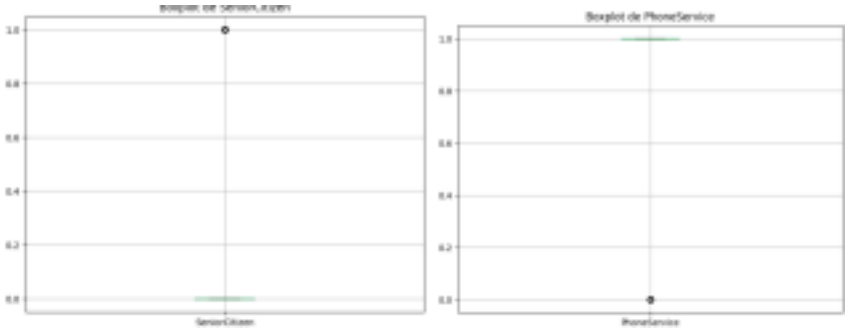
2ème nettoyage : repérer les valeurs nulles et les imputer avec la moyenne

6



On observe 10 valeurs de TotalCharges à null, à la place de les supprimer car cela réduira les performances de notre modèle, nous allons imputer ces valeurs à la moyenne.

3ème nettoyage : s'occuper des valeurs aberrantes (outliers) :

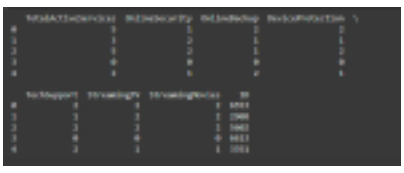


Une minorité représente des personnes agées, et nous retrouvons également un petit groupe d'outliers de client n'ayant pas de service téléphonique.

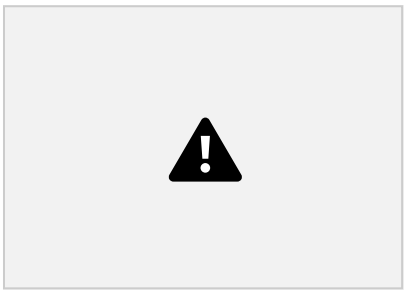
D'après la matrice de corrélation, leur corrélation avec les autres coefficient reste faible, nous allons donc les laisser telles qu'elles sont car ces outliers peuvent améliorer notre prédiction.

4ème nettoyage : fusionner des variables

La fusion des variables permet d'améliorer les prédictions et raccourcir le temps de travail, on a créé une variable 'TotalActiveServices' qui represente le total de service de chaque client (onlinesecurity, online backup etc), rappelons en haut que chaque service est à deux dans le csv quand le client en a, on a donc compté sur combien de services le 2 se répète, ce qui nous donne cet apperçu :



Ce qui nous donne ceci en aperçu total :



La séparation des données permet d'éviter le surapprentissage et d'être sur que le modèle est bien entrainé

7

Grâce à cela nous allons d'abord entrainer notre modèle afin qu'il s'adapte aux données d'entrées, ensuite on lui fera des tests afin d'évaluer sa performance une fois entrainée,

Puis on comparera le resultat avec les valeurs réelles afin de determiner la precision. Grace a ces étapes, on optimise le fonctionnement du modèle

Nous utiliserons ainsi le fichier csv x_test pour les entrainements

Nous choissirons ici le random forest, son rôle est d'afficher la liste des arbres de decisions, dans cet algorithme, chaque arbre fait des predictions independants

D'abord, nous allons effectuer une méthode de cross-validation avant tout afin d'évaluer la robustesse et la stabilité du modèle en utilisant la bibliotheque sklearn :

Le temps de traitement a duré plusieurs minutes, cela est normal l'algorithme doit passer par toutes les possibilités pour faire ses estimations

Les meilleurs paramètres trouvés pour notre modèle sont min_sample_leaf, min_samples_split et n_estimators, le meilleur score est de 81,1% représentant la performance moyenne du modèle avec les paramètres optimaux sur l'ensemble des folds et un écart tupe de 1,38% est détecté ce qui est faible, plus c'est faible et plus les performances sont cohérentes à travers les folds. Ces resultats nous permettent de savoir quels paramètres utiliser pour notre random forest afin d'avoir les meilleurs résultats possibles

Concernant le random forest, chaque noeuds comportera une caracteristique de notre jeu de données, et sur chaque branches nous auront les resultats de la condition

Tout en haut de l'arbre on trouvera la première caracteristique utilisée, et tout a la fond dans lesd feuilles on aura le valeur de la prediction

Le random forest est capable d'accepter et de s'adapter face aux données complexes comme des relations non linéaires et s'adapte face aux valeurs aberantes, il est donc très efficace et polyvant

De plus ayant discuté avec des data-analysts de notre entourage, nous avons retenu que le random forest est un algorithme très utilisé car très efficace, de plus cet algorithme-ci nous a permis d'augmenter notre score sur kaggle justiant son efficacité

Le random forest accepte les valeurs aberrantes, on a observé plusieurs outliers dans la question précedante cet algorithme est le plus approprié d'autant plus que nous avons fait exprès de ne pas supprimer les valeurs abérantes que l'on a vu tout à l'heure, notre modèle s'en charge de les traiter.

Par lecture de nos analyses, les valeurs ne semblent pas être lineaires car elles different en fonction chaque ligne, on ne peut

pas voir de prediction à l'oeil nu, il est donc preferable d'utiliser un modèle lineaire

Il existe plusieurs hyperparametres pour ce modele que l'on pourra optimiser tels que le nombre d'arbres qui augmentera la performance mais aussi le temps de calcul ou encore la profoneur maximale des arbres afin de limiter le surapprentissage. Rappelons que nous avons trouvé les bon hyperparamètres à utiliser grâce à la cross validation, c'est ce que nous utiliserons tout de suite.

Le sur-apprentissage definit l'appretissage excessif de notre algorithme, ou il enregistre tout les details, cela va donc nuire à la généralisation des données et le modèle de prédiction sera donc mauvais au fianl.

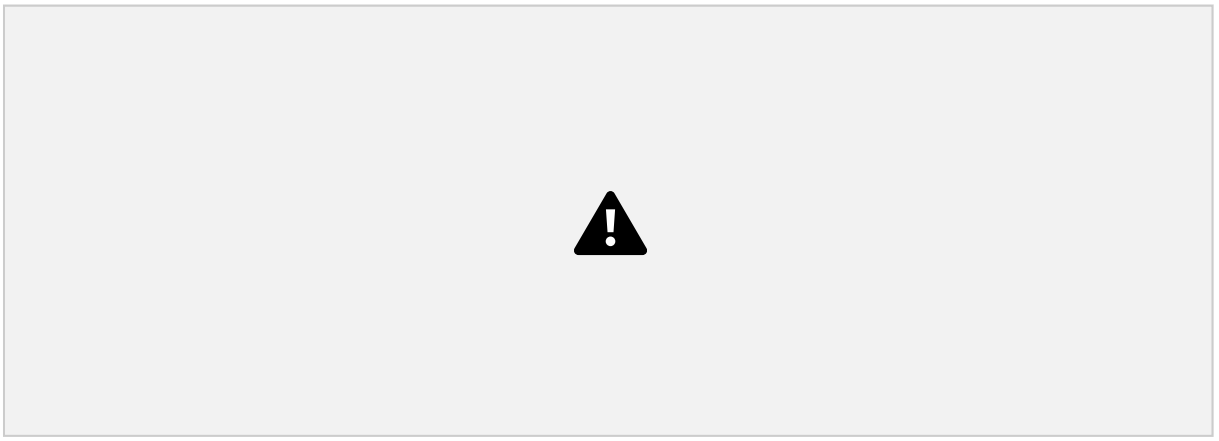
En ce qui concerne l'algorithme choisi, le moyen le plus simple de le faire sur entrainer est d'autoriser une profondeur limitée des arbres afin que l'algorithme retienne tout les details avec les exceptions inclues.

En plus de donner de faible généralisations, l'algorithme sera sensible à chaque changement et trop de modification mineures peuvent par exemple alterer de manière très significative les predictions et cela donnera plus de calcul à l'algorithme

8

Après ajustements de tout les parametres cités dessus (cross validation, suppression de colonne, imputation etc) nous avons généré le random forest en question :

Du fait du nombre de colonnes et de valeurs, on a un random forest très large, le voici en partie car il est trop long :



Nous avons par la suite créé la matrice de confusion qui a donné ceci :

On retrouve ici 478 vrais positifs, 28 faux positifs, 98 faux négatifs et 87 vrais négatifs

Détails : Accuracy ((VP+VN)/(VP+VN+FP+FN) = 0,817 soit 81,7% de prédictions globales correctes

Précision (VP/(VP+FP) = 0,944664 soit 94,5% de prédictions positifs

Recall (VP/(VP+FN) = 0,82986 soit 0,83% de détection de cas positifs réels

Nous avons plus de 80% de cas positifs trouvés, le modèle a fait une bonne prédiction, le modele est bien calibré pour identifier les instances positives tout en maintenant un bon équilibre entre précision et rappel.

Attardons-nous maintenant sur la courbe ROC, cette dernière permet d'illustrer les perfomance d'un modèle en traçant un taux de vrais positifs contre le taux de faux positifs à différents seuils de décision

Voici la courbe ROC

La courbe ROC s'incline fortement vers le coin supérieur gauche, indiquant que le modèle Random Forest a un bon taux de vrais positifs (TPR) et un bas taux de faux positifs (FPR). Cela signifie que le modèle est efficace pour distinguer entre les classes positives et négatives.

9

La ligne rouge diagonale représente un modèle aléatoire avec une AUC de 0.5. La courbe ROC en bleu se situe au-dessus de cette ligne, ce qui indique que le modèle est nettement meilleur qu'un modèle aléatoire.

Le modèle Random Forest a cette forme spécifique de courbe ROC en raison de sa capacité à capturer les relations complexes dans les données grâce à la combinaison de nombreux arbres de décision.

Une AUC de 0.86 signifie que le modèle a une bonne capacité de distinction entre les classes. Plus l'AUC est proche de 1, meilleure est la performance du modèle. Le modèle avec ses paramètres est donc efficace et au-dessus de la moyenne.

Il sera par la suite possible d'améliorer notre modèle avec les modifications des hyperparamètres du random forest : en effet c'est ce qu'a estimé la cross-validation, ou on peut modifier la profondeur des arbres par exemple, il est également possible d'augmenter le nombre de données d'entraînement par exemple.

Pour améliorer les performances de son modèle, il est essentiel de nettoyer les données et les filtrer les données tout en imputant les données null. Supprimer les variables qui ont peu de relation avec le jeu de données (on voit cela sur un heat Map). Changer de modèle classificateur peut améliorer la prédiction.

Pour atteindre une performance maximale du modèle, on peut rajouter plus de données pertinentes et les traiter.

Les hyperparamètres du modèle revient à optimiser les réglages du modèle peut ainsi améliorer la précision et réduire les erreurs du modèle tout en évitant le sur apprentissage.

Ici d'après nos résultats précédant quoi nous ont permis d'optimiser notre modèle, ce dernier évalue de très bons résultats et nous pensons l'avoir très bien optimisé. Il n'est donc pas nécessaire de changer d'algorithme par exemple, le random Forest est un bon choix pour ce genre de cas, il est puissant fiable et tolère les valeurs aberrantes permettant ainsi de le réutiliser à l'avenir si d'autres données sont à étudier au sien de l'entreprise

Ceci nous permet de donner notre réponse et nos conseils à propos de l'entreprise afin d'éviter les résiliations fréquentes :

Les caractéristiques les plus importantes grâce à la matrice de corrélation, sont les types de contrat et durée de l'abonnement, nous avons remarqué que les d'abonnement courtes sont celles en influence avec le taux de résiliation élevé

Les clients ayant un service téléphonique sont ceux qui vont le plus résilier

Les séniors résilient aussi et pour les anciens clients abonnés depuis longtemps également. Le genre n'influence pas réellement sur la décision de résiliation.

Explication à un public non technique avec nos recommandations :

D'après nos analyses, nous révélons plusieurs raisons justifiant la résiliation des clients, tout d'abord ceux qui résilient le plus souvent sont ceux ayant des contrats les plus courts (par mois), nous vous conseillons donc d'une part de proposer des « offres de bienvenue » afin de convaincre les clients à rester longtemps, car si ces derniers choisissent un contrat mensuel, cela vont dire qu'ils n'ont pas vraiment confiance et regardent si ça leur plait ou non, une offre de bienvenue et sans

engagement va déjà convaincre vos clients à rester plus longtemps.

De plus, pour les clients ayant beaucoup de service d'actif, proposez-leur une réduction tarifaire pour les inciter à conserver plus de service.

Les personnes âgées sont elles aussi impactées : ce sont celles qui résilient le plus, pour y remédier vous pouvez réduire la tarification pour les personnes âges, ou bien de mettre en place des barrières à la sortie (frais de résiliation), mais cette condition va peut-être devoir faire l'objet d'une nouvelle analyse de prédiction car cela altèrera le choix de tous les clients, le mieux serait donc de laisser une réduction pour les personnes âgées.

Enfin, les clients fidèles, soit ceux qui sont restés très longtemps abonnés, sont aussi susceptible de se désabonner, nous vous conseillons de réduire également la tarification si les clients restent longtemps, réduire par exemple de chaque année le prix de l'abonnement.

Avec toutes ces suggestions, cela va réduire un peu la valeur ajoutée de votre entreprise car vous allez réduire la tarification en fonction de certains paramètres, néanmoins en se projetons à long terme, vous allez par la suite garder vos clients, avoir plus de clients également et gagner en popularité trois éléments essentiels qui vous permettrons d'améliorer à long et très long terme votre chiffre d'Affaires.