Projet Classification supervisée

Happiness

Thomas Barat | Mathéo Richer

Table des matières

Introduction

Dans le cadre de notre formation, plus précisément du cours de classification supervisée, nous avons été amenés à réaliser un projet de modélisation à partir de données réelles. Pour cela, nous avons utilisé la base de données happiness, issue du package R wooldridge.

Cette base de données est composée d'informations sur le niveau de bonheur de différent individus , ainsi que diverses variables socio-économiques (âge, revenu, situation matrimoniale, etc.).

L'objectif principal de ce projet est de prédire le niveau de bonheur d'un individu à partir des variables explicatives. Nous mettons en œuvre plusieurs méthode de classification supervisée afin de comparer leurs performances et d'identifier les paramétres les plus pertinents.

Ce dossier présente successivement la base de données utilisée, les étapes de préparation des données, les différentes méthodes de classification utilisée, ainsi que les résultats obtenus.

Données

Présentation

Le jeu de données happiness, issu du package wooldridge sous R, regroupe des informations issues d'enquêtes menées auprès de nombreux individus résidant aux États-Unis. Il est initialement composé de 17137 observations et de 33 variables décrivant des caractéristiques socio-économiques, démographiques, familiales ou comportementales, telles que l'âge, le revenu, le niveau d'études, la situation matrimoniale ou encore le nombre d'heures passées devant la télévision.

La variable d'intérêt dans notre étude est happy, qui représente le niveau de bonheur déclaré par les individus. Cette variable, de type qualitative ordinale, constitue notre cible dans le cadre de la modélisation supervisée. L'objectif est ainsi de prédire si un individu peut être considéré comme heureux ou non à partir des autres variables disponibles.

L'ensemble des variables présentes dans la base est détaillé dans le tableau suivant.

Table 1: Description des variables de la base de données `happiness`

Nom de la variable	Type	Description
`year`	Numérique	Année de réponse au questionnaire
`workstat`	Facteur	Information sur l'état de travail
`prestige`	Numérique	Note de prestige du métier exercé
`divorce`	Facteur	L'individu est-il divorcé ou séparé
`widowed`	Facteur	L'individu est-il veuf
`educ`	Numérique	Nombre d'années d'études
`reg16`	Facteur	Région de résidence à l'âge de 16 ans
`babies`	Numérique	Enfants de moins de 6 ans
`preteen`	Numérique	Enfants de 6 à 12 ans
`teens`	Numérique	Enfants de 13 à 17 ans
`income`	Numérique	Tranche de revenu
`region`	Facteur	Région de résidence actuelle
`attend`	Facteur	Fréquence de pratique religieuse
`happy`	Facteur	Niveau de bonheur déclaré
`owngun`	Numérique	Possède une arme (0 ou 1)
`tvhours`	Numérique	Heures de télévision par jour
`vhappy`	Numérique	Très heureux (0 ou 1)
`mothfath16`	Numérique	Vivait avec père et mère à 16 ans (0 ou 1)

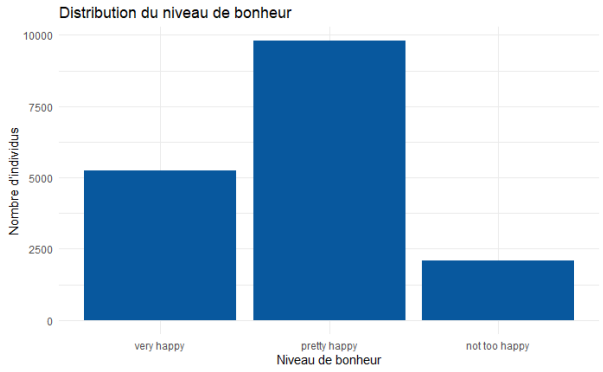
Suite à la page suivante

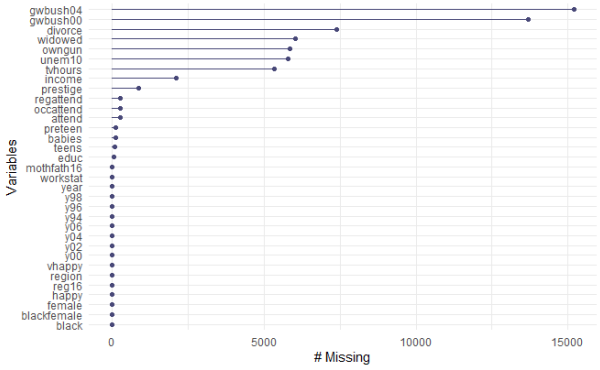
Table 1 – suite de la page précédente

Nom de la variable	Type	Description
`black`	Numérique	Est noir (0 ou 1)
`gwbush04`	Numérique	A voté Bush en 2004 (0 ou 1)
`female`	Numérique	Est une femme (0 ou 1)
`blackfemale`	Numérique	Femme noire (0 ou 1)
`gwbush00`	Numérique	A voté Bush en 2000 (0 ou 1)
`occattend`	Numérique	Pratique religieuse modérée (0 ou 1)
`regattend`	Numérique	Pratique religieuse régulière (0 ou 1)
`y94`	Numérique	Réponse en 1994 (0 ou 1)
`y96`	Numérique	Réponse en 1996 (0 ou 1)
`y98`	Numérique	Réponse en 1998 (0 ou 1)
`y00`	Numérique	Réponse en 2000 (0 ou 1)
`y02`	Numérique	Réponse en 2002 (0 ou 1)
`y04`	Numérique	Réponse en 2004 (0 ou 1)
`y06`	Numérique	Réponse en 2006 (0 ou 1)
`unem10`	Numérique	Au chômage dans les 10 dernières années (0 ou 1)

Analyse

Notre première étape consiste à analyser rapidement les données présentes et manquantes, afin de repérer d'éventuelles tendances ou déséquilibres visibles.
Commençons par les deux graphiques ci-dessous : l'un représente la distribution de la variable cible happy, et l'autre la proportion de valeurs manquantes pour l'ensemble des variables.

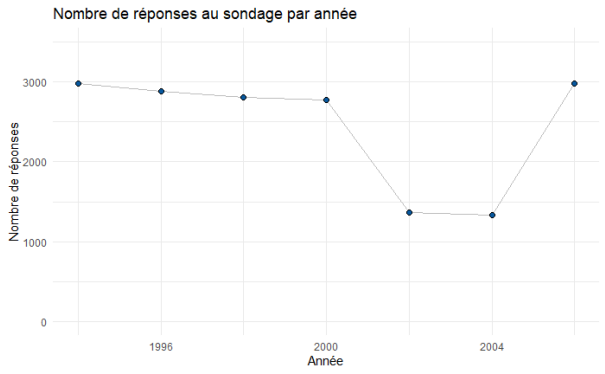


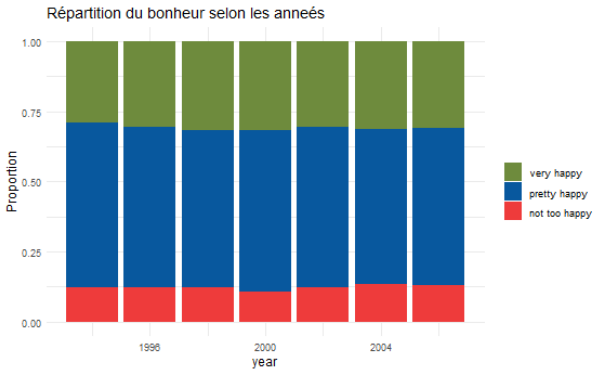


La première chose que l'on observe sur le premier graphique est un déséquilibre entre les classes, avec une nette dominance de la modalité "pretty happy". Ce déséquilibre, bien que pouvant sembler naturel dans le cadre d'une enquête sur le bonheur, reste à surveiller. En effet, une répartition inégale des classes peut influencer la performance des modèles de classification, en les incitant à favoriser la classe majoritaire au détriment des classes minoritaires.

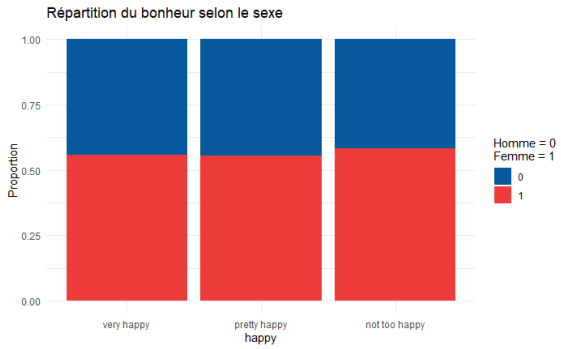
Sur le second graphique, on remarque un nombre particulièrement élevé de valeurs manquantes pour les variables gwbush00 et gwbush04, lié à la temporalité du questionnaire. D'autres variables présentent également un taux de valeurs manquantes conséquent. Pour certaines d'entre elles, une suppression pourra être envisagée si elles sont jugées non pertinentes. Dans le cas contraire, un traitement spécifique ou une imputation devra être fait.

Nous allons maintenant examiner les autres variables en fonction de la modalité happy, afin d'observer comment celles-ci se répartissent selon le niveau de bonheur déclaré.

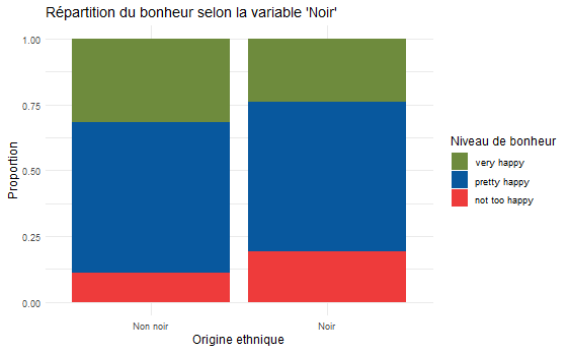




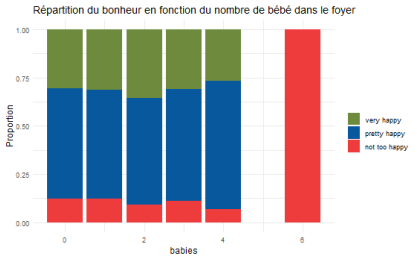
La première visualisation met en évidence l'évolution du nombre de réponses au sondage au fil des années. On observe une baisse progressive entre 1994 et 2002, avec un creux particulièrement marqué en 2002 et 2004, avant un retour à un niveau plus élevé en 2006. De son côté, le second graphique montre une certaine stabilité dans la répartition des modalités de la variable cible au cours du temps, suggérant que la perception du bonheur reste globalement constante d'une année à l'autre.

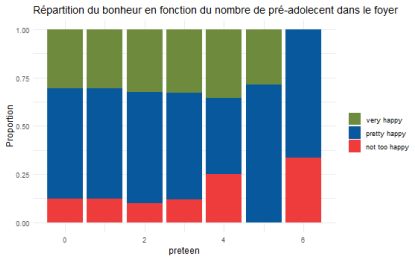


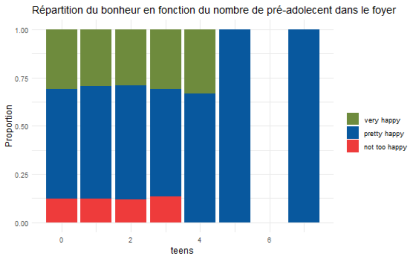
Concernant la répartition du bonheur selon le sexe, le graphique montre des proportions relativement similaires entre hommes et femmes dans chaque modalité de la variable happy. On note toutefois une légère surreprésentation des femmes dans la catégorie « not too happy ».



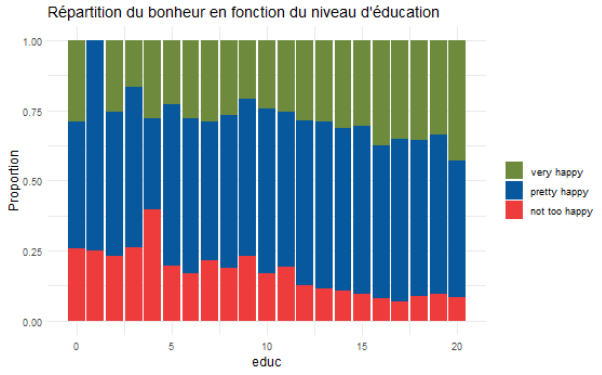
Concernant la répartition du bonheur en fonction de la variable ethnique black, on observe des proportions globalement similaires. Toutefois, on remarque une proportion légèrement plus élevée d'individus "not too happy" et une proportion plus faible de "very happy" parmi les personnes noires. Cela pourrait suggérer une perception du bonheur légèrement moins favorable au sein de cette partie de la population.

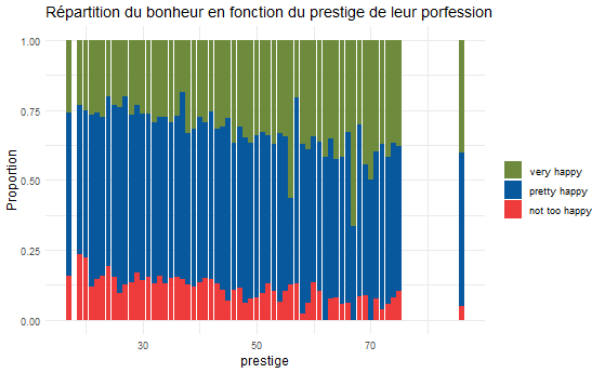




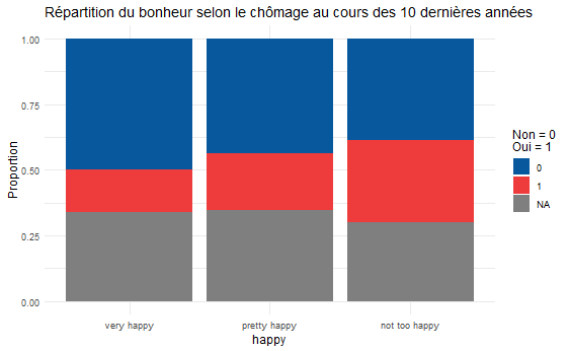


À la lecture des trois graphiques ci-dessus, représentant le niveau de bonheur en fonction du nombre de bébés, de préadolescents et d'adolescents dans le foyer, on observe que ni le nombre ni l'âge des enfants ne semblent avoir un impact majeur sur le niveau de bonheur déclaré. Cependant, quelques particularités ressortent : tous les individus ayant déclaré six bébés se classent comme "not too happy", tandis qu'à l'inverse, toutes les personnes ayant plus de cinq adolescents sont exclusivement "pretty happy". On remarque également une absence d'individus "not too happy" parmi ceux ayant cinq préadolescents ou quatre adolescents. Ces observations, bien que marginales, méritent d'être notées, même si elles peuvent s'expliquer par de faibles effectifs dans ces catégories.

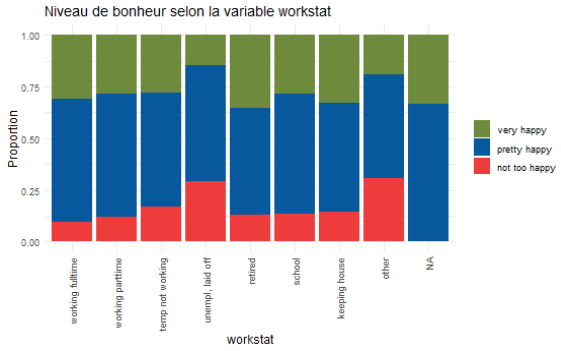




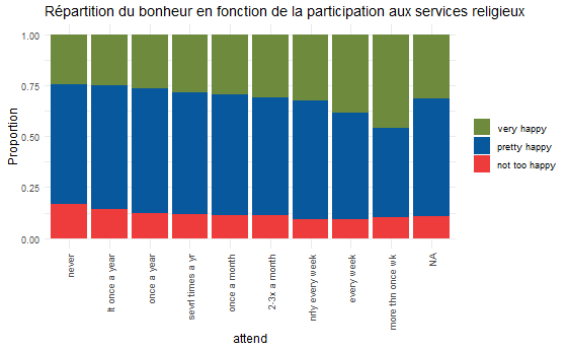
Nous commenterons ensemble les deux graphiques suivants, car les variables educ (niveau d'études) et prestige (prestige du métier exercé) sont fortement corrélées, comme nous le verrons dans la prochaine section, mais également car elles semblent suivre une tendance similaire. En effet, on observe une augmentation progressive du niveau de bonheur chez les individus ayant un niveau d'études ou un prestige professionnel plus élevé.



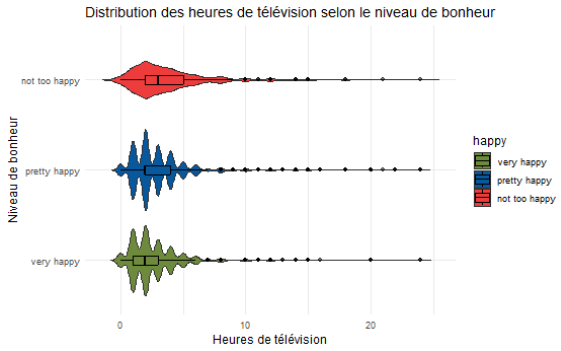
Sur ce graphique, plusieurs éléments ressortent clairement. Le premier, déjà évoqué précédemment, est le nombre important de données manquantes. Le second est la tendance observable : les individus ayant connu une période de chômage au cours des dix dernières années présentent un niveau de bonheur globalement plus faible. En d'autres termes, plus la personne a été confrontée au chômage, plus son niveau de bonheur tend à diminuer.



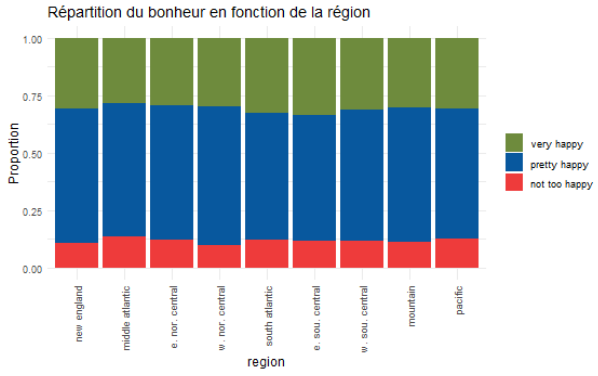
Ce graphique illustre la répartition du niveau de bonheur en fonction de la situation professionnelle des individus. Aucune tendance marquée ne se dégage de manière générale, à l'exception de certaines modalités. On observe que les personnes classées comme "non employé" ou "autre" présentent une proportion plus élevée d'individus "not too happy". Par ailleurs, on peut noter que parmi les personnes n'ayant pas répondu à cette question, aucune ne s'est déclarée "not too happy".

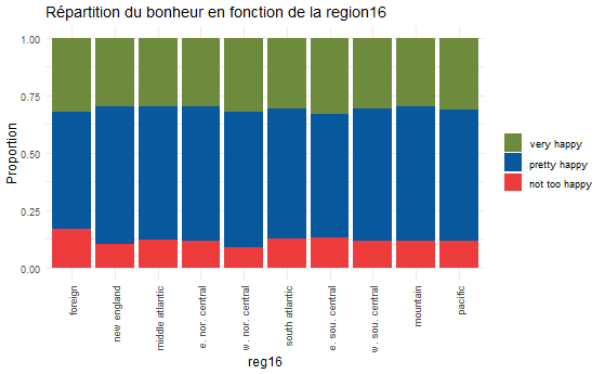


On observe ici que l'augmentation de la fréquence de participation aux services religieux a peu d'effet sur la réduction du taux de personnes se déclarant "not too happy". En revanche, elle s'accompagne d'une augmentation notable de la proportion d'individus se percevant comme "very happy".



Ce graphique montre que les individus "not too happy" regardent en moyenne plus de télévision que les autres, tandis que ceux "very happy" en consomment généralement peu. Cela pourrait suggérer une tendance inverse entre le niveau de bonheur et le temps passé devant la télévision.



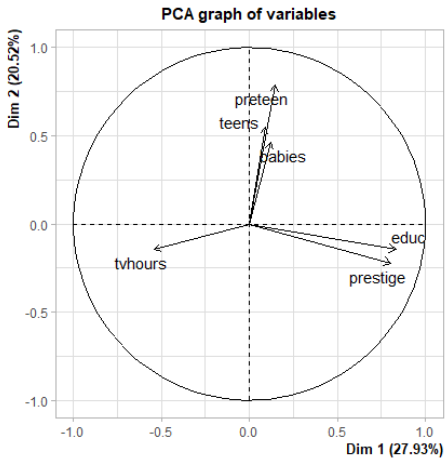


Les deux graphiques sur les régions d'habitation, qu'il s'agisse de la région actuelle ou de celle à l'âge de 16 ans, ne semblent pas influencer la perception du bonheur des individus. Les proportions de réponses sont relativement similaires d'une région à l'autre.

Corrélation

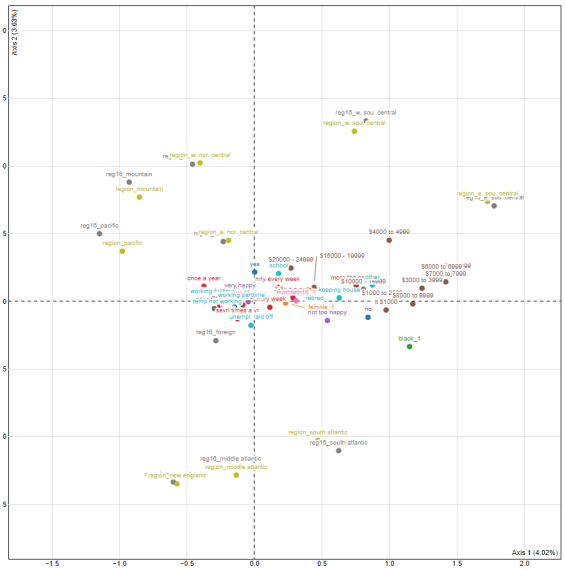
L'étape suivante de notre projet a consisté à vérifier les corrélations plausibles entre les différentes variables. Pour cela, nous avons eu recours à une ACP pour les variables quantitatives, et à une ACM pour les variables qualitatives.

Grace à l'ACP nous pouvons voir certaines corrélations se dessiner entre les variables.



On observe notamment une corrélation positive entre les variables 'educ' et 'prestige', ce qui semble cohérent : un niveau d'éducation plus élevé est souvent associé à une profession plus valorisée. Ces deux variables présentent également une corrélation négative avec tvhours, suggérant que les individus ayant un niveau d'éducation ou de prestige plus élevé passent en moyenne moins de temps devant la télévision. Par ailleurs, les trois variables liées aux enfants ('babies', 'teens', 'preteens') semblent elles aussi corrélées entre elles et négativement corrélées à tvhours, ce qui pourrait s'expliquer par un emploi du temps plus chargé lié aux responsabilités parentales.

Concernant l'ACM, elle a permis d'explorer les relations entre les variables qualitatives.



Les résultats indiquent peu de corrélations marquées entre ces variables, à l'exception notable des variables reg16 et region. Les modalités associées à ces deux variables sont souvent identiques, ce qui suggère que la majorité des individus n'ont pas changé de région entre leurs 16 ans et le moment du sondage.

Ces observations soulignent la présence de corrélations entre certaines variables, ce qui est à prendre en compte lors de la création de recettes de modèles. En effet, certains modèles, peuvent être impactés négativement par la présence de variables fortement corrélées.

Préparation des données & Recette de Base

Nettoyage

Après avoir observé et analysé les données brutes, l'étape suivante consiste à procéder à leur nettoyage. Cette phase est essentielle pour garantir la qualité des analyses à venir, en traitant les valeurs manquantes, les doublons éventuels, ainsi que les erreurs de typage des variables.

Premièrement, nous avons supprimé toutes les variables pouvant faire doublon, telles que 'Years' et les variables allant de 'y96' à 'y06', ou encore 'blackfemale' avec 'black' et 'female'. Pour ces cas, nous avons conservé les variables les plus générales, à savoir 'years', 'black' et 'female. Nous avons également supprimé la variable 'vhappy' afin d'éviter tout biais et problème lors des prédictions.

Ensuite, nous avons décidé de supprimer les variables 'gwbush00', 'gwbush04' et 'owngun' en raison de leur trop grande proportion de données manquantes, respectivement 80 %, 84 % et 34 %. De plus, nous avons estimé qu'elles n'étaient pas pertinentes pour expliquer les modalités du bonheur. C'est pour cette raison que 'owngun' a été retirée, contrairement à d'autres variables présentant des taux de données manquantes similaires, mais jugées plus pertinentes pour notre analyse.

Après cela, nous avons décidé de retirer les individus présentant des données manquantes pour les variables 'workstat', 'mothfath16', 'educ', 'teen', 'babies' et 'attend'. Nous nous sommes permis cette suppression car les proportions de valeurs manquantes étaient faibles. Cela engendre donc une assez faible perte d'observations tout en réduisant la quantité de données manquantes.

L'étape suivante a été un peu plus complexe, car nous avions deux variables : 'divorced' et 'widowed'. Ces variables sont particulièrement intéressantes mais présentent de forts taux de valeurs manquantes, respectivement 43 % et 35 %. Notre solution a été de croiser ces deux variables pour créer une nouvelle variable 'divorced_or_widowed', prenant la modalité "yes" si l'une des deux était positive, "no" si les deux étaient négatives, et "iap" sinon. Nous avons ensuite supprimé les individus pour lesquels des valeurs manquaient encore. Ce choix s'explique par le fait qu'après observation des données, nous avons constaté que les données manquantes des deux variables ne se recoupaient pas. Ainsi, nous avons pu réduire le nombre d'observations supprimées tout en diminuant le volume de valeurs manquantes.

Notre dernière étape du nettoyage des données a été le bon typage des variables, afin qu'elles soient interprétées correctement par les modèles et utilisables sans erreur. Nous avons ainsi transformé les variables numériques en entiers et les variables qualitatives en facteurs. La seule exception concerne la variable 'year', que nous avons également typée en facteur afin de traiter les années comme des catégories et non comme des valeurs numériques.

Voici la base de données après nettoyage.

Table 2: Description des variables de la base de données post nettoyage

Nom de la variable	Type	Description
`year`	Facteur	Année de réponse au questionnaire
`workstat`	Facteur	Information sur l'état de travail
`prestige`	Numérique	Note de prestige du métier exercé
`educ`	Numérique	Nombre d'années d'études
`reg16`	Facteur	Région de résidence à l'âge de 16 ans
`babies`	Numérique	Enfants de moins de 6 ans
`preteen`	Numérique	Enfants de 6 à 12 ans
`teens`	Numérique	Enfants de 13 à 17 ans
`income`	Facteur	Tranche de revenu
`region`	Facteur	Région de résidence actuelle
`attend`	Facteur	Fréquence de pratique religieuse

Suite à la page suivante

Table 2 – suite de la page précédente

Nom de la variable	Type	Description
`happy`	Facteur	Niveau de bonheur déclaré
`tvhours`	Numérique	Heures de télévision par jour
`mothfath16`	Facteur	Vivait avec père et mère à 16 ans (0 ou 1)
`black`	Facteur	Est noir (0 ou 1)
`female`	Facteur	Est une femme (0 ou 1)
`occattend`	Facteur	Pratique religieuse modérée (0 ou 1)
`regattend`	Facteur	Pratique religieuse régulière (0 ou 1)
`unem10`	Facteur	Au chômage dans les 10 dernières années (0 ou 1)
`divorce or widowed`	Facteur	L'individu est-il divorcé ,séparé ou veuf

Imputation

Après le nettoyage des données, certaines variables présentaient toujours des observations manquantes, dans leur cas, la suppression n'était pas une option envisageable, soit en raison de leur importance, soit en raison d'un taux de valeurs manquantes trop élevé pour éliminer les observations correspondantes. Ainsi, nous avons opté pour une imputation. Les variables concernées sont : 'prestige', 'income', 'unem10' et 'tvhours' avec respectivement 5 %, 12 %, 34 % et 31 % de données manquantes.

Plusieurs méthodes d'imputation ont été envisagées. L'imputation par moyenne ou médiane, qui consiste à remplacer les valeurs manquantes par la moyenne ou la médiane des valeurs observées, bien que simple à mettre en œuvre cette methode a tendance à lisser les données et à réduire la variance. L'imputation par régression utilise les autres variables pour prédire les valeurs manquantes via un modèle, offrant des imputations plus cohérentes mais pouvant introduire du biais en cas de mauvaise spécification. L'imputation par KNN, en prenant les observations les plus proches, permet de conserver les structures locales des données mais peut devenir coûteuse en temps de calcul. Et enfin, l'imputation par bagging, reposant sur un ensemble d'arbres de décision créés par bootstrap, est une méthode robuste qui limite la variance des imputations tout en étant plus complexe et couteuse egalement.

Finalement, la méthode retenue a été celle par bagging, car elle permet de mieux prendre en compte l'ensemble des variables, tout comme la méthode KNN. Cependant, bagging est plus rapide sur un grand jeu de données et offre généralement une meilleure précision.

Recette de base

Nous avons défini une recette pour prédire la variable "happy" à partir de toutes les autres variables dans le jeu de données data_train. Cette recette servira de base pour les différents modèles, tout en permettant d'ajouter certaines étapes spécifiques selon les besoins des modèle.

Tout d'abord, nous avons appliqué l'imputation des valeurs manquantes pour les variables citées précédement, à l'aide de la fonction step_impute_bag().

Ensuite, pour traiter le problème de déséquilibre des classes dans la variable cible happy, nous avons utilisé step_smotenc(). Nous avons laissé le paramètre over_ratio = tune() afin de tester différentes valeurs et choisir celle qui fonctionne le mieux pour notre modèle.

Apres cela nous avons appliqué les fonction step_center() et step_scale(). Ces deux étapes visent à rendre les variables comparables et à éviter que des variables à moyenne ou écart-type très différents n'influencent de manière disproportionnée l'apprentissage du modèle.

Enfin, nous avons ajouté une étape step_zv(), qui permet de supprimer toutes les variables ayant une variance nulle. Bien que cette étape soit redondante après le nettoyage des données, nous l'avons laissée en place comme sécurité supplémentaire.

Ainsi, voici la recette de base que nous avons construite, et qui servira pour certains modèles ou comme point de départ pour d'autres. Elle pourra être adaptée au fur et à mesure des besoins.

Différents modéles

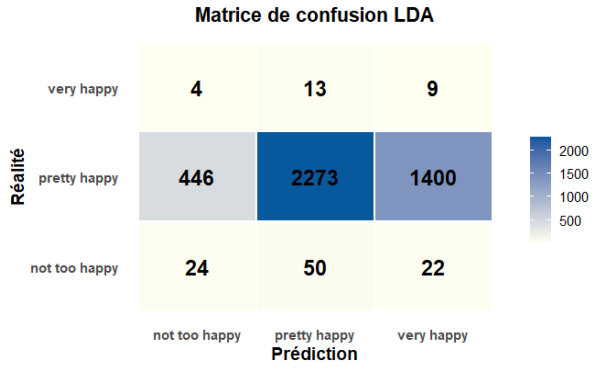
LDA

Notre premier modèle est une LDA (Linear Discriminant Analysis), une méthode de classification linéaire. Son principe est de projeter les données sur un espace de faible dimension tout en maximisant la séparation des classes. Pour cela deux hypotheses sont necessaires une distribution normale et une covariance identique entre deux classes.

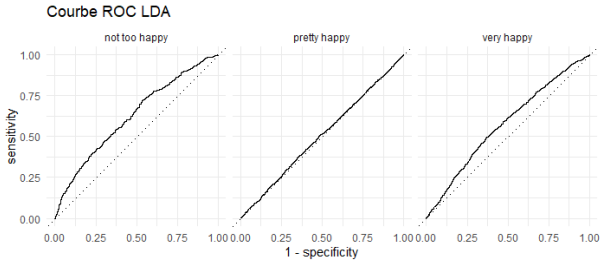
Ce modèle nécessitant que les variables catégorielles soient converties en variables numériques, nous avons ajouté à la recette la fonction step_dummy(). De plus, étant donné que le modèle est sensible à la multicolinéarité, nous utilisons également step_pca() afin de réduire les corrélations entre les variables.
Dans ce cas, un seul paramètre est véritablement à optimiser : l'over ratio. Il est aussi possible d'ajouter le paramètre neighbors dans la fonction step_smotenc(). Cependant, ce dernier a un impact limité sur les performances et augmente considérablement le temps de calcul.

Metric	Over ratio	.config
roc_auc	0.5	Preprocessor1_Model1
accuracy	0.5	Preprocessor1_Model1
precision	0.5	Preprocessor1_Model1
f_meas	0.75	Preprocessor2_Model1

Nous choisissons comme métrique d'évaluation le "roc_auc", car dans notre cas, il n'y a pas d'intérêt particulier à favoriser les faux négatifs plutôt que les faux positifs, ou inversement.L'accuracy elle peut être biaisée par le déséquilibre des classes.
Voici les résultats obtenus :



Métrique	Valeur (%)
Précision not too happy	5.06
Précision pretty happy	97.30
Précision very happy	0.63
Recall not too happy	25.00
Recall pretty happy	55.18
Recall very happy	34.62
Accuracy	54.37
AUC	58.44



Le modèle LDA présente une forte tendance à prédire la classe "pretty happy", ce qui se qui est visible dans la matrice de confusion ainsi que dans le tableau de résultats, avec une précision de 97,30 % pour cette catégorie. Par contre les classes "not too happy" et "very happy" sont largement sous-identifiées, affichant des précisions très faibles. Le recall suit la même logique, montrant une capacité limitée du modèle à détecter correctement ces deux classes. Avec une accuracy globale de 54,37 % et un AUC de 58,44 %, le modèle n'est que légèrement plus précis que le hasard. Enfin, les courbes ROC illustrent clairement les difficultés du modèle à prédire les classes "not too happy" et "very happy".

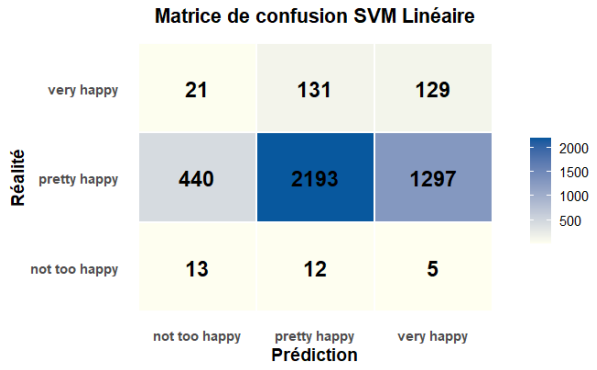
SVM Linéaire

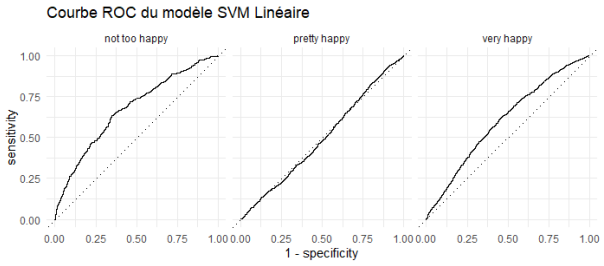
Notre second modèle est une SVM linéaire (Support Vector Machine), une méthode de classification reposant sur la recherche d'un hyperplan optimal pour séparer les classes. Son objectif est de maximiser la marge entre les observations des différentes classes les plus proches de la frontière de décision. La SVM ne fait pas d'hypothèse forte sur la distribution des données,contrairement à la LDA. Néanmoins, dans sa version linéaire, elle suppose que les classes sont séparables par une frontière linéaire.

Ce modèle nécessite également que les variables catégorielles soient converties en variables numériques. Pour cela, nous avons ajouté la fonction step_dummy() dans la recette, comme précédemment.
Ici encore, un seul paramètre est véritablement à optimiser : le over ratio. Il est aussi possible d'ajuster le paramètre neighbors via la fonction step_smotenc(), mais pour les mêmes raison que cités précédemment nous ne le faisons pas.

Metric	Over ratio	.config
roc_auc	0.4	Preprocessor4_Model1
accuracy	0.1	Preprocessor01_Model1
precision	0.4	Preprocessor4_Model1
f_meas	0.8	Preprocessor2_Model1

Ici encore, nous choisissons la métrique "roc_auc", car dans notre cas, il n'y a pas de raison particulière de privilégier les faux négatifs par rapport aux faux positifs, ou inversement. De plus, l'accuracy reste sensible aux déséquilibres de classes, ce qui pourrait biaiser le modèle.
Voici les résultats obtenus :





Métrique	Valeur (%)
Précision not too happy	2.74
Précision pretty happy	93.88
Précision very happy	9.01
Recall not too happy	43.33
Recall pretty happy	55.80
Recall very happy	45.91
Accuracy	55.06
AUC	60.73

Le modèle SVM Linéaire montre lui aussi une forte propension à prédire la classe "pretty happy", comme le montre la matrice de confusion et la précision élevée de 93,88 % pour cette catégorie. En revanche, les classes "not too happy" et "very happy" restent mal prédites, avec des précisions de seulement 2,74 % et 9,01 %. Le recall confirme ces déséquilibres, avec des valeurs de 43,33 % pour "not too happy" et 45,91 % pour "very happy". L'accuracy atteint 55,06 %, et l'AUC s'élève à 60,73 %, ce qui montre une amélioration par rapport au modèle LDA mais de trés peu. Les courbes ROC indiquent encore une limitation en ce qui concerne la capacité de discrimination du modèle.

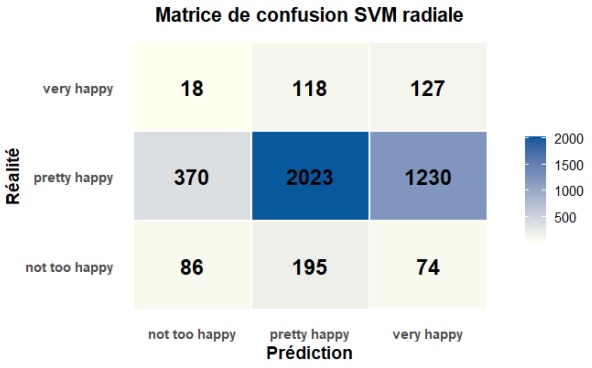
SVM Radiale

Notre troisième modèle est une SVM à noyau radial, une extension non linéaire de la SVM classique. Contrairement à la version linéaire, elle permet de capturer des frontières de décision complexes en projetant les données dans un espace de dimension supérieure grâce à une fonction noyau. Cette approche est plus adaptée lorsque les classes ne sont pas linéairement séparables dans l'espace d'origine. Le noyau radial permet une plus grande flexibilité dans la séparation des classes.

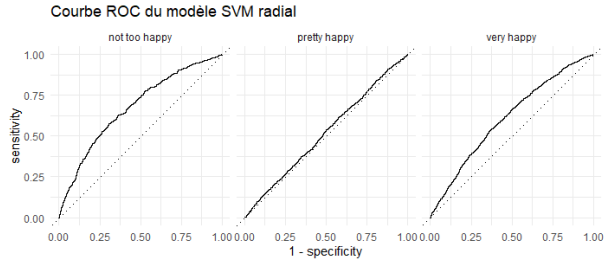
Ce modèle nécessite la même recette que la SVM linéaire, nous avons donc ici aussi ajouté la fonction step_dummy(). Deux paramètres sont à optimiser : l'over ratio, comme précédemment, et le paramètre cost, qui contrôle le compromis entre une marge large et une bonne classification des observations. Encore une fois, nous ne modifions pas le paramètre neighbors.

Metric	Cost	Over ratio	.config
roc_auc	0.5	0.5	Preprocessor1_Model1
accuracy	0.5	0.5	Preprocessor1_Model1
precision	0.5	0.5	Preprocessor1_Model1
f_meas	0.5	0.8	Preprocessor4_Model1

Comme précédemment, nous retenons le "roc_auc" comme métrique, pour les mêmes raisons qu'auparavant. Voici les résultats obtenus :



Métrique	Valeur (%)
Précision not too happy	18.14
Précision pretty happy	86.60
Précision very happy	8.87
Recall not too happy	24.23
Recall pretty happy	55.84
Recall very happy	48.29
Accuracy	52.72
AUC	62.63



Le modèle montre une amélioration notable par rapport aux modèles précédents. Bien qu'il prédise encore majoritairement la classe "pretty happy", il commence à mieux capturer les classes minoritaires. La précision atteint 86,60 % pour "pretty happy", mais reste faible pour "not too happy" (16,14 %) et "very happy" (8,87 %), bien qu'en hausse. Le recall suit cette tendance, atteignant 48,29 % pour "very happy" et 24,23 % pour "not too happy", indiquant une meilleure capacité de détection. L'accuracy globale est de 52,72 %, et l'AUC atteint 62,63 %, signalant un progrès dans la discrimination entre les classes. Les courbes ROC montrent également une séparation plus marquée des classes, notamment pour "very happy".

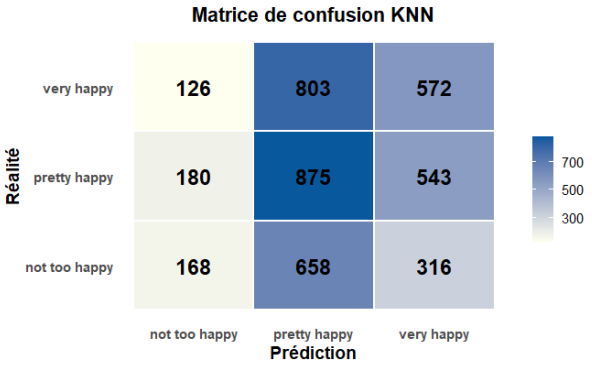
KNN

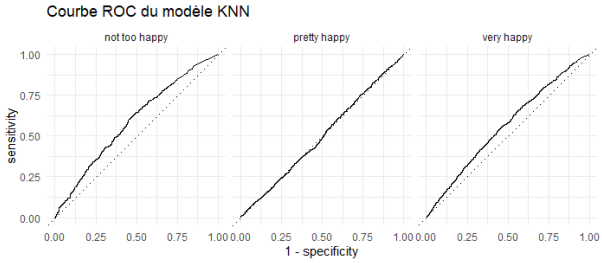
Notre quatrième modèle est un KNN (K-Nearest Neighbors), une méthode de classification basée sur la proximité des observations. Son principe repose sur l'attribution d'une classe à une observation en fonction des k observations les plus proches. Contrairement aux modèles précédents, KNN ne fait aucune hypothèse sur la distribution des données, mais sa performance dépend fortement du choix du paramètre k.

Dans le cas du modèle présent, la recette utilisée ne comprend pas d'étapes supplémentaires par rapport à la recette de base.
Contrairement aux deux premiers modèles, l'over ratio n'est pas le seul paramètre à optimiser. En effet, le paramètre neighbors, qui correspond au nombre k de voisins utilisés, est central dans les performances du modèle et doit donc être optimiser également.

Metric	Neighbors	Over ratio	.config
roc_auc	15	0.9	Preprocessor8_Model7
precision	13	0.2	Preprocessor1_Model6
f_meas	7	0.4	Preprocessor3_Model3

Comme précédemment, nous retenons le "roc_auc" comme métrique, pour les mêmes raisons qu'auparavant. On remarque également que le nombre de voisins utilisés (k = 15) est relativement élevé, ce qui permet de réduire les risques de surapprentissage.
Voici les résultats obtenus :





Métrique	Valeur (%)
Précision not too happy	35.23
Précision pretty happy	37.41
Précision very happy	40.18
Recall not too happy	14.64
Recall pretty happy	54.80
Recall very happy	38.21
Accuracy	38.10
AUC	56.07

La KNN montre des performances assez faibles. Contrairement aux précédents modèles, il ne privilégie pas clairement une seule classe, mais répartit ses prédictions de manière plus équilibrée, ce qui se voit dans la matrice de confusion. Cependant, les précisions restent basses pour toutes les classes, entre 35,23 % et 40,18 %. Le recall est particulièrement faible pour la classe "not too happy" (14,64 %), indiquant que cette catégorie est peu détectée. L'accuracy globale tombe à 38,10 %, et l'AUC à 56,07 %, ce qui en fait l'un des modèles les moins performants jusqu'à maintenant. Les courbes ROC confirment tout cela avec des séparations très limitées entre les classes.

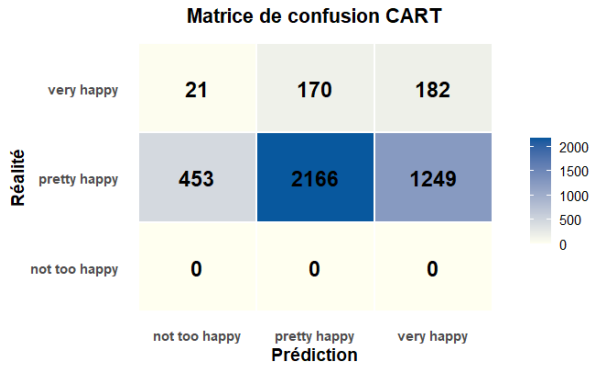
Arbre CART

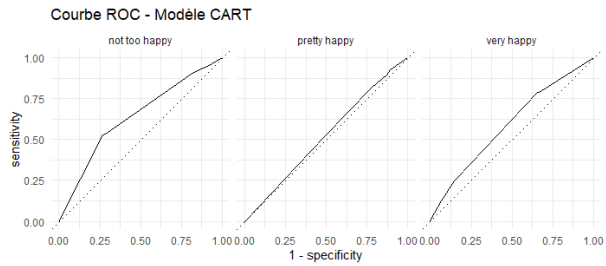
Notre cinquième modèle est un CART (Classification and Regression Trees), une modèle basé sur la construction d'un arbre de décision. À chaque nœud, l'algorithme choisit la variable et le seuil de coupure qui permettent de mieux séparer les classes. Ce modèle propose une interprétation intuitive, toutefois, il peut facilement sur-apprendre les données.

Avec ce modèle, nous utilisons uniquement la recette de base.
Deux paramètres sont ici à optimiser : comme toujours, l'over ratio, mais également le coût de complexité propre au modèle, qui permet de surveiller la profondeur de l'arbre et d'éviter le sur-apprentissage.

Metric	Cost_complexity	Over ratio	.config
roc_auc	0.002275846	0.3	Preprocessor1_Model12
precision	0.01	0.4	Preprocessor2_Model15
f_meas	0.01	0.4	Preprocessor2_Model15

Nous continuons d'utiliser la métrique "roc_auc", pour les mêmes raisons qu'auparavant. Toutefois, on remarque que le coût de complexité est ici particulièrement faible, ce qui peut indiquer un risque accru de surapprentissage.
Voici les résultats obtenus :





Métrique	Valeur (%)
Précision not too happy	0.00
Précision pretty happy	92.72
Précision very happy	12.72
Recall not too happy	NaN
Recall pretty happy	56.00
Recall very happy	48.79
Accuracy	55.36
AUC	59.96

Le modèle montre une forte tendance à prédire la classe "pretty happy", comme en témoigne la matrice de confusion et la précision très élevée pour cette catégorie (92,72 %). À l'inverse, la classe "not too happy" n'est jamais prédite, ce qui explique une précision et un recall de 0 % pour cette catégorie. La classe "very happy" est également sous-représentée, avec une précision de seulement 12,72 %. Malgré cela, le modèle atteint une accuracy de 55,36 % et un AUC de 59,96 %, des scores légers. Les courbes ROC confirment cette performance déséquilibrée, avec une bonne séparation pour "pretty happy", mais des résultats faibles pour les autres classes.

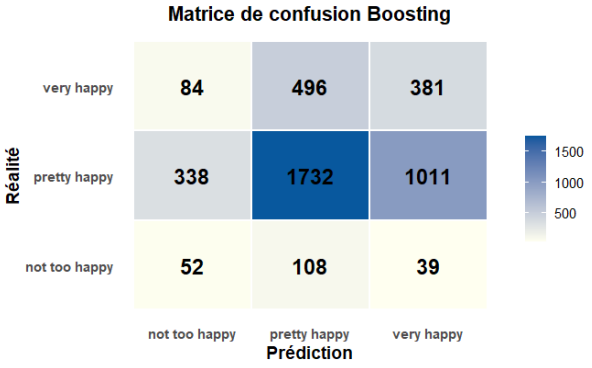
Boosting

Notre sixième modèle est un modèle de boosting, une méthode qui combine plusieurs arbres de décision faibles pour former un modèle robuste. Chaque nouvel arbre est entraîné pour corriger les erreurs commises par les arbres précédents. Cela permet souvent d'obtenir de bonnes prédictions, notamment sur des données complexes. Toutefois, elle peut devenir sensible au surapprentissage.

Ici nous n'ajoutons à la recette de base uniquement l'étape step_dummy().
Plusieurs paramètres sont optimisés : le nombre de variables prises en compte à chaque division (mtry), le nombre d'arbres (trees), la profondeur maximale des arbres (tree_depth), et le taux d'apprentissage (learn_rate), qui contrôle l'impact de chaque nouvel arbre sur le modèle final. L'over_ratio est toujours à optimiser également.

Metric	Mtry	Trees	Tree_depth	Learn_rate	Over_ratio	.config
roc_auc	2	100	5	1.023293	0.3	Preprocessor1_Model01
precision	2	100	5	1.023293	1	Preprocessor1_Model01
f_meas	2	100	7	1.023293	0.3	Preprocessor1_Model10

Nous choisissons à nouveau la métrique d'évaluation "roc_auc", pour les mêmes raisons que précédemment.
Voici les résultats obtenus :



Métrique	Valeur (%)
Précision not too happy	10.97
Précision pretty happy	74.14
Précision very happy	26.62
Recall not too happy	26.13
Recall pretty happy	56.22
Recall very happy	39.65
Accuracy	51.05
AUC	59.03

Le modèle présente une nette préférence pour la classe "pretty happy", comme on peut le voir sur la matrice de confusion, où la majorité des observations sont classées dans cette catégorie. Cela se traduit par une précision de 74,14 % pour cette classe. Les classes "not too happy" et "very happy" sont quant à elles bien moins bien identifiées, avec des précisions respectives de 10,97 % et 26,62 %. Le rappel suit la même tendance : le modèle détecte correctement 56,22 % des "pretty happy", mais seulement 26,13 % des "not too happy" et 39,65 % des "very happy". L'accuracy globale atteint 51,05 %, et l'AUC est de 59,03 %, ce qui montre une performance basse, légèrement au-dessus du hasard.

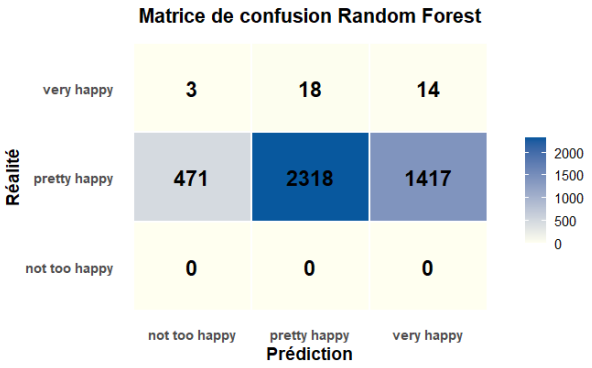
Random Forest

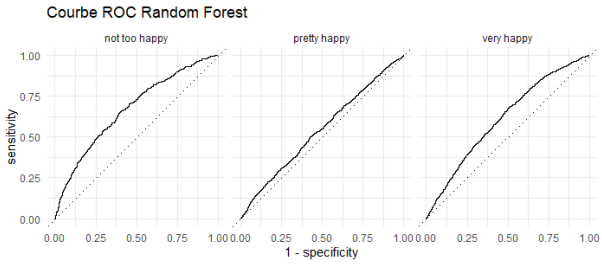
Notre dernier modèle est une Random Forest, une méthode d'ensemble basée sur l'assemblage de plusieurs arbres de décision. Chaque arbre est construit à partir d'un échantillon bootstrap, et à chaque nœud, seule une sous-partie des variables est considérée. Cette méthode permet de réduire la variance par rapport à un arbre seul, ce qui la rend plus robuste au sur-apprentissage.

Nous conservons pour ce modèle la recette de base, sans y ajouter d'étape supplémentaire. Trois paramètres sont à optimiser : l'over ratio, comme pour les modèles précédents, ainsi que mtry et trees, déjà présentés dans le modèle d'avant. Ces deux derniers contrôlent respectivement le nombre de variables à chaque nœud et le nombre total d'arbres construits.

Metric	Mtry	Trees	Over_ratio	.config
roc_auc	1	500	0.3	Preprocessor2_Model05
precision	1	500	0.2	Preprocessor1_Model05
f_meas	2	500	0.2	Preprocessor1_Model10

Nous choisissons comme métrique "roc_auc", pour une évaluation équilibrée de la performance du modèle. La mtry de 1 implique que chaque arbre ne sélectionne qu'une seule variable à chaque division, ce qui couplé au grand nombre d'arbres (500), présente un risque de sur-apprentissage à surveiller.
Voici les résultats obtenus :





Métrique	Valeur (%)
Précision not too happy	0.00
Précision pretty happy	99.19
Précision very happy	0.91
Recall not too happy	NaN
Recall pretty happy	55.07
Recall very happy	38.24
Accuracy	54.94
AUC	62.18

Ce modèle a tendance à prédire la classe "pretty happy", ce qui est visible dans la matrice de confusion,toutes les observations sont quasi-uniquement classées comme "pretty happy", y compris celles qui appartiennent aux deux autres classes. La classe "not too happy" n'est jamais prédite, ce qui entraîne une précision nulle pour cette classe. La précision pour "pretty happy" est très élevée (99,19 %), mais cela reflète un biais de prédiction énorme pour la classe majoritaire. Le rappel est correct pour "pretty happy" (55,07 %) mais très faible, voire inexistant, pour les autres classes. Le rappel pour "not too happy" est incalculable (NaN) car cette classe n'est jamais prédite. L'accuracy de 54,94 % donne une impression trompeuse de performance, alors que le modèle échoue à capturer la diversité des classes. Enfin, l'AUC de 62,18 % suggère une légère capacité discriminante, mais largement insuffisante au vus du déséquilibre observé.

Conclusion

Malheureusement, aucun des modèles testés ne se distingue réellement par de bonnes performances. Tous affichent une précision à peine supérieure à celle d'un tirage aléatoire, proche d'un lancer de pièce. Plusieurs raisons peuvent expliquer ces résultats décevants :

- La première, identifiée dès le début, est le fort déséquilibre des classes, qui biaise l'apprentissage de tous les modèles en les orientant vers la prédiction de la modalité majoritaire, "pretty happy".
- La seconde proviendrait probablement des variables explicatives elles-mêmes, parfois trop générales. Des variables plus ciblées comme "épanouissement au travail", "événement récent positif ou négatif ?", ou encore "niveau d'optimisme" auraient pu mieux capturer le bonheur.
- Egalement il est possible que les modalités de la variable cible posent problème : la perception de "very happy" peut fortement varier d'un individu à l'autre — pour certains cela correspondrait à un 8/10, pour d'autres à un 7/10.
- Enfin, la nature même de la variable à prédire pose des difficultés : le bonheur est un concept instable, hautement subjectif, influencé par le contexte, l'état psychologique, et des événements parfois récents. Deux individus ayant des caractéristiques identiques peuvent donner des réponses très différentes à une même question sur leur bonheur.

Face à ces limites persistantes, la modélisation du bonheur s'avère complexe. Malgré tout, les performances obtenues restent légèrement supérieures au hasard (d'environ 5 à 10 points de pourcentage), ce qui montre une légere utilité des modèles.

Si un choix devait malgré tout être fait, nous nous orienterions vers le modèle SVM radial. Bien qu'il soit plus long à entraîner, il est celui qui parvient le mieux à équilibrer la prédiction entre les classes, sans se limiter à une prédiction quasi-systématique de "pretty happy". Certes, sa précision globale est légèrement inférieure à celle du SVM linéaire, mais ce dernier prédit très mal la classe "not too happy", ce qui le rend moins pertinent.

Ainsi, notre choix se porterait sur le SVM radial, bien que ses performances restent globalement insuffisantes.

Sources

- Image de page de garde : https://www.freepik.com/free-photo/composition-with-different-expressions-copy-space_12558454.htm
- Information de la base de données : https://vincentarelbundock.github.io/Rdatasets/doc/wooldridge/happiness.html