DM de Machine Learning

Jean-Louis DU (IMPE)

6 avril 2025

Résumé

Ce devoir maison se décompose en 3 exercices. Dans chacun d'eux, on va devoir étudier une base de données et appliquer les méthodes vues en cours pour prédire une variable Y. Cette variable Y sera prédite à l'aide d'un prédicteur $\hat{Y} = f(X)$ avec X les autres variables appelées variables explicatives.
Pour mesurer la performance de notre prédicteur on prendra comme fonction de performances :

$$\begin{cases} \mathbb{P}(Y \neq \hat{Y}) & \text{pour une variable binaire} \\ \mathbb{E}\left[(Y - \hat{Y})^2\right] & \text{pour une variable continue} \end{cases}$$

On notera n le nombre d'observations et N le nombre de variables explicatives.

1 Présentation des méthodes de machine learning utilisées

1.1 Régression logistique (RLog) → Variable binaire

La régression logistique est une méthode de classification qui consiste à chercher notre prédicteur au sein d'une classe de fonction particulière. Pour un seuil s fixé, le prédicteur est le suivant :

$$\hat{Y} = \begin{cases} 1 & \text{si } P(Y = 1 \mid X) \geq s \\ -1 & \text{si } P(Y = 1 \mid X) < s \end{cases}$$

avec $P(Y = 1 \mid X) = \frac{e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_N X_N}}{1 + e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_N X_N}}$.

1.2 Régression linéaire multiple (RLin) → Variable continue

La régression linéaire multiple est une méthode de régression qui consiste à chercher un prédicteur de la forme :

$$\hat{Y} = \beta_0 + \beta_1 X_1 + \cdots + \beta_N X_N$$

Comme son nom l'indique on cherche une combinaison linéaire des différentes variables explicatives X_i pour approcher Y. Dans ce modèle, on suppose que les erreurs suivent une loi normale de moyenne 0 et de variance constante (homoscédasticité).

1.3 K plus proches voisins (KPPV) → Variable binaire/continue

La méthode des k plus proches voisins est une méthode de classification qui consiste à donner une prédiction basée sur une notion de distance.
Étant donné une certaine mesure de distance, on détermine les k voisins les plus proches selon cette distance et on prédit l'état majoritaire parmi ces voisins ou la moyenne des voisins. Généralement on prend un k impair.

1.4 Arbres → Variable binaire/continue

Les arbres de classification sont une autre méthode d'apprentissage intéressante. C'est une suite de conditions basée sur des seuils appliqués aux différentes variables explicatives. Avec toutes ces conditions on se retrouve avec une partition du domaine. On assigne ensuite une valeur (l'élément majoritaire en classification, la moyenne en régression par exemple) à chaque région.

1.5 Random Forest (RF) → Variable binaire/continue

Cette méthode se repose sur le principe des arbres de classifications. Au lieu de faire un seul arbre très riche qui peut provoquer un sur-ajustement, on va plutôt faire plusieurs arbres plus simples. Puis on va considérer le prédicteur global comme la moyenne des prédicteurs sur chacun des arbres en régression et la classe majoritaire en classification.
Pour chacun des arbres on va considérer un échantillon de toutes nos données, de plus on va se limiter à un sous aléatoire de paramètres.

1.6 Gradient Boosting (Gboost) → Variable binaire/continue

Le Gradient Boosting est une technique d'apprentissage supervisé qui améliore la performance des modèles en combinant plusieurs arbres de décision de profondeur faibles de manière itérative. À chaque itération, le nouvel arbre corrige les défauts de l'arbre précédent jusqu'à aboutir à un modèle performant. L'optimisation se fait par méthode de Gradient, d'où son nom. Le prédicteur est ainsi l'arbre obtenu après convergence.

1.7 Réseaux de neuronnes (RN) $\rightarrow$ Variable binaire/continue

C'est une méthode qui, à partir de l'assemblage de calculs simples (combinaison linéaire, application de fonctions), donne un résultat riche et complexe. Pour arriver au résultat final, il y a donc plein de calculs intermédiaires effectués sur ce qu'on va appeler des neurones.
Chaque neurone reçoit des informations, effectue un calcul, applique une fonction d'activation et transmet le résultat aux neurones des couches suivantes.
L'apprentissage se fait en ajustant les poids des connexions entre neurones. Notre prédicteur est donc le résultat obtenu en couche de sortie.

2 Exo 1

2.1 Analyse de la base de donnée

On analyse un jeu de données bancaires pour construire un prédicteur portant sur la possession de la carte Visa Premier des clients de cette banque. Pour cela, on a à notre disposition quelques variables qualitatives telles que la situation familiale, le sexe ou encore le département de résidence. On a aussi de nombreuses variables quantitatives portant sur l'âge, le nombre d'impayés en cours et des montants pour diverses opérations par exemple. On a au total 56 variables.

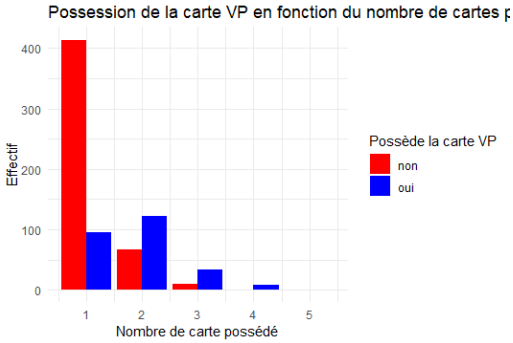
Pour commencer on va nettoyer la base de données, on va retirer des variables contenant beaucoup de valeurs manquantes (G25G26S, G03G04S, etc...). De plus, on va se limiter aux observations n'ayant pas de données manquantes pour les autres variables. On aurait pu aussi retirer les individus trop jeunes ou trop âgés comme proposé dans l'étude, mais on a préféré les laisser.
Après cela il nous reste 753 observations complètes.
Avant de continuer, on va créer des variables qualitatives basées sur plusieurs de nos variables quantitatives dans le but de tester nos méthodes d'apprentissage sur seulement les variables quantitatives (visa_cat), qualitatives(visa_num) ou les deux(visa_full).

2.2 Question 1

La classification supervisée dans le cas d'une variable à deux modalités {-1,1} consiste à donner un prédicteur $\hat{Y} = f(X)$ devinant la bonne modalité à partir des variables explicatives $(X_i)_{i\in 1,\dots,N}$. Pour ce faire, on se restreint généralement à une classe de fonctions dans laquelle on va optimiser des paramètres permettant un prédicteur performant. Pour mesurer la performance d'un prédicteur, on se donne une fonction de perte à laquelle est associée une fonction de performance. Dans le cas d'une variable binaire, on peut choisir comme fonction perte : $\mathbb{1}_{Y\neq\hat{Y}}$ et comme fonction performance : $P(Y \neq \hat{Y})$. Généralement, on ne connaît pas explicitement la fonction performance, on évalue ainsi la performance avec l'erreur empirique : $\frac{1}{n}\sum_{j=1}^{n}\mathbb{1}_{Y^j\neq\hat{Y}^j}$.
Notre indicateur de performance est donc l'erreur empirique qu'on cherche donc à minimiser. En le minimisant on s'assure que notre prédicteur se trompe moins.
Dans notre cas on veut prédire la variable Y qui vaut "oui" si le client possède la carte VP et "non" s'il ne la possède pas.

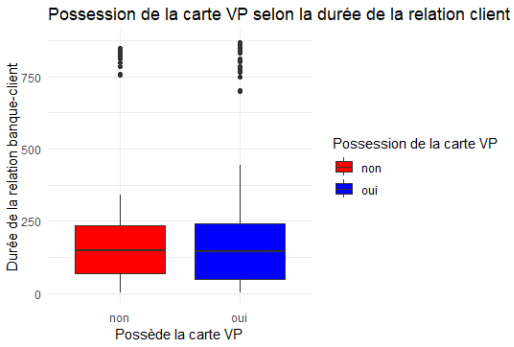
2.3 Question 2

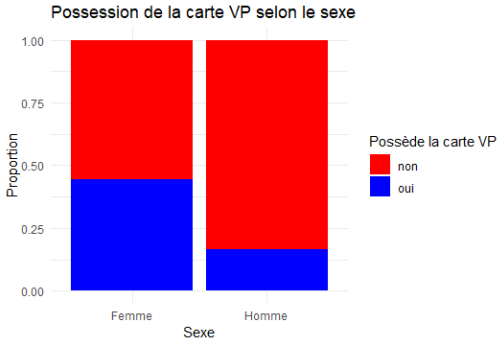
Après avoir regardé la liste des variables explicatives, on a certaines intuitions vis-à-vis de la corrélation entre certaines variables explicatives et la possession de la carte VP. On va donc vérifier si notre intuition ne se trompe pas en visualisant quelques graphiques.



Logiquement, on pense que plus un client possède de cartes, plus il a de chances que parmi ses cartes il y ait la carte VP. En effet, on remarque que c'est vrai. Sur ce graphique, bien qu'en effectif il y ait plus de personnes possédant peu de cartes, si l'on regarde en termes de proportion par nombre de cartes, on voit que le pourcentage de clients ayant la carte VP croît avec le nombre de cartes possédées.

De prime abord, on pense aussi que plus un client est depuis longtemps chez sa banque, plus il est fidèle et donc plus il a de chances de posséder la carte Visa Premier. Mais les boîtes à moustaches suivantes prouvent le contraire. En effet, on ne remarque aucune différence notable entre ceux possédant et ceux ne possédant pas de carte VP.





En étudiant ce dernier graphe, on remarque une différence assez importante entre les femmes et les hommes en termes de possession de la carte VP. Il y a presque trois fois plus de femmes en proportion que d'hommes possédant la carte VP. On observe ainsi un investissement assez genré.

2.4 Question 3

On se propose de tester plusieurs méthodes de classification supervisée pour ce problème. On analyse une variable binaire, alors on va tester les méthodes de régression logistique et des k plus proches voisins. Ces méthodes sont considérées assez simples, on choisit donc de compléter par des méthodes plus avancées telles que les arbres, RF, RN et Gboost. Pour chacune de ses méthodes, on va les tester sur les bases de données visa_cat, visa_num et visa_full. On va utiliser une validation croisée en 5 blocs sur nos différents modèles, puis on va comparer les erreurs sur nos données d'entraînement. Une validation croisée par bloc consiste à séparer nos données en plusieurs blocs. On va ensuite choisir un bloc, on entraîne nos données sur les autres blocs et on calcule notre erreur sur le bloc sélectionné. On répète la même opération sur chaque bloc et on calcule l'erreur moyenne obtenue. On sélectionne ainsi le modèle avec l'erreur moyenne obtenue la plus faible.
On aurait aussi pu choisir notre modèle sur l'aire de la courbe ROC puisque que l'on est dans le cas binaire. On va tout de même en parler très rapidement dans la suite.
On regroupe toutes nos observations dans le tableau ci-dessous.

	RL	KNNP	Arbre	RF	Gboost	RN
visa_full	23.0	29.8	12.1	8.9	7.1	30.9
visa_num	18.1	29.8	11.7	9.3	7.3	33.1
visa_cat	24.0	24.6	21.2	19.8	18.6	19.3

TABLE 1 – Erreurs (en %) observées selon la méthode et la base de données utilisée

On voit que la méthode de Gradient Boosting est la plus efficace sur notre base de données. On va choisir celle-ci pour la suite.

2.5 Question 4

Pour calculer l'erreur empirique, puisque l'on n'a pas de données de test, on sépare nos données en deux parties : une partie pour l'entraînement du modèle appelé "train" et une autre partie pour calculer l'erreur empirique appelée "test". On obtient une erreur empirique de %. De plus, on trace aussi la courbe ROC pour vérifier que notre modèle est efficace pour tout seuils. On voit que c'est le cas, on obtient une aire sous la courbe très proche de 1, elle est de 0.9992.
On va maintenant observer le comportement de notre prédicteur sur quelques observations des données tests.

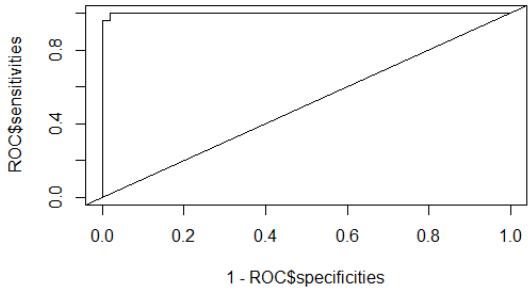


FIGURE 1 – Courbe ROC pour le Gradient Boosting

On a les scores suivants, les prédictions associées $\hat{Y}$(on a pris pour seuil s = 0.5) et les vraies valeurs Y. Sur ces trois valeurs notre prédicteur a deviné correctement toutes les valeurs. On

Score "non"	Score "oui"	$\hat{Y}$	Y
0.9838757	0.01612434	non	non
0.9875894	0.01241056	non	non
0.1220892	0.87791076	oui	oui

TABLE 2 – Analyse de 3 observations

remarque néanmoins que notre prédicteur se serait trompé pour un seuil plus élevé, par exemple s = 0.9.

3 Exercice 2

3.1 Présentation de la base de donnée

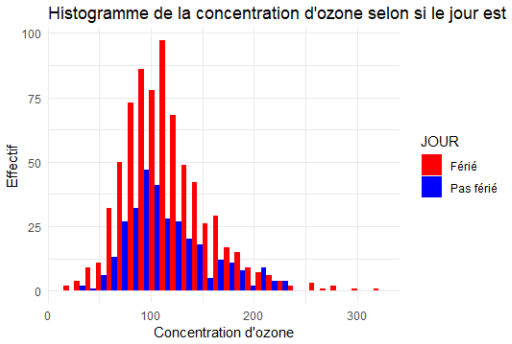
On s'intéresse maintenant à une base de données de Météo France portant sur la concentration d'ozone observée qui sera notre variable d'étude Y. On a pour variables explicatives des données de température, de diverses molécules, du site de mesure et des caractéristiques du vent. On a au total 10 variables Contrairement à la base de données sur la carte VP, cette base de données est complète (elle ne comporte aucune valeur manquante). On fait juste attention à mettre en variable factor les variables qualitatives et en variable numérique les variables numériques.

3.2 Question 1

La classification supervisée dans le cas d'une variable continue consiste à donner un prédicteur $\hat{Y} = f(X)$ devinant au mieux à partir des variables explicatives $(X_i)_{i\in 1,\dots,N}$. Pour ce faire, on se restreint généralement à une classe de fonctions dans laquelle on va optimiser des paramètres permettant un prédicteur performant. Pour mesurer la performance d'un prédicteur, on se donne une fonction de perte à laquelle est associée une fonction de performance. Dans le cas d'une variable continue, on peut choisir comme fonction perte : $(Y - \hat{Y})^2$ et comme fonction performance : $E((Y-\hat{Y})^2)$. Généralement on évalue la performance avec l'erreur empirique : $\frac{1}{n}\sum_{j=1}^{n}(Y^j - \hat{Y}^j)^2$. Notre indicateur de performance est donc l'erreur empirique qu'on cherche donc à minimiser. En le minimisant on s'assure que notre prédicteur se trompe moins.
Dans notre cas on veut prédire la variable Y qui indique la concentration d'ozone observée.

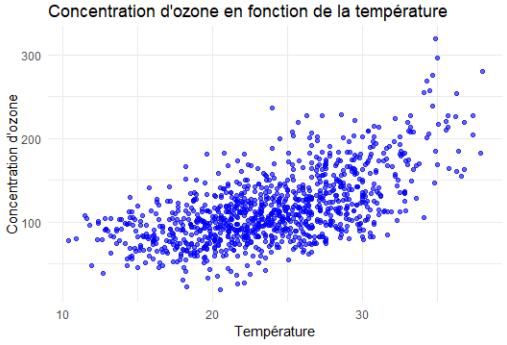
3.3 Question 2

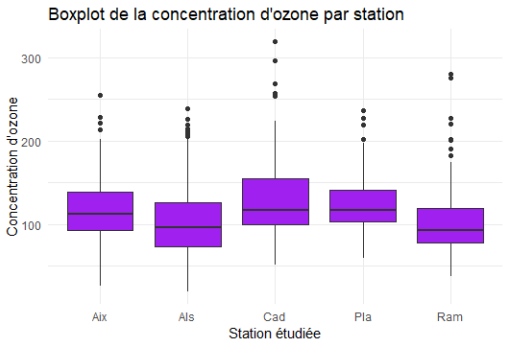
Puisque l'on a peu de variables explicatives, on se propose d'étudier leurs relations avec la concentration d'ozone. On va en afficher quelques unes qu'on trouve intéressantes.



Sur cet histogramme est représentée la concentration d'ozone en fonction de si le jour est férié ou non. On ne remarque pas de différences notables, on met en doute l'influence de cette variable.

Ensuite, dans ce nuage de points de la concentration d'ozone en fonction de la température, on observe une légère tendance. On remarque que lorsque la température croît, la concentration d'ozone croît aussi.





On voit aussi que selon la station où l'on a pris les mesures, la concentration d'ozone a l'air différente. En effet, les quantités n'ont pas l'air d'être au même niveau selon la station. La station Cad semble être la plus riche en ozone, à l'inverse la station Ram semble être la plus pauvre en ozone. On observe ainsi une influence du lieu de prise de la mesure.

3.4 Question 3

Pour commencer on considère une régression linéaire multiple sur toutes les variables explicatives. On obtient une variance expliquée ajustée R^2_{ajuste} de 52.7%. On s'intéresse au R^2_{ajuste} plutôt qu'au R^2 classique car on prend en compte le nombre de variables explicatives dans le modèle. Dans le cas classique lorsque l'on rajoute une variable on augmente toujours le R^2, ici on augmente le R^2_{ajuste} si et seulement si la variable aide à la prédiction.
Ainsi on va tenter plusieurs régressions linéaires multiples afin de trouver le meilleur R^2_{ajuste}. Pour cela on va se baser sur les graphes descriptifs de la partie précédente.
On commence par seulement considérer les variables TEMPE, MOCAGE et STATION car les graphiques ont montré une corrélation évidente avec notre variable d'intérêt. On trouve $R^2_{ajuste} =$ 50.9%. En rajoutant la variable JOUR, on remarque que le R^2_{ajuste} diminue à 50.7%, ainsi elle n'explique pas la concentration d'ozone. On décide donc de ne pas l'inclure dans notre régression. En rajoutant ensuite les mesures d'humidité et de monoxyde et dioxyde d'azote on améliore le R^2_{ajuste}, il passe à 51.8%. En rajoutant les mesures du vent dans notre régression on atteint 52.9%. C'est le meilleur score qu'on a pu avoir par régression linéaire multiple. Pour ce cas-là on note qu'on obtient un R^2 classique valant 53.4%.

3.5 Question 4

On teste d'autres méthodes d'apprentissage supervisé, notamment Random Forest, Gradient Boosting et Réseaux de neurones. Après quelques tests, on remarque qu'inclure la variable JOUR mène à un meilleur R^2_{ajuste}. Pour ces 3 méthodes, on décide donc de la conserver dans nos variables explicatives. On obtient pour variance expliquée les valeurs suivantes :

	RF	GBoost	RN
R^2	61.2	59.6	58.9

TABLE 3 – R^2 pour 3 méthodes d'apprentissage supervisées

On observe que la méthode expliquant le mieux la variance de nos données est Random Forest. Ainsi, on choisit cette méthode-là pour calculer son erreur empirique.

3.6 Question 5

Comme pour l'exercice 1, pour calculer l'erreur empirique, on sépare nos données en deux parties une partie pour l'entraînement du modèle appelé "train" et une autre partie pour calculer l'erreur empirique appelée "test". On obtient une erreur empirique de 566, ce qui nous donne à la racine 23.8. On remarque aussi que l'erreur est très dépendante de la partition faite sur les données. De plus, on voit que l'erreur de test est bien plus élevée que l'erreur d'apprentissage, cela nous laisse penser qu'on est en sur-apprentissage ou en sous-apprentissage. En faisant varier on trouve qu'une profondeur de 5 est optimale, malgré cela l'écart avec l'erreur d'entraînement reste très élevé.

On va maintenant observer le comportement de notre prédicteur sur quelques observations des données tests.

On a les prédictions suivantes $\hat{Y}$ et les vraies valeurs Y. On remarque des prédictions relativement proches pour les deux premières observations. Cependant, pour la dernière, on observe un

$\hat{Y}$	Y
100.0	103
112.1	116
92.4	73

TABLE 4 – Analyse de 3 observations

écart d'au moins 20. Ainsi, notre prédicteur a encore beaucoup à découvrir efficacement la valeur de concentration d'ozone.

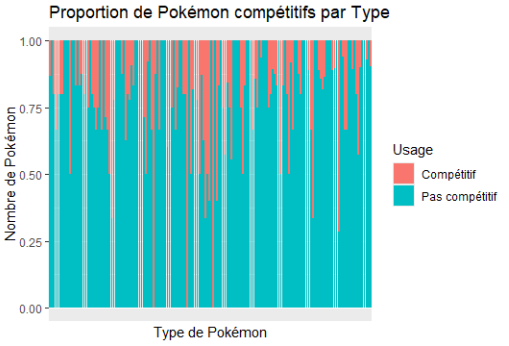
4 Exercice 3

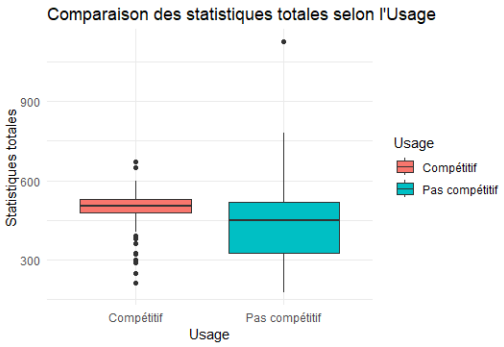
4.1 Présentation des données

On a choisi une base de données portant sur un jeu vidéo de l'entreprise Nintendo, à savoir Pokemon (https ://www.kaggle.com/datasets/danielsmdev/pokemon-competitive-usage-smogon-and-vcgworlds). Pokemon est une licence où l'on retrouve des créatures ayant des pouvoirs se battant aux côtés d'un dresseur pour explorer le monde. Chaque espèce de Pokemon est unique de par sa forme mais aussi de par un ensemble de caractéristiques (attaque, défense, vitesse, etc...).
En effet, on se propose de trouver un prédicteur permettant de déterminer la capacité d'un Pokemon à être performant d'un point de vue compétitif. Dans cette base de données on va considérer comme Pokemon compétitif ceux ayant été joués lors des mondiales en 2024.
On commence par travailler la base de données possédée. On enlève les données liées aux compétitions autres que les Worlds 2024. Ce que l'on cherche à faire est de déduire le caractère compétitif des statistiques du Pokemon et non de ses anciennes participations à d'autres tournois.
On choisit aussi de regrouper les types 1 et 2 en une seule catégorie double-type. Il est bien plus intéressant d'analyser une combinaison de types, cela donne plus d'informations sur le comportement défensif ou offensif du Pokemon.

4.2 Analyse descriptive

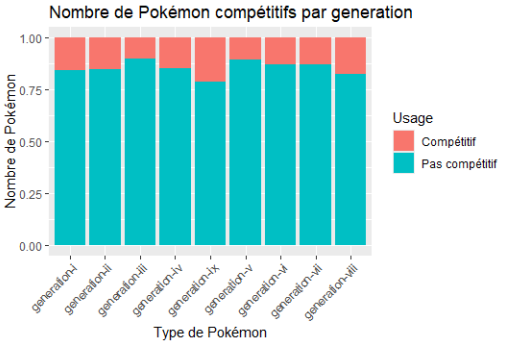
On remarque que le double-type des Pokemons influent grandement sur son caractère compétitif. On remarque que certains double-type ont l'air d'être très performants contrairement à d'autres. On a choisi de ne pas afficher le nom des doubles-types en abscisses car il y en a beaucoup trop. Cela s'explique par le fait que certains double-type tel que eau-sol possèdent beaucoup de résistance et très peu de faiblesses, tandis que roche-sol à l'inverse est très fragile.





Sur cette boîte à moustache on voit que la grande majorité des Pokemons compétitifs se concentrent autour d'une moyenne de statistiques totales aux alentours de 500. On le comprend très bien, car pour qu'un Pokemon puisse exceller en compétition il doit pouvoir avoir beaucoup d'atouts. Néanmoins il y a des restrictions dans les concours, en effet certains Pokemons considérés comme trop puissant sont bannis. C'est pour cela que l'on retrouve des grosses valeurs chez les Pokemons non compétitifs.

Les Pokemons sont classés en génération. Ils sont basés sur les jeux dans lesquels ils ont fait leur première apparition. On remarque que l'on a plus ou moins la même proportion de Pokemons compétitifs pour chaque génération, cela n'a pas l'air d'être un critère rendant le Pokemon puissant et compétitif.



4.3 Apprentissage supervisé

On se retrouve à vouloir déterminer une variable à deux modalités : "Compétitif" et "Non compétitif". On peut donc tenter une régression logistique. Cependant nos données comportent

les classes "ability1", "ability2" et "hidden_ability" qui sont des variables à plusieurs centaines de facteurs. On décide donc de les omettre pour gagner un temps considérable. On note qu'avec plus de temps on aurait pu les regrouper dans une sorte de tier-list pour leur assigner une nouvelle étiquette. Ainsi on obtiendrait moins de modalités pour cette nouvelle variable.

	RL	Arbre	knnp	RF	GBoost	RN
Erreur (en %)	18.6	14.3	14.8	14.4	13.3	14.4

TABLE 5 – Erreurs pour différentes méthodes d'apprentissage

On voit que le Gradient Boosting est la méthode la plus efficace.

4.4 Erreur empirique

Comme pour les exercices précédents on sépare nos observations en deux. On entraîne nos données sur "train". Puisque notre variable types contient beaucoup de double-type unique, on doit faire correspondre les levels des données de test et d'entraînement avec celui des données complètes. En l'entraînant sur "train", on observe une erreur d'entraînement de 14.4%. On remarque étrangement que notre erreur empirique est de 13.2%, cela n'a pas d'explication claire, hormis peut-être un tirage chanceux des données de test. En le lançant pour une autre seed, on observe le même résultat.

Y \ $\hat{Y}$	Compétitif	Pas compétitif
Compétitif	1	17
Pas compétitif	0	111

TABLE 6 – Matrice de confusion

On remarque que notre prédicteur a du mal à deviner le statut "Compétitif". Pour accepter les "Compétitif" plus facilement on peut diminuer le seuil d'acceptation (qui est de 0.5 par défaut), mais cela va se faire au risque d'accepter plus de faux positifs.

Score "Pas compétitif"	Score "Compétitif"	$\hat{Y}$	Y
0.5156568	0.4843416	Pas compétitif	Pas compétitif
0.4014944	0.59850563	Compétitif	Compétitif
0.9337374	0.06626263	Pas compétitif	Pas compétitif

TABLE 7 – Analyse de 3 observations pour un seuil s = 0.5

On voit que pour les deux premières observations les scores sont très proches, avec un seuil légèrement plus élevé ou moins élevé on aurait eu une prédiction différente.

4.5 Conclusion

Le prédicteur obtenu donne une bonne erreur empirique, mais cela est peut-être dû à la base de données qui n'est pas propice à une étude de la sorte. Peut-être qu'en étudiant les variables "hability", on aurait eu un meilleur prédicteur. Comme dit précédemment, on aurait peut-être pu créer une nouvelle variable catégorielle (selon leur puissance) pour réassigner les "hability" dans de plus grands groupes.
De plus, pour étudier de manière plus complète les Pokemons, il aurait fallu étudier le panel de compétences pouvant être apprises. On aurait peut-être pu les étudier selon leur rôle, si le Pokemon est plutôt offensif, défensif, en soutien.
Les modèles étudiés ne prennent pas en compte la compétition inter-Pokemon. Si deux Pokemons présentent des caractéristiques et des rôles identiques, avec l'un des deux qui est légèrement meilleur en termes de vitesse, par exemple, il paraît évidemment qu'on le préférera à son compagnon.

5 Bibliographie

— Base de donnée et inspiration pour l'exercice 3 :
https ://www.kaggle.com/datasets/danielsmdev/pokemon-competitive-usage-smogon-and-vcgworlds

— Base de donnée et inspiration pour l'exercice 2 :
https ://www.math.univ-toulouse.fr/ besse/Wikistat/pdf/st-scenarapp- ozone-meteoF.pdf

— Base de donnée et inspiration pour l'exercice 1 :
https ://www.math.univ-toulouse.fr/ besse/Wikistat/pdf/st-scenar-app-visa.pdf

— Aide pour l'exercice 1 :
https ://github.com/wikistat/Apprentissage/blob/master/GRC-carte_Visa/Apprent-Python-Visa.ipynb