19/01/2022

Prédiction de la durée d'un match de tennis du circuit ATP

Apprentissage automatique, Master SEP

HUGO CARLIN, SARA MADANI, MARIEM BOUHADDA, IANIS DORGHAM, PIERRE-EMMANUEL BADIN

Table des matières

Introduction

L'ATP (Association of Tennis Professionals) a été fondée en 1972 et organise depuis 1990 le circuit mondial masculin des tournois de tennis. De très nombreuses données ont ainsi été collectées concernant les matchs de ces tournois, notamment des informations sur les joueurs impliqués, ou bien sur le déroulement des matchs.

Nous allons ici tenter de construire un modèle qui nous permettra de prédire la durée d'un match de tennis. Nous utiliserons pour cela une base de données répertoriant les matchs du circuit professionnel entre 1968 et 2019. Celle-ci est constituée de 49 variables détaillant le déroulement du match et les caractéristiques des deux joueurs. Nous nous intéresserons en particulier aux variables suivantes :

- tourney_id : identifiant du tournoi
- tourney_name : nom du tournoi
- surface : surface sur laquelle se joue le tournoi
- draw_size : nombre de joueurs participants au tournoi final
- tourney_level : type de tournoi (F = finals, G = grand chelem, M = masters 1000, A = ATP250 ou ATP500, D = Davis Cup)
- tourney_date : date de début du tournoi
- match_num : numéro du match
- winner_id / loser_id : identifiant du vainqueur / du perdant
- winner_name / loser_name : nom du vainqueur / du perdant
- winner_hand / loser_hand : avec quelle main joue le vainqueur / le perdant
- winner_ht / loser_ht : taille du vainqueur / du perdant
- winner_age / loser_age : âge du vainqueur / du perdant
- best_of : nombre de sets maximum dans le match
- round : tour auquel correspond le match
- minutes : temps du match en minutes
- winner_rank / loser_rank : rang du vainqueur / du perdant au classement ATP

Avant de construire un modèle de prédiction, nous allons effectuer un pré-traitement des données afin de les rendre exploitables.

Pré – traitement des données

La première étape du pré-traitement des données est le retrait des 5200 premiers matchs de la base. Cela correspond au fait qu'une règle instaurée en 1970 a établi la règle du jeu décisif. Avant cette règle, un set ne pouvait être gagné que par deux jeux d'écarts, ce qui augmentait la durée du match de manière non négligeable. Depuis, un jeu décisif est joué en cas d'égalité 6 jeux partout, le gagnant remporte le set. Nous avons donc supprimé les matchs antérieurs à cette règle.

Nous retirons ensuite les variables qui ne nous intéressent pas, en particulier les variables concernant le déroulement du match, étant donné que le modèle que nous cherchons à mettre en place fera des prédictions avant le match.

Puis, Nous retirons les matchs comprenant des valeurs manquantes pour certaines variables, ce qui les rends inexploitables. Nous observons une stabilité dans les données à partir de 1991 et la réduction considérable des valeurs manquantes à partir de cette année-là.

Enfin, nous enlevons les matches interrompus par un abandon (mention « RET » dans le score) et certains matches irréalistes, issues d'erreurs humaines (par exemple des matches en 3 sets gagnants fini en moins de 30 minutes, et joué en 4 ou 5 sets).

Toutes ces modifications ont réduit le nombre de matches de 176 116 initialement à 77 545.

Une fois ce nettoyage effectué, nous pouvons nous attaquer à la prédiction.

Première tentative : k-plus proches voisins

Nous allons dans un premier temps tenter d'établir une prédiction en utilisant la méthode de classification supervisée des k-plus proches voisins (KNN). Celle-ci a pour but, pour une donnée x, de déterminer la valeur de la variable cible Y la plus probable en déterminant les données les plus ressemblantes à x. Il faut pour cela définir une distance ; nous utiliserons ici la distance euclidienne.

Nous avons codé une nouvelle variable « minutes_recod » pour créer des classes en fonction de la durée des matches. Nous avons ainsi distingué 4 classes : Courte durée (28min-1h15), Durée moyenne (1h15-1h36), Longue durée (1h36-2h05) et Très longue durée (2h05-12h00). C'est cette variable que nous allons tenter de prédire.

Pour la mise en place de notre méthode, il nous faut maintenant diviser notre base initiale en une base d'entrainement et une base de test. On choisit ici d'utiliser 80% des valeurs pour la base d'entrainement (soit 62 036 matchs) et 20% pour l'autre (soit 15 510 matchs).

Nous avons choisi ici un k égal à 200. Après l'application de l'algorithme de classification par KNN, nous pouvons comparer les prédictions obtenues avec les valeurs réelles. Nous obtenons une précision de 35,8%. Ce résultat est plutôt décevant car l'ensemble des prédictions montre une faible précision quant à la catégorie des matches en questions. Nous allons donc utiliser une autre méthode pour notre apprentissage supervisé.

Deuxième tentative : Arbre de décision

L'arbre de décision est un algorithme d'apprentissage automatique supervisé qui peut être utilisé pour effectuer la classification. La terminologie de base importante des arbres est la suivante.

- La racine : représente une population entière ou un ensemble de données qui est divisé en deux ou plusieurs ensembles purs. Il contient toujours une seule variable d'entrée.
- Le nœud terminal : Ce nœud n'est pas divisé davantage et contient la variable de sortie.

Ces arbres sont basés sur l'homogénéité des groupes formés. L'homogénéité ou l'impureté des données est quantifiée en calculant des métriques comme l'entropie, le gain d'information et l'indice de Gini.

La métrique la plus couramment utilisée est le gain d'information. Il s'agit de la mesure permettant de quantifier la quantité d'informations qu'une variable caractéristique fournit sur la classe.

Le meilleur modèle d'arbre de décision a été obtenu à la suite d'une division de 70% pour la base d'entrainement et 30% pour la base test. On n'a obtenu que 35,3% de précision. Nous allons par la suite essayer un autre modèle de classification supervisée. C'est l'algorithme « Random Forest » (ou « forets aléatoires »).

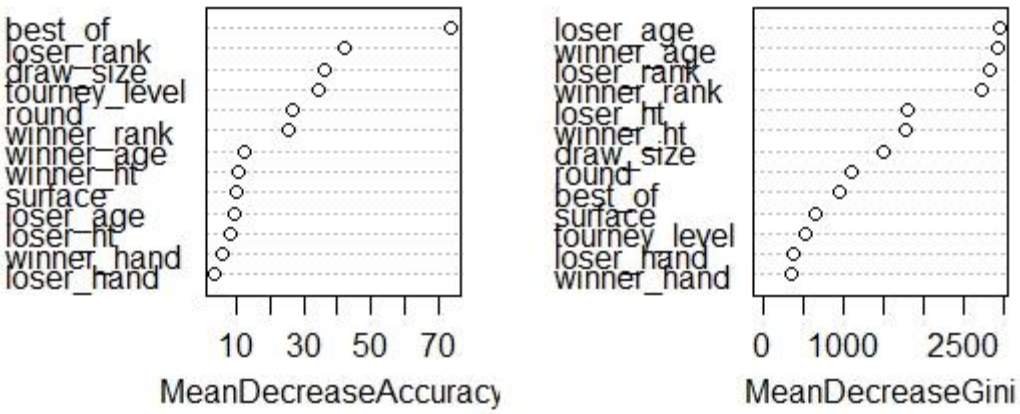
Prédiction de la durée d'un match : forêts aléatoires

Nous allons ici utiliser les forêts aléatoires pour essayer d'améliorer la précision de nos prédictions. Cet algorithme présente l'avantage de pouvoir traiter sans difficultés de très grandes bases de données et d'offrir une grande précision, comparativement à d'autres modèles. Le Random Forest trouve ici un intérêt

Pour les bases train et test nous avons ici diviser la base en 63,2% pour la base train et 36,8% pour la base test. Cette façon de diviser les bases train et test reste théoriquement la plus adaptée sur ce type de problématique, loin des 70-30 ou 80-20 que l'on observe dans le traitement d'images par exemple. En théorie, on montre que la probabilité de ne pas choisir l'échantillon est égale à (100-36,8)%. Ainsi, le rapport optimal montre une division train/test de l'ordre de 65-35 et plus précisément ici 63,2%-36,8%.

Nous pouvons désormais ajuster notre algorithme Random Forest. On choisit comme paramètres 1500 pour le nombre d'arbres de notre forêt et 2 pour le nombre de variables candidates à chaque coupure.

Figure 1 - Importance des variables

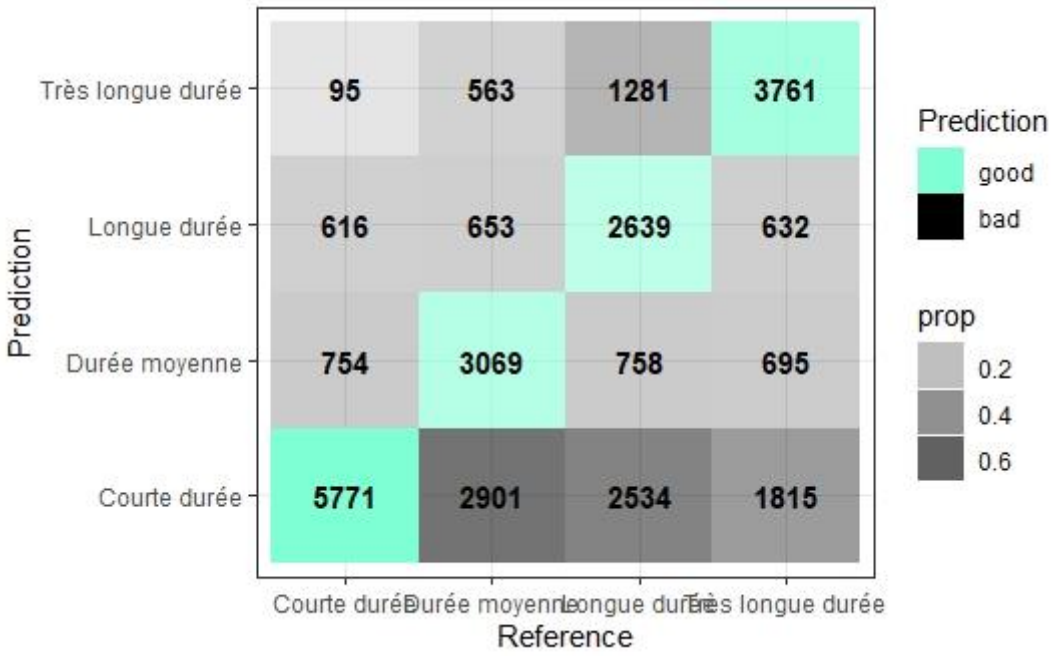


Source : graphique des auteurs, selon les données de l'ATP

On remarque ici que les variables qui offrent la plus grande précision sur la prédiction sont les variables best_of (ce qui est plutôt logique, un match en 3 sets gagnants dure en moyenne plus longtemps qu'un match en 2 sets gagnants), loser_rank (classement du perdant), draw_size nombre de tours dans le tournoi), tourney_level (niveau du tournoi) et dans une moindre mesure round (tour durant lequel se déroule le match) et winner_rank (classement du gagnant).

Lorsque l'on observe les diminutions de Gini, les variables les plus importantes sont loser/winner_age et loser/winner_rank.

Figure 2 - Matrice de confusion



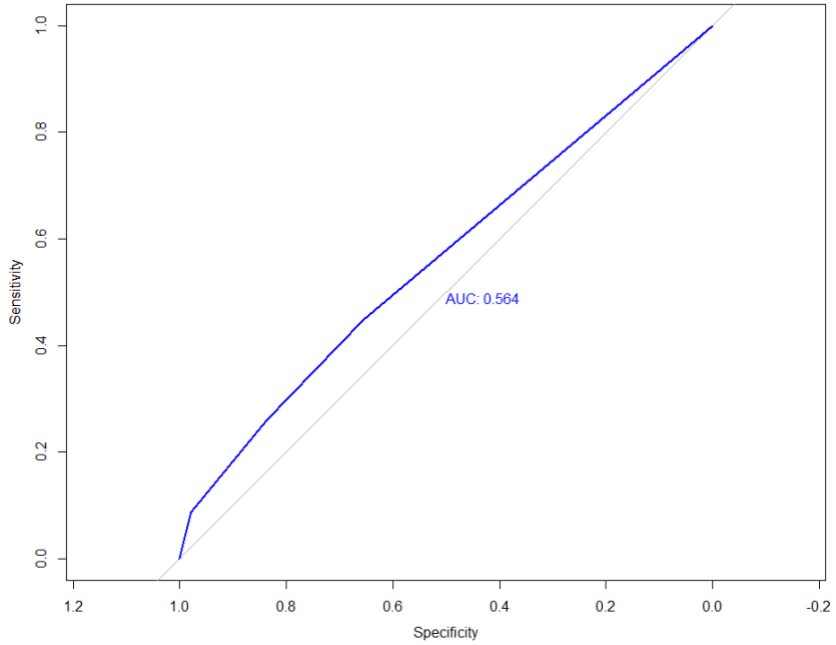
Source : graphique des auteurs, selon les données de l'ATP

Cette matrice de confusion nous montre que nous avons une bonne précision pour la prédiction de la durée des matchs de tennis répartis en 4 classes. Nous obtenons une précision de 53,4% dont nous pouvons justifier l'efficacité par l'observation de cette matrice. Effectivement, la prédiction pour chacun des matches semble efficace, toutes proportions gardées. Le nombre de prédictions justes est pour chaque catégorie la prédiction juste. Les prédictions pour les matches de très longue durée par exemple ont été correctement prédites dans 65% des cas quand pour la catégorie longue durée on ne dépasse pas 58%. Pour la catégorie durée moyenne cette-fois ci, nous sommes également à une précision de 58% pour la détection de la bonne catégorie et enfin nous restons à 44% environ pour la courte durée. Ces précisions relatives restent un aboutissement conséquent dans la mesure où la classe majoritairement prédite est toujours la bonne classe. Néanmoins, notre prédiction montre des limites sur les matches de courte durée principalement, ce qui a donc pour conséquence de réduire la précision globale à 53%.

Nous avions initialement une très bonne reconnaissance des catégories Très longue durée et courte durée (les extrêmes étaient bien reconnues) quand les longues et moyennes durées étaient extrêmement mal prédites. La polarisation de bonnes prédictions aux extrêmes rendait notre modèle quasiment caduc. En théorie, tout l'intérêt de construire des modalités catégorielles réside dans le fait de pouvoir les segmenter de manière claire. Or, notre algorithme confondait beaucoup trop moyenne et longue durées. Aussi, nous avons opté pour une refonte de notre Random Forest. Nous avons augmenté le nombre d'arbres, avons divisé nos bases en 65%-35% (puis 63,2%-36,8%) au lieu de 80-20 initialement. Le résultat est sans appel, la prédiction des classes moyenne durée/longue durée et très longue durée est bien meilleure. La problématique principale de notre projet résidait dans le fait de pouvoir prédire efficacement la durée d'un match. Ceci ayant été plus compliqué que prévu, nous obtenons ici une précision relativement bonne pour les 3 classes supérieures quand la prédiction de la classe courte durée montre une moins bonne précision. Il s'agit de la catégorie de de matches

apparemment la plus représentée (ce qui semble logique en raison du nombre de petits tournois se jouant en 2 sets gagnants face aux 5 GC seulement par an, se jouant en 3 sets gagnants).

Figure 3- Courbe ROC



Source : graphique des auteurs, selon les données de l'ATP

La courbe ROC est l'une des métriques les plus utilisés pour l'évaluation de la performance des modèles de classification. Il est possible de la tracer dans le cas multi-classe sous R grâce au package « pROC ». Par contre ce package trace la courbe de sensibilité en fonction de la spécificité (pas en fonction de 1-spécificité) c'est pour cela que l'origine de la courbe ne commence pas en (0,0) mais plutôt (1,0).Plus la surface sous la courbe de ROC est proche de 1 plus le modèle de classification est performant. Pour notre cas la valeur de l'AUC est de 0.56.

Conclusion

Après des résultats décevants obtenus par la méthode des k-plus proches voisins (35,8%) et de l'arbre de décision (35,3%), nous avons pu établir une prédiction satisfaisante grâce à nos forêts aléatoires (53,4%). Ce pourcentage de 53,4% ne doit d'ailleurs surtout pas éluder l'efficacité réelle de nos prédictions sur 3 des 4 classes de notre variable minutes_recod.

Ce modèle est cependant perfectible. Certaines variables pourraient être ajoutées pour apporter plus d'information ; l'état de forme, la santé mentale des joueurs concernés avant match ou encore des données climatiques sont par exemple des données qu'il pourrait être pertinent de prendre en compte.

Il est à noter que la puissance de calcul dont nous disposions a été un handicap et que des machines plus performantes pourraient en théorie faire tourner des modèles plus complexes plus facilement (avec un nombre d'arbres plus conséquent si cela peut être considéré comme une limite de précision de notre algorithme).

Bibliographie/Webographie et références

Cours d’apprentissage automatique dispensé par Messieurs Regnault et Blanchard.

https://towardsdatascience.com/machine-learning-basics-random-forest-regression-be3e1e3bb91a

https://towardsdatascience.com/what-is-out-of-bag-oob-score-in-random-forest-a7fa23d710