Université du Québec à Montréal

Prédictions des salaires des joueurs de la LNH selon 2 modèles

Pour le cours :

Analyse de données en actuariat - ACT6100

Présenté à:

M. Noureddine Meraihi

Par:

Laurence Beaudet (BEAL25589600)

Edgar Lanoue (LANE03059909)

Caleb Miller (MILC15089801)

Le 10 mai 2022

Introduction

Connor McDavid, capitaine des Oilers d'Edmonton, a été payé 16,4 M$ en 2021-2022. Pas étonnant que de nombreux jeunes joueurs rêvent de devenir des célébrités de hockey! Pourtant, les chances de toucher un tel salaire sont très faibles. Effectivement, si on calcule le salaire moyen gagné parmi tous les joueurs de hockey de ce monde et tous niveaux confondus, un joueur de hockey gagnerait environ 4 $ par année.

Nous avons alors trouvé intéressant de créer un modèle capable de prédire le salaire d'un joueur en se basant sur certaines caractéristiques. Afin de créer notre algorithme, nous avons recueillis les données de 874 joueurs professionnels, en excluant les gardiens de buts. Les données recueillies sont composées de variables quantitatives tel que le salaire et le poids d'un joueur et qualificatives tel que son côté dominant. Aléatoirement, nous avons séparé cette base de données en 2 groupes. Le premier groupe nous a servis à construire les modèles et à l'entraîner (environ 700 joueurs). Le deuxième nous a permis de les tester.

Les modèles construits seront de types supervisés. Notre variable réponse sera le salaire, qui sera prédit par plusieurs données entrantes et connues.

Les données

Dans cette section, nous traiterons des données qui seront utilisées pour prédire les salaires des joueurs de la LNH. Les données brutes se trouvent sur le site suivant :
http://www.hockeyabstract.com/testimonials/nhl2016-17playerdata

Le csv que nous avons conservé ce trouve dans la feuille Excel « All Sits », à laquelle nous avons tranché la première colonne. Cette colonne ne contient que la source initiale des données recueillies. Suite à cette transformation, nous finissons avec une base de données de 888 lignes, joueurs, par 167 colonnes, caractéristiques. Il y a donc énormément de variables explicatives et beaucoup de données à nettoyer.

Voici le nettoyage qui a été effectué :

- La variable explicative *Salary* d'abord été convertie en caractères numériques, puis divisée par 1 000 000 afin d'être convertie en échelle normale. De plus, nous avons supprimé toutes les lignes d'observations contenant un salaire inconnu ;

```
str(data$Salary)
```

```
##  chr [1:888] "$575,000.00 " "$5,500,000.00 " "$842,500.00 " "$892,500.00 " ...
```

```
data$Salary <- gsub("\\$","", as.character(data$Salary))
data$Salary <- gsub(",","", as.character(data$Salary))

data$Salary <- as.numeric(data$Salary)
data$Salary <- data$Salary/1000000
```

#Creation de la variable age

```
dateMoitieSaison2016 <- as.Date("01/01/2017", "%m/%d/%Y")
Born <- as.Date(data$Born, "%m/%d/%Y")

age <- floor(difftime(dateMoitieSaison2016, Born, units= "weeks")/52.25)
data <- cbind(data, age)
```

- Nous avons ensuite créée la variable *Âge* à partir de la date de naissance :

```
dateMoitieSaison2016 <- as.Date("01/01/2017", "%m/%d/%Y")
Born <- as.Date(data$Born, "%m/%d/%Y")

age <- floor(difftime(dateMoitieSaison2016, Born, units= "weeks")/52.25)
data <- cbind(data, age)
```

- La variable *Position* a été simplifiée pour contenir seulement « ATT » pour attaquants, « D » pour défenseurs et « G » pour gardien. Originalement, la source de données nous fournissait l'information sur le côté dominant des joueurs, soit allier droit/gauche, centre, …. Ces informations, biens qu'importantes dans un autre contexte, sont souvent interchangeables et nous jugeons que la différence n'est pas assez significative pour la prendre en compte.

```
data$Position[which(data$Position != "D" & data$Position != "G"  )] <- "ATT"
```

- La variable *DftRd* (qui représente la ronde de repêchage) a été modifiée afin de traiter les données manquantes. Les NA présents ont été imputés à 8. Il s'agit de joueurs qui n'ont jamais été repêcher. La ronde maximale de repêchage a alors été fixée à +1, donc à 8 pour ces joueurs. Et, pour une raison similaire, nous avons imputé les NA de la variable *Ovrl* (représentant le rang de repêchage) à un rang de repêchage maximale de +1, donc de 293.

```
data$DftRd[is.na(data$DftRd)] <- 8

data$Ovrl[is.na(data$Ovrl)] <- max(data$Ovrl, na.rm = TRUE)+1
```

- Également, les variables contenant un % ont dues être formattée puisqu'elles étaient en caractères. Ces variables sont les suivantes :
  - **TOI%**
  - **IPP%**
  - **SH%**
  - **Pct%**
  - **FO%**
  - **X%FOT**
  - **BLK%**

```
data$TOI. <- as.numeric(gsub("%","", as.character(data$TOI.)))/100
```

```
## Warning: NAs introduits lors de la conversion automatique
```

```
data$IPP. <- as.numeric(gsub("%","", as.character(data$IPP.)))/100
```

```
## Warning: NAs introduits lors de la conversion automatique
```

```
data$SH. <- as.numeric(gsub("%","", as.character(data$SH.)))/100
```

```
## Warning: NAs introduits lors de la conversion automatique
```

```
data$Pct. <- as.numeric(gsub("%","", as.character(data$Pct.)))/100
data$FO. <- as.numeric(gsub("%","", as.character(data$FO.)))/100
data$X.FOT <- as.numeric(gsub("%","", as.character(data$X.FOT)))/100
```

```
## Warning: NAs introduits lors de la conversion automatique
```

```
data$BLK. <- as.numeric(gsub("%","", as.character(data$BLK.)))/100
```

```
## Warning: NAs introduits lors de la conversion automatique
```

- Nous avons ensuite converti les données numériques en caractères numériques puisque certaines étaient des entiers et les variables *Hand* et *Position* ont été converties en facteurs.

```
Hand <- data$Hand
Position <- data$Position

data <- sapply(data, as.numeric)
```

- Finalement, nous avons retiré de notre base de données les variables suivantes :
  - ***Born***
  - ***DftYr***
    - L'information de la date de naissance (*Born)* est contenue approximativement dans la nouvelle variable *Âge.* La variable *DftYr* ne donnait pas d'information supplémentaire utile à nos modèles.

  - ***City***
  - ***Pr.St***
  - ***Cntry***
  - ***Nat***
    - Les quatre variables précédentes décrivent l'origine du joueur. Nous avons posé l'hypothèse que cette information n'était pas pertinente pour déterminer le salaire d'un joueur. Nous savons que les probabilités de faire partie de la LNH sont fortement reliées aux origines d'un joueur. Cependant, nous considérons que le salaire du joueur sera davantage influencé par ses performances une fois admis dans la LNH.

  - ***Cap.Hit***
  - ***CHIP***
    - Ces variables s'agissent de technicalité. Ce sont des variables énormément corrélées à celle que nous tentons de prédire, soit le salaire, donc nous les retirons.

  - ***Status***
    - Nous avons décidé de négliger celle-ci puisqu'elle représente le statut des joueurs à l'approche de la période des signatures d'agent libre. Nous ne voyons pas de lien direct à la variable à prédire.

  - ***NMC***
    - Cette variable représente une clause dans les contrats des joueurs. Elle pourrait possiblement avoir un effet sur la valeur monétaire d'un contrat. Cependant, nous cherchons à évaluer si le salaire d'un joueur peut être prédit par des variables portant sur ses performances.

- ***Injuries***
- ***MGL***
    - Ces variables traitent de blessures encourues par un joueur. Ce qui n'influence pas le salaire courant d'un joueur. Elles pourraient toutefois avoir un impact sur les futurs contrats.

- ***X3rd***
- ***X2nd***
- ***X1st***
    - Les trois variables précédentes sont respectivement le compte de 3^{e}, 2^{e} et 1^{ière} étoile d'un joueur. Ces variables sont subjectives. Dans l'optique de conserver uniquement les caractéristiques qui reposent sur les performances, nous avons décidé de les retirer.

- ***First.Name***
- ***Last.Name***
    - Les noms des joueurs sont des caractéristiques non pertinentes.

- ***NHLid***
    - Nous avons conservé cette colonne dans le but d'identifier les joueurs, bien qu'il est évident que celle-ci n'est pas pertinente pour déterminer le salaire des joueurs.

- ***Team***
    - Sachant que la LNH fonctionne avec un plafond salarial égal pour chaque équipe, l'équipe ne devrait pas ou peu influencer le salaire d'un joueur.

- ***SOS***
- ***SOG***
- ***SOGDG***
    - Imputation des NAs des ces variables à 0 puisque ces variables représentent des occurrences, et donc en absence d'occurrence, on remplace par 0.

Avec les modifications précédentes, nous avons passé de 888 lignes par 167 colonnes à 874 lignes par 149 colonnes.

Plusieurs données peuvent paraître farfelues, mais il s'agit de jargon de hockey. Si jamais vous voulez consulter la légende, vous n'avez qu'à vous référer à la base de données initiale qui la contient.

Plusieurs données semblent également se répéter. Elles viennent cependant de sources différentes et peuvent parfois être étonnamment distinctes. Il serait intéressant d'éventuellement évaluer si une de ces sources recueillent de meilleures données pour prédire le salaire d'un joueur.

Il est finalement bien important de se rappeler que nous tentons d'évaluer le salaire d'un joueur à l'aide des statistiques de l'année courante, mais que souvent les contrats sont signés une ou plusieurs années auparavant et eux-mêmes ne dépendent pas de la performance de l'année courante.

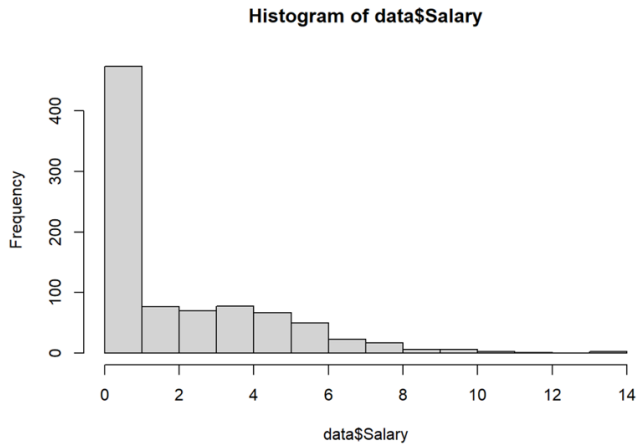
Statistique descriptive

Étudions rapidement la variable salaire.

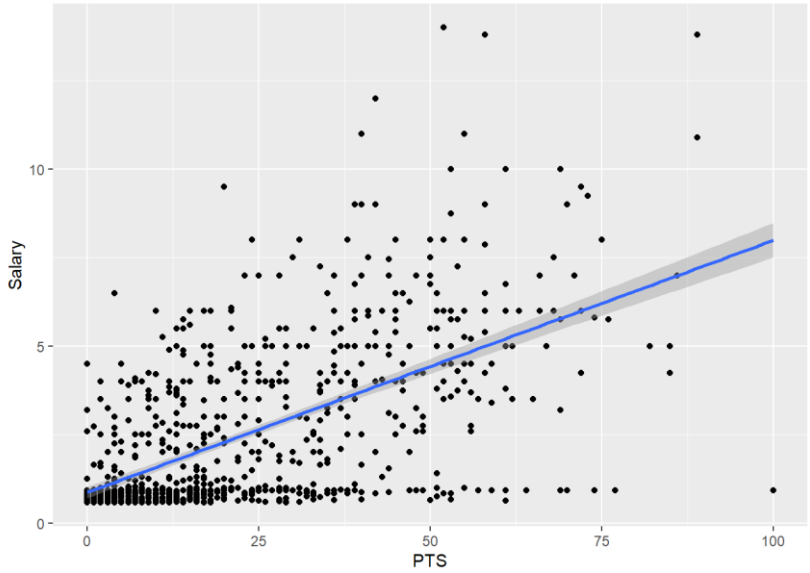
La moyenne salariale est 2,32 millions.

Le salaire médian en revanche est de 0,925.

On s'attend donc à une distribution centrée à gauche (Figure 1). En effet, bien que la médiane nous donne déjà une bonne indication, on voit bien que majorité des joueurs font moins de 1M $.



Maintenant, si l'on regarde le salaire en fonction des points, ce qui correspond à la statistique la plus importante pour les attaquants au hockey. On remarque effectivement une tendance vers le haut avec quelques valeurs aberrantes. Ces valeurs aberrantes peuvent être expliquées par plusieurs raisons, mais les principales sont probablement de bons défenseurs moins performants à l'attaque, mais gagnant également beaucoup d'argent, ou bien des joueurs vieillissants qui ont de bons contrats mais qui sont sur la pente descendante de leurs habilités.



De plus, nous avons créé un graphique qui met en relation le salaire des joueurs en fonction de l'âge. Il est évident que celui-ci démontre une pente ascendante de cette relation, mais les valeurs aberrantes proviennent des

joueurs dans la fleur de l'âge, ce qui est logique puisque les meilleurs joueurs sont au summum de leur maturité et de leur forme[1].

Essayons maintenant de construire des modèles prenant en compte la totalité de nos variables explicatives. Pour ce faire, nous avons créé deux sous-bases de données, l'une d'entraînement et l'autre de test. Celle d'entraînement contient 80 % des données et celle de test, le 20 % restant. Le fractionnement est fait de manière aléatoire. Pour les besoins de ce projet, nous avons défini un « seed » particulier pour être en mesure de comparer les deux modèles décris plus bas.

Les modèles

Modèle 1

Le modèle de Random Forest a été choisi puisque ce dernier, bien que peu élégant, est très efficace. De plus, ce modèle fait la sélection de variables automatiquement, ce qui peut s'avérer utile lorsque nous avons 148 variables explicatives. Le peu d'hyperparamètres à régler est aussi intéressant. Finalement, à la base, nous aimions le fait de ne pas avoir à faire de sous-bases de données de test et d'entraînement puisque nous n'avons que 874 données. En effet, chaque arbre utilisé dans la forêt crée ses propres ensembles de test et d'entraînement. Cependant, pour les besoins de ce travail, nous avons tout de même utilisé l'ensemble d'entraînement mentionné plus haut pour entraîner la forêt aléatoire.

Étant donné que le modèle de Random Forest ne peut admettre de données NA, la première étape est de faire un nettoyage final et d'imputer les NA. Cette imputation sera faite selon la technique de Breiman et Cutler.

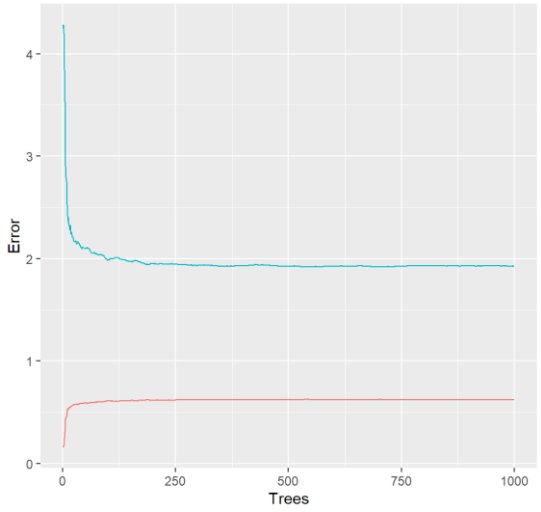
```
dataTestImpute <- rfImpute(Salary ~ ., data = dataTest, iter=6)
```

```
dataImpute <- rfImpute(Salary ~ ., data = dataTrain, iter=6)
```

Cette technique utilise une Random Forest pour imputer les NA. On remarque que le MSE oscille sans vraiment augmenter ou diminuer, il est donc inutile de faire plus de 6 itérations.

[1] https://www.linternaute.fr/expression/langue-francaise/14170/fleur-de-l-age/#:~:text=%C3%8Atre%20dans%20la%20fleur%20de,d%C3%A9clin%20li%C3%A9%20%C3%A0%20la%20vieillesse.

Il est maintenant possible de faire le modèle à l'aide de la fonction randomForest du paquetage R portant le même nom. Nous avons utilisé 1 000 arbres pour créer cette forêt. Nous pouvons maintenant observer s'il s'agit d'un nombre suffisant d'arbre à l'aide de ce graphique qui montre l'erreur quadratique moyenne et le pourcentage de variance expliquée.



On remarque que les deux semblent se stabiliser assez tôt, nous aurions tendance à dire autour de 150. Pour être certain de cette stabilisation, nous prendrons un nombre de 300 arbres et puisque le modèle est assez rapide à entraîner, nous ne perdons pas grand-chose à en mettre un peu plus qu'il n'en faut.

Pour ce qui est maintenant du nombre de caractéristiques aléatoires à considérer à chaque nœud, la fonction de base prend p/3, donc $148/3$ = 48. Regardons s'il s'agit du p qui minimise l'erreur quadratique moyenne en cherchant les p dans un rayon de 15 autour de 48, donc entre 33 et 63.

```
mse <- vector(length = 31)
for(i in 33:63) {
  temp.model <- randomForest(Salary ~ ., data=dataImpute, mtry=i, ntree=300)
  mse[i-32] <- temp.model$mse[length(temp.model$mse)]
  print(i)
}
mseValues <- data.frame(c(33:63), mse)
min(mseValues)

for(i in 41:71) {
  temp.model <- randomForest(Salary ~ ., data=dataImpute, mtry=i, ntree=300)
  mse[i-40] <- temp.model$mse[length(temp.model$mse)]
  print(i)
}

mseValues <- data.frame(c(41:71), mse)
min(mseValues)
```

On trouve que p = 56 et on remarque qu'entre 33 et 63 il semble y avoir une baisse, surtout au début. Réévaluons donc autour de 56. Cette fois-ci, on voit qu'on se stabilise autour d'un EQM de 2. Gardons 56 qui semble être une bonne estimation du nombre de caractéristiques aléatoires à considérer.

On se retrouve donc avec les hyperparamètres suivants : mtry = 56 et ntree = 300. Évaluons donc notre modèle sur l'ensemble test. On obtient finalement un EQM totale de 2,30625.

```
salary.rf <- randomForest(Salary ~ ., data = dataImpute, ntree = 300, mtry = 56)
```

Modèle 2

Puisque nous cherchons à expliquer le salaire des joueurs selon de nombreux facteurs, le choix du deuxième modèle nous paraissait simple puisque c'est exactement l'utilité d'une régression linéaire. Effectivement, une régression linéaire est un modèle qui cherche à établir des relations linéaires entre la variable que nous cherchons à expliquer et des variables explicatives. Ici la variable à expliquer est le salaire et nous avons plus d'une centaine variables explicatives potentielles. Nous savons que plus nous avons de variables explicatives potentielles, plus le modèle risque d'être précis.

```
salaryPred <- predict(salary.rf, dataTestImpute)
error <- dataTest$Salary - salaryPred
quadError <- error^2

#quadErrorId
mse <- mean(error^2)
```

```
reg1 = lm(dataTrain$Salary~ . , data=dataTrain)
summary(reg1)
```

Le modèle donne un r carré assez satisfaisant soit au-dessus de 0,65. Évidement dans le modèle nous avons énormément de variables explicatives. Si nous avions voulu faire un modèle plus approfondi nous aurions pu faire une série de régression linéaire simple avec chaque variable explicative et exclure les variables dont la p value était au-dessus d'un certain seuil (souvent 5%).

Dans notre base de données nous avons probablement énormément de données fortement corrélées.

Nous aurions pu éliminer un maximum de variable selon la corrélation de sorte de ne pas conserver deux variables très corrélées.

En somme le modèle n'est pas mauvais, mais on remarque que certaines valeurs prédites avec le modèle sont négatives. Cela est impossible puisqu'un salaire ne peut être négatif.

Comparaison des modèles

Étant donné qu'un de nos deux modèles, celui de la forêt fait beaucoup moins appel aux mathématiques et n'a pas de distribution, nous allons comparer les modèles à l'aide des résultats suivants (par rapport au salaire dans l'ensemble test):

- La moyenne de l'écart absolu entre le salaire prédit le salaire connu
- L'erreur quadratique moyenne entre le salaire prédit le salaire connu

Moyenne Ecart RF	**0.993**
Moyenne Ecart LM	**1.147**
mseRF	**2.266**
mseLM	**2.591**

On remarque donc que le modèle forêt aléatoire prédit mieux le salaire que le modèle de régression linéaire multiple, avec les deux estimateurs utilisés. Il y aurait très certainement moyen de « tuner » les deux modèles pour obtenir de meilleurs résultats, mais ceci nécessiterait aussi probablement une meilleure sélection initiale des variables explicatives.

Conclusion

En conclusion, il serait intéressant de voir à quel point les modèles peuvent s'ajustés dans le temps. Sachant qu'environ 6 ans se sont écoulés entre les données recueillies et aujourd'hui, nous pourrions étudier à quel point les modèles peuvent prédire les salaires des joueurs aujourd'hui.

Dans ce travail, nous avons été confrontés à plusieurs embûches. Tout d'abord, le nettoyage de données a été bien plus considérable que ce que nous avions anticipé. Il nous a en effet fallu regarder presque chaque variable explicative pour voir ce qu'il était préférable de faire. En même temps, il s'agit d'une base de données avec une grande proportion de variables explicatives, donc nous aurions pu nous y attendre. Nous avons aussi eu plus de difficulté de prévu avec le modèle Random Forest, avant de découvrir la fonction rfImpute. Nous sommes bien heureux d'avoir pu mettre en pratique les concepts vus au courant de la session, nous comprenions la base, mais, en soi, le mettre en exécution a été bien plus ardu que prévu.

Nous sommes heureux que notre modèle « plus avancé », soit la forêt aléatoire ait été le modèle qui prédisait le mieux le salaire des joueurs. En effet, nous aurions été un peu déçus du contraire.

À bien y penser, et surtout, en nous renseignant sur les modèles, on aurait finalement trouvé plus intéressant d'essayer la régression Elastic Net que la régression linéaire multiple. En effet, cette régression aurait fait une sélection de variables automatiquement, ce qui nous aurait été bien utile. En plus, il est supposément rapide à entraîner. Cette caractéristique ne nous aurait pas nécessairement beaucoup aidé ici, mais elle sera certainement utile dans notre parcours d'analyse de données.

Bibliographie

Combien gagne réellement un joueur de hockey professionnel? *Le journal de Montréal*, le 2 octobre 2020 (page consultée le 7 mai 2022), [En ligne], adresse URL : https://www.journaldemo ntreal.com/2020/10/02/combien-gagne-reellement-un-joueur-de-hockey-professionnel

NHL 2016-17 Player Data, (page consulté le 6 mai 2022), [En ligne], adresse URL : http://www.hockeyabstract.com/testimonials/nhl201617playerdata?fbclid=IwAR2Tbay92iX14CVyjhil5dpZASmmjYyBOQ1gE8rVMz4oBxZzdGq4T4HZrLs

MERAIHI, NOUREDDINE, *Notes de cours ACT6100H22*, (page consulté le 6 mai 2022), [En ligne], adresse URL : https://nmeraihi.github.io/act6100book/sommaire.html

Pas une année faste pour les joueurs les mieux payés de la LNH. *Radio-Canada*, le 27 octobre 2021 (page consultée le 6 mai 2022), [En ligne], adresse URL : https://ici.radio-canada.ca/sports/1834906/connor-mcdavid-joueur-salaire-lnh-forbes-carey-price-matthews-cro sby-karlsson-marner