CNAM_STA211-synthese_rapport_fabrice_dunan_PROBTP

February 13, 2019

1 Projet CNAM-STA211 - Analyse de 10 ans de résultats de matchs de tennis professionnel

Dépôt Git du projet :

- https://gitlab.com/logrus_fr/CNAM-projets/

Auteur :

- Fabrice DUNAN fabrice.dunan@laposte.net

Tuteurs CNAM :

- Ndeye NIANG KEITA ndeye.niang_keita@cnam.fr
- Emmanuel JAKOBOWICZ ej@stat4decision.com

1.1 SOMMAIRE


Total Points Won
Serve Speed
132mph
Matches Played
First serve aces
32%
Aces
Unforced Errors
Breakpoint conversion
67%
Receiving Points Won
Winning % on 1st Serve
Total Games

Logo

2. Section **??**

4. Section **??**
5. Section **??**

    1. Section **??**
    2. Section **??**
    3. [Construction d'une variable auxiliaire : le "MOIS" du tournoi](#Construction d'une variable auxiliaire : le "MOIS" du tournoi)
    4. Section **??**
    5. Section **??**

        1. Section **??**

    6. Section **??**

6. Section **??**

    1. Section **??**
    2. Section **??**
    3. Section **??**

        1. Section **??**
        2. Section **??**
        3. Section **??**
        4. Section **??**

    4. Section **??**

        1. Section **??**
        2. Section **??**
        3. Section **??**

    5. Section **??**
    6. Section **??**

        1. Section **??**
        2. Section **??**
        3. Section **??**
        4. Section **??**
        5. Section **??**

    7. Section **??**

    8. Section **??**

        6. Section **??**
        7. Section **??**

    9. Section **??**
    10. Section **??**

7. Section **??**

    1. Section **??**

        1. Section **??**
        2. Section **??**

    2. Section **??**

1. Section **??**
2. Section **??**

3. Section **??**
4. Section **??**
5. Section **??**
6. Section **??**

8. Section 1.9

1. Section **??**

1. Section 1.9.1
2. Section 1.9.1
3. Section 1.9.1

1.2 INTRODUCTION

Ce document est le rapport d'étude de cas **STA211**, Entreposage et fouille de données, visant à l'obtention du certificat de données massives du **CNAM**.

Ce rapport présentera la synthèse de l'étude de données regroupant les résultats de matchs de tournois de tennis professionnel. Cette synthèse sera illustrée par des extraits de deux "notebooks".

Ils sont disponibles ici, sous réserve d'habilitation par l'auteur : * https://gitlab.com/logrus_fr/CNAM-projets/blob/master/STA211/atp-non_supervise.ipynb * https://gitlab.com/logrus_fr/CNAM-projets/blob/master/STA211/atp-supervise.ipynb

Ceux-ci peuvent fournir plus de détails pour le lecteur souhaitant approfondir ou avoir un complément d'informations. Si la synthèse doit permettre de valider les résultats et les démarches, les détails peuvent éclairer la vérification des savoir-faire et l'estimation du travail accompli.

Au travers des données pré-citées, on tentera de résoudre deux problématiques tennistiques. Chaque problématique mettra en oeuvre une des deux grandes familles de méthodes et d'outils étudiés en cours.

1.3 ANALYSE METIER

Les tournois, dont découlent les résultats de matchs étudiés, sont organisés par l'association de tennis professionnel (ATP). Cette "association" n'organise que des matchs internationaux masculins. Le périmètre couvert par les données est circonscrit aux résultats des matchs en simple.

Ces résultats sont tirés de données librement mises à disposition sur Internet. Elles étaient inconnues avant l'analyse. Par conséquent, on se place d'ores et déjà dans un cas réel d'analyse de données. On suppose que les motivations de la personne ayant mis à disposition les données sont d'optimiser des paris. Cette hypothèse découle de la nature des variables, variables décrites un peu plus tard dans le document.

1.3.1 Les objectifs métier

Le but de l'étude est d'extraire de l'information de données extraites de résultats des matchs de tennis professionnel pour répondre aux deux problématiques suivantes :

Structure de la saison de tennis professionnel ATP : Découverte de groupes de tournois On souhaite, à partir des données fournies, dégager des groupes de tournois pour classer les tournois au cours de l'année de tennis. Cela permettra de tirer des conclusions à la fois sur les caractéristiques des tournois eux-mêmes mais aussi sur la stratégie de l'ATP quant à la structure de la saison de tennis.

En effet, les amateurs de tennis professionnel qui assistent au tournoi local occasionnellement, par exemple feu l'Open de Nice, ne savent pas forcément qu'il est organisé dans l'optique de la préparation à Roland Garros dans une période du planning ATP proche, sur des surfaces extérieures similaires. Donc que ce tournoi ferait partie d'un groupe de tournois similaires par certains critères inventoriés dans l'étude. Au dela de l'amateur, toute personne s'interessant à la structure de la saison professionnelle, quelquesoit sa motivation, devra trouver des informations sur des regroupements de tournois dans les conclusions de l'étude. L'étude tentera donc de donner un résultat probant sur cette classification.

Prédiction du gain du match "en perf" L'objectif est de pouvoir prédire, à partir des données tirées des matchs passés, si un match, impliquant deux joueurs donnés, lors d'un tournoi donné, sur une surface donnée, etc... est gagné par le plus mal classé des deux joueurs.

L'étude tentera de donner un outil de prédiction quelquesoit la motivation : ludique, pédagogique...ou autre.

Critères de succès des deux objectifs

- Obtenir une bonne compréhension de la structure de la saison de tennis professionnel et des catégories de tournois.
- Créer un modèle qui est capable de prédire le résultat d'un match entre deux adversaires. On souhaite que cette prédiction soit meilleure qu'une prédiction aléatoire.
- Dans les deux cas suivre une démarche valide, que ce soit dans les différentes étapes de l'étude et de leur rapport, mais aussi dans la statistique et l'application des algorithmes de fouille de données.

1.3.2 La situation actuelle

Les buts initiaux de ce projet n'ont d'objectifs que pédagogiques. Le fait que le sujet soit le tennis n'est dû qu'aux goûts de l'auteur et au fait que le sport professionnel en général génère un grand nombre de données statistiques. Les amateurs de paris en ligne pourraient néanmoins être vivement intéressés par la deuxième partie de l'étude...

Inventaire des ressources Cette étude a été réalisée sous la tutelle des enseignants du CNAM. Les réponses à mes questions ont permis d'éviter des écueils méthodologiques et ont donné des pistes pour enrichir l'étude ou tenter de la sortir d'impasses. Le datascientist de PROBTP, E.Kouadio, a également instilé des conseils sur à ce travail.

Outre les enseignements...et Internet, un certain nombre de références bibliographique ont contribué à cette étude. Elles sont inventoriées en annexe.

L'étude a été réalisée avec un laptop lenovo T430s fonctionnant avec la distribution linux Ubuntu et les outils standard du DataScientist (python, anaconda, jupyter...).

Prérequis, hypothèses et contraintes Ce projet initialement prévu en binôme a dû évoluer sur son sujet : le sujet précédent mettait en oeuvre des algorithmes hors périmètre de l'enseignement reçu. Même si le projet a accusé un retard conséquent au changement de sujet, ce document a pu voir le jour grâce aux échanges avec les enseignants, notamment sur les algorithmes et les méthodes alternatives à mettre en place.

L'apprentissage de python et de ses bibliothèques (Pandas, numpy, scikit-learn…) a pris une part importante de l'étude. Les rudiments divulgués dans le cours ont nécessité une recherche complémentaire.

De plus, sans expérience, ni pratique quotidienne dans le travail des données, il a fallu choisir des données sans réelle expérience. On verra par la suite si cela, ainsi que les échéances du rendu des projets, auront prêté à conséquence. L'intérêt est de se mettre en situation d'un Data Scientist qui, même s'il connaît partiellement le sujet, reçoit des données qu'il ne produit pas. L'étude part donc de données initialement inconnues, si ce n'est leur thème, librement disponibles sur internet.

La taille du rapport projet est fixée à *~ 20 pages*.

Terminologie et "concepts" du tennis En cas de méconnaissance du tennis, on renvoie : * au site http://www.fft.fr pour tout ce qui a trait au jeu de tennis. * au site https://fr.wikipedia.org/wiki/ATP_World_Tour pour tout ce qui a trait à l'association du tennis professionnel et à ses tournois. Ce dernier site permet de vérifier un certain nombre d'informations dans cette étude; notamment le nombre de tournois "master 1000", si à l'époque contemporaine il existe encore des tournois sur moquette….etc… * à un exemple de saison ATP : https://fr.wikipedia.org/wiki/Saison_2017_de_l%27ATP

Coût et profits Le coût du projet STA211 est évalué à **15 jours soit 120 heures de travail**.

De ce travail découle l'assimilation de l'usage des outils, langages, bibliothèques, méthodes, théories et algorithmes en Datascience.

1.3.3 Les objectifs en fouille de données

La méthode de découverte de groupes de tournois Pour définir ces classes, on utilisera des algorithmes de fouille de données non supervisés adaptés à la problématique de classification. En effet, cette famille permet l'étude de données inconnues, sans cible quant aux caractéristiques (les colonnes du tableau) particulières.

- Les prérequis

Il est nécessaire de s'appuyer sur des caractéristiques des tournois stables. En effet, l'ATP fait évoluer pour diverses raisons les catégories de tournois. Cela fera peser une contrainte sur les individus sélectionnés pour la création des groupes. On construira une variable qui situera le mois du tournoi dans l'année. On excluera les informations relatives aux joueurs et aux matchs puisqu'on ne s'occupe que des tournois.

- Le choix de l'algorithme

Un candidat naturel est la classification hiérarchique ascendante (CAH), qui en plus de répondre à notre problématique de "clustering" c'est à dire de création de classe, permet une visualisation du résultat. Pour affiner les résultats, on utilisera une analyse des correspondances multiples avant l'application de la CAH. On comparera avec les résultats d'un k-means sur les mmes données.

- Critères de succés de la fouille

En plus de corroborer avec un autre algorithme et les méthodes classiques de validation, on jugera du succés de la classification par la capacité à donner un sens aux groupes découverts.

La méthode de prévision de la victoire du "moins bien classé" Pour définir et prévoir le résultat, on évaluera plusieurs algorithmes de fouille de données supervisés adaptés à la classification binaire. On pourra a priori construire l'étiquette "match gagné face à un mieux classé" lors de la phase préparatoire.

- Les prérequis

Les données d'apprentissage deviennent de moins en moins représentatives au fil du temps. On sélectionnera donc des données "fraîches" pour éviter de mettre à jour le modèle. Certaines variables seront exclues car liées à ce que l'on veut prédire : Le nombre de sets gagnés par le gagnant, les différents scores...On reconduira le mois en écartant les informations spécifiques aux tournois : la localisation, le nom du tournoi....

- Le choix de l'algorithme

On comparera la performance de plusieurs algorithmes (Forêt aléatoire, Support Vector Machine, K plus proches voisins...) et on justifiera des algorithmes écartés (régressions plus adaptées à un contexte de variables qualitatives, arbre de décision incapable en python de gérer les variables qualitatives sans transformation...)

- Critères de succès de la fouille

Il faudra obtenir un modèle ayant une efficacité prédictive supérieure au choix au hasard.

1.4 ANALYSE DE LA DONNEE

1.4.1 La source de la donnée

Les données en format CSV sont librement disponibles à l'URL suivante : https://data.opendatasoft.com/explore/dataset/atp%40public/analyze/?flg=fr

L'hébergeur est opendatasoft : une entreprise de valorisation et de divulgation de données "ouvertes".

Les données compilent les informations d'un certain nombre de sites, officiels ou non, dont le sujet est le sport.

- ATPtennis.com - http://www.atptennis.com/
- ATP Tour Rankings and Results Page - http://www.stevegtennis.com/
- Xscores - http://www.xscores.com/
- Livescore - http://www.livescore.net/
- odds - http://oddsportal.com

1.4.2 Décrire la donnée : Les variables et leur explication

Même si c'est anticipé par rapport aux résultats de l'étude, pour une meilleure compréhension du lecteur, on donnera des exemples de valeurs des noms de colonnes du tableau à chaque fois que possible. Ces modalités seront traduites en francais.

- **ATP** : Identifiant d'ordre du tournoi dans la saison ATP (Ce n'est pas une pré supposée clé unique ce dont on s'est apercu plus tard dans l'étude... variant de 1 à 69 donc supérieur aux 47 semaines de la saison ATP. On en deduit déjà qu'il y a superposition des plannings des tournois aux dates de début et de fin proche)
- **Location** : Ville où le tournoi se déroule.
- **Tournament** : nom (unique) du tournoi "commercial" du tournoi.
- **Date** : Date au format YYYY-MM-DD.
- **Series** : Type du tournoi au sens ATP (Grand chelem, Master 1000...).
- **Court** : Si les courts sont en extérieur ou en intérieur.
- **Surface** : Type de surface (Terre battue, dur, gazon et même moquette jusqu'en 2009 !).
- **Round** : Numéro du tour dans le tableau du tournoi. Se reporter au site www.fft.fr pré cité pour savoir ce qu'est un tableau dans un tournoi de tennis et les finesses scientifiques de leur élaboration. (qualifications, 1er , 2eme, 3eme, 4eme tours, quart, demi et finale)
- **Best of** : Matchs au meilleur du nombre de manches indiquées (3 ou 5 manches)
- **Winner** : Nom et initiale du prénom du gagnant.
- **Loser** : Nom et initiale du prénom du perdant.
- **WRank** : Classement du gagnant au moment du match. Le classement ATP comporte plus de 7000 joueurs.
- **LRank** : Classement du perdant au moment du match.
- **WPts** : Nombre de points octroyés au gagnant.
- **LPts** : points octroyés au perdant.
- **W1 L1 W2 L2 W3 L3 W4 L4 W5 L5** : Scores des différents sets, W(n) pour nième set gagné, L(n) pour nième set perdu. Se reporter au site www.fft.fr pré cité pour savoir ce qu'est un score de tennis.
- **Wsets** : Du point de vue du gagnant nombre de sets gagnés.
- **Lsets** : Du point de vue du perdant nombre de sets gagnés.
- **Comment** : Informations supplémentaires sur le déroulement du match (Fini, gagné par abandon, par forfait, disqualifié, ...).
- **MaxW** : Cote max sur match gagné par le gagnant final.

- **MaxL** : Cote max match gagné par le perdant final.
- **AvgW** : Cote moyenne match gagné par le gagnant final.

- **AvgL** : Cote max match gagné par le perdant final.

1.5 EXPLORATION DE LA DONNEE

Cette section permet de donner une idée du contenu du fichier. Le tableau ci-dessous donne un apercu des premières lignes. On remarque que certaines valeurs sont absentes, ce dont il faudra s'occuper par la suite.

```
In [194]: tour_DS.head()
```

```
Out[194]:   ATP Location                          Tournament        Date               Series
          0  25  Houston  U.S. Men's Clay Court Championships  2005-04-21        International
          1  26  Estoril                        Estoril Open  2005-04-27  International Series
          2  28     Rome          Telecom Italia Masters Roma  2005-05-03              Masters
          3  28     Rome          Telecom Italia Masters Roma  2005-05-04              Masters
          4  29  Hamburg                         Hamburg TMS  2005-05-11              Masters
```

Ci dessous un résumé numérique des différentes variables du jeu de données.

Le jeu de données brut initial comprend plus de 40 000 résultats de matchs de tournois et chaque match est décrit par 32 variables. On remarque que les classements, les points gagnés lors des matchs, les scores et les paris sont incomplets. A quelques exceptions (Wsets, Lsets, MaxW, MaxL, AvgW, AvgL), les variables sont **qualitatives**.

```
In [195]: tour_DS.describe(include="all")
```

```
Out[195]:            ATP Location Tournament        Date         Series    Court Surface      Ro
          count    44065    44065      44065       44065          44065    44065   44065      44
          unique      69      104        192        4047              9        2       4
          top          6    Paris    US Open  2001-01-15  International  Outdoor    Hard  1st Ro
          freq      2032     2784       2032         127          10490    36117   23169      20
```

1.6 PREPARATION DE LA DONNEE : De la donnée à l'information

1.6.1 Correction de valeurs incorrectes

En faisant l'inventaire des valeurs des modalités, des erreurs ont été detectées puis corrigées. Exemples : 'Comment' valeurs 'Walover' 'Retied' 'Compleed', . . . cette colonne n'étant pas utilisée dans la suite de l'étude ! On verra plus loin le traitement des valeurs NR pour les classements.

1.6.2 Formatage de la donnée

On transforme les types des données brutes, notamment pour rendre exploitable la date des matchs.

1.6.3 Construction d'une variable auxiliaire : le "MOIS" du tournoi

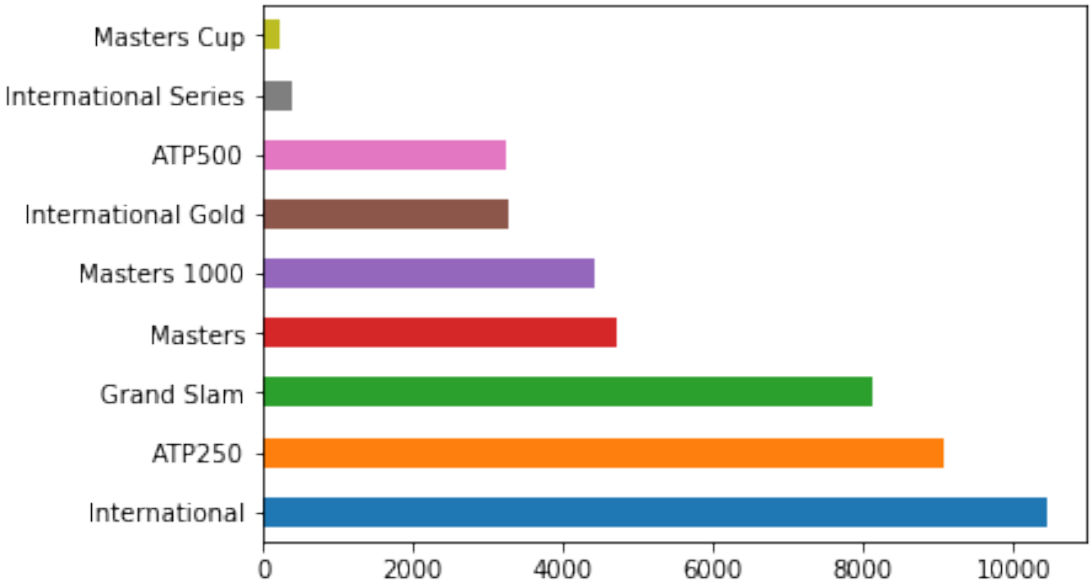
On simplifie ici la date en mois pour etre le marqueur unique du planning d'une saison de tournois. On fait l'**hypothèse** qu'un tournoi appartient au mois du dernier match en date (la finale dans le cas général). On devra tenir compte de cette approximation lors de l'interprétation : la répartition temporelle des tournois est approximative.

1.6.4 Analyse unidimensionnelle

On réalise ci après, l'étude des distribution des modaliés pour certaines variables qualitatives : 'Series', 'Court', 'Surface', 'Best of','Mois'

```
In [32]: matches_prediction['Series'].value_counts().plot(kind='barh')
```

```
Out[32]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8ad0dfe240>
```

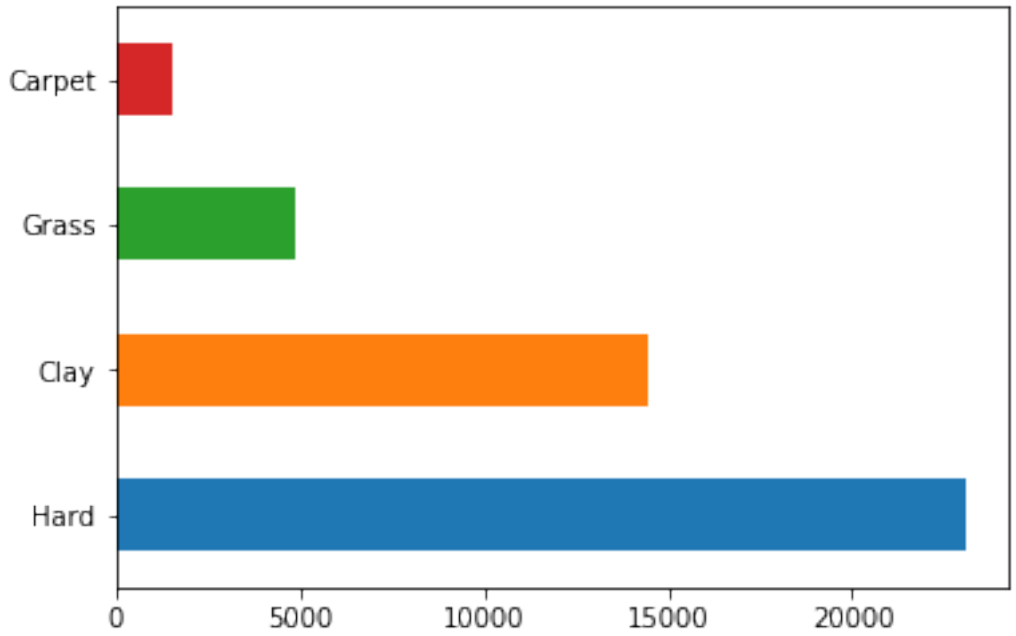



```
In [308]: matches_prediction['Court'].value_counts().plot(kind='pie')
```

```
Out[308]: <matplotlib.axes._subplots.AxesSubplot at 0x7f7458c16f60>
```

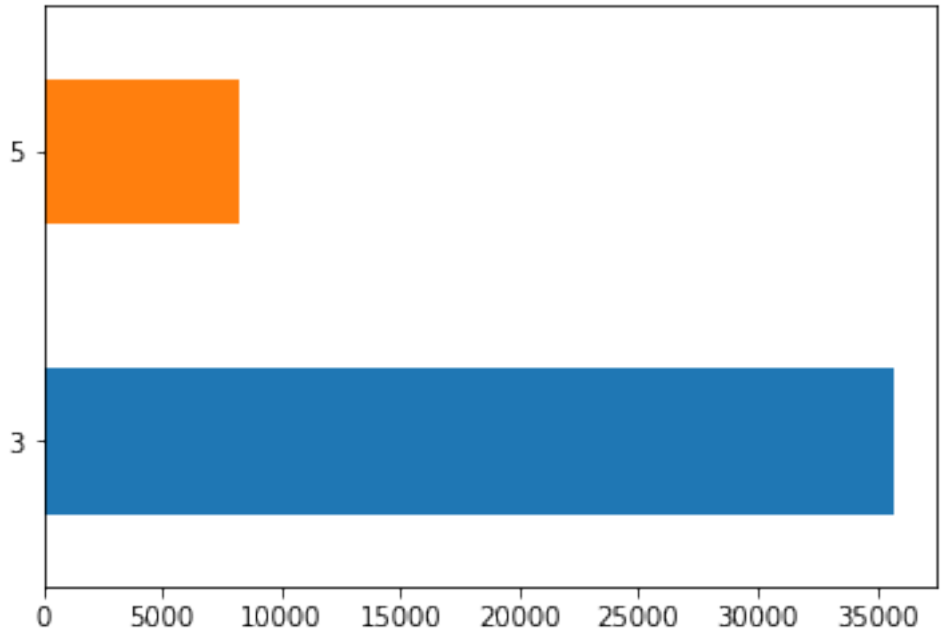

```
In [309]: matches_prediction['Surface'].value_counts().plot(kind='barh')
```

Out[309]: <matplotlib.axes._subplots.AxesSubplot at 0x7f743ac75828>



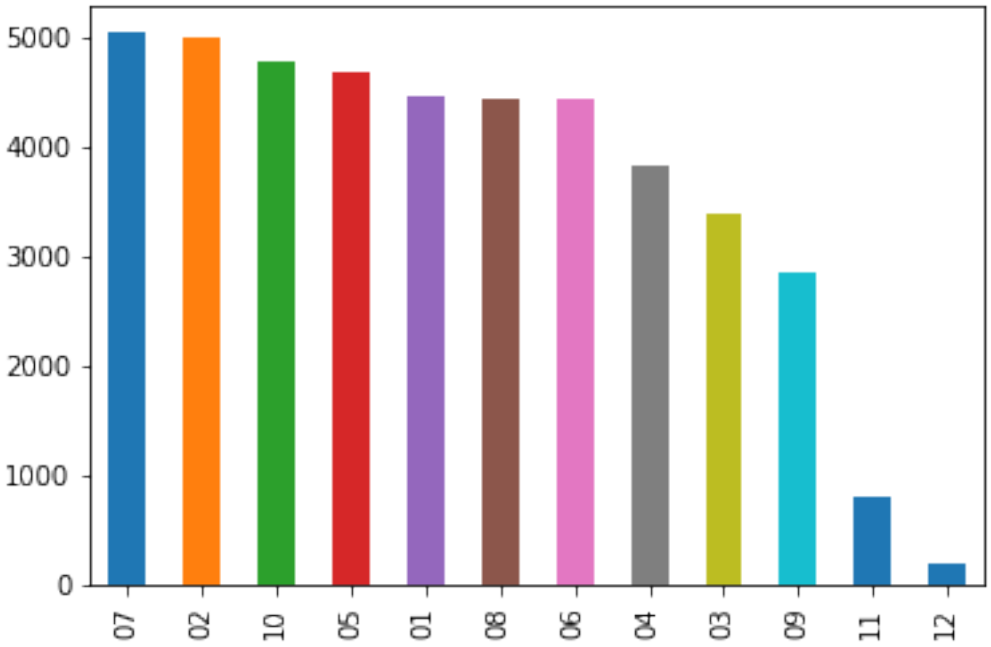
In [310]: matches_prediction['Best of'].value_counts().plot(kind='barh')

Out[310]: <matplotlib.axes._subplots.AxesSubplot at 0x7f743aeddf60>



```
In [32]: matches_prediction['Mois'].value_counts().plot(kind='bar')
```

```
Out[32]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8896f76ef0>
```



1.6.5 Etude bidimensionnelle

Etude de la corrélation Dans cette section, on étudie les relations entre les variables qui vont participer aux deux études pour compléter l'analyse unidimensionnelle précédente. Pour ce faire, on construira les tableaux de contingences (non representés) pour le calcul du khiš de contingence. Pour ce dernier test, à chaque fois que la "p-value" sera supérieure à 5%, on pourra conclure à l'indépendance des deux variables. La contingence sera représentée par un histogramme "mosaïque".

Par souci de synthèse, on ne représentera que les variables corrélées.

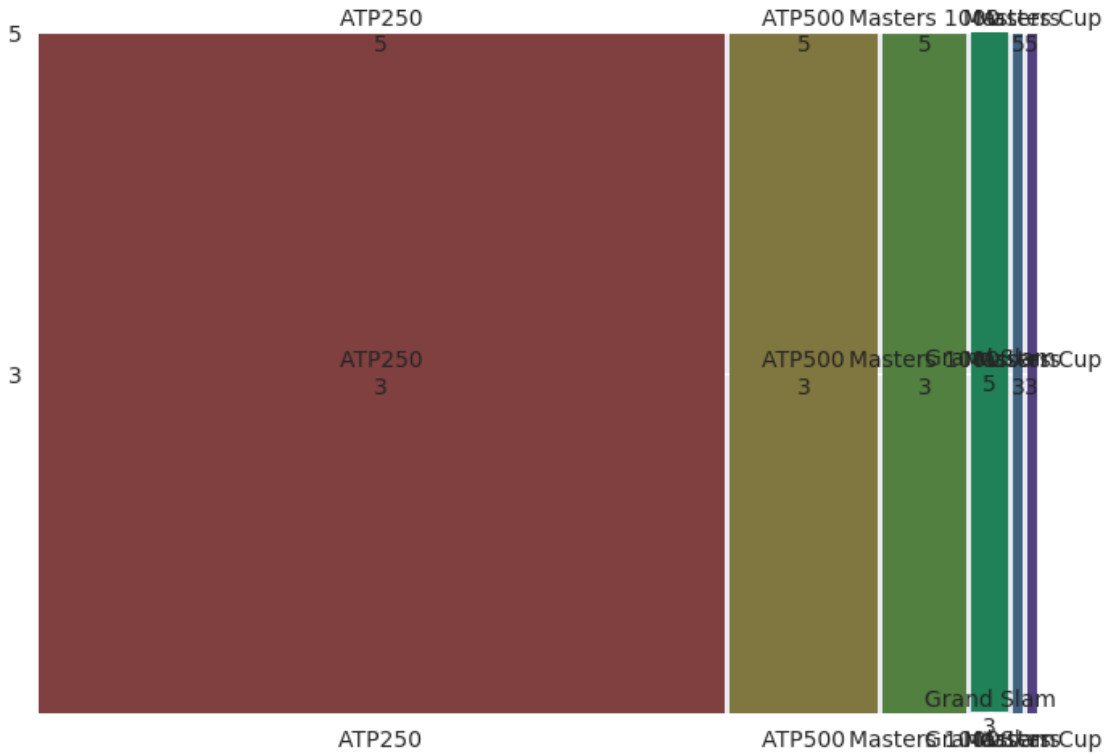
Sans tomber dans le piège : "La corrélation n'implique pas la causalité", on essaiera de trouver un sens tennistique aux éventuelles corrélations, à chaque fois que possible.

SERIES / BEST OF

```
In [98]: chi2,p,dof,expected=chi2_contingency(obs3)
         p
```

```
Out[98]: 4.66112473720682e-21
```

```
In [530]: mosaic(tournois_clustering,["Series","Best of"])
          plt.show()
```



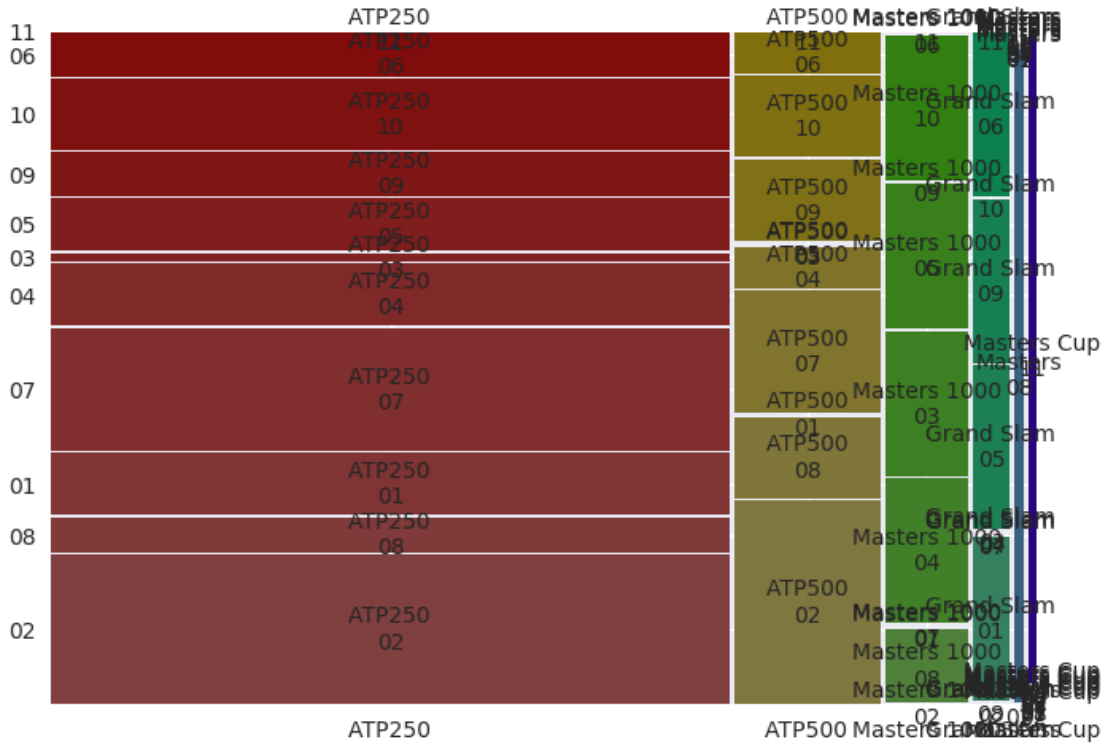
'Series' et 'Best of' sont très corrélées ! En effet, seuls les grand chelems se jouent en 5 sets, le reste des tournois se joue en 3.

SERIES / MOIS

```
In [97]: chi2,p,dof,expected=chi2_contingency(obs4)
         p

Out[97]: 1.0908950419296423e-12
```

```
In [531]: mosaic(tournois_clustering,["Series","Mois"])
          plt.show()
```



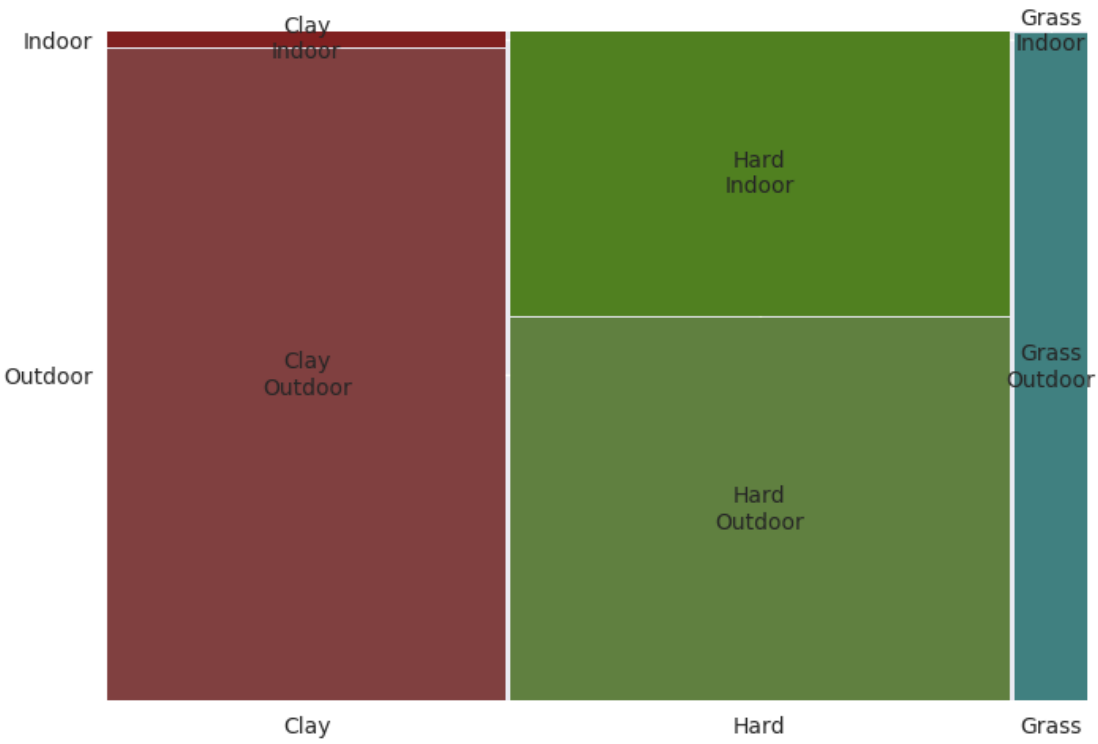
Là encore **'Series' et 'Mois' sont très corrélés !** Les masters 1000, grand chelems et masters, ont des dates fixes dans la saison. Quant aux autres tournois, leur catégorie permet d'émettre des hypothèses sur leur nombre par mois et leur proximité aux dates des grands tournois.

SURFACE / COURT

```
In [96]: chi2,p,dof,expected=chi2_contingency(obs5)
         p

Out[96]: 4.600989349950896e-06

In [532]: mosaic(tournois_clustering,["Surface","Court"])
          plt.show()
```



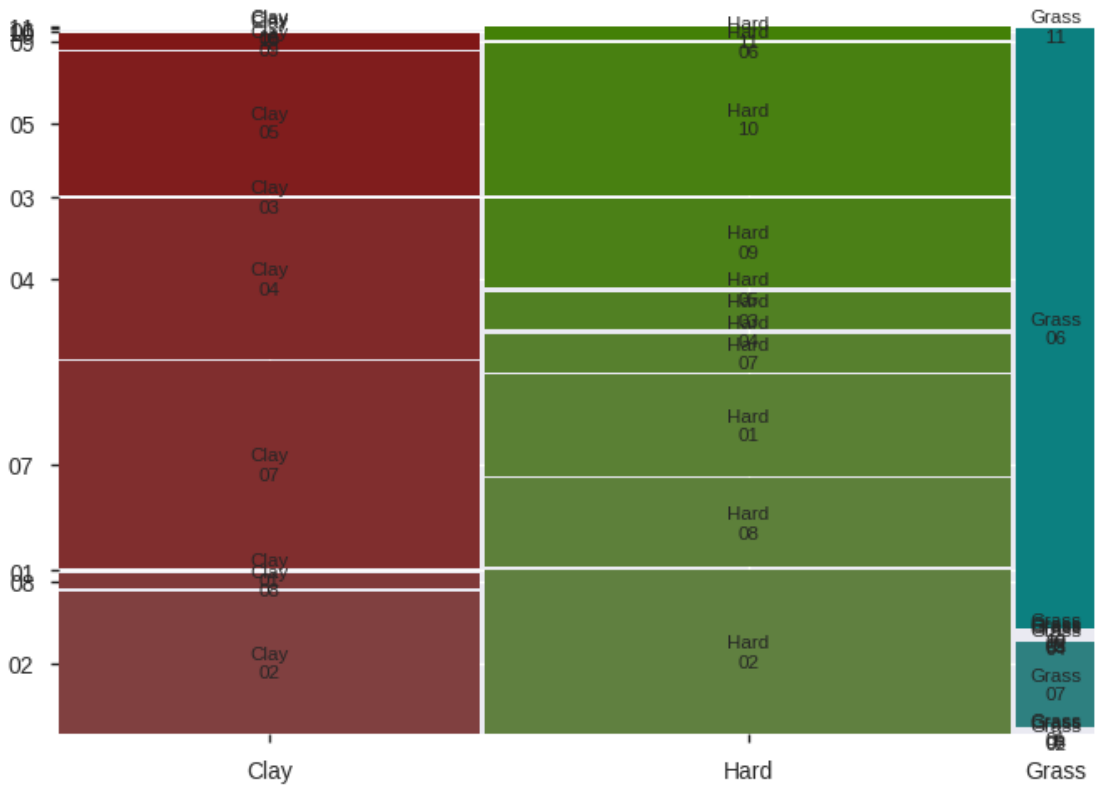
Ici aussi **'Surface' et 'Courts' sont très corrélés !** Les tournois sur terre battue et gazon se déroulent presque exclusivement en extérieur.

SURFACE / MOIS

```
In [154]: chi2,p,dof,expected=chi2_contingency(obs7)
          p

Out[154]: 9.11814521135037e-23

In [83]: mosaic(tournois_clustering,["Surface","Mois"])
         plt.show()
```



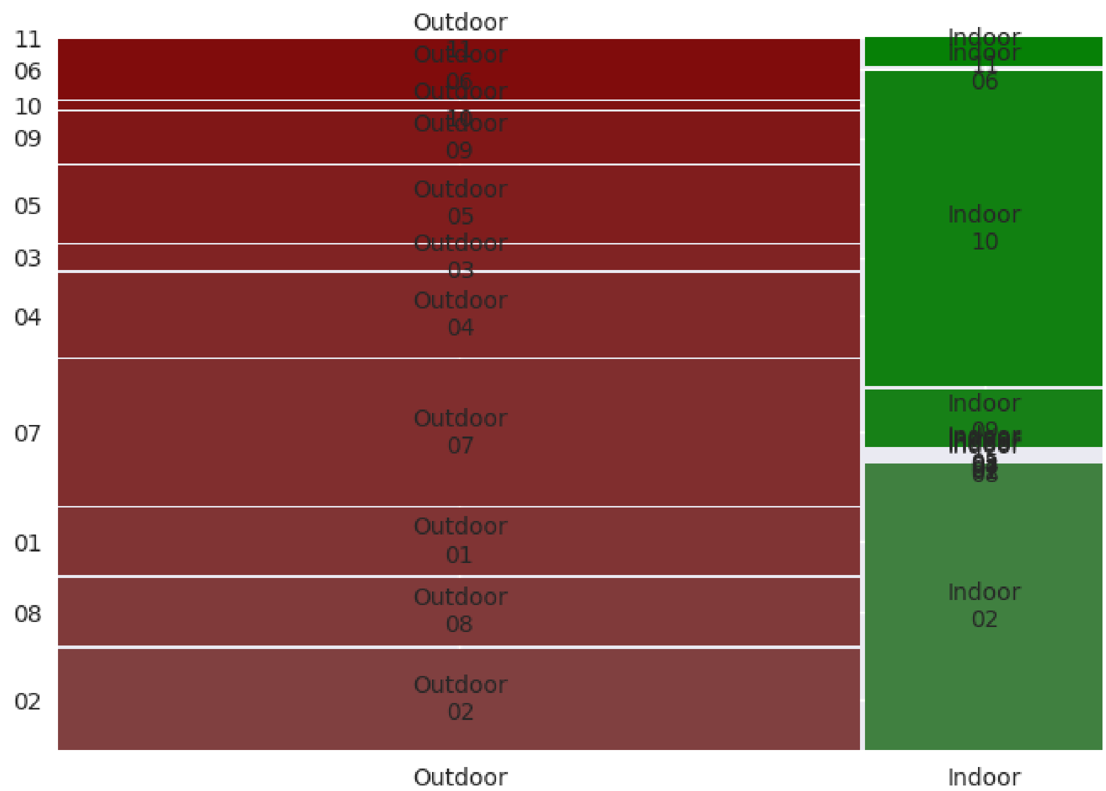
Fait extrêmement intéressant, notamment pour la partie classification qui va suivre, **'Surface' et 'Mois' sont très corrélés !** Chaque surface a sa période de l'année. Chronologiquement au cours des mois de l'année, la saison commence par le dur, suit la terre battue qui s'interrompt pour laisser place au gazon et toute la fin de saison se déroule sur dur.

COURT / MOIS

```
In [103]: chi2,p,dof,expected=chi2_contingency(obs9)
          p

Out[103]: 3.097927793259093e-09

In [537]: mosaic(tournois_clustering,["Court","Mois"])
          plt.show()
```



De même : **'Court' et 'Mois' sont très corrélés !** Si chaque surface a sa periode de l'année, cela a des répercussions sur le fait que le match se joue en extérieur ou en intérieur… puisque court et surface sont corrélés.

1.6.6 Sélection des individus commune aux deux études

Un premier objectif de filtrage est de supprimer les tournois 'obsolètes', qui correspondent à une classification ATP qui n'a plus cours ou des tournois qui ont disparu, changés de catégorie ou de planning. La solution choisie est de ne prendre que les x dernières années, on cherche à récupérer au moins 69 tournois (nombre de valeurs différentes de la colonne ATP) pour avoir une saison complète.

Le choix se porte sur l'**année 2010**. En effet, 2009 correspond côté ATP :

- à une réforme visant à modifier le nom des types de tournois : plus d'international series au profit des ATP 250 et 500, ce nombre étant l'ordre de grandeur de la dotation en milliers de $.
- La surface "moquette" sera interdite dans les futurs tournois.

Par conséquent, on ne partira en réalité, dans les deux parties de l'étude, que de matches réalisés sur une période de **7 ans**, du **1/1/2010** au **18/11/2016**.

1.7 PREMIERE PARTIE : CREATION DES CLASSES DE TOURNOIS PAR METHODE NON SUPERVISEE DANS LE CAS DE DONNEES QUALITATIVES

1.7.1 Sélection des individus spécifiques

Il est trés important de ne sélectionner, pour le problème présent, que les dernières occurrences. En effet, l'objectif est de répertorier les tournois récents avec leurs caractéristiques (type de tournoi, surface...) actualisées. Un tournoi, comme le grand prix de Lyon, a pu se jouer sur moquette jadis. Il se joue sur terre battue aujourd hui. Certains tournois disparaissent comme l'Open de Nice en 2017. Ces deux facteurs au moins pourraient fausser les classes que l'on pourrait construire. On supprime donc les doublons et on conserve les dernières occurences. On note une grosse réduction de notre jeu de données: de 40 000 lignes et 32 colonnes, on passe à 105 lignes et 6 colonnes.

1.7.2 Sélection des variables

On sélectionne les données en écartant tout ce qui concerne les matchs, scores, côtes et joueurs pour ne conserver que ce qui concerne les tournois : *'Tournament'*, *'Date'*, *'Date_t'*, *'Series'*, *'Court'*, *'Surface'*, *'Best of'*, *'Mois'*.

'Location' aurait pu être pris en compte au prix d'un travail supplémentaire de transformation longitude/latitude. Par souci de simplification, on l'écartera.

1.7.3 Choix de l'algorithme et application sur les données

Une première approche consiste à utiliser la Classification Ascendante Hiérarchique (CAH) pour construire des groupes de tournois avec des données sans étiquette. Le graphique qui en résulte permet de se faire une idée visuelle des différents regroupements et de faire un choix sur le nombre de classes.

. Deux possibilités : * Appliquer la CAH. * Appliquer une Analyse en composantes multiples sur les données avant d'appliquer la CAH.

Dans les deux cas, la CAH se fondant sur des distances entre individus, il est nécessaire de transformer chaque modalité en variables binaires. On procédera à un encodage disjonctif complet.

Construction de l'information : Conversion modalités Le profiling après sélection des individus nous apprend : * tournament 105 modalités : on choisira cette variable comme index * series : 6 * court : booleen * surface : 4 * best of : booleen * mois : 12

```
In [235]: tournois_clustering_d.head()
```

Out[235]:

Tournament	Series_ATP250	Series_ATP500	Series_Grand Slam	Series_Mast
Copa Telmex	1	0	0	
Garanti Koza Sofia Open	1	0	0	
Pilot Pen Tennis	1	0	0	
ASB Classic	1	0	0	
Studena Croatia Open	1	0	0	

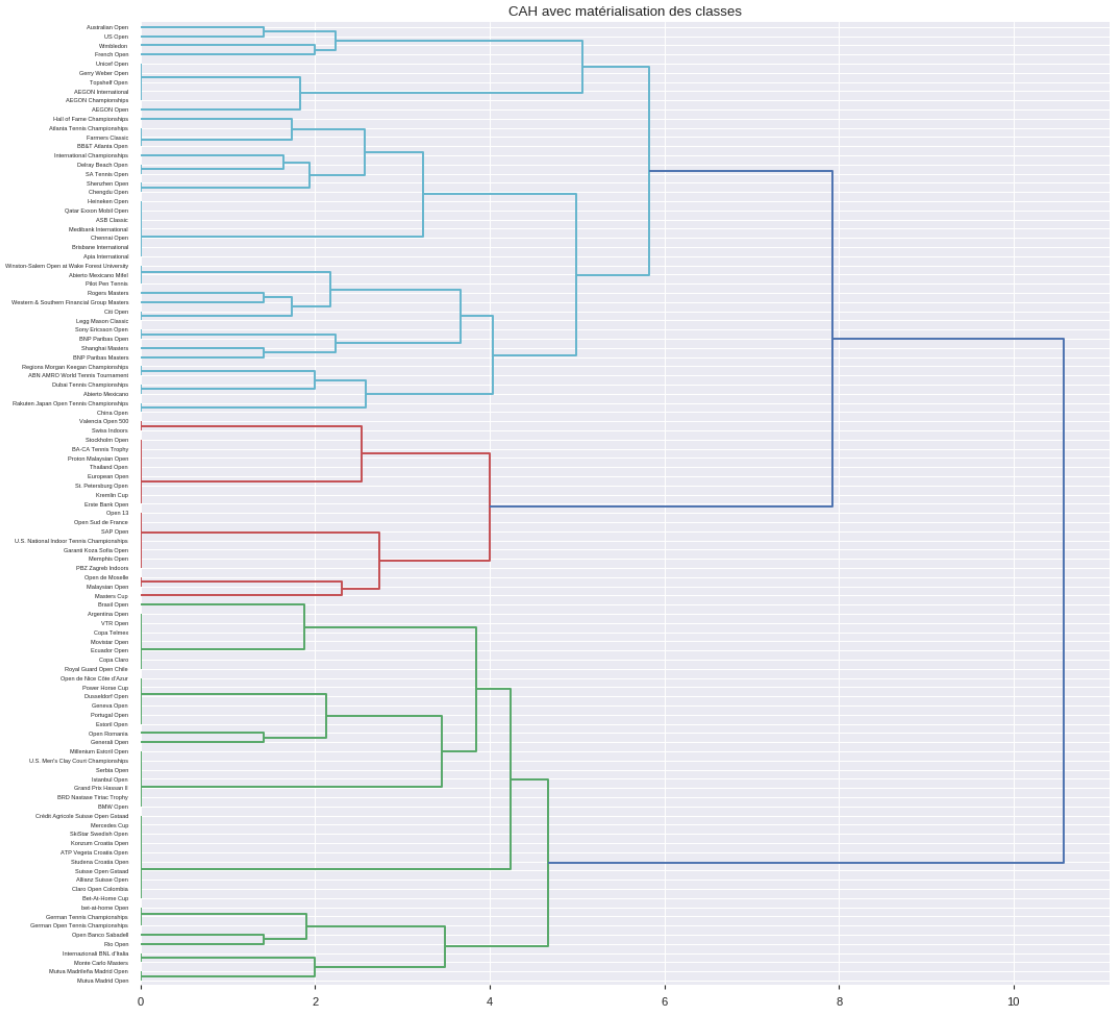
CAH et ses "hyperparamètres" D'après http://maths.cnam.fr/IMG/pdf/Classification-2008-2.pdf la CAH doit s'effectuee avec la méthode de Ward et la distance du š entre lignes. Après échanges, il vaut mieux éviter la distance euclidienne, utilisée par defaut, pour des variables qualitatives. Du coup, la méthode Ward est inutilisable sans distance euclidienne : "Methods 'centroid', 'median' and 'ward' are correctly defined only if Euclidean pairwise metric is used" cf https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.pdist.html

On choisit donc, en l'absence de distance du chiš en scipy, le couple méthode pondérée et distance cosinus.

Affichage du dendrogramme On génére deux aspects du même dendogramme. Le premier donne un aperçu du résultat et permet de faire un choix sur la "hauteur de découpe de l'arbre" qui déterminera le nombre de classes. Le second permet de dégager le nombre de classes. Seul ce dernier est représenté.

```
In [314]: #choix "metier" de la valeur de la hauteur
          hauteur = 6

In [321]: pic= plt.figure(figsize=(15, 15), dpi=80)
          dendrogram(Z,labels=tournois_comp.index,orientation='right',color_threshold=hauteur)
          plt.title('CAH avec matérialisation des classes')
          plt.show()
```



La colonne 'group' matérialise les classes de tournois fournies par l'algorithme utilisé.

```
In [255]: pandas.DataFrame(tournois_comp.index[idg],groupes_cah[idg]).head()
```

```
Out[255]:                       Tournament
          1              Mutua Madrid Open
          1  Mutua Madrileña Madrid Open
          2           Monte Carlo Masters
          2  Internazionali BNL d'Italia
          3                      Rio Open
```

Caractéristiques des groupes

Effectif de chaque groupe

```
In [505]: tournois_groupes.group.value_counts().sort_values()
```

```
Out[505]: 2    20
          1    42
          3    43
          Name: group, dtype: int64
```

Caractérisation en variables des groupes Dans cette section, on découvre les caractéristiques en "portion de variable" des différentes classes données par la CAH. C'est grâce à cette information que l'on pourra décrire les différents groupes et les interpréter.

Type de tournois

```
In [459]: pd.crosstab(tournois_groupes['group'],tournois_clustering['Series'],margins=True, no
```

```
Out[459]: Series     ATP250  ATP500  Grand Slam  Masters  Masters 1000  Masters Cup        All
          group
          1       44.594595   31.25         0.0      0.0     44.444444          0.0  40.000000
          2       22.972973   12.50         0.0      0.0      0.000000        100.0  19.047619
          3       32.432432   56.25       100.0    100.0     55.555556          0.0  40.952381
```

Répartition du meilleur des manches

```
In [460]: pd.crosstab(tournois_groupes['group'],tournois_clustering['Best of'],margins=True, n
```

```
Out[460]: Best of          3      5        All
          group
          1        41.584158    0.0  40.000000
          2        19.801980    0.0  19.047619
          3        38.613861  100.0  40.952381
```

Extérieur / intérieur

```
In [461]: pd.crosstab(tournois_groupes['group'],tournois_clustering['Court'],margins=True, norm
```

```
Out[461]: Court     Indoor    Outdoor        All
          group
          1       4.166667  50.617284  40.000000
          2      83.333333   0.000000  19.047619
          3      12.500000  49.382716  40.952381
```

Répartition des surfaces de jeu dans les classes

```
In [462]: pd.crosstab(tournois_groupes['group'],tournois_clustering['Surface'],margins=True, n
```

```
Out[462]: Surface       Clay  Grass       Hard        All
          group
          1        97.674419    0.0   0.000000  40.000000
          2         0.000000    0.0  37.037037  19.047619
          3         2.325581  100.0  62.962963  40.952381
```

Répartition dans l'année

```
In [463]: pd.crosstab(tournois_groupes['group'],tournois_clustering['Mois'],margins=True, norma
```

```
Out[463]: Mois      01         02     03     04         05     06         07    08    09
          group
          1        0.0  40.909091    0.0  100.0  88.888889    0.0  76.470588  12.5  12.5   0.0
          2        0.0  31.818182    0.0    0.0   0.000000    0.0   0.000000   0.0  25.0  83.3
          3      100.0  27.272727  100.0    0.0  11.111111  100.0  23.529412  87.5  62.5  16.6
```

1.7.4 Evaluation et interprétation de la première partie

Le résultat consistent en un découpage de la saison de tennis professionnel en 3 groupes :

Groupe 1 : La saison de terre battue : EUROPE et AMERIQUE DU SUD :

- *40% des tournois, ~40% des Master1000, et ATP250, 30% des ATP500*
- *50% des tournois outdoor, 5% indoor*
- *moitié février, avril, mai, juillet et une fraction d'août et septembre*
- *quasiment tous les tournois sur terre battue de la saison et ce à l'exclusion d'autres surfaces*

Ce groupe propose exclusivement des tournois sur terre battue. C'est le groupe de tournois qui, aux beaux jours, sur une surface exigente pour l'entretien et le physique des joueurs va aboutir au grand chelem Roland Garros (sans que celui ci appartienne à ce groupe). C'est ce groupe qui rassemble énormement (40%) de tournois master 1000 : parmi eux les gros tournois européens sur terre battue : Rome, Madrid, Monte Carlo. Mais c'est aussi le groupe de la saison sud américaine pour ceux qui ne "s'alignent" pas sur les tournois sur herbe qui suit la saison sur terre battue comme on pourra le voir ci-aprés. On notera que l'Open du Brésil de février, seul tournoi en terre battue indoor de la saison, appartient bien à ce groupe.

Groupe 2 : Le noyau des tournois dur indoors masters compris :

- *20% des tournois*
- *20% des ATP250, 10% des ATP500 et le masters*
- *80% des tournois indoor*
- *1/3 de février, 1/4 septembre, octobre, novembre*
- *1/3 des tournois sur dur*

Ce (petit) groupe propose exclusivement des tournois sur dur intérieur. C'est le groupe des tournois de fin de saison qui se finit par le masters: sorte de 5e grand chelem. Celui-ci se déroule à Londres depuis quelques années à une période où il vaut mieux jouer en intérieur...

Groupe 3 : Le groupe mastodonte : Les grands chelems et les tournois majeurs sur surfaces rapides :

- *40% des tournois*
- *30% des ATP250, 50% des ATP500, 50% des master1000, tous les grands chelems*
- *50% outdoor, 10%indoor*
- *janvier, 1/5 fevrier, mars, juin, août, septembre*
- *2/3 des tournois sur dur, tous les tournois sur herbe et Roland garros sur terre battue*

Ce groupe pourrait résumer à lui seul toute la saison. Il "contient" les 4 tournois majeurs dits du grand chelem : chronologiquement l'open d'Australie, Roland Garros, Wimbledon et l'US Open, qui se jouent sur les 3 surfaces, le dur étant representé deux fois.

Ce groupe a pour identité les surfaces rapides : toute la saison sur herbe, débouchant sur Wimbledon, y réside. Mais on y trouve toute la "tournée américaine" (les Masters 1000 d'Indian Wells et Miami en mars avant le groupe 1, Toronto et Cincinatti aprés le groupe 1 et avant le groupe 2)

Il contient quelques tournois indoor mineurs : Memphis et Rotterdam en février dont on peut légitimement se poser la question de leur présence dans ce groupe plutôt que dans le groupe 2. Mais surtout, et là c'est plus cohérent, la présence du (gros) master 1000 indoor de Paris Bercy.

1.7.5 Conclusion partielle

Le premier découpage proposé à l'intérêt d'être synthétique. Il est interprétable donc valide. Par contre, comme on a pu le voir plus haut, sa cohérence "métier" est discutable. On va voir si le changement de méthode d'analyse de données permet une amélioration.

1.7.6 Amélioration du découpage : Application d'une CAH sur une ACM

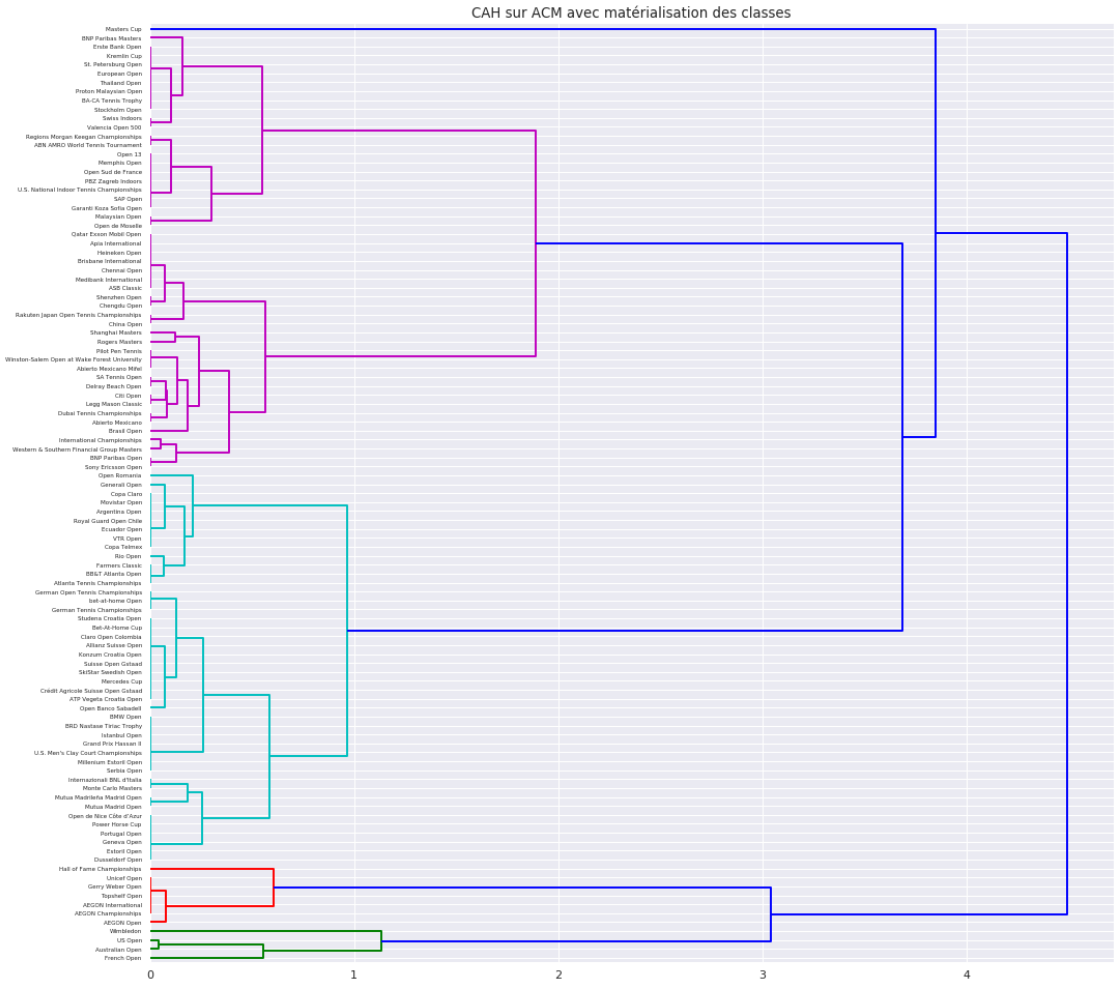
Principe Effectuer la classification hiérarchique sur le tableau des coordonnées factorielles des n individus après A.C.M. et des variables prédictives. Les approches CAH sur tableau disjonctif et CAH sur ACM sont équivalentes si on utilise tous les facteurs de l'A.C.M. L'objectif est de comparer les résultats de cette nouvelle méthode à la methode précédente et, après interprétation, de juger l'éventuelle amélioration.

Mise en place On applique une ACM au tableau disjonctif que l'on a fourni précédemment en entrée de la CAH. Par défaut, l'implémentation sélectionnée utilise une fonction de correction de la variance et des valeurs propres nommées 'Benzecri'. On projette sur l'espace factoriel puis on applique la CAH.

```
In [138]: mca_ben = MCA(tournois_clustering_d, ncols=ncols)
          projete = mca_ben.fs_r()
```

Résultat de l'application de la CAH sur ACM

```
In [322]: #matérialisation des classes (hauteur cf param 'threshold')
          pic= plt.figure(figsize=(15, 15), dpi=80)
          plt.title('CAH sur ACM avec matérialisation des classes')
          dendrogram(Z_ACM,labels=tournois_clustering.index,orientation='right',color_threshol
          plt.show()
```




```
<matplotlib.figure.Figure at 0x7f9911afd2b0>
```

Comparaison avec la CAH sur disjonctif On reprend la même étude que précédemment : effectifs puis tableau croisé groupes / variables initiales avec cette fois 5 groupes.

Caractéristiques des groupes

Effectif de chaque groupe

```
In [503]: tournois_groupes_ACM.group_ACM.value_counts().sort_values()

Out[503]: 5     1
          1     4
          2     7
          3    44
          4    49
          Name: group_ACM, dtype: int64
```

Caractérisation en variables des groupes Dans cette section, on découvre les caractéristiques en "portion de variable" des differentes classes données par la CAH. C'est grâce à cette information que l'on pourra décrire les différents groupes et les interpréter.

Type de tournoi

```
In [363]: pd.crosstab(tournois_groupes_ACM['group_ACM'],tournois_clustering['Series'],margins=
```

Out[363]:

Series	ATP250	ATP500	Grand Slam	Masters	Masters 1000	Masters Cup	
group_ACM							
1	0.000000	0.00	100.0	0.0	0.000000	0.0	3.809
2	8.108108	6.25	0.0	0.0	0.000000	0.0	6.666
3	47.297297	31.25	0.0	0.0	44.444444	0.0	41.904
4	44.594595	62.50	0.0	100.0	55.555556	0.0	46.666
5	0.000000	0.00	0.0	0.0	0.000000	100.0	0.952

Répartition du "meilleur" des manches

```
In [364]: pd.crosstab(tournois_groupes_ACM['group_ACM'],tournois_clustering['Best of'],margins=
```

Out[364]:

Best of	3	5	All
group_ACM			
1	0.000000	100.0	3.809524
2	6.930693	0.0	6.666667
3	43.564356	0.0	41.904762
4	48.514851	0.0	46.666667
5	0.990099	0.0	0.952381

Extérieur / intérieur

```
In [365]: pd.crosstab(tournois_groupes_ACM['group_ACM'],tournois_clustering['Court'],margins=Tr
```

Out[365]:

Court	Indoor	Outdoor	All
group_ACM			
1	0.000000	4.938272	3.809524
2	0.000000	8.641975	6.666667
3	0.000000	54.320988	41.904762
4	95.833333	32.098765	46.666667
5	4.166667	0.000000	0.952381

Répartition des surfaces de jeu dans les classes

```
In [366]: pd.crosstab(tournois_groupes_ACM['group_ACM'],tournois_clustering['Surface'],margins=
```

Out[366]:

Surface	Clay	Grass	Hard	All
group_ACM				
1	2.325581	12.5	3.703704	3.809524
2	0.000000	87.5	0.000000	6.666667
3	95.348837	0.0	5.555556	41.904762
4	2.325581	0.0	88.888889	46.666667
5	0.000000	0.0	1.851852	0.952381

Répartition dans l'année

```
In [367]: pd.crosstab(tournois_groupes_ACM['group_ACM'],tournois_clustering['Mois'],margins=Tru
```

```
Out[367]: Mois          01         02     03     04         05         06         07    08    09
          group_ACM
          1           12.5   0.000000    0.0    0.0  11.111111  14.285714   0.000000   0.0  12.5
          2            0.0   0.000000    0.0    0.0   0.000000  85.714286   5.882353   0.0   0.0
          3            0.0  36.363636    0.0  100.0  88.888889   0.000000  94.117647  12.5  12.5
          4           87.5  63.636364  100.0    0.0   0.000000   0.000000   0.000000  87.5  75.0
          5            0.0   0.000000    0.0    0.0   0.000000   0.000000   0.000000   0.0   0.0
```

1.7.7 Evaluation et interprétation de l'optimisation

Grace à la visualisation de la CAH, on propose cette fois de découper les individus en 5 groupes.

- **GROUPE 1 : LES GRAND CHELEMS** :
- *4 tournois, 100% des grands chelems, 100% 5sets, tous en extérieur, toutes les surfaces, janvier (OA) - mai (RG) - juin (wimbledon) - septembre (USO)*
- **GROUPE 2 : LES TOURNOIS "MINEURS" SUR GAZON** :
- *~7% des tournois*
- *~45% des ATP250, ~30% des ATP500, ~45% des Masters1000
- *Exclusivement en 3 sets*
- *Exclusivement en extérieur*
- *Tous les tournois sur gazon et seulement ceux-ci à l'exception de Wimbledon*
- *La majorité des tournois de juin*
- **GROUPE 3 : LA SAISON SUR TERRE BATTUE** :
- *~40% des tournois*
- *30% des ATP250, 50% des ATP500, 50% des master1000*
- *Exclusivement en 3 sets*
- *Exclusivement en extérieur*
- *30% des tournois de février, avril et mai, juillet et une faible fraction d'août à septembre*
- *Tous les tournois sur terre battue à l'exception de Roland Garros... plus quelques tournois sur dur*

Interprétation des exceptions :
D'apres les données ci-dessous, l'Open de Los Angeles(farmer classics) et d'Atlanta ont lieu en juillet en plein "boom" de la saison de terre battue. Cette originalité leur vaut leur incorporation dans le groupe 3. C'est discutable d'un point de vue métier car ils pourraient figurer dans un groupe de surface rapide comme le groupe 2, groupe qui propose des tournois pendant cette

période. Néanmoins, on ne peut pas blâmer l'algorithme d'avoir incorporé ces tournois dans le plus gros groupe. De plus, la similarité gazon / dur n'apparaît pas dans les données étudiées !

A noter que Atlanta Tennis Championships et BB&T Atlanta Open sont des dénominations marketing distinctes d'un même tournoi.

- **GROUPE 4 : LE GROUPE DES SURFACES RAPIDES INTERIEUR ET EXTERIEUR** :
- *~45% des tournois*
- *~45% des ATP250, ~60% des ATP500, 55% des master1000*
- *30% des tournois outdoor, tous les tournois indoor hormis le masters*
- *janvier, février, mars d'une part et août, septembre, octobre*
- *tous les tournois sur dur à l'exception des grands chelem sur ces surfaces... plus un tournoi sur terre battue*

Interpréation de l'exception :

D'apres les données ci dessous, l'Open du Brésil est l'unique tournoi indoor sur terre battue. Cette originalité lui vaut son incorporation dans le groupe 4, ce qui est discutable d'un point de vue métier. Il pourrait prendre place dans le groupe 3 qui propose des tournois à cette période. Encore une fois, l'algorithme a tenu compte du caractère indoor plus important dans le groupe 4. Par ailleurs, il ne disposait pas d'une information de hiérarchie surface > indoor/outdoor, qui lui aussi est discutable d'un point de vue métier.

- **GROUPE 5 : LE MASTERS** :
- *1 tournois, 100% des masters (cup), exclusivement en dur intérieur, fin de saison en novembre*

Seul tournoi par poules en fin de saison, il fait l'objet d'un groupe à lui tout seul vu ses spécificités.

Remarques Cette fois-ci, nous avons une répartition orientée sur les critères prédominants d'importance du tournoi et de surface. On remarque que si le second critère est explicite dans les variables fournies à l'algorithme, le premier a été déduit en fonction des autres variables. Les groupes sont une fois encore interprétables donc valides et d'un point de vue "métier" plus cohérent. Il reste néanmoins quelques scories.

```
In [324]: tournois_groupes_ACM[(tournois_groupes_ACM.Surface_Clay == 1)&(tournois_groupes_ACM.g
```

```
Out[324]:              group_ACM  Series_ATP250  Series_ATP500  Series_Grand Slam  Series_Maste
          Tournament
          Brasil Open          4              1              0                  0
```

```
In [325]: tournois_groupes_ACM[(tournois_groupes_ACM.Surface_Hard == 1)&(tournois_groupes_ACM.g
```

```
Out[325]:                              group_ACM  Series_ATP250  Series_ATP500  Series_Grand S
          Tournament
          Atlanta Tennis Championships          3              1              0
          Farmers Classic                       3              1              0
          BB&T Atlanta Open                     3              1              0
```

1.7.8 Comparaison avec un autre algorithme au fonctionnement distinct

On propose pour valider les résultats du premier algorithme d'utiliser un autre algorithme ayant le même but (créer des groupes dans de grandes données) mais avec un fonctionnement différent. En effet, les k-means sont des algorithmes orientés partition alors que les CAH sont des algorithmes hiérarchiques.

Choix du nombre de classes, ici on s'adapte au nouveau nombre de classes (5) de la CAH sur ACM.

Exécution de l'algorithme

```
In [295]: kmeans_ACM = cluster.KMeans(n_clusters=k)
          kmeans_ACM.fit(tournois_clustering_d)

Out[295]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
              n_clusters=5, n_init=10, n_jobs=1, precompute_distances='auto',
              random_state=None, tol=0.0001, verbose=0)

In [379]: pandas.crosstab(groupes_cah_ACM,kmeans_ACM.labels_)

Out[379]: col_0   0   1  2   3  4
          row_0
          1       0   2  1   0  1
          2       0   0  0   0  7
          3       0   3  9  32  0
          4      23  26  0   0  0
          5       1   0  0   0  0
```

Conclusion de la comparaison Au numero des classes et à quelques individus prés, il y a coïncidence entre les classes produites par la CAH et le k-means. Celui-ci est paramétré avec le choix arbitraire mais assisté par la visualisation du nombre de classes suggéré donc par la CAH.

Cela corrobore en premiere approximation les résultats précédents.

Utilisation d'un indice pour éclairer sur le nombre de classes choisi lors de la CAH

```
In [276]: from sklearn import metrics
          #faire varier le nombre de clusters de 2 à 10
          res = np.arange(9,dtype="double")
          for k in np.arange(9):
              km = cluster.KMeans(n_clusters=k+2)
              km.fit(tournois_clustering_d)
              res[k] = metrics.silhouette_score(tournois_clustering_d,km.labels_)
          print(res)
          #graphique
          import matplotlib.pyplot as plt
          plt.title("Silhouette")
          plt.xlabel("# of clusters")
          plt.plot(np.arange(2,11,1),res)
          plt.show()
```

```
[0.25531526 0.29501036 0.3378127  0.31713963 0.33269589 0.34410889
 0.38073804 0.42031212 0.46042234]
```

On remarque que la décision “métier” de choisir un nombre de groupes égal à 5 n’est pas forcément optimal. Du point de vue du k-means, l’indice silhouette montre qu’un découpage avec un nombre supérieur de classes eut été meilleur. Par contre, avec 9 ou 10 classes, l’analyse n’aurait pas forcément été synthétique. Le graphique suggère que k=4 eut été un meilleur choix en terme de ratio synthèse / restitution de l’information. En effet, regrouper le groupe 1 et 5 pour former le groupe des tournois majeurs est cohérent d’un point de vue métier.

1.7.9 Conclusion de la première partie

L’analyse a atteint son but :

Du point de vue de l’analyse de données, l’étude décrit l’amélioration de la classification par une méthode combinant l’analyse factorielle et un outil de classification classique pour rendre un résultat plus cohérent au sens métier.

Du point de vue métier, cette analyse, aidée par les outils de fouille de données, rend bien une idée de la saison de tennis professionnel. Pour un non-initié, elle permet de voir que le découpage correspond à la saisonalité de l’hémisphère nord. En effet, on joue des tournois en extérieur avec des surfaces exigentes en terme d’entretien (gazon, terre battue) ou en terme d’investissement physique du joueur (terre battue) pendant les “beaux jours”. Sans doute que le fait que le marché, notamment pour venir physiquement voir le match, soit plutôt dans l’hémisphère nord ne doit pas être sans effet. Néanmoins, l’audiovisuel relativise ce fait.

1.7.10 Améliorations possibles

Variables

- Intégrer “Location” et corréler le continent auquel “location” appartient. Cela ajouterait un critère géographique. http://geopy.readthedocs.io/en/latest/
- Ajouter les points ATP octroyés aux joueurs pour mieux distinguer les tournois.
- Ajouter le nombre de tours des tournois et les tours de qualification pour mieux distinguer les tournois.
- Considérer toutes les occurences du tournoi s’il a changé de dates, pondérer négativement les tournois ayant disparus...
- Affiner les dates des tournois : du mois à la semaine.

Individus

- Ajouter les tournois des autres années en construisant une variable année... il faudrait comparer par décennie vu les disparitions et les changements de surface voire dégradation au sens “série ATP” au cours du temps.

Données Enrichir les données :

- Ajouter une notion de similarité des surfaces : Le dur et le gazon sont plus proches d’un point de vue tennis que la terre battue.
- Ajouter les résultats des matchs des tournois “challenger” et “future” qui font aussi partie de la saison ATP. Ces tournois sont moins connus car moins dotés et concernent les joueurs mal classés ou en devenir.

Méthode

- Envisager d’autres algorithmes et d’autres critères de validation

Implémentation Choix des bibliothèques :

- “Disjonctiver” avec ScikitLearn.LabelBinarizer()
- Comparer les résultats avec AgglomerativeClustering de scikit learn

Extension de la problématique

- Suivre la tendance des tournois au cours des années (faire une double ACP pour l’étudier)

1.8 DEUXIEME PARTIE : PREDICTION DU RESULTAT D'UN MATCH PAR METHODE SUPERVISEE SUR UN ENSEMBLE DE VARIABLES QUALITATIVES

1.8.1 Préparation des données spécifiques

Choix des prédictives On supprime tout ce qui n'a pas trait aux matchs ou qui découle du résultat (les scores des gagnants et des perdants). Après la construction de la variable que l'on souhaite prédire et de la variable mois, on n'oubliera pas de supprimer les variables qui nous ont permis de les construire.

```
In [296]: attributs_prediction=['Series', 'Court', 'Surface', 'Best of', 'Mois', 'Round']
          attributs_complementaires_a_supprimer=['WRank','LRank','Date_t']
```

Choix des Individus On supprime les individus à valeur manquante et les classements non renseignés.

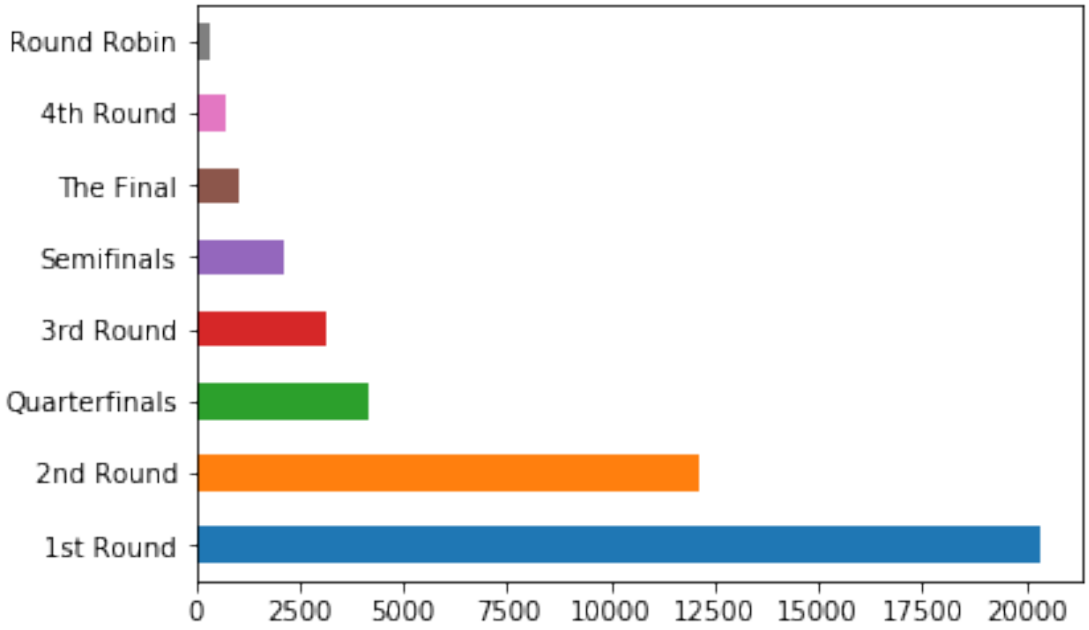
```
In [30]: matches_prediction.shape
```

```
Out[30]: (43962, 9)
```

1.8.2 Analyse unidimensionnelle spécifique et distribution des variables

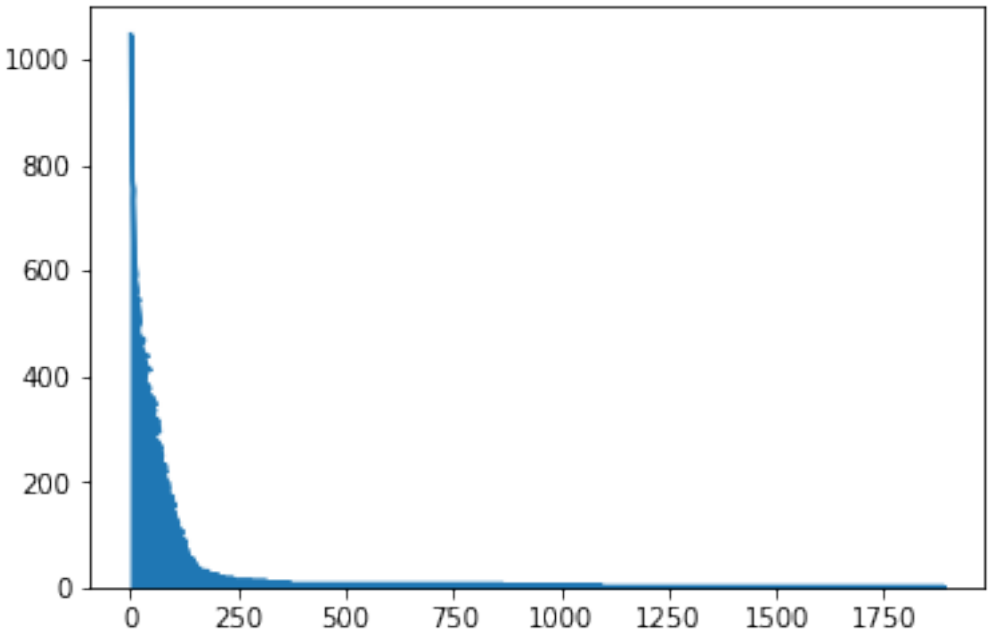
```
In [257]: matches_prediction['Round'].value_counts().plot(kind='barh')
```

```
Out[257]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8895954e10>
```



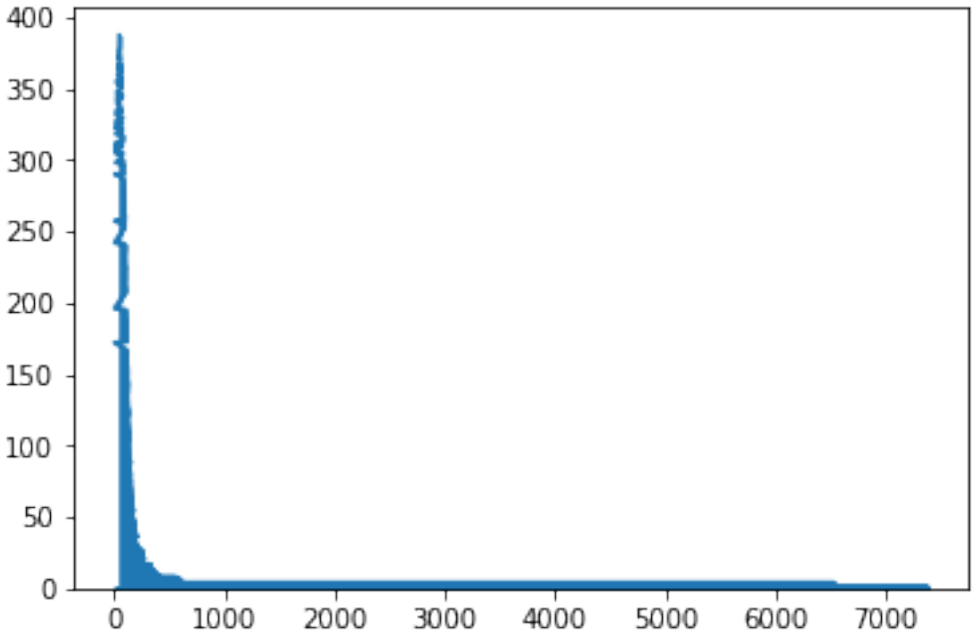

```
In [311]: matches_prediction['WRank'].value_counts().plot(kind='area')
```

Out[311]: <matplotlib.axes._subplots.AxesSubplot at 0x7f743ac45828>



In [313]: matches_prediction['LRank'].value_counts().plot(kind='area')

Out[313]: <matplotlib.axes._subplots.AxesSubplot at 0x7f743af3bd30>



Les données comportent plus de résultats de matchs pour les bien classés mais peu pour les mal classés ... C'est inhérent à la structure des tournois de tennis et à l'évolution des classements au cours des saisons de tennis successives: 1. les premiers sont souvent exemptés des premiers tours et sont donc plus frais donc favorisés sportivement => leur probabilité de gagner est améliorée par l'effet bénéfique de leur classement 2. les règles de constitution des tableaux font qu'ils jouent contre les moins bien classés... qu'ils battent généralement faute de perdre en classement l'année suivante En résumé, les mieux classés gagnent plus de matchs donc sont plus représentés dans les feuilles de matchs avec un effet cumulatif de saison en saison.

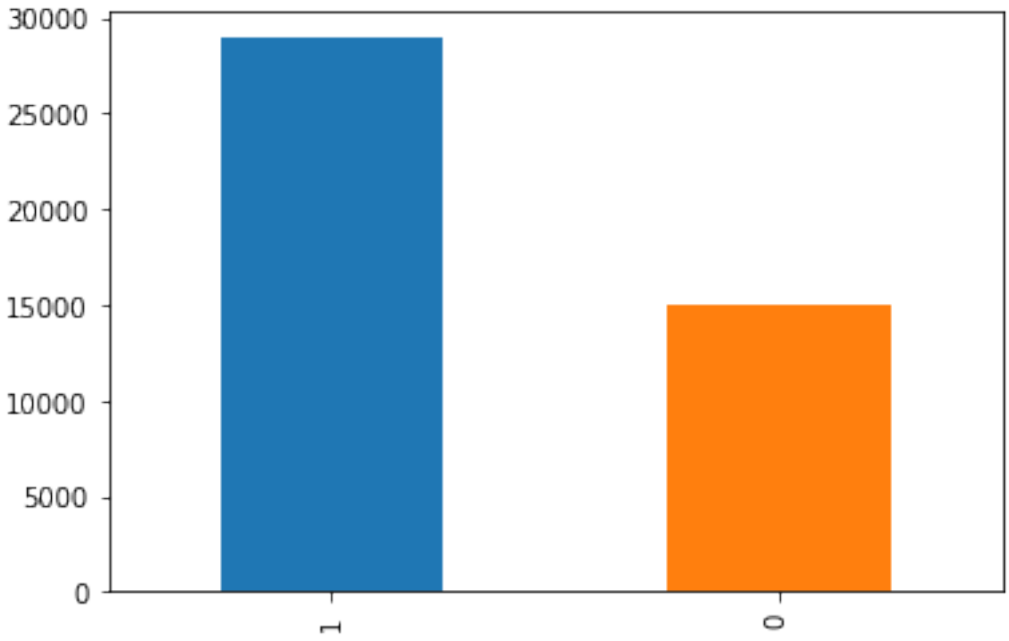
Construction de la variable à prédire La "perf" en tennis signifie que *le joueur le moins bien classé bat le joueur le mieux classé* : C'est ce que nous allons tenter de prédire.

```
In [322]: matches_prediction['y']=np.where(matches_prediction['LRank']>matches_prediction['WRa
```

Distribution des modalités de la variable construite

```
In [51]: matches_prediction['y'].value_counts().plot(kind='bar')
```

```
Out[51]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8ac07d3ef0>
```




```
In [329]: matches_prediction.describe()
```

```
Out[329]:                    y
          count  43962.000000
```

```
mean        0.341977
std         0.474377
min         0.000000
25%         0.000000
50%         0.000000
75%         1.000000
max         1.000000
```

On remarque que **l'effectif des "perf" est le plus important !**

1.8.3 Analyse bidimensionnelle

```
In [142]: pd.crosstab(matches_prediction['Series'],matches_prediction['y'],margins=True,normali
```

```
Out[142]: y                              0         1
          Series
          ATP250                 0.359833  0.640167
          ATP500                 0.336609  0.663391
          Grand Slam             0.278380  0.721620
          International          0.361340  0.638660
          International Gold     0.367521  0.632479
          International Series   0.366834  0.633166
          Masters                0.375318  0.624682
          Masters 1000           0.327001  0.672999
          Masters Cup            0.283333  0.716667
          All                    0.342000  0.658000
```

On remarque que les "perf" sont plus nombreuses en grand chelem et en Masters Cup... mais attention ! L'effectif des matchs est moindre car seuls 5 tournois par saison produisent ces matchs.

```
In [259]: pd.crosstab(matches_prediction['Round'],matches_prediction['y'],margins=True,normaliz
```

```
Out[259]: y                      0         1
          Round
          1st Round       0.365930  0.634070
          2nd Round       0.313000  0.687000
          3rd Round       0.308283  0.691717
          4th Round       0.258065  0.741935
          Quarterfinals   0.336543  0.663457
          Round Robin     0.342949  0.657051
          Semifinals      0.366651  0.633349
          The Final       0.345351  0.654649
          All             0.342000  0.658000
```

Les "perf" sont plus nombreuses au 4ème tour d'un tournoi.

Une hypothèse "métier" est qu'à partir du 4ème tour, les tes de series du tableau commencent à se rencontrer. par conséquent, l'écart de classement retrecit et la perf' devient plus probable. Par contre, après le 4ème tour, les favoris "protégés" par un début de tournoi a priori simple font respecter la logique du classement.

1.8.4 Méthode de construction et d'évaluation de modèles prédictifs

1) CONSTRUCTION DES QUALITATIVES DISJONCTIVEES

Ici encore, il est nécessaire de procéder à un encodage disjonctif complet (one hot encoding).

2) CONSTITUTION DU JEU DE TESTS

Suivant les méthodes, on séparera le jeu d'apprentissage et le jeu de tests : * dans le cas standard dans une proportion de 80% / 20% et aléatoirement * en utilisant la validation croisée à 10 passes (suffixe _CV dans le tableau ci dessous)

3) APPRENTISSAGE DES MODELES

4) PREDICTION SUR LES JEUX DE TESTS

5) EVALUATION

Les métriques utilisées sont : * précision * rappel * f1-score * ROC
On choisit de privilégier les scores précision et rappel par rapport à l'aire sous la courbe efficacité du récepteur (ROC). Si l'on se positionne dans l'optique d'un d'un pari faute de meilleur besoin, on souhaitera se focaliser sur minimiser les faux positifs par rapport à minimiser les faux négatifs. En effet, une victoire en "perf" aura sans doute une meilleure côte et rendra le pari intéressant. On préfèrera un vrai positif rare (gain important mais rare) plutôt que de nombreux faux postifs (nombreuses pertes, même faibles). Il sera donc moins grave de déconsidérer le rappel. On note tout de même que la classe positive est fréquente.

6) OPTIMISATION

Pour le meilleur algorithme, on fera une recherche des meilleurs hyperparamètres accompagnée de nouveau d'une k-fold cross validation. Ceci fait, on comparera présision et rappel et on essaiera de modifier le seuil de décision pour améliorer la précision.

7) RESULTATS

Les résultats des scores des différents algorithmes seront présentés dans le tableau ci dessous.
Tableau de synthèse des résultats des algorithmes supervisés sur les données de tennis ATP
Table 1. Scores par algorithmes du résultat du calcul de la 'perf'
méthode & algorithme/fonction de score
Precision
Rappel
F1-SCORE
ROC_AUC
Random Forest
0,661
0,983
0.791
0,501
Random Forest_CV
0.656

0,979
0,786
0,547
Adaptative boosting
0,485
1
0,795
0,5
Adaptative boosting_CV
0,657
1
0,793
0,551
Gradient boosting
0,660
0,999
0,795
0,499
Gradient boosting_CV
0,657
0,999
0,793
0,552
Gradient boosting_CV_GS
0,660
1
0,795
0,5
Support Vector Machine
0,660
1
0,795
0,5
Support Vector Machine_CV
0,657
1
0,793
0,504
Naïf bayesien
0,692
0,462
0,554
0,531
Naïf bayesien_CV
0,689
0,485
0,569
0,541

Regression logistique
0,660
1
0,795
0,5
Regression logistique_CV
0,657
1
0,793
0,545
K-plus proches voisins
0,668
0,794
0,726
0,513
K-plus proches voisins_CV
0,659
0,820
0,731
0,513

On a choisi de ne pas fournir : * de matrice de confusion car c'est redondant avec le score ROC * de courbe ROC car les résultats sont tous proche de 0,5, donc de la diagonale

Il faut, par contre, afficher les courbes précision en fonction du rappel d'une part pour essayer d'optimiser la précision, et la courbe précision fonction du seuil de décision d'autre part pour déduire l'impact sur le seuil de décision de l'algorithme.

CALCUL PRECISION RAPPEL ET SEUILS DE DECISION

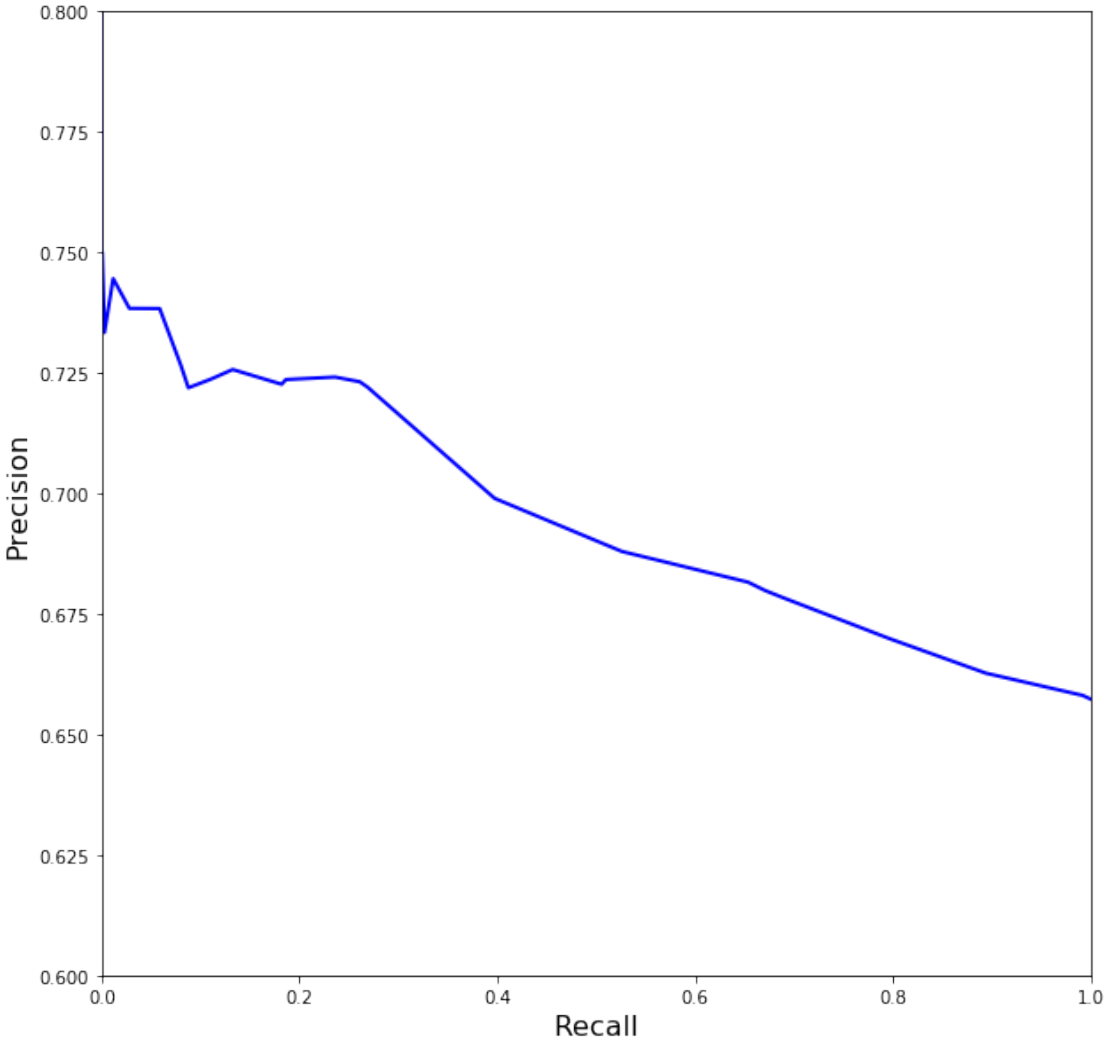
```
In [271]: modele_seuils= cross_val_predict(modele, x_train, y_train, cv=3,method="decision_fun
          modele_seuils

Out[271]: array([0.65040217, 0.72011473, 0.62898332, ..., 0.65057629, 0.65057629,
                 0.63433295])

In [272]: precisions, recalls, thresholds = precision_recall_curve(y_train, modele_seuils)
```

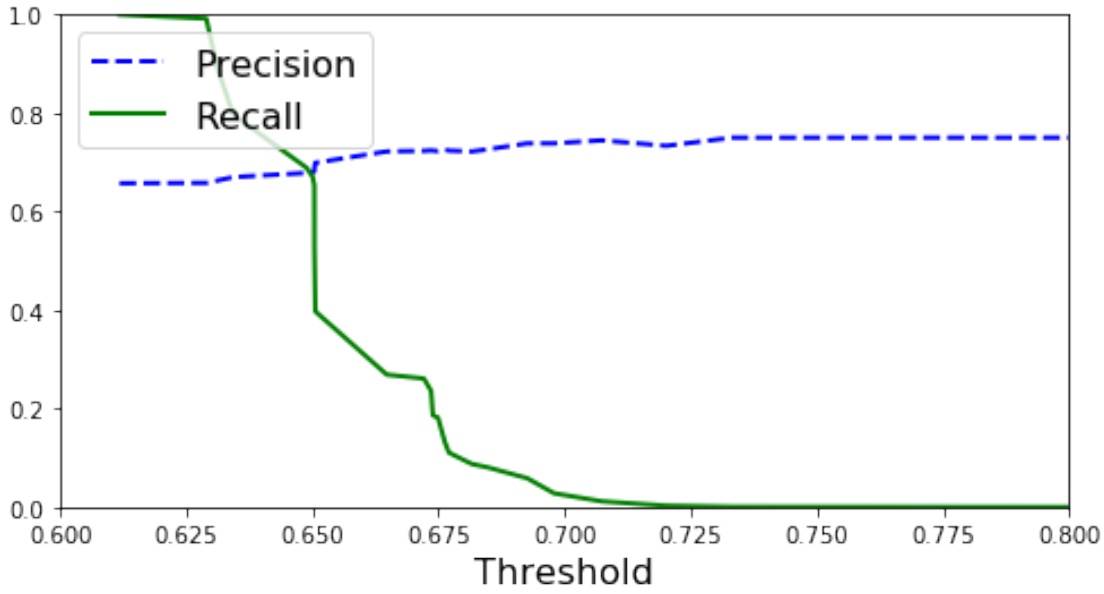
COURBE PRECISION FONCTION DU RAPPEL

```
In [284]: plt.figure(figsize=(10, 10))
          plot_precision_vs_recall(precisions, recalls)
          plt.ylim([0.6, 0.8])
          #save_fig("precision_vs_recall_plot")
          plt.show()
```



COURBE PRECISION ET COURBE RAPPEL EN FONCTION DU SEUIL

```
In [277]: plt.figure(figsize=(8, 4))
          plot_precision_recall_vs_threshold(precisions, recalls, thresholds)
          plt.xlim([0.6, 0.8])
          #save_fig("precision_recall_vs_threshold_plot")
          plt.show()
```



INTERPRETATION Si l'on souhaite augmenter la précision, c'est au prix d'une grande baisse du rappel ! Un "compromis" serait de faire baisser le rappel de **0,99** à…**0,25** pour augmenter la precision de **0,657** à **0,725** ! On comprend qu'il est très difficile d'améliorer ce modèle…ou du moins au détriment de sa qualité. La conséquence serait une modification du seuil de décision de l'arbre. En observant grossièrement les courbes, on déduit que pour obtenir une precision ~0,725 avec un rappel de ~0,25, il faut augmenter le seuil de décision à ~0,67. Un exemple de décision avec une perf' est donnée ci dessous.

```
In [297]: score=modele.decision_function(une_perf)
          score

Out[297]: array([0.65050326])

In [301]: seuil=0.60
          prediction_une_perf=(score>seuil)
          prediction_une_perf

Out[301]: array([ True])

In [303]: #precision ~0,725 avec un rappel de ~0,25 => seuil ~0,67
          seuil=0.67
          prediction_une_perf=(score>seuil)
          prediction_une_perf

Out[303]: array([False])
```

1.8.5 Conclusion de la deuxième partie

Malgré la mise en compétition de divers algorithmes, l'optimisation du jeu de données d'apprentissage, la recherche des meilleurs paramètres de l'algorithme et enfin la modification des résultats de ces derniers, la performance des modèles reste relativement faible : on peut taxer le meilleur des modèles proposés de *weak classifier*.

Cependant, si l'objectif metier n'est pas atteint, chacune des mises en oeuvre de bonnes pratiques méthodologiques a eu un effet positif sur la qualité du modèle.

1.8.6 Pistes d'améliorations

- Objectif métier
  - Prédire une "perf" dans un groupe de tournois découvert en première partie
- Méthodologie
  - Effectuer uen campagne de tests avec des données réelles à venir
  - Echantillonnage stratifié pour rééquilibrer :
    - le fait que les "perf" soient plus probables
    - que les mieux classés soient plus representés en terme de matches
- Algorithmes
  - Utiliser R pour bénéficier d'arbres de décision gérant les qualitatives nativement
- Variables
  - Considérer les dates exactes des matches.

1.9 ANNEXES

1.9.1 Références

Bibliographiques

- "ML avec scikit-learn" A.Geron Dunod
- "Data science par la pratique"
- "Python for data analysis"
- "Analyse de données avec R"
- GLMF HS nř94 "Machine learning"

Internet

- Tennis

http://www.fft.fr
https://fr.wikipedia.org/wiki/ATP_World_Tour
https://fr.wikipedia.org/wiki/Saison_2017_de_l%27ATP

Projet

- Dépôt git

https://gitlab.com/logrus_fr/CNAM-projets/

- notebooks

https://gitlab.com/logrus_fr/CNAM-projets/blob/master/STA211/atp-non_supervise.ipynb
https://gitlab.com/logrus_fr/CNAM-projets/blob/master/STA211/atp-supervise.ipynb