Aline Giordano
An Toàn Neyraud
ET4 Info

RAPPORT PROJET APPRENTISSAGE AUTOMATIQUE

Table des matières

Aline Giordano
An Toàn Neyraud
ET4 Info

Exploration des données

Pour ce projet nous avons choisi un dataset sur kaggel sur le thème des matchs NBA. Nous avions plusieurs choix de fichier csv possibles et les prendre tous nous aurait fait un trop grand nombre de données à traiter. Nous avons donc choisi de prendre celui qui répertoriait les statistiques des matchs (https://www.kaggle.com/nathanlauga/nba-games).

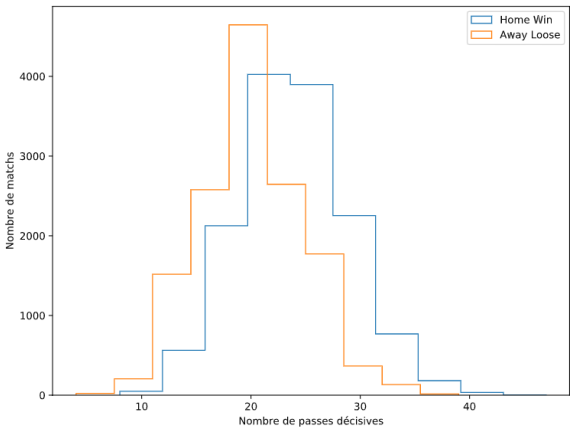
Le dataset original possède les informations suivantes :

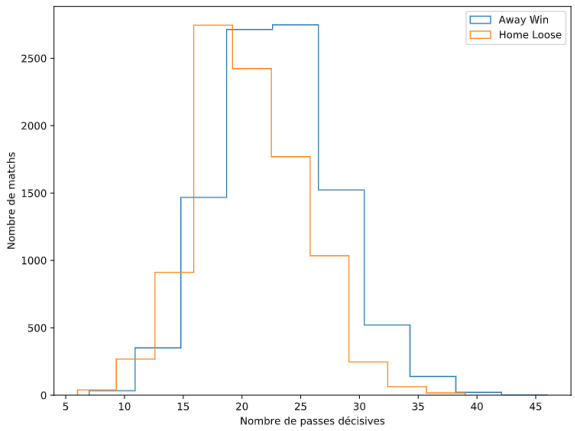
GAME_DATE_EST, GAME_ID, GAME_STATUS_TEXT, HOME_TEAM_ID, VISITOR_TEAM_ID, SEASON, TEAM_ID_home, PTS_home, FG_PCT_home, FT_PCT_home, AST_home, REB_home, TEAM_ID_away, PTS_away, FG_PCT_away, FT_PCT_away, FG3_PCT_away, AST_away, REB_away, HOME_TEAM_WINS.

Dans un premier temps, nous avons donc regardé les features disponibles et fait un premier tri en enlevant celles qui ne nous paraissaient pas pertinentes c'est-à-dire celles qui n'influent pas directement sur le résultat du match. Nous avons donc retiré les ID des équipes, la saison, la date du match, le statut du match ainsi que l'id du match. On obtient ainsi un dataset de 23520 lignes pour 12 features.

A partir de ce nouveau dataset, nous avons comparé les valeurs de différentes features en fonction de si le match est gagné ou perdu par les joueurs à domicile afin de déterminer la pertinence de celles-ci. Nous obtenons les graphes ci-après.

- Graphiques associés au nombre de passes décisives :

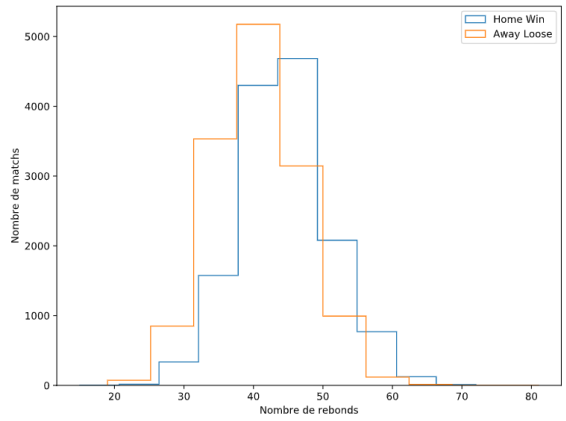


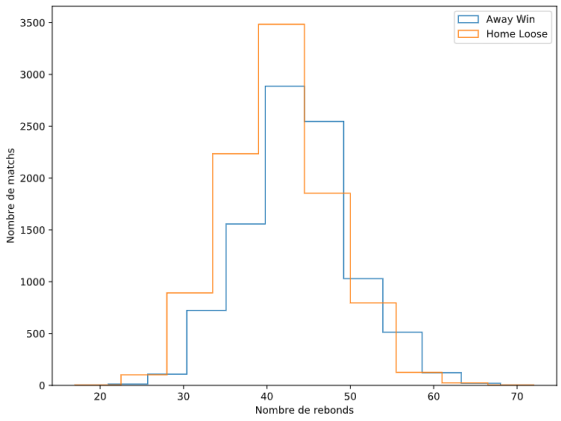


Observations : On remarque que lorsque home gagne les matchs, elle a fait plus de passes décisives que les visiteurs qui ont perdu. De même quand les visiteurs (away) gagnent, leur nombre de passes décisives par match est plus grand que celui des joueurs à domicile vaincus. On peut en conclure que les passes décisives sont importantes pour le résultat d'un match.

Aline Giordano
An Toàn Neyraud
ET4 Info

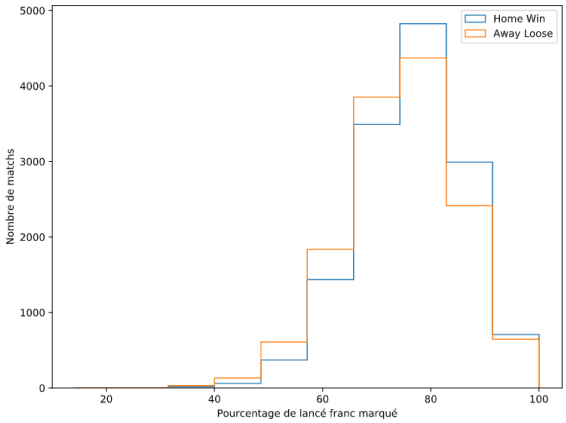
- Graphiques associés au nombre de rebonds par match :

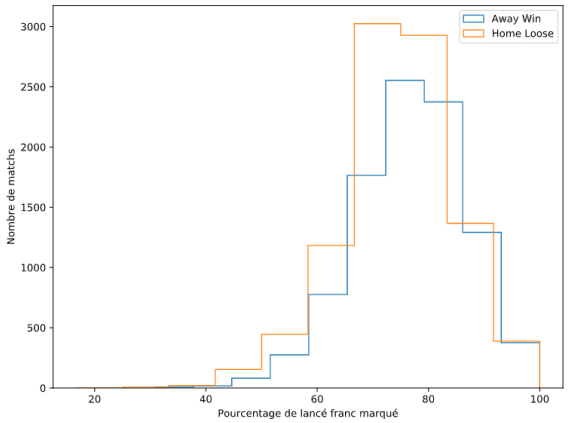




Observations : On remarque que les joueurs à domicile (home) récupèrent plus de rebonds que les visiteurs lors des matchs qu'ils gagnent. Et inversement, pour les visiteurs, lors d'un match gagné ils ont un nombre de rebonds plus élevé que les joueurs à domicile. On peut donc dire que le nombre de rebonds influence pas mal le résultat du match.

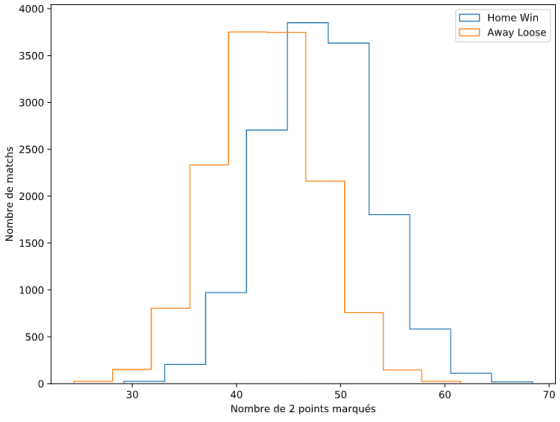
- Graphiques associés au nombre de lancers francs :

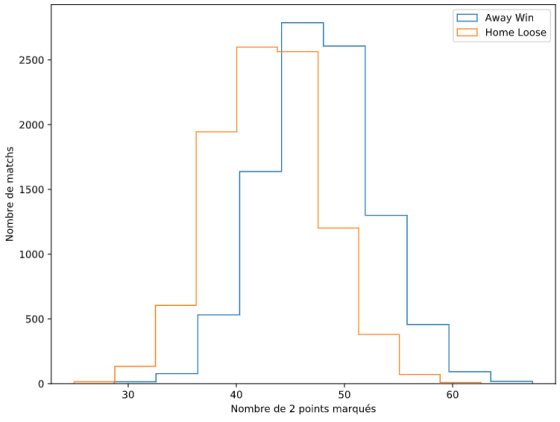




Observations : Comparé aux deux features précédentes, la réussite aux lancers francs ne joue pas un rôle déterminant pour l'issue du match. Cette feature ne sera pas forcément utile à garder pour la suite.

- Graphiques associés au nombre paniers à 2 points marqués par match :





Aline Giordano
An Toàn Neyraud
ET4 Info
Observations : Comme nous pouvions nous en douter, le nombre de paniers à 2 points marqués a de l'importance pour déterminer le vainqueur d'un match. Il en va relativement de même pour les paniers à 3 points.

Définition de la tâche

A partir des observations que nous avons faites, nous avons pu déterminer l'objectif de ce projet : prédire si les joueurs à domicile gagnent le match ou pas. Nous allons donc faire une classification binaire supervisée car nous avons déjà les résultats des matchs.

Pre-processing 1

Avant de réellement commencer à traiter notre dataset nous nous sommes assurés qu'il soit complet et correctement utilisable. Nous avons remarqué qu'une bonne quantité de données étaient NaN, pour pallier cela et sans pour autant trop 'fausser' les données nous avons normalisé X avec une fonction min-max. Puis, nous avons fait la moyenne des valeurs normalisées pour chacune des features afin de remplacer les valeurs Nan par des valeurs cohérentes.

De plus, nous avons choisi de ne pas prendre en compte les données sur le score final de chacune des équipes car cela nous a semblé trop faciliter la prédiction du résultat. De plus, d'après les observations que nous avons faites, nous avons aussi retiré les lancers francs. Nous nous retrouvons donc avec un dataset de 23520 lignes pour 8 colonnes.

On attribue les données à X et y, X va prendre toutes les colonnes sauf la dernière qui n'est autre que celle des résultats des matchs et qui va donc correspondre à y.

Optimisations

Architecture

Pour notre objectif de faire une classification binaire supervisée, nous avons choisi la méthode SVC et plus précisément la méthode SVM de manière linéaire. C'est un algorithme qui est efficace en dimension élevée.

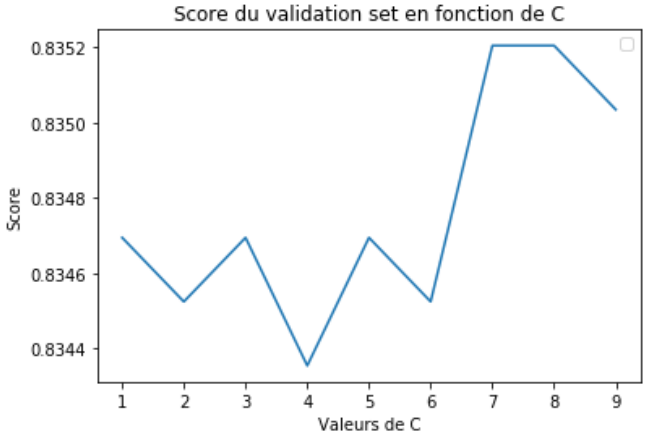
Choix d'hyper-paramètres

Pour l'optimisation des paramètres, nous avons choisi de faire une cross-validation en 4 parties, c'est-à-dire que le validation set représente 25% du dataset. Le reste sert pour l'entraînement.

Optimisation C :

Ensuite, nous l'avons fait pour plusieurs valeurs de C (paramètre de régularisation).

Aline Giordano
An Toàn Neyraud
ET4 Info



Ici, la valeur 1 est intéressante, jusqu'à 6 il n'y a pas d'augmentation. Mais la valeur 7 est aussi potentiellement intéressante, même si en observant l'échelle, la différence de score n'est pas grande, nous avons donc décidé de comparer la différence de performance entre C=1 et C=7.

```
start_time1 = time.time()
clf1 = svm.SVC(C = 1,kernel = 'linear')
scores = cross_val_score(clf1, X_normalized, newY, cv=4)
print("Validation score : ", scores[3], "& calcul en --- %s seconds ---" % (time.time() - start_time1))
```

```
Validation score :  0.8346938775510204 & calcul en --- 7.956667184829712 seconds ---
```

```
start_time2 = time.time()
clf2 = svm.SVC(C = 7,kernel = 'linear')
scores = cross_val_score(clf2, X_normalized, newY, cv=4)
print("Validation score : ", scores[3], "& calcul en --- %s seconds ---" % (time.time() - start_time2))
```

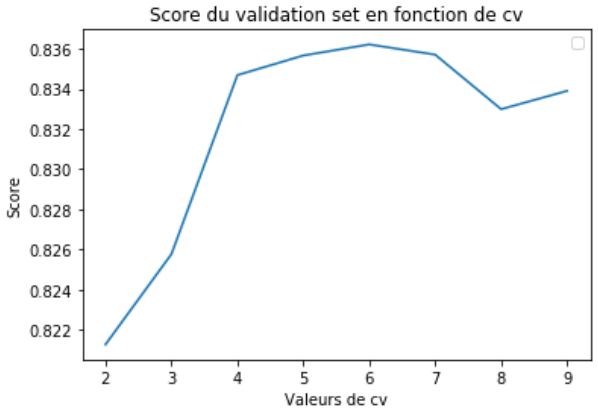
```
Validation score :  0.835204081632653 & calcul en --- 8.668288707733154 seconds ---
```

De cette façon, on observe que pour C=7 le gain de score (0,0007) n'est pas très élevé comme prévu. De même, on remarque que le temps augmente de 0,71 secondes.

Nous avons choisi la valeur 1 pour C car le gain de score n'est pas assez élevé pour le temps d'exécution en plus.

Optimisation nv:

Puis, nous avons voulu tester la cross validation pour différente valeur de cv. Cv est le paramètre permettant de définir la taille des segments de données réservés pour les sets d'entrainements et celui de validation.



On observe sur le graphique que 4 est la valeur qui permet le meilleur compromis entre le score et l'efficacité du code (Plus cv est grand, plus nombreux sont les tableaux de données). Nous avons donc choisis 4.

Avec les valeurs de ces hyper-paramètres, nous obtenons un score de 83,5%

Ensuite, nous avons voulu comparer la précision de la cross-validation. On utilise la méthode train_test_split, avec encore un validation set de 25%.

On obtient un score de 84%.

On en conclut que la cross-validation nous permet d'avoir des valeurs plus précises en faisant varier la portion choisie pour le validation set mais ne nous apportent pas d'avantages considérables sur le score en ce qui concerne ce data set.

Pre-processing 2

Nous nous sommes par la suite demandé, si en se basant uniquement sur les statistiques de matchs des équipes qui jouent à domicile, nous pouvions réussir à prédire l'issue du match. Pour cela, nous avons pris en compte toutes les features liées à l'équipe home. Nous avons donc maintenant un dataset de 23520 lignes pour 6 colonnes. Nous avons normalisé les données de la même façon qu'au pre-processing précédent.

Nous avons de nouveau fait une cross-validation avec toujours 4 parties et donc un validation set qui représente 25% du dataset. Avec cette méthode et ces features on obtient un score de : 0,74.

Ce résultat nous permet de voir l'importance d'avoir aussi des informations sur les statistiques des équipes visiteurs car en effet, la prédiction des vainqueurs est moins bonne avec seulement les données sur l'équipe à domicile. Cette observation est cohérente avec la réalité car il se peut que dans un match le score final ne soit pas trop élevé mais que l'équipe à domicile gagne tout de même. Ainsi notre modèle aurait pu prédire leur défaite au vu des statistiques pas très bonnes de l'équipe et cela car il n'avait aucune information sur la performance de l'équipe visiteur.

Ouverture/Conclusion

Pour conclure, les résultats obtenus sont cohérents et bons. Nous aurions pu nous attendre à un score plus élevé au vu de la quantité de données que nous avions.

Afin d'augmenter le score, nous aurions pu prendre des données encore plus précises qui donne par exemple le pourcentage de victoire à domicile/extérieur pour chacune de équipes. Davantage de données nous aurait permis d'affiner notre prédiction.