Ordonner pour personnaliser : cas d'usage dans le monde du pari sportif

Paul Steffen [1] & Bertrand Beaufils [2]

[1] *Betclic Group, p.steffen@betclicgroup.com*
[2] *Betclic Group, b.beaufils@betclicgroup.com*

Résumé. Betclic est un des leaders européens des paris sportifs et des jeux en ligne. Proposer une expérience utilisateur fluide et adaptée à chacun de ses utilisateurs est un défi majeur, notamment pour une application proposant des paris sur plus de 450 évènements sportifs par jour répartis sur près de 50 sports différents. Face à ce challenge, la personnalisation de la page d'accueil de l'application est un enjeu réel, pouvant être solutionné à l'aide d'une modélisation statistique visant à filtrer puis ordonner les évènements sportifs de manière pertinente.
Afin d'étudier la prise de paris de nos utilisateurs sur notre page d'accueil, une base de données de plusieurs milliards d'observations relevées sur 2 ans a été construite. Cette dernière a permis l'entraînement d'un système de recommandation hybride utilisant une factorisation matricielle afin de scorer l'ensemble des éventuelles interactions utilisateur-match, et de proposer un ordonnancement pertinent pour chacun de ces utilisateurs.
Après une présentation de la métrique retenue pour évaluer ce modèle ainsi que du design expérimental choisi pour représenter au mieux la qualité prédictive du modèle une fois mis en production, les alternatives permettant d'éventuellement améliorer cette méthode de classement seront évoquées.

Mots-clés. méthode de classement, système de recommandation, pari sportif, factorisation matricielle

Abstract. Betclic is one of Europe's leading sports betting and online gaming companies. Providing a smooth and tailored user experience to each of its users is a major challenge, particularly for an application offering bets on more than 450 sporting events a day across almost 50 different sports. To meet this challenge, personalising the application's home page is a real issue, which can be solved with the help of statistical modelling aimed at filtering and ordering sporting events in a relevant way on this page.
In order to study how our users take bets on our home page, a database of several billion observations over 2 years was built and used to train a hybrid recommender system using matrix factorisation to score all possible user-game interactions, and to propose a relevant order for each user.
After a presentation of the metrics used to evaluate this model and the experimental design chosen to best represent the predictive quality of the model once it is in production, we will look at alternatives for improving this ranking method.

Keywords. ranking method, recommender system, sports bet, matrix factorization

1 La problématique

La personnalisation de la page d'accueil de notre application revient à un problème de classement des différents évènements sportifs disponibles au moment de la consultation de cette page. Ainsi, un premier filtre concernant la date des matchs permet de ne considérer que les évènements non terminés, éligibles au pari sportif. Un second, basé sur une analyse mettant en avant l'intérêt de nos utilisateurs pour les matchs ayant lieu dans un futur proche, a été défini afin de ne conserver que les matchs débutant dans les 7 jours à venir.

2 Les données

Avec plus de 2 milliards de paris sportifs uniquement sur les années 2022 et 2023, réalisés par plus de 3,8 millions d'utilisateurs différents, le jeu de données à disposition pour répondre à ce problème est suffisamment important et nécessite même une attention particulière quant aux méthodes utilisées afin de rendre les traitements réalisables sous contrainte de capacité de calcul et de temps.
L'ensemble de ces interactions entre utilisateurs et rencontres sportives (ou *items*) peut être caractérisé à l'aide de différentes variables comme le sport, la compétition et les opposants de la rencontre ou encore la différence temporelle entre le début de la rencontre et le moment où le pari a été effectué. Ainsi, en simplifiant au cas où uniquement 2 opposants se rencontrent lors d'un évènement sportif, on peut obtenir le diagramme de base de données suivant :

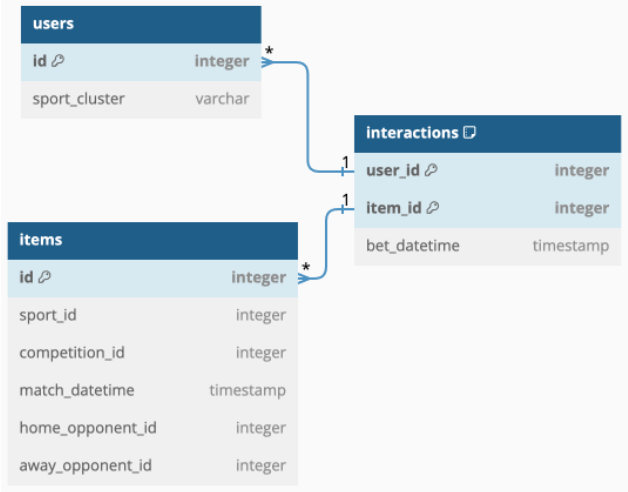

Figure 1: Diagramme de base de données

Nous pouvons alors comprendre que ces données ne représentent que l'observation du comportement de nos utilisateurs. Contrairement aux interactions explicites, provenant d'un retour de la part des utilisateurs et caractérisant à quel point ce dernier a pu apprécier ou non l'objet de la recommandation, nous n'utilisons ici que des interactions implicites. Ne nécessitant aucun retour d'information de l'utilisateur, ce dernier type d'interactions a

l'avantage d'être plus fréquent, bien qu'il ne précise pas l'appréciation ou non de l'utilisateur, en considérant chaque interaction comme positive.
Avec des pics horaires pouvant atteindre plus de 550 000 paris, une quantité importante de données informatives peut être ajoutée en très peu de temps. Les nouveaux utilisateurs s'inscrivant chaque jour et le catalogue de matchs évolutif, imposent que l'ensemble de données utilisé doit régulièrement être mis à jour. Enfin, la particularité d'intégrer régulièrement de nouveaux évènements jusqu'alors inconnus (comme avec le tennis, sport pour lequel il est impossible de connaître les caractéristiques exactes des matchs du tour suivant, sans que les matchs du tour actuel ne soient terminés) et de rendre d'autres matchs inéligibles une fois arrivés à leur terme, implique un contrôle de la fraîcheur des données important.

3 Modélisation statistique

3.1 Système de recommandation hybride et classement

Face à cette problématique, et avec le jeu de données à notre disposition, un premier choix fut d'utiliser un système de recommandation hybride, combinant le meilleur du filtrage collaboratif et du filtrage basé sur le contenu. Contrairement au premier type de filtrage, utilisant la similarité des utilisateurs basée sur leur historique d'interactions, un système hybride permet d'apporter plus de diversité que ce type de recommandation du fait qu'il ne soit pas limité à recommander des matchs populaires auprès d'utilisateurs similaires. À l'inverse d'un filtrage basé sur le contenu, un système hybride rend possible une certaine sérendipité[1] et évite un effet de bulle de recommandation, en proposant du contenu différent de l'historique d'un utilisateur.
Ainsi, le framework *LightFM* de Kula (2015), implémente un modèle hybride de factorisation de matrices représentant chaque utilisateur et chaque match comme une représentation linéaire de facteurs latents de leurs caractéristiques. Ainsi, pour notre cas, n'ayant pas identifié de caractéristique suffisamment pertinente pour cette problématique, un utilisateur $u \subset U$ n'a été représenté que par un facteur latent, propre à chaque utilisateur : q_u de dimension 30 et un biais b_u.
Les matchs $i \subset I$, décrits par des caractéristiques $c_i \subset F^I$ beaucoup plus informatives dans ce problème de recommandation, ont été représentés par un facteur latent composé des *embedding* [2] e_f^I, eux aussi de dimension 30 pour chaque caractéristique c:

$$p_i = \sum_{j \in c_i} e_j^I$$

et un biais :

$$b_i = \sum_{j \in c_i} b_j^I$$

[1] sérendipité : fait de faire par hasard une découverte inattendue qui s'avère ensuite fructueuse
[2] embeddings: représentation numérique d'une caractéristique

La dimension de ces *embeddings* n'a actuellement pas été ajustée afin d'optimiser la performance du modèle et a été laissée par défaut comme conseillé lors d'une première approche. Ainsi, le score d'une interaction entre un utilisateur u et un match i est obtenu par le produit de leur représentation, ajusté par leurs biais respectifs :

$$\hat{r}_{ui} = f(q_u \cdot p_i + b_u + b_i)$$

où $f(\cdot)$ est la fonction sigmoïde : $f(x) = \frac{1}{1+exp(-x)}$ et l'optimisation du modèle s'effectue par maximum de vraisemblance :

$$L(e^U, e^I, b^U, b^I) = \prod_{(u,i)\in S^+} \hat{r}_{ui} \times \prod_{(u,i)\in S^-} (1 - \hat{r}_{ui})$$

avec S^+ et S^- représentant respectivement les interactions positives (paris placés sur le match) et négatives (pari non placé sur un match disponible) entre les utilisateurs et les matchs.
À l'aide de ces représentations latentes des matchs, le problème de démarrage à froid (caractérisant le manque de connaissance d'un nouvel arrivant dans le système de recommandation) des nouveaux matchs du catalogue a ainsi pu être écarté, du fait que le sport, la compétition, et les équipes caractérisant ces matchs soient déjà connus par le modèle.

Avec, en moyenne, plus de 250 000 utilisateurs journaliers différents, proposer un contenu spécifique à chacun d'entre eux représente un réel défi technique pour la fluidité de notre application. C'est pourquoi, les utilisateurs ont été regroupés par une suite de règles déterminées à l'aide de leur historique de paris, réduisant fortement le volume et la complexité de la tâche. Ainsi, il a été possible d'obtenir le classement médian de chacun des matchs, pour un groupe d'utilisateurs similaires, et de proposer un unique classement pour ce même groupe.

3.2 Méthode d'évaluation et résultats

Afin d'évaluer la pertinence de notre modèle prédictif, il a été nécessaire de choisir une métrique d'évaluation adaptée à ce problème de classement. Du fait qu'il n'y ait généralement que quelques matchs pertinents pour un utilisateur dans un catalogue disponible pouvant atteindre près de 500 matchs et qu'il est préférable que ces derniers soient mis en avant, le rappel à 5 a été une métrique privilégiée afin de répondre au mieux à cet objectif.

$$Rappel@5 = \frac{nombre\ d'éléments\ pertinents\ dans\ le\ top\ 5}{nombre\ total\ d'éléments\ pertinents}$$

Cependant, le comportement des utilisateurs peut être très varié. Certains parieurs peuvent atteindre un grand nombre d'interactions à l'aide de paris combinés. Ainsi, pour rééchelonner cette métrique et la rendre comparable entre les différents utilisateurs, le dénominateur a été borné à 5. On obtient alors pour notre cas d'usage la formule suivante :

$$Hit@5 = \frac{nombre\ de\ matchs\ pariés\ dans\ le\ top\ 5}{min(nombre\ total\ de\ matchs\ pariés, 5)}$$

Enfin, pour s'assurer du caractère de généralisation du modèle étudié, une validation croisée temporelle a été choisie. Le modèle est alors entraîné avant de faire les prédictions. De cette façon, le modèle utilise toutes les informations disponibles.
Il s'agit d'une variante de la validation croisée standard mais, au lieu de procéder à une distribution aléatoire des observations, l'ensemble d'entraînement est augmenté de manière séquentielle, en conservant l'ordre temporel des données.
Dans l'optique d'optimiser la performance du modèle, la fraîcheur des données joue un rôle essentiel. Ainsi, filtrer l'ensemble des matchs terminés et inclure les nouveaux matchs disponibles toutes les heures, amène à recalculer un classement toutes les heures et permet d'obtenir un résultat au plus proche de son utilisation en production.
En agrégeant les $Hit@5$ sur un ensemble de périodes temporelles de l'année 2023, permettant de représenter au mieux la diversité que peut connaître le calendrier sportif sur une année, un $Hit@5$ d'environ 0,22 a été observé. Cette performance ne peut malheureusement pas être comparée à d'autres cas d'usage similaires dans le secteur d'activité par manque de disponibilité de l'information.

4 Conclusion

Bien que ces résultats soient encourageants, ils ne permettent pas à l'heure actuelle d'utiliser un système de recommandation hybride sur notre page d'accueil. Un modèle analytique, calculant un score pour chaque groupe d'utilisateurs basé sur le nombre d'interactions observé par match permet à l'heure actuelle d'obtenir de meilleurs résultats selon la méthodologie d'évaluation précédemment décrite. Un réglage plus fin du système de recommandation laisse penser à de possibles meilleurs résultats dans le futur. Enfin, l'utilisation d'algorithmes de classement tel que *LambdaMART* ou de systèmes de recommandation utilisant des réseaux de neurones comme *DLRM* permettent d'envisager des alternatives face à ce problème d'ordonnancement.

Bibliographie

Maciej Kula (2015), Metadata Embeddings for User and Item Cold-start recommendations, *Proceedings of the 2nd Workshop on New Trends on Content-Based Recommender Systems co-located with 9th ACM Conference on Recommender Systems (RecSys 2015), Vienna, Austria, September 16-20, 2015.*

Burges, Christopher J. C. (2010), *From RankNet to LambdaRank to LambdaMART: An Overview.*

Naumov, Maxim and Mudigere, Dheevatsa and Shi, Hao-Jun and Huang, Jianyu and Sundaraman, Narayanan and Park, Jongsoo and Wang, Xiaodong and Gupta, Udit and Wu, Carole-Jean and Azzolini, Alisson and Dzhulgakov, Dmytro and Mallevich, Andrey and Cherniavskii, Ilia and Lu, Yinghai and Krishnamoorthi, Raghuraman and Yu, Ansha and Kondratenko, Volodymyr and Pereira, Stephanie and Chen, Xianjie and Smelyanskiy, Misha

(2019), *Deep Learning Recommendation Model for Personalization and Recommendation Systems.*