Apprendre à enchérir : prédiction d'évènement rare et choix de stratégie

Slimane Makhlouf

HESAM UNIVERSITÉ

le cnam

ÉCOLE DOCTORALE Sciences des Métiers de l'Ingénieur

CEDRIC

THÈSE

présentée par : **Slimane Makhlouf**
soutenue le : **19 avril 2023**

pour obtenir le grade de : **Docteur d'HESAM Université**

préparée au : **Conservatoire national des arts et métiers**

Discipline : **Section CNU 27**

Spécialité : **Informatique**

Apprendre à enchérir : prédiction d'évènements rares et choix de stratégie.
Learning to bid : rare events prediction and auction policy study.

THÈSE dirigée par :
M. BAR-HEN Avner Professeur, Conservatoire national des arts et métiers
et co-encadrée par :
M. JOLLOIS François-Xavier Maître de conférences, Université Paris Cité

Jury

Mme. Samia BOUZEFRANE	Professeure, Conservatoire national des arts et métiers	Présidente
M. Cyril DE RUNZ	Maître de conférences, Université de Tours	Rapporteur
Mme. Fatma BOUALI	Professeure, Université de Lille	Rapporteure
M. Catherine FARON	Professeure, Université Côte d'Azur	Examinatrice
M. Nicolas TRAVERS	Maître de conférences, École supérieure d'ingénieurs Léonard-de-Vinci	Examinateur
M. Avner BAR-HEN	Professeur, Conservatoire national des arts et métiers	Directeur
M. François-Xavier JOLLOIS	Maître de conférences, Université Paris Cité	Directeur

THÈSE

Affidavit

Je soussigné, Slimane Makhlouf, déclare par la présente que le travail présenté dans ce manuscrit est mon propre travail, réalisé sous la direction scientifique de Avner BAR-HEN (directeur) et de François-Xavier JOLLOIS (co-directeur), dans le respect des principes d'honnêteté, d'intégrité et de responsabilité inhérents à la mission de recherche. Les travaux de recherche et la rédaction de ce manuscrit ont été réalisés dans le respect de la charte nationale de déontologie des métiers de la recherche. Ce travail n'a pas été précédemment soumis en France ou à l'étranger dans une version identique ou similaire à un organisme examinateur.

Fait à Paris, le 30 janvier 2023

Slimane Makhlouf

Affidavit

I, undersigned, Slimane Makhlouf, hereby declare that the work presented in this manuscript is my own work, carried out under the scientific direction of Avner BAR-HEN (thesis director) and of François-Xavier JOLLOIS (co-thesis director), in accordance with the principles of honesty, integrity and responsibility inherent to the research mission. The research work and the writing of this manuscript have been carried out in compliance with the French charter for Research Integrity. This work has not been submitted previously either in France or abroad in the same or in a similar version to any other examination body.

Place Paris, date 30th january 2023

Slimane Makhlouf

Remerciements

Je tiens particulièrement à remercier Avner Bar-Hen et François-Xavier Jollois, mes deux tuteurs pour leur engagement et leur aide qu'il m'ont apporté tout au long de ce projet. Ils ont su me soutenir et me pousser à aller plus loin dans mes réflexions mais aussi à me remotiver et à reprendre confiance dans les moments de doutes.

Je remercie également Abdoul Gambo et François Le Corre qui sont à l'origine de ce projet au sein de la société Velvet Consulting et qui m'ont offert cette belle opportunité. Je tiens à Anne-Charlotte Argand pour sa bonne humeur et son aide concernant les aspects administratifs parfois lourds à gérer. Plus généralement, je remercie toutes les personnes au sein de Velvet Consulting qui ont su être présent et qui m'ont apporté leur soutien quotidien et leur bonne humeur.

Je tiens à remercier ma famille pour leur soutien ainsi tous mes amis pour les moments de détente et de joie sans lesquels ce travail aurait été bien plus difficile à porter.

Résumé

Le marché de l'affichage publicitaire en ligne est en très forte croissance et est en passe de devenir le canal de diffusion le plus important en termes de valeur. Parmi les mécanismes de diffusion publicitaire sur internet, le RTB (Real-Time Bidding, enchères en temps réel) est celui qui reçoit le plus grand intérêt. Cette méthode permet d'automatiser les achats et ventes d'encarts publicitaires entre les sites web et les annonceurs au moyen d'un mécanisme d'enchères. Cela permet un affichage individuel d'une publicité à un visiteur et donc un ciblage fin, expliquant la grande popularité de cette approche auprès des annonceurs.

Le mécanisme du RTB repose sur la mise aux enchères des encarts publicitaires disponibles sur les pages web. Ces enchères sont organisées de manière standardisée pendant le chargement de la page du visiteur qui se connecte. Des algorithmes d'enchères spécialement développés sont chargés de placer les ordres afin de garantir aux annonceurs le meilleur revenu possible.

Nous présentons dans ce document l'ensemble de nos travaux portant sur l'étude et l'amélioration des approches d'enchères en temps réel effectués durant ces trois ans de thèse doctorale. Le problème du RTB (coté annonceur) consiste en une maximisation sous contrainte : on souhaite développer un algorithme permettant de maximiser le nombre de clics obtenus durant la campagne d'affichage sous la contrainte d'un budget limité. Ces travaux nous ont amenés à considérer le problème à travers deux grandes questions : la prédiction de la probabilité de clic permettant d'obtenir une estimation de la valeur d'un affichage, et l'optimisation des enchères qui doit, en se basant sur cette estimation, gérer les montants des ordres et le budget afin d'obtenir le maximum de clics possibles.

La prédiction de la probabilité joue ainsi un rôle crucial : celui de permettre l'estimation de l'utilité d'une impression. Avec environ seulement une publicité cliquée pour mille affichages, ce problème de classification binaire tombe dans le domaine de la prédiction d'événements rares. La prédiction de

ce type d'événements requiert l'utilisation de modèles et de fonctions d'évaluation spécifiques. Nous présentons dans ce sens, une étude sur les performances et les biais de modèles de classification classiques et nous explorons les moyens de réduire ces biais. Pour cela, nous comparons les performances de prédiction de la probabilité de clic de trois modèles : la régression logistique classique, la régression logistique pondérée pour les événements rares ainsi qu'un modèle d'apprentissage profond faisant référence dans la communauté de recherche sur la prédiction du clic. Nous étudions ces performances sous plusieurs fonctions d'évaluation nous permettant de montrer certains biais induits par les mesures de performances classiques. Nous présentons une mesure de performance spécifique au RTB permettant de corriger ces biais mais également de donner des indications sur la rentabilité des campagnes d'affichage afin d'aider à la décision.

En utilisant ces travaux sur la prédiction de la probabilité de clic, nous étudions l'optimisation des campagnes d'enchères en temps réel. Nous formulons ce problème sous la forme d'un processus de décision markovien nous permettant de développer plusieurs stratégies d'enchères plus ou moins élaborées : la stratégie naïve d'enchères constantes, la stratégie d'enchère linéaire consistant à enchérir proportionnellement à la probabilité de clic et sa variante, la stratégie d'enchères linéaires avec contrôle du rythme de dépense. Nous ajoutons à cela, une stratégie d'apprentissage profond par renforcement permettant, en théorie, l'apprentissage d'une stratégie d'enchère dynamique, s'adaptant aux conditions de la campagne en continu. Nous étudions les performances de ces stratégies sur un jeu de données de référence et montrons que malgré sa grande popularité dans la communauté de recherche, l'apprentissage par renforcement n'apporte pas d'amélioration significative par rapport aux autres approches entre autres à cause des problèmes de convergence de ce type d'approche notamment dues à la formulation des états du processus de décision markovien.

Nous présentons une étude sur la convergence de l'apprentissage par renforcement et l'apprentissage de la formulation des états en utilisant un jeu de casse-brique comme une simulation simplifiée du RTB. Nous explorons l'utilisation d'autoencoders afin de synthétiser une formulation des états qui permettrait une meilleure convergence de l'apprentissage par renforcement.

Mots-clés : Enchères en temps réel, Prédiction du clic, stratégie d'enchères, Prédiction d'événements rares, Apprentissage par renforcement, apprentissage de la représentation automatique.

Abstract

The online display advertising market is growing rapidly and is becoming the most important distribution channel in terms of value. Among the advertising mechanisms on the Internet, Real-Time Bidding (RTB) is the most widely used. This method automates the buying and selling of advertisements between websites and advertisers through an auction mechanism. This allows for individual display of advertisement to visitors and thus a fine targeting, explaining the great popularity of this approach.

The RTB mechanism is based on the auctioning of available ad slots on the web pages. These auctions are organized in a standardized way during the loading of the web page. Bidding algorithms are responsible for placing the bids in order to guarantee the advertisers the best possible revenue.

In this document, we present our work on the study and improvement of real-time bidding approaches carried out during the three years of this Ph.D. The RTB problem (on the advertiser side) consists in a constrained optimization problem : we want to develop an algorithm that maximizes the number of clicks obtained during the display advertising campaign under the constraint of a limited budget. This work led us to consider the problem through two main issues : the prediction of the clicks probability to obtain an estimate of the value of a given ad slot, and the optimization of the bidding campagn which, based on this estimate, should regulate the bids and the budget in order to maximize the number of clicks.

Click probability prediction hence plays a crucial role in enabling the utility estimation of an impression. With only about one clicked ad per thousand impressions, this binary classification problem falls into the rare event prediction domain. The prediction of such events requires the use of specific models and evaluation functions. We thus present a study on the performances and biases of classical classification models and we explore ways of reducing these biases. To this end, we compare the

performance of three models : classical logistic regression, weighted logistic regression for rare events, and a reference deep learning model. We study these performances under several evaluation functions allowing us to show some biases induced by classical performance measures. We present a performance measure specific to RTB to correct these biases but also to give indications on the profitability of ad display campaigns in order to help decision making.

Using this work on click probability prediction, we study the optimization of real-time bidding campaigns. We formulate this problem as a Markov decision process and develop several bidding strategies : the naive constant bidding strategy, the linear bidding strategy consisting in bidding proportionally to the click probability and its variant, the linear bidding with budget pacing. We also study a deep reinforcement learning strategy theoretically enabling to learn a dynamic bidding strategy, adapting to the conditions of the campaign continuously. We study the performance of these strategies on a benchmark dataset and show that despite its great popularity in the RTB research community, reinforcement learning does not bring significant improvement compared to other approaches, amongst others because of the convergence and stability issues of this type of approach, notably due to the formulation of the states of the Markovian decision process.

We finally present a study on the convergence of reinforcement learning and state formulation learning using a game as a simplified simulation of RTB. We explore the use of autoencoders to learn a state formulation that would allow better convergence of reinforcement learning.

Keywords : real-time auctions, click prediction, bidding strategy, rare event prediction, reinforcement learning, state representation learning.

Table des matières

Liste des tableaux

Table des figures

Chapitre 1

Introduction

Contenu

Depuis l'avènement de l'ère Internet dans les années 2000, les campagnes publicitaires digitales et leurs enjeux économiques ne cessent de prendre de l'importance. Le développement des technologies de *Real-Time Bidding* (Enchères en temps réel) depuis 2009 a révolutionné ce secteur en apportant aux régies publicitaires une grande précision quant au ciblage des audiences. Ce ciblage fin et individuel offert par le *RTB* en fait un canal très attractif pour les annonceurs. D'après une étude [1], ce marché pesait en 2019 environ 6.6 milliards d'euros et pourrait d'après les projections des auteurs, connaître sur la période 2019 - 2024 un taux de croissance annuel moyen ($CAGR$) de 32.9% pour atteindre un poids de 27.2 milliards d'euros.

Le RTB automatise les processus permettant la vente, l'achat et l'affichage des publicités en ligne à travers des algorithmes développés pour proposer les meilleurs revenus d'affichage aux éditeurs. Cela permet aussi aux annonceurs d'adresser leurs publicités à l'audience souhaitée avec une très fine granularité et au meilleur prix. Pour ce faire, au moment de la connexion d'un utilisateur sur une page web, chaque emplacement disponible sur cette page est proposé sous la forme d'une requête d'enchère (bid request) sur une plateforme d'échange (AdExchange, Adx) où les algorithmes de RTB placent leurs ordres et qui permet au gagnant d'afficher sa publicité. Ce processus se déroulant durant le chargement de la page, il est critique que celui-ci ne dégrade pas l'expérience utilisateur et ne doit pas dépasser les 100 millisecondes. Il est donc important de limiter la complexité des frameworks, surtout depuis que les entreprises tendent à développer leurs propres DSP en interne [Interactive Advertising Bureau (IAB), 2018] et peuvent disposer des ressources de calcul limitées. Nous décrivons dans cette section l'ensemble des mécanismes composant le RTB.

La *Supply-side platform (SSP)* correspond à la partie vente d'emplacements : les sites Web ou les intermédiaires qui vendent des emplacements pour le compte des sites Web. L'objectif de ce côté est de tirer le maximum de l'inventaire qu'ils ont à vendre.

La *Demand-side platform (DSP)* représente le côté acheteur. Les annonceurs cherchent à maximiser l'efficacité de leur campagne publicitaire en limitant les coûts tout en maximisant un indicateur de performance (KPI) qui est dans la plupart des cas soit les impressions soit les clics et plus rarement les conversions c'est à dire l'acte d'achat.

La *Ad-Exchange platform (AdX)* est la partie centrale de l'architecture RTB. Elle présente et met aux enchères les offres d'impression des SSP pour que les DSP les achètent.

1. `https://www.researchandmarkets.com/research/ndpb8n/worldwide?w=5`

Ces trois parties communiquent entre elles au cours du processus RTB de manière standardisée. Le processus, dans sa version simplifiée est présenté sur le diagramme 1.1.

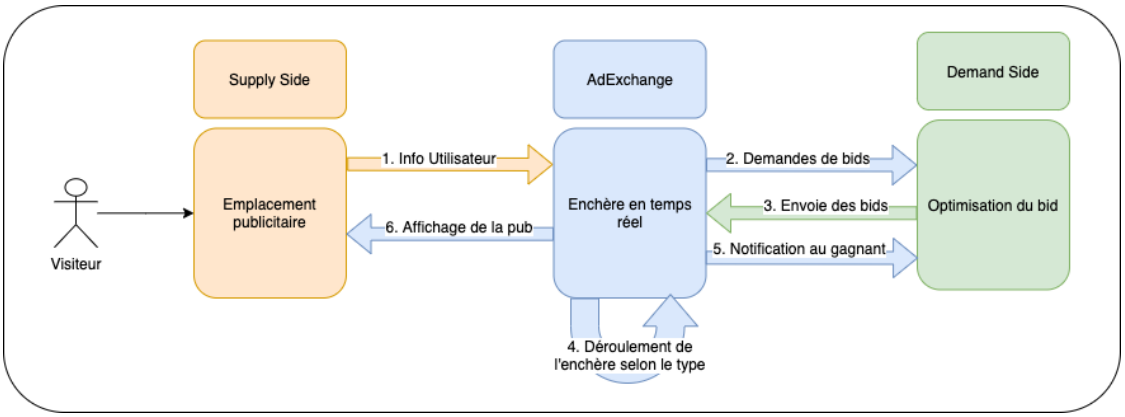


FIGURE 1.1 – Diagramme de fonctionnement général du Real-Time Bidding

Lorsque qu'un utilisateur accède à un site sur lequel des emplacements publicitaires sont présents, le mécanisme d'enchères se met en marche :

1. Les informations de connexion du visiteur (Heure de la visite, continent, adresse IP, appareil et navigateur utilisé, etc....) ainsi que des informations concernant l'emplacement en lui-même (taille, position, etc...) sont transmises à la plateforme d'échange.
2. Ces informations sont envoyées à tous les annonceurs enregistrés sur la plateforme d'échange sous la forme de demandes d'enchères (*Bid Request*).
3. Compte tenu des informations reçues, les annonceurs vont calculer la valeur qu'ils sont prêts à payer afin de faire afficher leur publicité au moyen d'algorithmes plus ou moins complexes. Les données reçues peuvent aussi être complétées par des données personnelles achetées à une DMP (*Data Management Plateform*). Ces enchères (*bids*) sont renvoyées à la AdExchange.
4. La plateforme ayant reçu un certain nombre de bids, décernera la victoire à un participant selon le mode d'enchères prédéfini (First ou second price auction, c.f. détails ci-après).
5. La notification de victoire ainsi que le prix à payer (*prix du marché*, *market price*) sont envoyés au gagnant de l'enchère, qui enverra en retour la publicité à afficher.
6. Enfin, la publicité est affichée au visiteur sur l'emplacement ainsi vendu. Si un clic est enregistré sur cette pub, l'information est remontée à l'annonceur à des fins de suivi des performances de la campagne d'affichage et d'amélioration des algorithmes.

1.1 Théorie des Enchères pour le Real-Time Bidding

Les processus d'enchères sont des mécanismes millénaires permettant l'achat et la vente d'objets par la mise en concurrence de plusieurs acheteurs. Il en existe plusieurs types. Le plus connu, les enchères à l'anglaise, consiste à fixer un prix de départ et laisser les participants poser des ordres faisant petit à petit monter le prix jusqu'à ce que plus personne n'enchérisse : l'objet est remporté par le dernier enchérisseur. Il existe aussi des systèmes d'enchères descendants où l'on part d'un prix de départ élevé, que l'organisateur de la vente baisse jusqu'à ce qu'un participant se porte acquéreur. Ces types d'enchères sont dits publiques car les participants ont continuellement accès à l'entièreté de l'information : les prix, les ordres de leurs concurrents, etc... Dans le cadre du RTB, les mécanismes d'enchères utilisés sont dits "sous pli caché" (*sealed bid auction*) ce qui signifie que les participants n'ont accès qu'à des informations partielles.

Les enchères en second prix (Second-price) ou enchères de Vickrey [Vickrey, 1961] sont un type d'enchères à un tour où plusieurs participants prennent part et posent un seul ordre afin de remporter un objet x_i. Les participants ne sont pas informés des montants des ordres placés par les autres. Le gagnant est le plus offrant mais le prix à payer n'est que celui de la deuxième meilleure enchère appelé *prix du marché* c_i et seul le gagnant aura accès à ce prix. Les autres participants n'auront donc accès qu'à une borne inférieure de ce prix correspondant au montant de leur propre enchère. Pour ce type d'enchères il peut être démontré que l'optimalité, qui permettrait à l'ensemble des participants de maximiser conjointement leurs bénéfices, est atteinte lorsque tous choisissent comme montant, la valeur que chacun associe réellement à l'objet mis en vente v_i (*Truthful bidding*)[2] [Jackson, 2013].

Les enchères en premier prix (first-price auction) est assez similaire aux enchères au second prix, le gagnant est toujours le plus offrant, mais dans ce cas, le *prix du marché* est fixé au même prix que l'offre. Bien que cette différence puisse sembler minime, elle a un impact important sur les meilleures stratégies. Ainsi, on peut prouver que l'équilibre pour ce type d'enchères est atteint lorsque chaque participant m offre $b_{m,x} = \frac{n-1}{n} * v_{m,x}$ où $m \in \{1, \ldots, n\}$, n étant le nombre de participants [Jackson, 2013].

En pratique, dans le contexte du RTB, ces stratégies optimales ne sont pas possibles à appliquer en raison d'une part, de la difficulté pour les acteurs d'évaluer précisément la valeur qu'ils associent à

2. `https://en.wikipedia.org/wiki/Vickrey_auction#Proof_of_dominance_of_truthful_bidding`

l'encart publicitaire mis en vente et d'autre part, à cause de la contrainte budgétaire limitant les marges de manoeuvre des enchérisseurs. Cela créé de l'irrationalité chez les acteurs et empêche l'application des stratégies de *Truthful bidding*. Par conséquent, les DSP ont cherché des moyens d'optimiser les revenus de leurs clients en utilisant des processus plus ou moins avancés allant de la fonction d'enchères paramétrée manuellement par un expert, aux algorithmes d'enchères basés sur les réseaux de neurones profonds.

Dans la suite de cette introduction, nous donnons tout d'abord une définition du revenu et décrivons les différentes issues possibles d'une enchère. Nous développons ensuite les défis techniques liés à l'optimisation des campagnes d'enchères en temps réel puis nous finissons en décrivant les contributions que nous apportons afin de répondre à certains de ces défis.

1.2 Revenu et issues des enchères

Dans cette section nous présentons les différentes issues d'une enchère et leurs implications sur le revenu engrangé lors d'une campagne.

Notation	**Description**
x_i	Objet mis en vente
v_i	Valeur associée à x_i
c_i	Coût adjugé de x_i
p_i	Probabilité de clic estimée
b_i	Montant misé pour x_i
y_i	Clic : 0 si x_i n'entraîne pas de clic, 1 sinon
z_i	Enchère remportée : 0 si $b_i < c_i$, 1 sinon
$\widehat{E}_i = p_i v_i$	Valeur de x_i espérée
$r_i = y_i v_i - c_i$	Revenu réel si enchère remportée

TABLE 1.1 – Notations pour les systèmes d'enchères en temps réel

Le tableau 1.1 présente les notations liées au RTB. Ainsi, l'objectif final d'un agent au cours d'une campagne de RTB comportant T opportunités d'affichage est de maximiser son revenu sous contrainte d'un budget limité B :

$$\max \sum_{i=0}^{T} r_i \quad \text{t.q.} \quad \sum_{i=0}^{T} c_i \leq B \tag{1.1}$$

Afin de réussir à maximiser ce revenu, il est important de voir que tous les cas ne se valent pas et que l'on peut séparer les issues possibles d'une enchère de RTB en quatre cas de valeurs différentes,

TABLE 1.2 – Tableau de coûts spécifiques au RTB

		Vraie valeur : 0	Vraie valeur : 1
valeur prédite	0	Vrai négatif (TN) : Ne pas prédire de clic et qu'il n'y en ait pas	Faux négatif (FN) : Échouer à prédire un clic
	1	Faux Positif (FP) : Prédire un clic quand il n'y en a pas	Vrai positif (TP) : Réussir à prédire un clic

résumées dans le tableau 1.2 :

- L'enchérisseur perd une enchère qui ne donne pas lieu à un clic. L'agent n'a pas dépensé d'argent et n'a pas laissé passer de clic, ce cas peut être considéré comme un cas neutre de statu quo.
- L'enchérisseur perd une enchère qui donne lieu à un clic. L'agent n'a pas dépensé de budget mais il est passé à côté d'un clic, réduisant le nombre de clics maximal possible. La valeur d'un tel cas est donc négative.
- L'enchérisseur gagne une enchère qui ne rapporte pas de clic. Il perd du budget pour ne rien gagner, ce cas est donc négatif.
- L'enchérisseur gagne une enchère qui rapporte un clic : il utilise une partie de son budget mais obtient un clic en contrepartie. Ce cas est positif, à condition que le prix à payer (*market price*) soit inférieur à la valeur associée à x_i : $c_i < v_i$.

Nous développons cet aspect du RTB concernant la prise en compte des différentes valeurs de l'issue d'une enchère plus en détail en section 3.2.2.

1.3 Défis techniques inhérents au RTB

Après avoir présenté ce qu'est le RTB dans les grandes lignes, nous présentons ici les deux grands défis techniques qui interviennent dans l'optimisation d'une campagne d'affichage publicitaire en ligne si l'on se place du coté DSP. Le premier est l'estimation de la probabilité de clic (Click-Through rate prediction, CTR prediction). Une fois la requête x_i reçue, il faut estimer la probabilité d'occurrence du clic si cette requête était remportée : cela est nécessaire pour obtenir une estimation de la valeur de cette requête $\hat{E}_i = \hat{p}_i v_i$. Cette estimation d'utilité est cruciale afin d'évaluer le montant à proposer et a déjà été largement étudiée notamment dans la communauté de recherche sur les moteurs de recommandation ainsi que pour le sujet qui nous abordons ici : l'affichage publicitaire en ligne [Guo et al., 2017, Qu et al., 2016, 2018]. La principale difficulté de cette tâche est le très gros déséquilibre

des données : il n'y a que environ 0.1% de clics parmi toutes les impressions (publicités affichées). Ce déséquilibre nous amène à étudier la prédiction du clic sous l'angle de la prédiction d'événements rares. Les modèles de classification ainsi que les méthodes d'évaluation de performance classiques sont conçus pour fonctionner sur des données équilibrées, c'est-à-dire, des échantillons de taille comparable pour chaque classe à prédire. La faible présence d'échantillons de la classe d'intérêt, ici les clics, biaise les modèles et les fonctions d'évaluation [Ranjan, 2020, Tomz et al., 2003, Van Den Eeckhaut et al., 2006, Weiss and Hirsh, 1998]. Ainsi, pour un ensemble de données contenant 99% d'échantillons appartenant à la classe 1, un modèle prédisant uniquement cette classe là obtiendra en apparence de très bons scores malgré le fait qu'il sera totalement incapable de prédire les cas rares qui sont souvent les cas que l'on souhaite prédire.

Une fois la probabilité de clic estimée en appliquant des techniques de prédiction d'évènements rares, la seconde grande question qui se pose est l'optimisation de la campagne d'enchères en elle-même. Le but de cette phase est de formuler une proposition de prix pour chaque demande d'affichage de manière à maximiser le nombre de clics obtenus dans la limite du budget B. Pour cela il faut définir une stratégie (*politique*) qui est chargée de proposer un montant d'enchères étant donnée une requête d'enchère x_i contenant les informations qualifiant l'enchère et permettant la prédiction de la probabilité de clic. À partir de l'état actuel de la campagne : budget restant, vitesse de consommation du budget, nombre d'opportunités d'enchères restants jusqu'à la fin de la campagne, etc..., la *politique* devra adapter les montants proposés pour, par exemple, ne pas dépenser trop rapidement et passer à côté d'affichages plus intéressants plus tard dans la campagne ou bien, au contraire, ne pas assez dépenser et finir la campagne avec du budget restant mais avec très peu de clics remportés.

Les travaux présentés dans ce document tentent d'apporter différentes contributions concernant ces deux grandes questions.

1.4 Contributions

Nous présentons dans ce manuscrit les contributions apportées au domaine des enchères en temps réel appliquées au RTB. Ces contributions peuvent être regroupées en trois parties.

La première concerne la prédiction de la probabilité de clic et plus largement les modèles de prédiction des événements rares ainsi que leur évaluation. Nous étudions les modèles de classification binaire

et les biais qu'ils présentent lorsque appliqués à la prédiction d'évènements rares. Nous travaillons avec trois modèles : la régression logistique, la régression logistique pondérée ainsi que la Deep Factorization Machine (DFM) et les appliquons à la prédiction de la probabilité de clic afin de mettre en évidence leurs potentiels biais sur les événements rares.

Nous étudions également les biais induits par ce type de données dans les méthodes classiques d'évaluation et montrons qu'elles peuvent être trompeuses dans certains cas. Afin de pallier cela, nous présentons une mesure d'évaluation propre au RTB, intégrant les coûts réels ainsi que les valeurs associées aux clics. Nous démontrons que cette fonction d'évaluation permet bien de corriger les biais propres à la prédiction d'événements rares. Nous montrons également que l'intégration du paramètre de valeur des clics permet une meilleure projection de la rentabilité des campagnes d'affichage publicitaire en ligne. Grâce à cette nouvelle mesure d'évaluation de la prédiction du clic nous montrons la supériorité de l'approche par régression logistique pondérée sur les deux autres méthodes étudiées.

Les données de RTB et particulièrement de prédiction de la probabilité de clic sont de hautes dimensions et majoritairement catégorielles. Lors de nos recherches nous avons remarqué que la phase de pré-traitement et la question de la réduction de dimension n'étaient que très peu abordées par les travaux dans le domaine. Ainsi, nous présentons le problème de la réduction de dimensionnalité sur des données catégorielles. Le jeu de données de référence Ipinyou utilisé dans tous les travaux de recherche sur le RTB ou la prédiction du clic (pCTR) peut être, selon les pré-traitements, de très haute dimensionalité (jusqu'à 900k variables) et très éparse une fois encodé. Nous proposons une méthode de regroupement de modalités intra-variable basée sur les coefficients de la régression logistique permettant la réduction de la dimensionalité sans toutefois dégrader significativement les performances de classification. Cette approche tend vers l'automatisation du pré-traitement et la sélection automatique de variables. Le but est de produire des regroupements de modalités intra-variables ayant un sens et restant explicables, ce qui reste une question de recherche ouverte et active.

Dans un second temps et en nous appuyant sur nos travaux sur la prédiction de la probabilité de clic, nous étudions la question de l'optimisation de la stratégie d'enchères pour la maximisation des clics sous contrainte du budget. Nos contributions dans ce domaine se concentrent sur la formulation de la représentation du problème et sur différentes fonctions d'enchères : constante, dynamique, ainsi que sur les méthodes d'apprentissage par renforcement. Nous présentons le processus d'optimisation des enchères et sa formulation sous forme de processus de décision markovien permettant l'application

des différentes fonctions d'enchères et proposons une étude comparative des performances de ces différentes approches, certaines déjà largement utilisées dans l'industrie et d'autres s'inspirant des dernières approches d'apprentissage profond par renforcement. Nous présentons les résultats, tout d'abord sur un jeu de données de référence contenant déjà la prédiction du clic utilisé dans les travaux de référence du domaine à des fins de comparaison. Nous présentons également les résultats croisés entre les approches de prédiction de la probabilité du clic et d'optimisation de stratégie d'enchères. Nos résultats montrent que les approches par renforcement ne produisent pas de résultats significativement meilleurs que les autres approches. La sensibilité de ce type d'algorithme aux hyperparamètres ainsi qu'à la formulation du problème, en plus de leur instabilité d'apprentissage fait qu'il est très compliqué en pratique d'obtenir des résultats de qualité sur un problème aussi large et complexe que le RTB.

Par conséquent, dans une troisième partie, nous nous concentrons sur l'étude de la convergence de l'approche par renforcement du Deep Double Q-Network (DDQN). Afin de mettre en évidence les comportements de cet algorithme, nous utilisons une abstraction du problème du RTB en appliquant nos études sur le jeu de casse-brique *Atari-Breakout* disponible dans la bibliothèque de référence pour l'apprentissage par renforcement *Gym*[3]. Ce jeu nous permet de nous affranchir d'une partie de la complexité du RTB afin de mieux mettre en lumière les propriétés de convergence du DDQN. Nous présentons ainsi une étude sur l'influence de la formulation du problème et notamment sur la taille de l'espace d'action sur la convergence et une autre sur l'apprentissage automatique de la formulation des états du processus de décision markovien.

1.5 Organisation du manuscrit

Nous présentons l'ensemble de nos travaux en quatre chapitres. Le chapitre 2 présente le jeu de données de référence que nous avons utilisé dans la plupart de nos expériences. Le chapitre 3 présente les travaux effectués sur les événements rares et la prédiction de la probabilité de clic. Le chapitre 4 présente les travaux sur l'optimisation des campagnes d'enchères. Le chapitre 5 présente quant à lui nos travaux sur la convergence de l'apprentissage par renforcement ainsi que l'apprentissage automatique de la représentation des états. Enfin, dans le chapitre 6, nous présentons les conclusions ainsi que les perspectives liées à nos travaux.

3. https ://www.gymlibrary.ml/

Chapitre 2

Jeu de données : IpinYou

Contenu

2.1 Contexte

Pour réaliser nos études, nous utilisons l'ensemble de données Ipinyou [Zhang et al., 2014a] qui est largement utilisé dans la littérature sur l'optimisation des enchères et la prédiction de la probabilité du clic [Huang et al., 2020, Qu et al., 2016, Ren et al., 2018, Zhang et al., 2014a]. Cet ensemble de données fut initialement partagé par Ipinyou, l'une des principales sociétés chinoise de publicité en ligne, pour son concours international d'enchères en temps réel en 2013. À l'origine, la compétition se déroulait sur 3 saisons : la première sur avril et mai 2013, la seconde entre juin et septembre et la dernière entre octobre et décembre 2013. À chaque saison, une phase offline était organisée où les équipes participantes avaient accès à des données de logs et devaient proposer les meilleurs algorithmes de RTB possibles. Ils étaient ensuite testés sur un jeu de test gardé privé par Ipinyou et les meilleurs algorithmes étaient déployés au sein de la DSP online de Ipinyou. L'équipe gagnante était choisie selon les scores obtenus en offline et en online.

Les jeux de données d'entraînement fournis sont organisés en deux fichiers de logs : le fichier d'ordre (bidding log) et le fichier d'impressions, clics et conversion. Les deux fichiers contiennent les informations qualifiant la demande de bid. Le fichier d'ordre contient les ordres passés par la DSP de Ipinyou et le deuxième fichier contient le résultat de ces ordres si l'enchère a été remportée par la DSP. La réunification de ces deux fichiers se fait sur l'attribut *Bid ID*.

Dans l'ensemble de nos travaux, nous n'utilisons pas cette version d'origine des jeux de données offline, mais une version unifiée des logs d'ordres et de clics. Le jeu de données que nous utilisons [1], contient uniquement les données des saisons 2 et 3, celles de la saison 1 n'étant pas basées sur le même modèle de données [Liao et al., 2014]. Nous présentons ce modèle dans le tableau 2.1. Le jeu de données contient un total de 15 395 258 opportunités d'enchères qui ont conduit à un total de 11 557 clics répartis en 9 campagnes différentes. Chacune de ces campagnes correspond à un annonceur différent, par exemple l'annonceur 3476 vend des pneus alors que l'annonceur 2259 vend de la poudre de lait. La catégorie industrielle de chaque annonceur peut être trouvée dans le tableau 2.2.

1. https ://github.com/wnzhang/make-ipinyou-data

TABLE 2.1 – Description des variables du jeu de données Ipinyou

Colonne	Description	Type	Exemple	Nombre de modalités
click	Si un clic a eu lieu ou non	Booléen	1	2
weekday	Jour de la semaine	Entier	2	7
hour	Heure du jour	Entier	12	24
region	ID de la région	Entier	210	
city	ID de la ville	Entier	102	
adexchange	ID de l'ad exchange	Entier	1	3
domain	domaine du site (anonymisé)	Chaine de charactère	"trqRTJkrBoq7JsNr5SqfNX"	32010
slotid	ID du de l'emplacement	Chaine de charactère	"mm_34022157_3445226_11175096"	88741
slotwidth	Largeur de l'emplacement	Entier	336	17
slotheight	Hauteur de l'emplacement	Entier	280	11
slotvisibility	Position de l'emplacement	Chaine de charactère	"First view", "Second view", ...	11
slotformat	Type de l'emplacement	Chaine de charactère	"Pop", "Fixed", ...	3
slotprice	Prix de réserve	Entier	10	265
creative	ID de la publicité à afficher	Chaine de charactère	"77819d3e0b3467fe5c7b16d68ad923a1"	30
payprice	Prix du marché	Entier	200	301
usertag	Segmentation utilisateur de Ipinyou	Liste de tags	10006,10010	
User-Agent	information sur l'os et le navigateur	Chaine de charactère	"Chrome_Android"	39

TABLE 2.2 – Statistiques sur le jeu d'entraînement

Camp ID	Business	Impressions	Clicks	CTR (%)
1458	e-commerce chinois	3,697,694	2969	0,08
2259	Lait en poudre	835,556	280	0.03
2261	Télecom	687,617	207	0.03
2821	Chaussure	1,322,561	843	0.06
2997	E-commerce Mobile (application)	312,437	1,386	0.44
3358	Logiciel	1,742,104	1,358	0.07
3386	E-commerce international	2,847,802	2,076	0.07
3427	Hydrocarbures/Carburant	2,593,765	1,926	0.07
3476	Pneus	1,970,360	1,027	0.05

2.2 Jeu de données utilisé et pré-traitement

Nous présentons la version du jeu de données Ipinyou de petite dimension que nous construisons grâce à la sélection de variables. Elle est obtenue en sélectionnant un ensemble de 11 variables : adexchange, city, creative, hour, region, slotformat, slotheight, slotvisibility, slotwidth, usertag, weekday. Nous prenons également la variable User-Agent et la scindons en deux : os et browser en suivant Zhang et al. [2014a]. Nous avons également divisé les *usertags*, qui étaient à l'origine une liste de tags caractérisant le visiteur selon la segmentation interne d'Ipinyou. Ce pré-traitement permet ainsi d'encoder l'information sur ces usertags afin qu'elle soit prise en compte par les modèles de prédiction.

Cette sélection de caractéristiques a été faite en écartant les caractéristiques ayant un nombre élevé de modalités, parfois autant que d'exemples de données, ce qui les rend inutiles pour un modèle de prédiction et aurait créé un ensemble de données très clairsemé. Le tableau 2.3 présente le nombre de dimensions pour chaque campagne. Le nombre de modalités présentes dans les campagnes étant

TABLE 2.3 – Dimensions du jeu de données Ipinyou_lowDim pour chaque campagne

Campagnes	Dimensions
1458	525
2259	191
2261	577
2821	550
2997	511
3358	541
3386	533
3427	550
3476	533

variable, la dimension de chaque jeu de données en codage disjonctif complet est également différente. Le jeu de données obtenu conserve les informations significatives tout en réduisant grandement sa dimension une fois encodé, facilitant ainsi les calculs.

2.3 Description du jeu de données

Dans cette section, nous présentons une rapide analyse descriptive des données. Nous présentons en figure 2.1 les distributions cumulatives des prix des clics et non-clics pour chacune des neufs campagnes (exprimés en Yuan x 1000). Cette figure nous permet de vérifier que la plupart du temps les clics valent plus cher que les non-clics. Néanmoins, nous pouvons voir que pour certaines campagnes, comme la 2821 ou la 2259, les deux distributions de rapprochent voire se touchent pour des prix bas. Cela indique un nombre similaire de clics et de non-clics à bas coût pour ces campagnes et pousse l'enchérisseur à se concentrer sur des enchères peu élevées.

La figure 2.2 représente les distributions non-cumulatives des prix gagnants (winning prices) pour les clics et les non-clics. Nous savons d'après Lin et al. [2020] que les prix gagnants suivent approximativement une distribution log-normale. Nous pouvons également voir sur cette figure que les distributions ont pratiquement la même forme, à ceci près que celles des clics est légèrement décalée vers des prix plus élevés. Par conséquent, cela signifie qu'il y a tout de même un nombre non négligeable d'impressions à bas prix menant à des clics. Ces impressions ont donc une valeur élevée et devraient idéalement être privilégiées par l'enchérisseur.

Pour aller plus loin dans l'analyse des prix, nous présentons la distribution des prix par variable

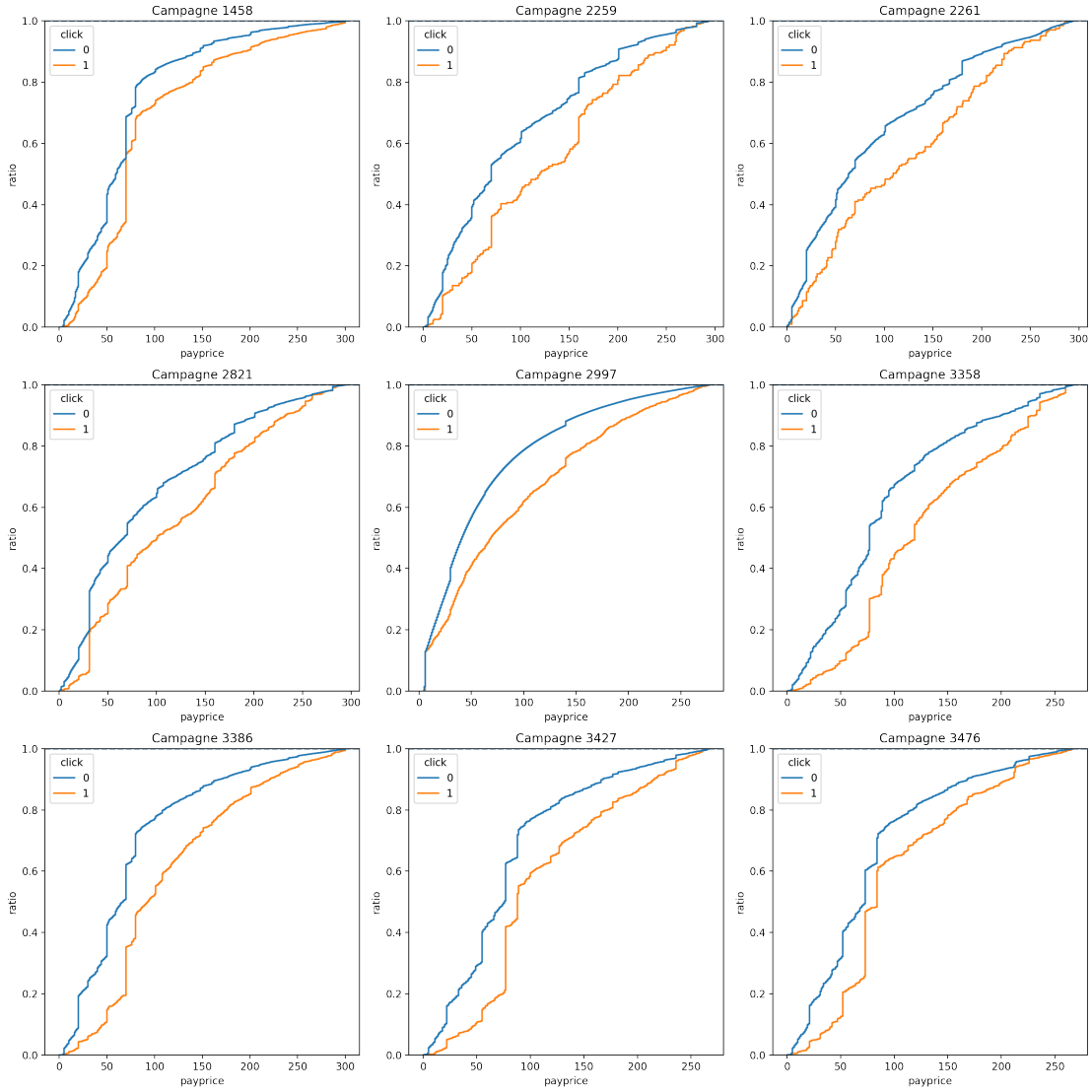


FIGURE 2.1 – Distribution cumulative des prix pour les clics et non clics

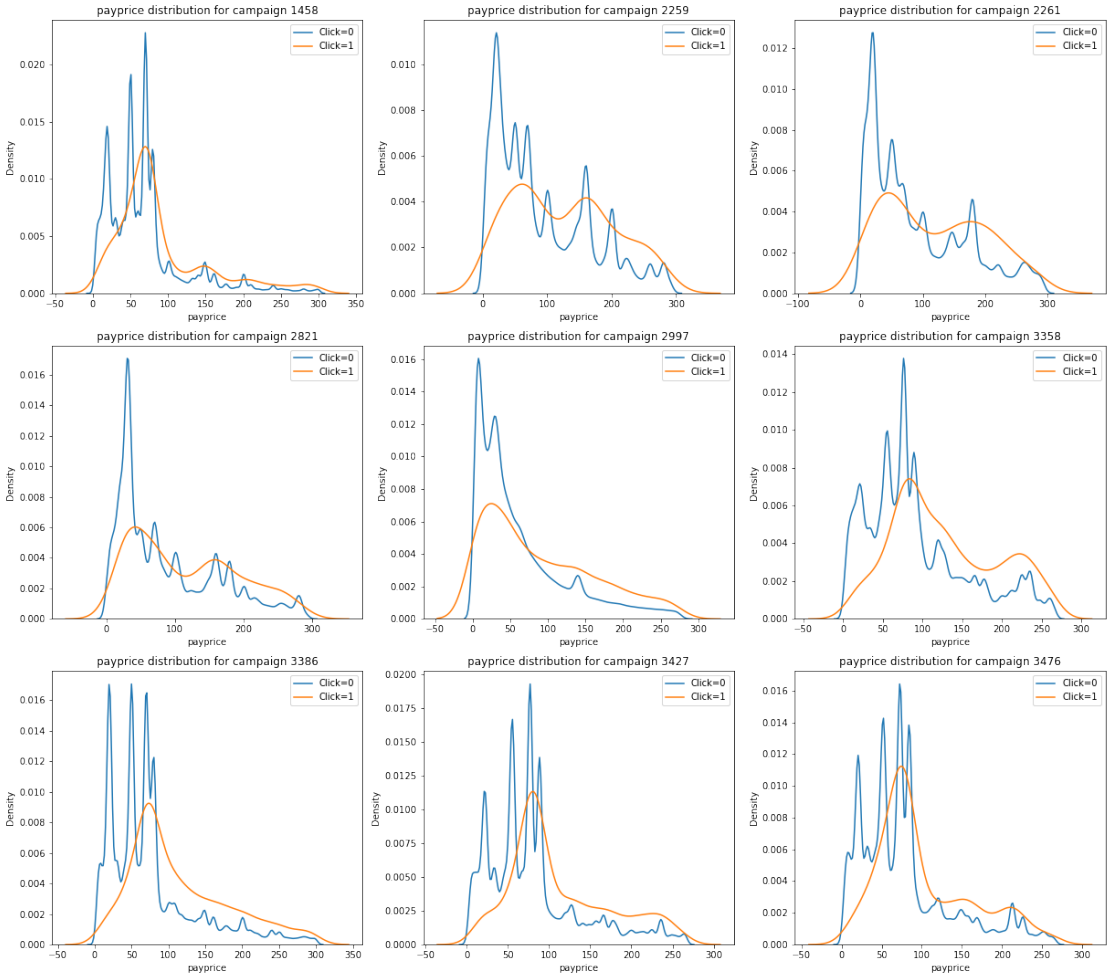


FIGURE 2.2 – Distribution des prix gagnants pour les clics et non-clics pour chaque campagne

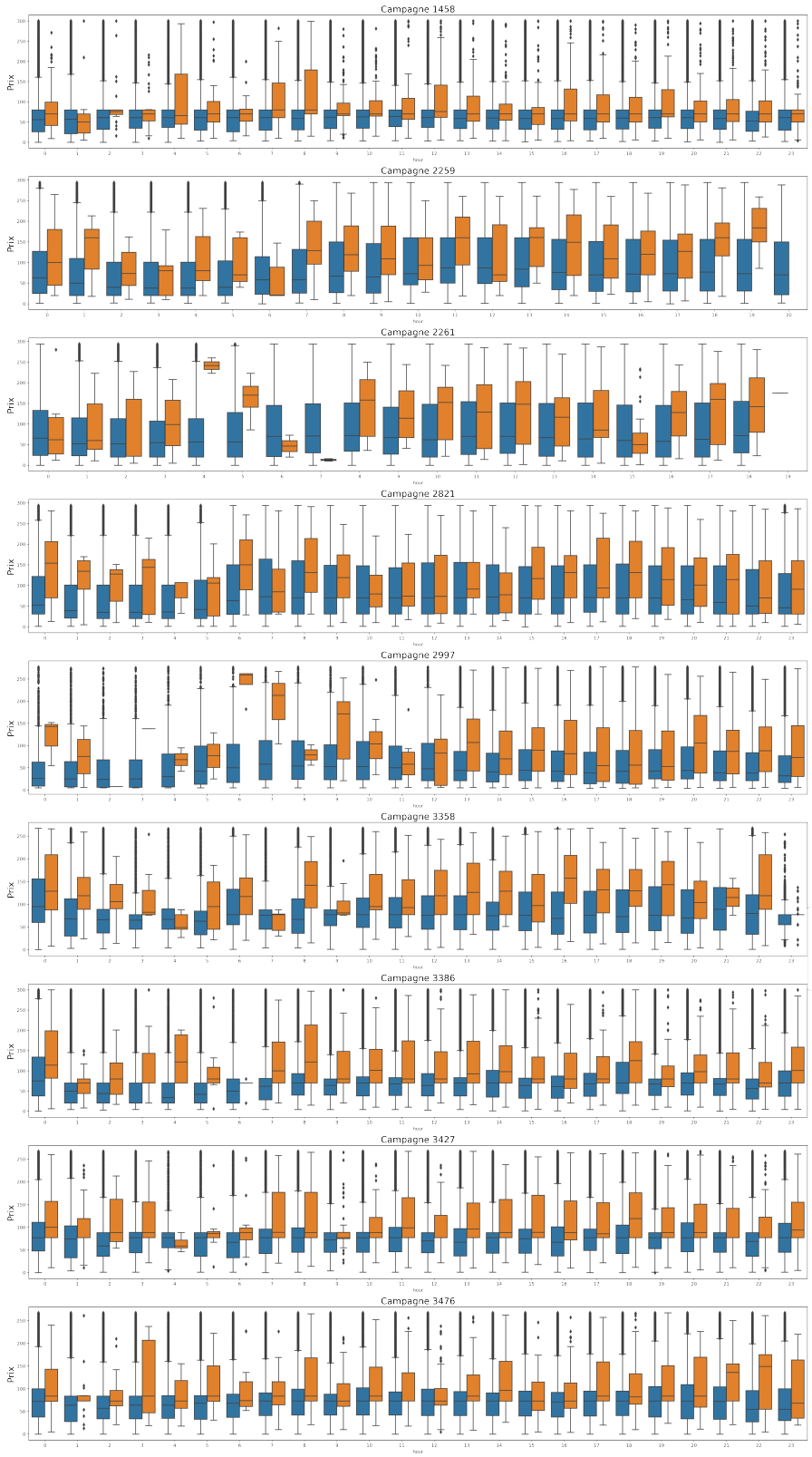


FIGURE 2.3 – Distribution des prix pour les clics (orange) et non-clics (bleu) selon les heures

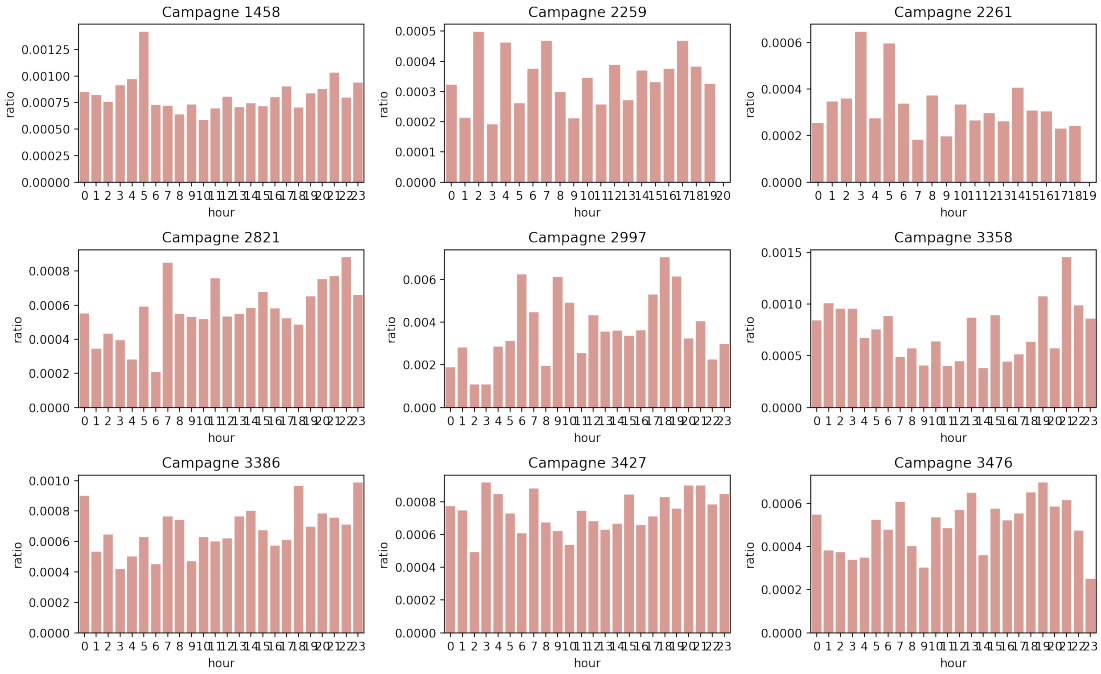


FIGURE 2.4 – Histogramme des ratio entre clic et non clics du nombre d'occurrences des heures

pour chaque campagne (fig. 2.3 pour la variable *heure*, les autres graphes sont donnés en annexe A). La figure 2.3 montre que les prix des clics sont logiquement plus élevés que ceux des non-clics, mais nous observons également que pour les clics se produisant pendant la nuit, les prix deviennent assez instables. Cela peut être dû à une taille d'échantillon plus faible, car il y a beaucoup moins de clics la nuit. L'analyse des distributions de prix pour le reste des caractéristiques nous indique que les prix sont toujours plus élevés pour les clics que pour les non-clics, quelle que soit la campagne.

La figure 2.4 représente le ratio entre le nombre de clics et de non-clics par heure. Sur cette figure, il est intéressant de voir que la forme générale du graphe diffère d'une campagne à l'autre : cette différence est particulièrement voyante entre les campagnes 2997 et 3358. Cela indique des différences de distributions entre les campagnes et cela se vérifie sur toutes les autres visualisations. Par conséquent, il faudra traiter les campagnes séparément et entraîner des modèles et des agents à l'échelle d'une campagne.

Il est aussi intéressant de noter certains pics récurrents de clics au petit matin entre 4 et 8 heures. Cela traduit un relativement faible nombre de non-clics par rapport au nombre de clics. Il pourrait donc être rentable de se positionner sur les impressions à ces heures-là. D'après la figures 2.3, on peut

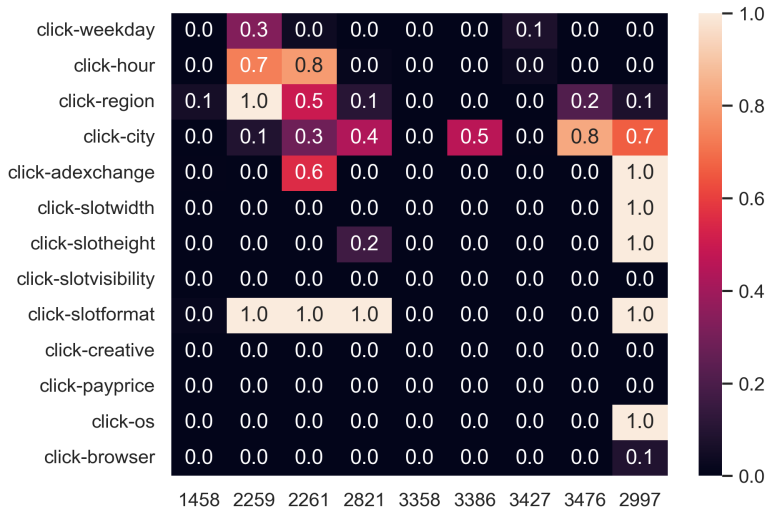


FIGURE 2.5 – Matrice de significativité présentant les p-values du test chi-square entre la variable clic et chacune des autres variables.

voir que ces impressions peuvent être assez peu chères pour certaines campagnes comme la 2259 ou la 2821 mais au contraire assez chères comme pour la 3386. Cela va aussi dans le sens de considérer les campagnes séparément.

La figure 2.5 montre la matrice des p-values du test statistique du chi-square entre la variable clic et chacune des autres variables pour toutes les campagnes.

On remarque sur cette figure que la plupart de ces coefficients sont à zéro traduisant une grande significativité des variables dans la plupart des cas. Cela est assez étonnant car l'on pouvait s'attendre à ce que certaines de ces variables comme os et browser n'atteignent pas une p-value aussi basse. On peut aussi observer que la variable city n'est pas significative pour six des neuf campagnes.

Les p-valeurs qui sont égales à dans cette figure correspondent aux variables qui ne prennent qu'une seule valeur ou bien même qui ne sont pas du tout renseignées dans le jeu de données.

Nous fournissons en annexe les graphiques pour l'ensemble des variables considérées.

Chapitre 3

Événements rares et prédiction du clic

Contenu

Comme souligné en introduction, la première tâche dans le processus d'optimisation des enchères est la prédiction du clic. Déjà très étudiée dans la littérature, cette tâche tombe dans la catégorie de la prédiction d'événements rares, étant donnée la rareté des clics par rapport au nombre d'affichages (moins d'un clic pour mille affichages).

Comme souligné en introduction, la gestion de jeu de données comportant un grand déséquilibre dans la répartition des classes des échantillons requiert une attention et des traitements particuliers. L'ensemble de ces techniques est regroupé sous le terme de prédiction d'événements rares. Nous décrivons dans ce chapitre l'ensemble de nos travaux concernant ce sujet et en particulier sur la prédiction de la probabilité de clic.

Dans un second temps, nous présentons une approche visant à réduire la dimensionnalité des données d'entrée de la prédiction du clic. Les données d'entrée dans le cadre du RTB sont de nature catégorielle et de très haute dimension. Cela tient à la très grande cardinalité des ensembles de valeurs possibles des variables qui une fois encodées (disjonctif complet) augmentent grandement la dimension et le caractère épars du jeu de données. L'augmentation de dimension pose ainsi des questions quant au respect des temps de réponse autorisés par le RTB ($\approx 100ms$). De plus, nous remarquons également que la rareté d'occurrence des différentes valeurs possibles d'une variable a des conséquences sur les performances des algorithmes de prédiction.

Dans cette optique, nous présentons une approche de pré-traitement automatique visant la réduction de dimension des données par regroupements itératifs des modalités intra-variable. Dans cette approche, les regroupements entre deux modalités sont effectués successivement à l'intérieur d'une variable selon la proximité des coefficients de régression logistique qui leurs sont associés. Nos résultats initiaux montrent que nombreux regroupements peuvent être effectués avant que la perte d'information ainsi provoquée ne dégradent la qualité de prédiction du modèle. Les regroupements successifs effectués ne présentent pas de structure en rapport avec la variable correspondante et ne permettent donc pas de faire apparaître de nouvelles informations sur la structure du jeu de données. Par exemple, les regroupements opérés sur la variable région ne sont pas effectués dans une logique géographique qui pourrait exercer une influence sur les clics. Cette propriété sémantique sur les regroupements est pourtant souhaitable pour aller vers l'automatisation du pré-traitement dont les résultats resteraient facilement interprétables. L'étude présentée ici n'est donc qu'une première étape de recherche dans cette direction et devra être approfondi dans de futurs travaux.

3.1 Classification binaire et prédiction d'événements rares

Prédire des événements rares comme les guerres, les crises, les maladies rares et, dans notre cas les clics, est une tâche complexe : les algorithmes d'apprentissage automatique nécessitent généralement une proportion raisonnable d'événements, c'est-à-dire de cas d'intérêt que l'on veut apprendre à prédire. Par conséquent, lorsque les événements sont rares, les modèles de classification classiques sont biaisés et de nombreux travaux ont été réalisés pour étudier et corriger ces biais [Ranjan, 2020, Tomz et al., 2003, Van Den Eeckhaut et al., 2006, Weiss and Hirsh, 1998].

De manière générale, le problème de la classification binaire consiste à classer des individus dans une classe ou l'autre. À partir d'un ensemble de données contenant N individus définis par m variables de sorte que X, l'ensemble de données, s'écrive :

$$X = \begin{bmatrix} x_{11} & \dots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{N1} & \dots & x_{Nm} \end{bmatrix}$$

Chacun de ces individus appartient à une classe. On note $y_i \in \{0, 1\}$ la classe de l'individu i dans le cas de la classification binaire. On note Y le vecteur contenant les classes de chaque individu :

$$Y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$$

Le but de la classification binaire est d'estimer les paramètres $\hat{\theta}$ d'un modèle h_θ permettant de minimiser l'erreur de prédiction. Le modèle donne une estimation de la classe de l'individu x_i tel que $\hat{y_i} = h_{\hat{\theta}}(x_i)$. Les estimations du modèle sont comparées avec les vraies classes grâce à une fonction d'erreur comme l'erreur quadratique moyenne définie par :

$$MSE(y_i, \hat{y_i}) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y_i})^2$$

On utilise l'erreur afin de mettre à jour les paramètres θ d'un modèle grâce à des algorithmes d'optimisation tels que le gradient stochastique qui met à jour les paramètres du modèle dans le sens de la réduction de la fonction objective tel que :

$$\hat{\theta}_{new} \leftarrow \hat{\theta} - \eta \nabla E(\hat{\theta})$$

où $E(\hat{\theta})$, est la fonction d'erreur (fonction objective) et le paramètre η est le taux d'apprentissage (*learning rate*) et contrôle la magnitude de la correction appliquée aux paramètres du modèle. Ces

étapes sont répétées jusqu'à convergence du modèle, à savoir, jusqu'à ce que l'erreur ne baisse plus ou tombe sous un seuil acceptable.

Pour cette phase d'entraînement, seule une partie X_{train} du jeu de données X est utilisée. Le reste des données X_{test} est utilisé afin d'évaluer la performance du modèle sur des données sur lesquelles il n'a pas été entraîné. Cela permet d'évaluer sa capacité de généralisation, c'est-à-dire, sa capacité à apprendre la structure sous-jacente des données plutôt que d'apprendre X_{train} par coeur (sur-apprentissage). Pour la classification binaire, on distingue quatre issues possibles que nous présentons dans le tableau 3.1. Il existe ainsi de nombreuses mesures d'évaluation des modèles de classification binaires [Hossin and Sulaiman, 2015] comme la précision ($\frac{tp}{tp+fp}$), qui mesure la capacité du modèle à correctement prédire les cas positifs ou le rappel ($\frac{tp}{tp+fn}$) qui mesure la capacité du modèle à détecter les cas positifs (voir la figure 3.1. Ou encore l'aire sous la courbe ROC (AUC) mesurant la capacité d'un modèle à mieux classifier que l'aléatoire. Nous reviendrons plus en détail sur les mesures de performance dans la section 3.2.2.

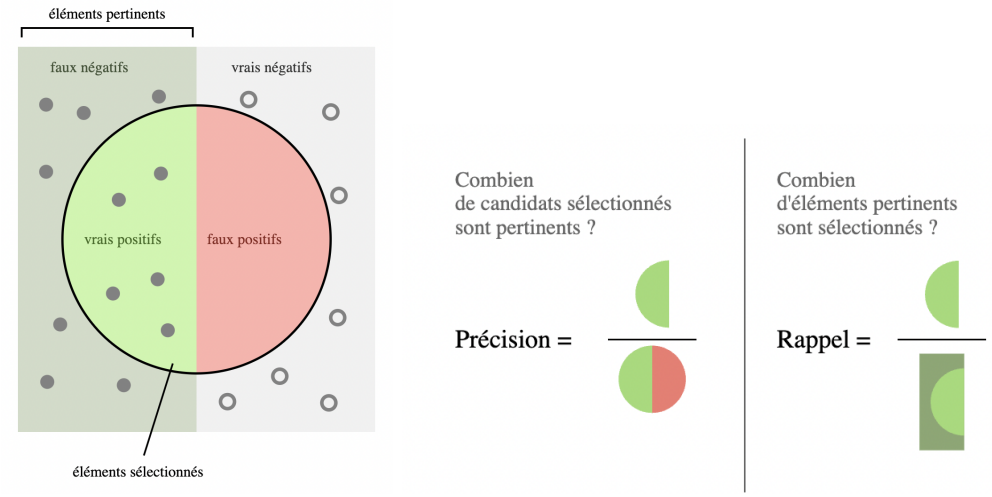


FIGURE 3.1 – Illustration des mesures de performance Rappel et Précision [1]

	$y_i = 0$	$y_i = 1$
$\hat{y_i} = 0$	Vrai négatif (tn)	Faux négatif (fn)
$\hat{y_i} = 1$	Faux positif (fp)	Vrai positif (tp)

TABLE 3.1 – Tableau de confusion pour la classification binaire

Dans le contexte des événements rares cependant, l'entraînement ainsi que l'évaluation sont remis

1. https://upload.wikimedia.org/wikipedia/commons/f/ff/Precisionrappel.svg

en question : si la classe positive ne représente que 0.1% des individus, alors prédire uniquement la classe négative donnera de bons résultats avec les méthodes classiques. C'est pourquoi il est crucial d'étudier les biais et les corrections possibles des modèles de classification et des méthodes d'évaluation afin de les appliquer aux événements rares. Pour cela, nous sélectionnons trois modèles que nous présentons ci-après : la régression logistique classique (LR), la régression logistique pondérée (wLR) ainsi que la Deep Factorization Machine (DFM). Le choix de ces trois modèles nous permet de comparer un modèle classique (LR), un modèle ayant été adapté à la prédiction d'événement rare (wLR) ainsi qu'un modèle d'apprentissage profond spécialement développé pour la prédiction du clic [Guo et al., 2017]. Ce choix de modèle est également justifié par la différence de complexité entre les modèles de régression logistique et DFM : la nécessité de temps réel en RTB conditionne le choix du modèle en fonction de la puissance de calcul disponible. Sur ce point, les modèles de régression sont bien moins complexes que l'apprentissage profond et nécessitent donc beaucoup moins de puissance de calcul. Comparer ces modèles pour mettre en évidence les gains potentiels de performance donne une indication forte sur le modèle à choisir : le modèle DFM ne devra être sélectionné que si le gain est assez élevé pour compenser la complexité supplémentaire.

3.1.1 Régression logistique

L'entraînement d'une régression logistique consiste à estimer les paramètres θ d'un modèle logistique exprimé comme suit :

$$h_\theta(x_i) = \frac{1}{1 + e^{-\theta^T x_i}}$$

où $x_i = \{x_{i1}, x_{i2}, \ldots, x_{iP}\}$. x_i est un échantillon de données à m variables, $h_\theta(x_i)$ exprime, étant donné θ, les paramètres ajustés du modèle, la probabilité que x appartienne à la classe positive. Pour estimer les paramètres $\hat{\theta}$, nous utilisons la descente de gradient stochastique classique avec la perte logistique (*LogLoss*) définie tel que :

$$L(h_\theta(x_i), y_i) = -y_i \log(h_\theta(x_i)) - (1 - y_i) \log(1 - h_\theta(x_i)) \tag{3.1}$$

La régression logistique a déjà été appliquée à la prédiction du clic. Bien que cette approche soit assez puissante et rapide à entraîner pour modéliser les combinaisons linéaires [Chapelle et al., 2015], elle échoue à modéliser les combinaisons entre les variables. Une telle combinaison de variables peut être modélisée au prix d'efforts fastidieux puisque le nombre d'interactions par paires augmente

quadratiquement avec le nombre de variables [Xu et al., 2016].

3.1.2 Régression logistique pondéré (wLR)

Ce type de modèle repose sur la régression logistique avec une modification du modèle afin d'améliorer les performances sur la prédiction d'événements rares [King and Zeng, 2001, Maalouf and Siddiqi, 2014, Maalouf and Trafalis, 2011]. La modification que nous utilisons ici consiste à pondérer l'erreur de prédiction correspondant à chaque individu par le ratio de présence de sa classe dans l'échantillon. Cela équivaut à une perte d'entropie croisée pondérée (weighted cross-entropy) :

$$L(h_\theta(x_i), y_i) = -y_i w_0 \log(h_\theta(x_i)) - (1 - y_i) w_1 \log(1 - h_\theta(x_i))$$

où :

$$w_0 = \frac{n}{n_{classes} * n_{events}},$$
$$w_1 = \frac{n}{n_{classes} * n_{nonevents}}$$

avec n le nombre total d'individus dans l'échantillon, $n_{classes} = 2$ dans le cas d'une classification binaire, n_{events} et $n_{nonevents}$ respectivement le nombre d'événements et de non-événements dans le jeu de données.

3.1.3 Réseaux de neurones artificiels

Maintenant bien connus, les réseaux de neurones artificiels sont des structures d'apprentissage permettant théoriquement d'approximer n'importe quelle fonction [Hornik et al., 1989]. Pris individuellement, un neurone artificiel, illustré figure 3.2, est un modèle de régression linaire recevant une ou plusieurs variables en entrée $x_{1,..,m}$, leur associant un poids $w_{1,..,m}$, et renvoyant une sortie selon la somme des produit linaires de ses entrées et de ses poids associés :

$$\hat{y} = \sum_{i}^{m} w_i x_i + b_i$$

où le biais b_i agit comme un seuil d'activation du neurone.

En combinant des neurones entre eux, on obtient un réseau de neurones (figure 3.3). On retrouve trois types de couches dans ces structures : la couche d'entrée, correspondant aux données d'entrée, les couches cachées qui regroupent les couches successives de neurones artificiels, et la couche de sortie,

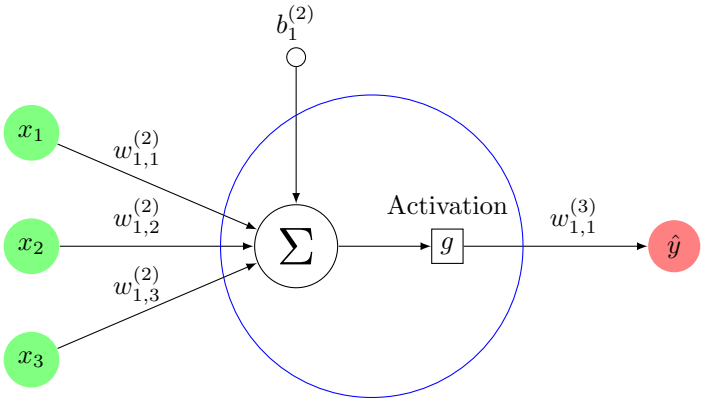


FIGURE 3.2 – Illustration d'un neurone artificiel

correspondant à la dernière couche de neurones, depuis laquelle on récupère le résultat de l'estimation $\hat{y}_i$.

On note o_q^k, la sortie du neurone q de la couche k, w_{qj}^k est le poids que le neurone q de la couche k associe à son prédécesseur j de la couche $k-1$. Par simplification on note aussi le biais $w_{0q}^k = b_q^k$. Cela permet d'écrire l'activation (somme des produits + biais) de chaque neurone q, de la couche k :

$$a_i^k = g_i^k(\sum_{j=0}^{l_{k-1}} w_{ji}^k o_j^{k-1})$$

Avec l_{k-1} étant le nombre de neurones dans la couche $k-1$ et g^k la fonction d'activation de la couche k. L'entraînement se fait en calculant l'erreur $L = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) - (1-y_i)\log(1-\hat{y}_i)$. Pour chaque poids w_{oj}^k, il faut calculer $\frac{\partial L}{\partial w_{qj}^k}$. Afin de simplifier les calculs et comme la dérivée d'une somme est équivalente à la somme des dérivées, nous pouvons poser les calculs pour chaque individu i puis les ajouter à la fin. En appliquant la règle de la chaine (*chain rule)*, on peut appliquer la décomposition :

$$\frac{\partial L}{\partial w_{qj}^k} = \frac{\partial L}{\partial a_j^k}\frac{\partial a_j^k}{\partial w_{qj}^k}$$

On peut noter que :

$$\frac{\partial a_j^k}{\partial w_{qj}^k} = \frac{\partial \left(\sum_{j=0}^{l_{k-1}} w_{qj}^k o_j^{k-1}\right)}{\partial w_{qj}^k} = o_j^{k-1}$$

Ainsi en prenant $\delta_j^k = \frac{\partial L}{\partial a_j^k}$:

$$\frac{\partial L}{\partial w_{qj}^k} = \delta_j^k o_q^{k-1}$$

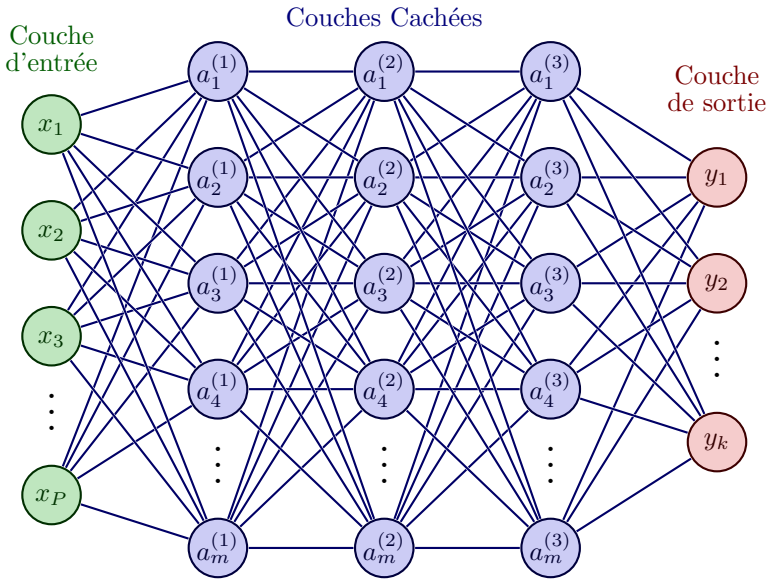


FIGURE 3.3 – Illustration d'un réseau de neurones artificiels

Nous pouvons maintenant calculer la valeur de la propagation arrière (*back-propagation*) pour la couche de sortie m avec g_o la fonction d'activation pour la dernière couche, en faisant une nouvelle fois intervenir la règle de la chaîne :

$$\delta_j^m = (\hat{y} - y) g_o'(a_j^m) = (g_o(a_i^m) - y) g_o'(a_j^m)$$

Et enfin :

$$\frac{\partial L}{\partial w_{qj}^m} = (\hat{y} - y) g_o'(a_j^m) o_q^{m-1}$$

De la même manière nous pouvons écrire la valeur de la propagation arrière pour les couches cachées :

$$\delta_j^k = g'(a_j^k) \sum_{s=1}^{l_{k-1}} w_{js}^{k+1} \delta_s^{k+1}$$

Grâce à ces deux équations, nous pouvons mettre à jour les poids du réseau de neurones en appliquant la descente de gradient : $w \leftarrow w - \eta \frac{\partial L}{\partial w_{qj}^k}$.

Après avoir présenté la théorie générale derrière les réseaux de neurones, nous présentons dans la suite les modèles que nous avons utilisé dans nos travaux.

3.1.3.1 Factorization Machine (FM)

FM (illustrée en figure 3.4) modélise les interactions linéaires mais aussi les interactions de caractéristiques par paires (pair-wise interactions). Comme nous l'avons vu dans la section sur la régression

logistique, le nombre d'interactions par paires augmente de manière quadratique. FM contourne ce problème en estimant des vecteurs de variables cachées (latent variables) pour chaque variable et en factorisant la matrice d'interaction par paire comme le produit VV^T où $V \in \mathbb{R}^{n \times k}$, V^T sa transposée, n le nombre d'individus d'apprentissage et k un hyperparamètre contrôlant la dimension de la représentation vectorielle cachée qui doit être choisie par validation croisée. L'estimation de la matrice V contenant les variables cachées est effectuée grâce à l'erreur quadratique moyenne et la descente de gradient stochastique pour ajuster les valeurs afin de minimiser l'erreur.

Ainsi, la prédiction d'une FM est donnée par :

$$\hat{y}(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \langle v_i, v_j \rangle x_i x_j \tag{3.2}$$

Où les deux premiers termes correspondent à un modèle linéaire classique, $\langle v_i, v_j \rangle$ est le produit scalaire tel que :

$$\langle v_i, v_j \rangle = \sum_{f=1}^{k} v_{i,f} \cdot v_{j,f} \tag{3.3}$$

Ce dernier terme peut être réécrit [Rendle, 2010] tel que :

$$\begin{aligned}
\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \langle v_i, v_j \rangle x_i x_j &= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \langle v_i, v_j \rangle x_i x_j - \frac{1}{2} \sum_{i=1}^{n} \langle v_i, v_i \rangle x_i x_i \\
&= \frac{1}{2} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{f=1}^{k} v_{i,f} v_{j,f} x_i, x_j - \sum_{i=1}^{n} \sum_{f=1}^{k} v_{i,f} v_{i,f} x_i x_i \right) \\
&= \frac{1}{2} \sum_{f=1}^{k} \left(\left(\sum_{i=1}^{n} v_{i,f} x_i \right) \left(\sum_{j=1}^{n} v_{j,f} x_j \right) - \sum_{i=1}^{n} v_{i,f}^2 x_i^2 \right) \\
&= \frac{1}{2} \sum_{f=1}^{k} \left(\left(\sum_{i=1}^{n} v_{i,f} x_i \right)^2 - \sum_{i=1}^{n} v_{i,f}^2 x_i^2 \right)
\end{aligned}$$

Cette formulation permet de modéliser les interactions paires à paires à travers le terme $\langle v_i, v_j \rangle$.

3.1.3.2 Deep Factorization Machine (DeepFM)

DFM [Guo et al., 2017] est une extension de FM qui estime à la fois les interactions de variables d'ordre inférieur et supérieur. DFM est composé d'un module linéaire qui estime les interactions d'ordre 1, d'un module FM qui apprend les interactions d'ordre 2 comme nous l'avons vu précédemment, et

2. Illustration reprise depuis Guo et al. [2017]

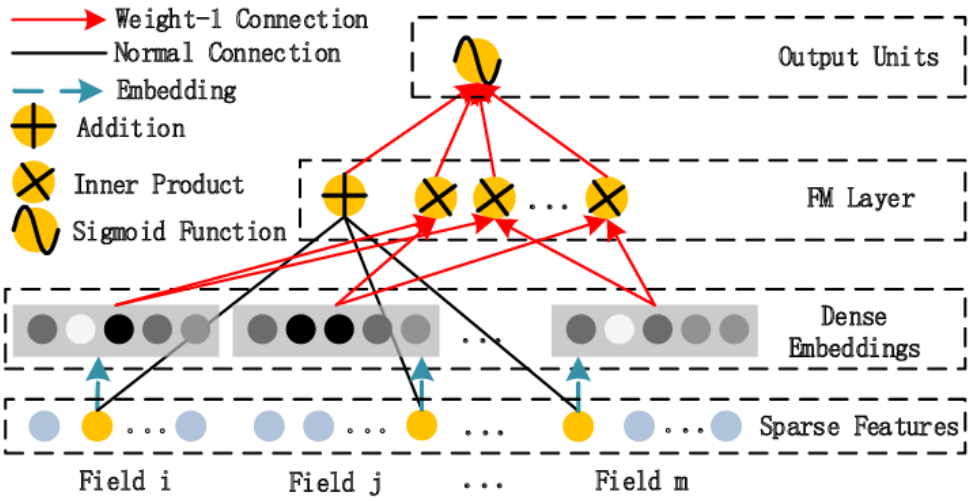


FIGURE 3.4 – Illustration de la Machine à Factoriser (FM) [2]

d'un module réseau neuronal profond (DNN) capable de modéliser les interactions d'ordre supérieur. La prédiction du modèle *DFM* s'écrit :

$$y_{DFM} = \sigma(y_{FM} + y_{DNN})$$

où $\sigma = sigmoid$, la fonction d'activation de la couche de sortie. Nous incluons une illustration du modèle en figure 3.6 [Guo et al., 2017]. Comme les données d'entrée pour la prédiction du CTR sont très éparses et peuvent être de taille variables à cause du nombre différent de modalités par variable, le modèle utilise une couche d'embedding pour les projeter dans un espace dense de dimension inférieure. Chaque variable représentée par un vecteur binaire (codage disjonctif complet de la modalité) est présentée en entrée de la couche de embedding qui l'encode en m vecteurs de taille fixe k (voir figure 3.5).

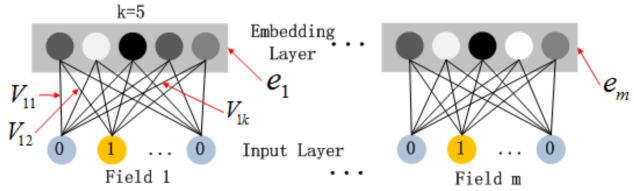


FIGURE 3.5 – Illustration de la couche d'embedding de la Deep Factorization Machine (DFM) [3]

3. Illustration reprise depuis Guo et al. [2017]

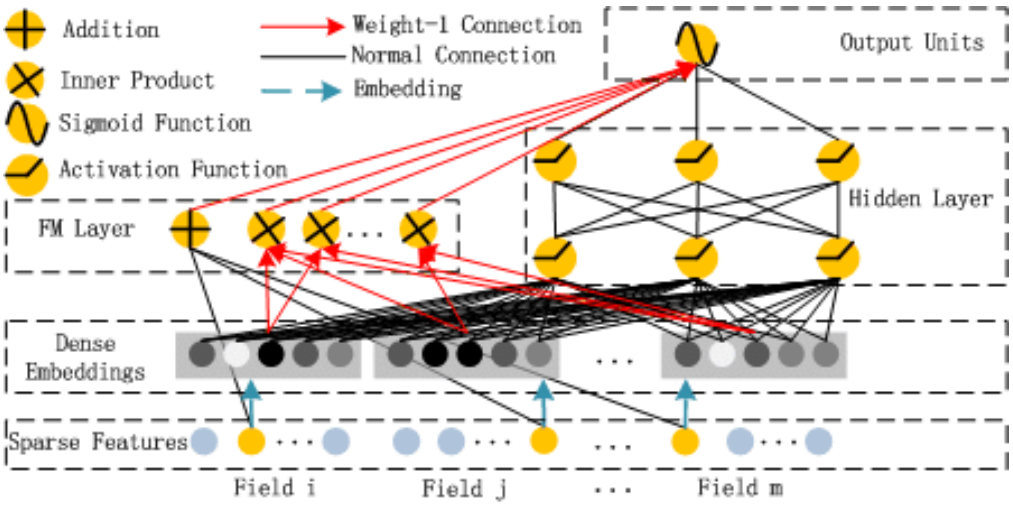


FIGURE 3.6 – Illustration du modèle Deep Factorization Machine (DFM)[3]

Cette couche d'embedding sert d'entrée aux composants FM et DNN et ses poids sont appris conjointement au reste du modèle, ce qui permet à DFM de ne pas nécessiter d'ingénierie sur les variables ni de pré-entraînement. Cette couche d'embedding permet également de remettre toutes les variables d'entrée à la même dimension : en codage disjonctif par exemple, les variables peuvent avoir des dimensions très différentes. La couche d'embedding prend chaque variable en entrée et les projette chacune dans une dimension fixe commune à toutes les variables.

Les modèles présentés dans cette section permettent de couvrir l'éventail des approches type utilisées dans la prédiction de le probabilité de clic : de l'approche linéaire classique légère et scalable avec la régression logistique [Chapelle et al., 2015, Maalouf and Siddiqi, 2014], pondérée pour les événements rares ou non, aux approches plus récentes et très populaires dans la communauté, basée sur des réseaux de neurones profonds [Guo et al., 2017, Zhang et al., 2016].

3.2 Évaluation de la prédiction du clic

L'évaluation des performances d'un modèle de prédiction est cruciale et non triviale. On trouve de nombreuses fonctions d'évaluation présentant chacune leurs avantages et inconvénients et il convient de choisir la, ou les mieux adaptées au problème sur lequel on souhaite évaluer un modèle. Nous présentons dans cette section les mesures de performance les plus utilisées dans les travaux de prédiction de la probabilité de clic en pointant certaines limites pouvant entraîner de mauvaises décisions dans le choix

du meilleur modèle de prédiction.

Afin de palier ces limites, nous présentons une nouvelle fonction d'évaluation spécialement conçue pour la prédiction de la probabilité de clic appliquée au RTB. Cette fonction prend ainsi en considération les différents cas de prédiction ainsi que les coûts ou bénéfices qu'ils entraînent. Nous montrons que cette fonction d'évaluation permet bien de palier certaines limites des mesures classiques permettant ainsi une meilleure sélection de modèle.

3.2.1 Méthodes classiques

3.2.1.1 AUC ROC

Une des métriques d'évaluation les plus utilisées est l'aire sous la courbe ROC (AUC). La courbe ROC est la courbe représentant le taux de vrais positifs (TPR), également appelé sensibilité ou rappel et défini comme suit :

$$TPR = \frac{TP}{TP + FN} \tag{3.4}$$

avec FN les faux négatifs, contre un taux de faux positifs (FPR) défini par

$$FPR = \frac{FP}{FP + TN} \tag{3.5}$$

pour tous les seuils de séparation. L'aire sous la courbe ROC mesure la qualité du modèle par rapport à une estimation aléatoire.

L'AUC donne une importance égale aux faux positifs (FP) et aux faux négatifs (FN) : en RTB, prédire un clic qui ne se produit pas à un coût différent de celui de ne pas prédire de clic lorsqu'il y en a un. Cela impacte directement la comparaison des modèles : deux modèles de prédiction du CTR peuvent avoir le même AUC_{ROC} mais conduire à des performances d'enchères différentes. Drummond and Holte [2004] montrent comment AUC_{ROC} échoue à prendre en compte les coûts. Dans des contextes d'événements rares, il est très courant que les événements et les non-événements aient des répercussions très différentes et il est donc hautement souhaitable d'utiliser une fonction d'évaluation sensible aux coûts.

De plus, la courbe ROC est calculée pour chaque seuil possible, ses régions extrêmes sont prises en compte dans le AUC_{ROC} alors qu'elles ne sont pas pertinentes dans le contexte de la classification. Le AUC_{ROC} ne peut également prendre en compte que l'ordre des probabilités prédites et non sur

TABLE 3.2 – Tableau de coûts spécifiques au RTB

		Vraie valeur	
		0	1
valeur prédite	0	Vrai négatif (TN) : Ne pas prédire de clic et qu'il n'y en ait pas	Faux négatif (FN) : Échouer à prédire un clic
	1	Faux Positif (FP) : Prédire un clic quand il n'y en a pas	Vrai positif (TP) : Réussir à prédire un clic

les probabilités absolues elles-mêmes [Lobo et al., 2008]. Or, la plupart des algorithmes d'optimisation des enchères se basent sur le pCTR pour proposer des ordres.

3.2.1.2 AUC_PR

Une autre métrique courante pour évaluer les modèles d'événements rares est l'aire sous la courbe de précision/rappel (AUC_{PR}). Elle mesure l'aire sous la courbe représentant la précision ($P = \frac{TP}{(TP+FP)}$) en fonction du rappel (TPR). Contrairement à la AUC_{ROC}, elle ne tient pas compte du taux de faux positifs (FPR) qui tend à être très faible dans les applications d'événements rares en raison du grand nombre d'exemples négatifs.

Elle s'est avérée efficace dans de tels contextes [Davis and Goadrich, 2006, Saito and Rehmsmeier, 2015, Sofaer et al., 2018]. Sofaer et al. [2019] comparent AUC_{ROC} et AUC_{PR} montrant que pour les événements rares, AUC_{PR} est moins enclin à surestimer la performance de classification des modèles en négligeant le taux de vrais négatifs. Néanmoins, AUC_{PR} calcule toujours un score général en considérant tous les seuils, y compris les régions extrêmes, et peut donc être biaisé.

3.2.2 Fonction d'évaluation spécifique au RTB avec prise en compte des coûts

Pour surmonter les limites des métriques présentées plus haut, nous présentons une fonction d'évaluation spécifique au RTB qui tient compte des coûts ainsi que de la valeur v que les annonceurs eux-mêmes associent à un clic. Elle donne également des valeurs différentes pour les faux négatifs et les faux positifs. Elle est assez simple par rapport à des mesures d'évaluation plus complexes pour les données déséquilibrées [Bekkar et al., 2013] et peut donc être facilement utilisée et manipulée par des utilisateurs n'ayant pas de compétences statistiques avancées. Nous décrivons chaque cas de classification possible dans le tableau 3.2.

Les détails de la valeur pour chaque cas sont donnés dans le tableau 3.2. Parmi les quatre cas

présentés dans ce tableau, seuls deux sont applicables : les faux positifs et les vrais positifs. En effet, par la nature semi-censurée du système d'enchères à l'œuvre dans le RTB, un enchérisseur qui ne remporte pas l'enchère n'est pas notifié du clic éventuel. Cela dit, la non-prise en compte de ces deux cas n'influe pas de manière significative sur la mesure de performance proposée, leurs coûts étant soit nuls dans le cas des vrais négatifs, soit très peu élevés pour les faux négatifs (le nombre de clics étant très faible).

Étant donné que les algorithmes d'optimisation des enchères reposent généralement sur les probabilités de clics prédites et non sur des prédictions binarisées [Lee et al., 2013a, Zhang et al., 2014b], nous définissons la fonction de valeur de manière probabiliste telle que :

$$\begin{aligned} V(x,v) &= \underbrace{-\left(\sum_i^N c_i p(x_i)(1-y_i)\right)}_{\text{coût FP}} + \underbrace{\left(\sum_i^N (v-c_i)p(x_i)\times y_i\right)}_{\text{coût TP}} \\ &= \sum_i^N \left(-c_i p(x_i) + v p(x_i) y_i\right) \end{aligned} \tag{3.6}$$

Où $y_i \in \{0,1\}$ est une variable binaire correspondant au clic. Dans l'équation 3.6, seuls les coûts liés aux faux positifs et aux vrais positifs sont pris en compte : les coûts liés aux deux autres cas, faux négatifs et vrais négatifs ne peuvent être pris en compte à cause du système d'enchères en deuxième pris en vigueur dans la plupart des plateformes d'échanges. En effet, les prix du marché et les notifications de clics ne sont envoyés qu'au gagnant de l'enchère. Ainsi, dans ces deux cas, n'ayant pas accès aux informations, nous ne pouvons les intégrer dans notre fonction de valeur.

3.3 Étude de cas

3.3.1 Étude comparative sur le jeu de données à dimension réduite (Ipinyou_LowDim) et fonction d'évaluation probabiliste

Nous présentons dans cette section les résultats obtenus sur l'étude de la capacité des modèles à prédiction des événements rares. Nous comparons ces performances selon plusieurs mesures de performance afin de mettre en lumière les éventuels biais présentés précédemment. Nous incluons dans cette étude les modèles de régression logistique LR et wLR ainsi que le modèle DFM.

Le modèle DFM est implémenté en Pytorch [4]. Nous avons empiriquement fixé la taille du DNN à

4. `https://pytorch.org/`

(400, 400, 400), ce qui correspond également à l'implémentation présentée par Guo et al. [2017]. Nous fixons un taux de dropout à 0,5 sur ce DNN et un terme de régularisation Ridge sur la couche de embedding à $1e-5$. Le dropout [Srivastava et al., 2014] est une méthode de régularisation permettant de limiter le sur-apprentissage des réseaux de neurones en nullifiant un certain pourcentage des poids du réseau. Cela permet également d'accélérer l'apprentissage. Nous fixons la dimension de la représentation latente à $k = 6$. Le modèle est entraîné pendant 500 époques sur chaque fold et 20% du jeu d'entraînement est utilisé pour la validation.

Pour les mesures d'évaluation, nous utilisons des implémentations scikit-learn pour la cross-entropy, également appelée logistic loss (logloss), et AUC_{ROC}. Pour AUC_{PR}, nous utilisons le *average_precision_score*[5] qui approxime l'AUC_{PR} défini par :

$$AUC_{PR} = \sum_n (R_n - R_{n-1}) P_n$$

Où P_n et R_n sont la précision et le rappel au seuil n.

Les résultats présentés sont obtenus par validations croisées 10 fois. Nous avons divisé le jeu de données en 10 sous-ensembles (folds) avec un ratio train/test fixé à 90%. Comme souligné dans l'analyse du jeu de données, les campagnes concernent différents annonceurs, les probabilités de clic et les annonces sont très différentes, qu'il s'agisse d'annonces pour des voitures ou des vêtements par exemple. Nous formons donc un modèle différent pour chaque campagne.

Nous suivons Gary King [2001] en sous-échantillonnant les non-événements pour obtenir un ratio de 1 pour 10 (événements :non-événements). Pour chaque fold, nous sélectionnons tous les événements (c.-à-d. les clics) et échantillonnons aléatoirement 10 fois plus de non-événements. Nous présentons les statistiques de chaque campagne dans le tableau 3.3.

3.3.1.1 Résultats

D'après le tableau de résultats 3.4, la régression logistique classique semble donner les meilleurs résultats dans chaque campagne et pour les trois métriques de $LogLoss$, AUC_{ROC} et AUC_{PR}.

Nous traçons dans la figure 3.7 la valeur du coût en fonction du v. Ces résultats montrent un résultat différent concernant le classement des modèles : la version équilibrée de la régression logistique surpasse les deux autres modèles. Cela confirme les biais des mesures d'évaluation classiques qui ne reflètent

5. `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.average_precision_score.html`

TABLE 3.3 – Taille de chaque plis pour chaque campagne

Campagne	Taille du sous-échantillon	Clics	Taille d'origine
1458	26 994	2 454	3 083 056
2259	3 080	280	835 556
2261	2 277	207	687 617
2821	9 273	843	1 322 561
2997	15 246	1 386	312 437
3358	14 938	1 358	1 742 104
3386	22 836	2 076	2 847 802
3427	21 186	1 926	2 593 765
3476	11 297	1 027	1 970 360

TABLE 3.4 – Performance du taux de clics dans différentes métriques pour chaque modèle considéré et pour les campagnes de publicité par affichage en ligne.

Campagne	Algorithme	Logloss	AUC_{ROC}	AUC_{PR}
1458	LR	**0.11**	**0.94**	**0.81**
	wLR	0.22	**0.94**	**0.81**
	DFM	0.39	0.60	0.12
2259	LR	**0.31**	**0.63**	0.18
	wLR	0.62	0.62	**0.19**
	DFM	0.48	0.50	0.11
2261	LR	**0.31**	**0.62**	**0.17**
	wLR	0.56	**0.62**	0.15
	DFM	0.55	0.54	0.10
2821	LR	**0.31**	**0.60**	**0.16**
	wLR	0.63	0.58	0.15
	DFM	0.44	0.51	0.10
2997	LR	**0.29**	**0.61**	**0.14**
	wLR	0.64	0.60	0.13
	DFM	**0.29**	0.50	0.08
3358	LR	**0.14**	**0.93**	**0.77**
	wLR	0.27	0.92	0.76
	DFM	0.40	0.62	0.18
3386	LR	**0.27**	**0.69**	**0.30**
	wLR	0.58	0.68	0.29
	DFM	0.39	0.58	0.12
3427	LR	**0.15**	0.90	**0.75**
	wLR	0.32	**0.91**	0.70
	DFM	0.36	0.56	0.11
3476	LR	**0.20**	**0.85**	**0.59**
	wLR	0.44	**0.85**	0.38
	DFM	0.60	0.52	0.10

pas les coûts appropriés pour les résultats possibles. La fonction de valeur proposée est linéaire et nous voyons que la régression logistique pondérée a la pente la plus élevée, ce qui signifie que c'est le modèle le plus rentable quelle que soit la valeur v.

À des fins de comparaison, nous ajoutons l'oracle qui est le modèle de prédiction de CTR parfait et qui n'a donc que des vrais positifs. Nous constatons que la fonction de valeur de l'Oracle est toujours plus élevée que les algorithmes considérés. Cela confirme que la fonction d'évaluation reflète bien les performances de prédiction du CTR et que nous pouvons nous y fier pour choisir le meilleur modèle, wLR en l'occurrence.

En plus de permettre une meilleure évaluation des modèles, la fonction proposée tenant compte des coûts fournit des indications utiles d'un point de vue appliqué : elle donne à l'annonceur le coût par clic (CPC) le plus élevé qu'il pourrait accepter de l'éditeur pour que sa campagne soit rentable. Cette caractéristique fait partie des avantages liés à l'utilisation de la fonction de valeur proposée pour évaluer et classer les modèles de prédiction du taux de clics.

En examinant les différences de performance entre les campagnes, nous constatons, en comparant les tableaux 3.3 et 3.4 et la figure 3.7, une corrélation entre la performance de prédiction et le nombre d'événements dans la campagne, surtout en termes de V : les campagnes 2259 et 2261 ont beaucoup moins d'événements et aussi les valeurs les plus faibles en termes de fonction "cost-aware". Cette remarque vaut pour les autres métriques du tableau 3.4 mais seulement à un degré moindre.

En ce qui concerne le modèle d'apprentissage profond DFM, les résultats montrent des performances décevantes, quelle que soit la métrique d'évaluation. Bien qu'il ait été développé pour cette tâche spécifique, ce modèle semble être très spécifique aux données et n'est pas compétitif ici. Dans notre cas, il ne vaut pas le temps et la puissance de traitement supplémentaires requis par rapport à la régression logistique pondérée.

3.3.1.2 Conclusion sur les événements rares

Nous avons étudié les fonctions d'évaluation pour les événements rares et proposé une fonction d'évaluation spécifique à la prédiction du taux de clics, tenant compte des coûts. Nous montrons que la perte logistique, AUC_{ROC} mais aussi AUC_{PR} ne tiennent pas compte de la nature déséquilibrée des données et notre proposition apporte une correction à ce biais. De plus, en considérant les coûts et les

6. `https://www.wordstream.com/cost-per-click`

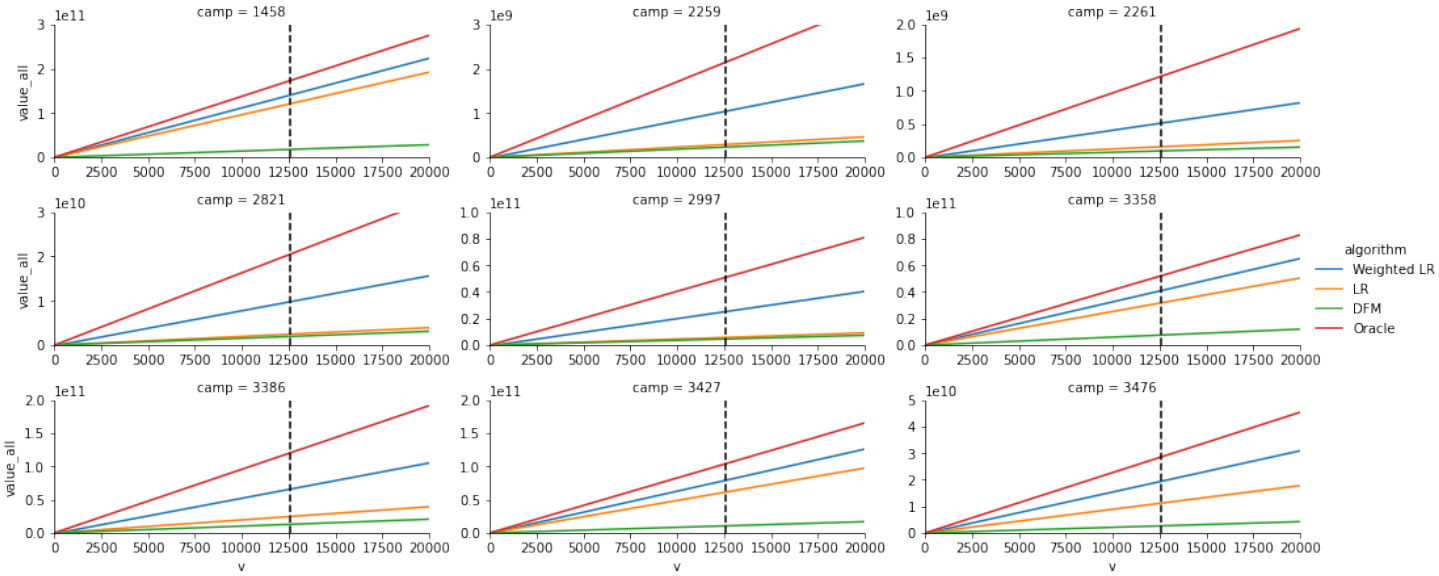


FIGURE 3.7 – Fonction de valeur pour chaque campagne. La ligne en pointillé correspond au $CPC_{moyen} \approx \$2 \approx 12.73CN$ qui désigne le prix moyen d'un clic sur le marché [6]. Cela donne une indication sur la rentabilité des modèles par rapport à la valeur du marché.

valeurs, cette nouvelle métrique permet d'évaluer le rendement potentiel d'un modèle de prédiction du taux de clics.

Nous avons appliqué la régression logistique pondérée pour les événements rares à la prédiction du taux de clic et avons comparé ses performances avec une base de régression logistique classique et un modèle d'apprentissage profond. Cela conduit à une amélioration significative des prédictions du taux de clics. D'un point de vue économique, l'approche proposée de régression logistique pondérée est la plus rentable, sur la base de la fonction de valeur avec prise en compte des coûts.

3.3.2 Étude sur le sous-échantillonnage

Le problème de la prédiction d'évènements rares réside dans la différence de proportion entre les classes à prédire. Une approche souvent utilisée afin de rééquilibrer les données consiste à supprimer aléatoirement une partie des exemples de la classe majoritaire. Wang [2020] montrent que dans une proportion contenue, le sous-échantillonage peut être opéré sans influence sur les estimations. Cette approche constitue donc un bon moyen d'équilibrer les données dans un contexte d'événements rares et peut être utilisée en complément des approches déjà présentées dans ce document.

Nous étudions ici l'influence du sous-échantillonnage sur les performances des algorithmes de pré-

diction. Cette méthode consistant à supprimer aléatoirement des échantillons de la classe majoritaire. Nous souhaitons, par cette étude, montrer les différentes sensibilités des algorithmes au sous-échantillonnage et à la perte d'information dans l'échantillon de non-événements.

Nous présentons en figure 3.8, l'AUC pour différents pourcentages de sous-échantillonnage de non-événements : 0%, 30%, 50% et 80%. La procédure de sous-échantillonnage consiste à éliminer de manière aléatoire un pourcentage de non-événements, c'est-à-dire d'impressions qui ne donnent pas lieu à des clics. Sur cette figure, LR_{900K} correspond aux performances obtenues par une régression logistique entraînée sur une version du jeu de données Ipinyou de très grande dimension ($\approx 900kdimensions$) Cette figure montre que pour la régression logistique pondérée et la machine à factoriser (FM), le sous-échantillonnage détériore les performances en termes d'AUC. Étonnamment, l'AUC de la régression logistique non pondérée chute lorsque le pourcentage de sous-échantillonnage diminue. Cependant, nous remarquons que ces variations de l'AUC sont très légères, même pour des pourcentages de sous-échantillonnage très élevés. Cela va dans le sens de Wang [2020], qui argumente que le sous-échantillonnage peut être fait tant que nous avons suffisamment d'événements.

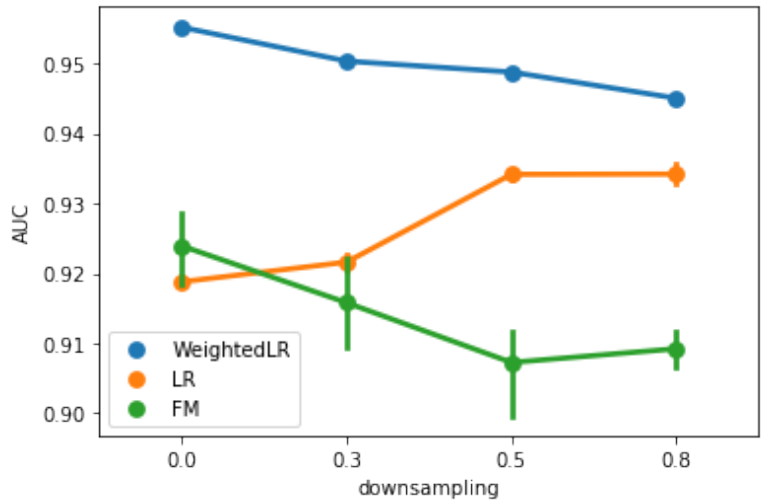
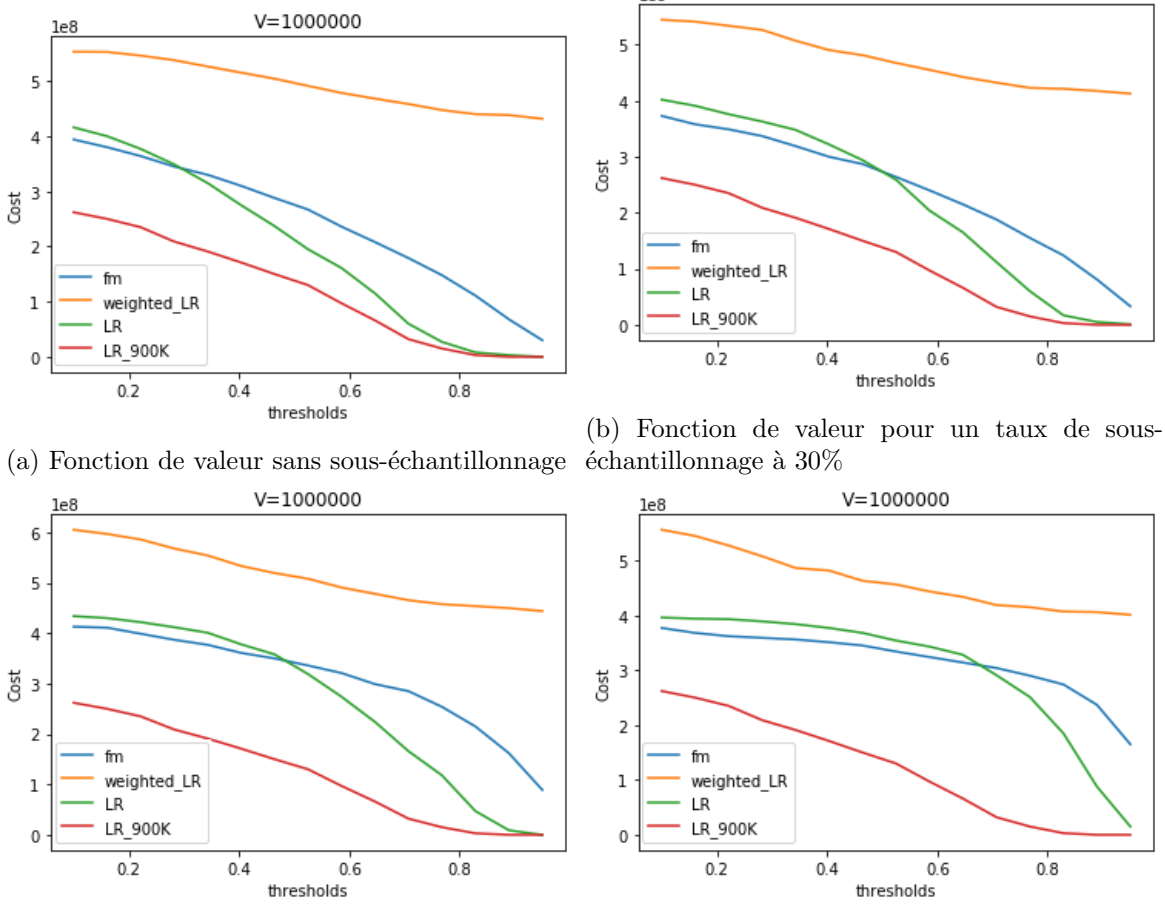


FIGURE 3.8 – AUC pour différents taux de sous-échantillonnage pour chaque algorithme

Nous vérifions l'influence très relative du sous-échantillonnage des non-événements sur les performances des modèles que nous considérons en évitant également les biais de l'AUC. Nous fournissons dans la figure 3.9 les graphiques de la fonction de valeur pour les différentes valeurs de sous-échantillonnage. À partir de ce graphique, nous pouvons voir que le sous-échantillonnage n'a pas eu d'effets significatifs sur les performances de classification du point de vue de notre fonction personna-

lisée non plus.



(a) Fonction de valeur sans sous-échantillonnage

(b) Fonction de valeur pour un taux de sous-échantillonnage à 30%

(c) Fonction de valeur pour un taux de sous-échantillonnage à 50%

(d) Fonction de valeur pour un taux de sous-échantillonnage à 80%

FIGURE 3.9 – Fonction de valeur pour différents taux de sous-échantillonnage

Dans de futurs travaux, ces résultats devront être complétés par les performances exprimées dans la mesure de performance sensible aux coûts proposée.

3.4 Réduction de dimension et pré-traitement automatique

Nous abordons dans cette section les approches existantes pour la réduction de dimension et présentons l'approche par regroupement de modalités intra-variable selon les coefficients de la régression logistique que nous proposons. Nous présentons dans un premier temps les méthodes de régularisation dont certaines permettent la sélection de variables. Nous présentons également les méthodes de

réduction de dimension par autoencoder et discutons des similitudes et différences avec la méthode bien connue de l'Analyse en Composantes Principales (ACP). Ces méthodes s'appliquent toutes à la réduction de dimension à l'échelle de l'ensemble de données mais, dans leurs formulations d'origine, ne permettent pas d'opérer les regroupements de manière intra-variable. De plus, elles peuvent agir sur le nombre de paramètres (et donc de modalités) pris en compte mais ne permettent pas de former des regroupements ascendants de modalités.

Ce sont ces deux aspects que nous souhaitons retrouver avec l'approche proposée. Celle-ci se base sur les coefficients d'une régression logistique afin d'opérer itérativement des regroupements entre modalités d'une même variable. D'après nos études, cette approche en clustering de modalités ascendant permet de nombreux regroupements sans dégrader les performances du modèle de prédiction. Ces études devront être approfondis afin d'étudier en détail les regroupements produits : il serait en effet hautement souhaitable de réussir à agir sur l'algorithme afin que les regroupements produits aient une valeur sémantique, par exemple que les modalités d'une variable *ville* soient regroupées de manière géographique. Ce genre de propriétés serait non seulement utile pour aller vers l'automatisation des pré-traitements mais aussi dans un but d'extraction d'information et d'analyse de données.

3.4.1 Régularisation

La régularisation désigne un ensemble de procédés permettant de réduire la complexité d'un modèle de prédiction. Cette complexité repose souvent sur le nombre de paramètres que comporte le modèle. Plus il y a de paramètres plus le modèle est sujet au sur-apprentissage, consistant en l'apprentissage par cœur des données par le modèle ce qui est absolument à éviter car cela réduit à néant la capacité d'un modèle à correctement prédire sur de nouveaux échantillons qu'il n'aurait pas vus durant l'apprentissage. Les méthodes de régularisation permettent de prévenir ce phénomène. En agissant sur les paramètres des modèles, ces méthodes permettent en fait aussi la sélection de variables et donc la réduction de dimensionnalité. Nous présentons dans cette section les méthodes de régularisation les plus connues.

3.4.1.1 Least Absolute Shrinkage and Selection Operator (Lasso)

La régularisation Lasso [Tibshirani, 1996] consiste en l'ajout d'un terme de pénalisation de norme L_1 sur les paramètres du modèle tel que, dans le cas d'un modèle linéaire :

$$min \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{D} w_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{D} |w_j| \tag{3.7}$$

L'ajout de ce terme permet le contrôle de la norme des paramètres du modèle. En d'autres termes, cela contrôle la magnitude des paramètres du modèle et peut, dans le cas de Lasso, les réduire jusqu'à 0. L'importance de la pénalisation est contrôlée par le paramètre λ. Un λ proche de 0 ne pénalisera que très peu le modèle et plus ce paramètre sera proche de 1, plus la magnitude des paramètres sera pénalisée : plus le nombre de paramètre mis à 0 sera grand ce qui équivaut à l'élimination des variables associées et donc à une diminution de la dimensionnalité.

3.4.2 Clustering de modalités en régression logistique

Les approches par régularisation ou auto-encodage se concentrent sur la sélection ou le regroupement de variables et ne s'intéressent pas à la nature de la variable. Par exemple, une donnée géographique est souvent disponible à plusieurs échelles et peut être plus pertinente à l'échelle de la région que de la ville. De même il est fréquent que le nombre de modalités intéressantes pour discriminer entre deux conditions soit assez faible. Par exemple tous les jours de la semaine ne sont pas nécessairement aussi informatifs et il peut être intéressant de ne considérer que les deux modalités week-end/jour de semaine.

Nous présentons dans cette section une méthode de réduction de dimension basée sur le regroupement de modalités avec un focus particulier sur les variables catégorielles ayant un grand nombre de modalités. Même si ce sujet a été moins travaillé que la sélection de variables, on peut noter quelques articles reliés à ce sujet. Par exemple, Abdallah and Saporta [1998] s'intéressent aux distances entre modalités, toutefois, ils ne font pas de distinction entre les modalités intra-variables et les modalités appartenant à des variables différentes. De même, Chavent et al. [1999] cherchent une partition de modalités dans un but de classification. Comme noté par les auteurs, l'utilisation de méthodes descendantes est peu adaptée à un nombre important de modalités en raison de leur trop grande complexité algorithmique. D'autres travaux sur la réduction de dimensionalité ont utilisé des termes de pénalisation, basés notamment sur la régularisation Lasso [Tibshirani, 1996]. Ainsi, en présence de variables ordinales, des approches de fusion de coefficients ont été développées, en utilisant un terme de pénalisation basée sur la distance entre les coefficients d'un modèle [Chiquet et al., 2017, Tibshirani

et al., 2005, Tutz and Gertheiss, 2016].

Nous adoptons une approche similaire consistant à regrouper les modalités dont les coefficients dans la régression logistique sont proches. Nous montrons que cette question peut s'écrire soit sous forme de classification hiérarchique ascendante, soit sous forme d'une régression logistique pénalisée avec une pénalité de type Lasso.

Soit $X = \{x_1, \ldots, x_n\}$, un jeu de données contenant n observations et p variables catégorielles, avec un nombre de modalités différent pour chaque variable $s = \{s_1, \ldots, s_p\}$. Nous utilisons le codage disjonctif complet (*one-hot encoding*) :

$$\begin{aligned} x_i &= \underbrace{(0,1,\ldots,0)}_{\text{variable 1}}, \underbrace{(0,0,\ldots,1)}_{\text{variable 2}}, \ldots, \underbrace{(1,0,\ldots,0)}_{\text{variable p}}, \\ &= (x_i^{je})_{i=1,\ldots,n;j=1,\ldots,p;e=1,\ldots,s_j} \end{aligned} \tag{3.8}$$

avec x_i^{je} variable indicatrice, égale à 1 si l'individu i a la modalité e pour la variable j. Soit n_{je} le nombre d'individus ayant la modalité e (de la variable j).

Notre but est de trouver des regroupements de modalités d'une même variable afin de réduire la dimensionalité du jeu de données, sans dégrader les performances du modèle de prédiction. Nous développons une approche de clustering hiérarchique des modalités, basée sur les coefficients d'une régression logistique que nous détaillons dans cette section. Nous présentons également le pseudo-code de cette approche en figure 3.10.

Nous entraînons tout d'abord une régression logistique sur l'ensemble du jeu de données $X = \{x_1 \ldots, x_n\}$ définie par :

$$p(y_i = 1|x_i) = \frac{1}{1 + e^{-(\beta_0 + \beta x_i)}} \tag{3.9}$$

Puis, pour chacune des variables j, nous trouvons les deux modalités e et e' les plus proches en terme de coefficient β, tel que $e, e' = argmin_{e,e'=1\ldots,s_j} |\beta_{je} - \beta_{je'}|$. Nous les regroupons en une seule nouvelle modalité e". Le regroupement qui dégrade le moins les performances du modèle en terme d'AIC est gardé et le modèle est recalculé.

Nous pouvons réécrire cette procédure sous la forme d'une adaptation de la régression pénalisée fused Lasso [Tibshirani et al., 2005], où les auteurs introduisent un terme de pénalisation contraignant la distance entre deux coefficients voisins à un modèle linéaire :

$$min \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{D} \beta_j x_{ij} \right)^2 + \lambda_1 \sum_{j=1}^{D} |\beta_j| + \lambda_2 \sum_{j=2}^{D} |\beta_j - \beta_{j-1}| \tag{3.10}$$

Calculer les paramètres β du modèle complet.
Répéter
  Pour toute variable j **Faire**
    Rechercher le couple de modalités (e, e') le plus proches en terme de coefficient.
    Créer une nouvelle modalité e" en fusionnant les modalités e et e'
    Recalculer le modèle.
  Fin Pour
  Garder la variable groupée minimisant l'AIC.
Jusque Stabilité du modèle

FIGURE 3.10 – Clustering de modalités en régression logistique

Comme indiqué par Tibshirani et al. [2005], cette formulation est adaptée à des variables ordinales. La proximité des modalités dans le jeu de données signifie que ces modalités sont proches en termes de sens. Les auteurs discutent aussi de la manière d'appliquer cette pénalisation à des données catégorielles non ordonnées en utilisant la plus petite distance euclidienne ou bien la plus forte corrélation entre coefficients. Notre approche peut être vue sous cet angle en utilisant donc la norme entre les plus proches coefficients de la régression logistique. Nous cherchons donc le maximum de vraisemblance sous contrainte de fusion des coefficients les plus proches (au sens de la distance L_1) :

$$\max_{\beta} \sum_{i=1}^{n} \Big[ln(1 + e^{\beta x_i}) - y_i \beta x_i \Big] - \lambda_1 \sum_{j=1}^{p} \sum_{e=1}^{s_j} |\beta_{je}| - \lambda_2 \sum_{j=1}^{p} \sum_{e=1}^{s_j} |\beta_{je} - \beta_{je'}| \tag{3.11}$$

avec $\beta_{je-'} = argmin_{e' \neq e} |\beta_{je} - \beta_{je'}|$ et $\lambda_i > 0$. Les paramètres λ_1 et λ_2 contrôlent respectivement le nombre de variables éliminées et le nombre de modalités regroupées. Nous répétons ces étapes jusqu'à ce que toutes les modalités de chaque variable aient été regroupées. Nous obtenons ainsi un arbre hiérarchique des modalités. Les modèles de régression logistique étant emboîtés, il est possible d'utiliser un critère de sélection de modèles de type vraisemblance pénalisée. Ceci permet d'obtenir le niveau de découpage optimal. Afin de conserver le modèle ayant la meilleure capacité prédictive, nous utilisons le Critère d'AIC (Akaike Information Criterion) défini par $AIC = 2k - 2ln(L)$ où L est la vraisemblance du modèle testé et k est le nombre de paramètres.

Nous présentons en figure 3.11 les résultats obtenus sur le dataset d'entraînement Ipinyou_lowDim à 525 dimensions (campagne 1458). On observe que les groupements successifs ne dégradent pas les performances du modèle avant un nombre important d'itérations. Cette relative robustesse au regroupement du modèle est contre-intuitive mais peut être expliquée en partie par le faible nombre d'occurrences de chaque modalité ainsi que par leur proximité en terme de coefficients. À partir d'un

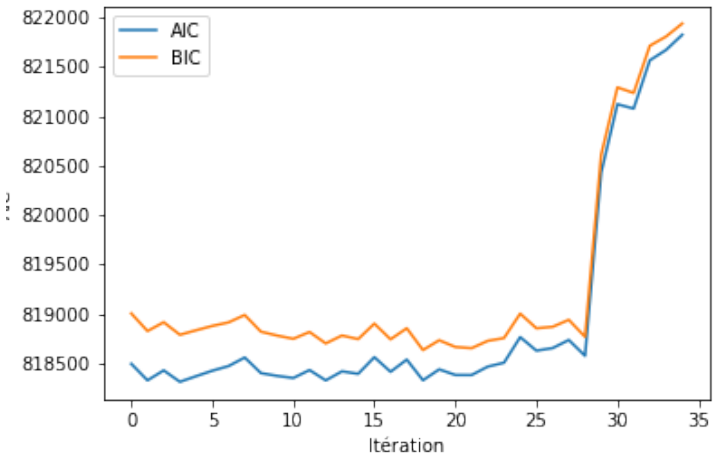


FIGURE 3.11 – Évolution des performances du modèle appliqué à la variable région.

certain nombre d'itérations cependant, les regroupements provoquent une trop grande perte d'information dans les données d'entraînement provocant une chute rapide des performances du modèle.

Les travaux présentés dans cette section sur le regroupement de modalité ne représentent qu'une ébauche et ont vocation à être continués. En outre, l'analyse des regroupements successifs ne nous a pas permis de mettre en évidence des structures dans les données. Une telle propriété serait pourtant souhaitable. Cela permettrait d'opérer des regroupements explicables et ainsi automatiser le travail fastidieux de pré-traitement des données. Nous proposons cet axe de recherche dans les perspectives de travaux.

3.5 Conclusions

Dans ce chapitre, nous avons présenté l'ensemble de nos travaux portant sur la gestion des événements rares dans le contexte de la prédiction de la probabilité du clic pour les enchères en temps réel. Nous avons montré les biais de prédiction des algorithmes d'apprentissage classiques. Nous avons également montré que les mesures d'évaluations classiques peuvent être trompeuses dans de telles conditions de rareté des événements. Nous montrons que les méthodes de correction des biais sont bel et bien efficaces pour la prédiction de la probabilité du clic. Nous validons ces conclusions par la présentation d'une mesure d'évaluation de la prédiction du clic reflétant les coûts associés aux différentes issues des enchères. Dans un deuxième temps et toujours dans une optique de réduction des

biais entraînés par la rareté des événements, nous avons présenté nos travaux sur l'étude des effets du sous-échantillonnage des non-événements sur les capacités prédictives des modèles. Les résultats obtenus montrent que ces effets sont très contenus et dégradent assez peu les performances des modèles de prédiction du clic étudiés ici nous permettant d'affirmer qu'appliquer une telle procédure pendant la phase de pré-traitement est souhaitable. Dans un troisième temps, nous avons présenté nos travaux sur la réduction de dimension. Nous avons notamment présenté un algorithme proche des approches de type fused lasso permettant la fusion de modalités intra-variable basées sur les coefficients d'une régression logistique. Ces travaux ont montré que dans le cadre du jeu de données Ipinyou, de nombreux regroupements peuvent être opérés avant de dégrader significativement les performances prédictives du modèle, permettant ainsi de drastiquement baisser la dimensionalité du jeu de données. Ces travaux constituent une piste de réflexion encourageante vers l'élaboration d'approches de réduction de dimension automatique et devront être approfondis en particulier dans l'étude du critère de choix de la coupe de l'arbre des regroupements mais également de la parallélisation de l'algorithme qui permettrait de grandement améliorer les temps d'entraînement.

Chapitre 4

Stratégies d'enchères et apprentissage par renforcement

La tâche de prédiction du clic, développée dans le chapitre précédent permet d'estimer l'utilité pour chaque requête d'enchère. Elle donne une probabilité de clic permettant à un enchérisseur d'estimer l'espérance du gain associé à un encart publicitaire en ligne. Cela dit, dans le cadre du RTB, il ne s'agit pas de remporter une seule enchère mais bien de maximiser le nombre de clics sous contrainte d'un budget donné et sur une période donnée. L'estimation de la probabilité de clic n'est ainsi que la première partie du problème plus large de l'optimisation d'une campagne d'affichage publicitaire en ligne. Dans un second temps, il faut proposer un prix en tenant compte de cette estimation mais aussi d'autres facteurs. Dépenser de gros montants pour remporter des clics très probables mais peu nombreux peut-être sous optimal, comme démontré par Zhang et al. [2014b], il peut être plus profitable de parier sur des impressions peu chères. Il peut aussi être souhaitable de ne pas dépenser le budget trop rapidement pour ne pas rater de meilleures opportunités en fin de campagne. Les annonceurs souhaitent également que les affichages s'étalent dans le temps permettant ainsi de toucher une audience plus large. Cela nécessite que la stratégie d'enchère contrôle également la vitesse de consommation du budget [Wu et al., 2018]. Le caractère hautement compétitif du RTB rend la distribution des prix hautement dynamique et doit aussi être pris en compte dans l'élaboration de la stratégie : celle-ci doit pouvoir adapter ses enchères aux changements dans la distribution des prix.

Sur ce point, les approches basées sur des modèles traditionnels [Perlich et al., 2012a, Zhang et al., 2014b] échouent à appréhender le caractère dynamique du RTB et peuvent donc rapporter moins de clics qu'espéré. Un deuxième type d'approche gagnant rapidement en popularité dans la communauté

de recherche en RTB repose sur l'apprentissage profond par renforcement (Deep Reinforcement Learning, DRL). L'apprentissage par renforcement est un domaine de l'apprentissage automatique où un agent interagit avec un environnement. Celui-ci doit choisir une action a_t parmi un ensemble d'actions possibles selon l'état de l'environnement s_t. Une fois l'action choisie, l'environnement transitionne vers un nouvel état s_{t+1} et renverra également la récompense r_t liée à cette transition. L'idée générale des approches par renforcement est d'apprendre de ces interactions afin d'optimiser le choix des actions permettant de maximiser les récompenses obtenues. Pour cela, le problème sur lequel on souhaite appliquer ce type d'algorithme est formulé sous la forme d'un processus de décision markovien (MDP) dont nous donnons la définition ci-après. Dans le cas du RTB, l'agent est l'enchérisseur, les actions disponibles contrôlent les montants des enchères et les récompenses correspondent aux clics. Pour ce qui est des états, cela devient plus compliqué : au contraire de certaines applications de l'apprentissage par renforcement comme le plateau d'une partie d'échecs où les états donnent une représentation complète de l'environnement, l'environnement du RTB n'est pas entièrement accessible. La formulation des états pour le RTB relève d'un choix. Cette formulation doit regrouper assez d'information afin de refléter les conditions actuelles de la campagne d'affichage (l'environnement) mais doit rester assez compacte pour qu'un agent puisse apprendre des actions prises et converger vers une stratégie optimale maximisant les clics. De nombreuses recherches ont appliqué l'apprentissage par renforcement afin d'optimiser les campagnes RTB [Cai et al., 2017, Jin et al., 2018b, Wu et al., 2018]. Ce type d'approche permet des stratégies d'enchères prenant en compte la dynamique de l'environnement ainsi que la gestion du budget et a montré de bons résultats aussi bien en offline qu'en online. Cependant, ces méthodes sont souvent difficiles à mettre en place de manière efficace, nécessitant notamment le choix d'une formulation des états pertinente mais aussi une recherche des meilleurs hyperparamètres fastidieuse ainsi que des temps d'entraînement longs.

Dans ce chapitre nous présentons nos travaux sur l'optimisation des campagnes RTB. Nous présentons différentes approches, certaines très répandues dans l'industrie et d'autres plus élaborées, notamment par renforcement. La première approche est l'enchère constante, consistant naïvement à toujours enchérir le même montant quelle que soit la probabilité du clic ou l'état de la campagne. La seconde approche que nous incluons dans nos recherches est l'approche que nous appelons bid linéaire qui consiste à enchérir proportionnellement à la probabilité du clic. Cette stratégie est répandue dans l'industrie et permet une meilleure adaptation des enchères à l'espérance du gain, elle ne peut

cependant pas s'adapter aux conditions changeantes au fur et à mesure de la campagne. La stratégie optimale peut en effet varier selon le budget restant, le nombre d'opportunités d'affichage restant ou encore la vitesse de dépense du budget, autant de facteurs que la stratégie du bid linéaire est incapable de prendre en compte. La stratégie de bid linéaire avec contrôle de la vitesse de dépense (Linear biding with budget pacing, LBBP) que nous incluons également dans nos études permet de palier cela, du moins pour la gestion du rythme de dépense du budget. Dans la même optique d'adaptation de la stratégie d'enchères aux dynamiques de l'environnement d'enchères, nous proposons également une approche utilisant l'apprentissage par renforcement afin d'optimiser le facteur d'ajustement des enchères de la stratégie linéaire à partir de l'expérience accumulée au fil des campagnes.

Nous étudions les performances obtenues par ces différentes approches sur le jeu de données Ipinyou. Nous combinons ces approches d'optimisation de la campagne d'enchères en temps réel avec les travaux sur la prédiction du clic présentés dans le chapitre précédent.

Dans une dernière partie, nous présentons un travail sur l'étude de l'adaptation des algorithmes d'apprentissage par renforcement sur le passage d'un système d'enchères en second prix à un système en premier prix (second to first price auction). Ce travail est motivé par le changement probable du mécanisme d'enchères dans l'industrie. Comme souligné en introduction, les enchères en temps réel pour l'affichage publicitaire en ligne sont historiquement basées sur un mécanisme d'enchères en second prix, ce qui signifie que le meilleur enchérisseur remporte l'objet mis en vente mais ne paye que le montant de la deuxième enchère la plus élevée. De plus en plus de plateformes d'échange se tournent vers un système en premier prix, où celui qui remporte l'objet paye effectivement le montant de son enchère, pour des raisons de transparence notamment [Despotakis et al., 2019]. Nos travaux sur cet aspect, cherchent à étudier la capacité des algorithmes d'enchères en temps réel développés pour du second prix à s'adapter naturellement à un système en premier prix.

4.1 Les enchères en temps réel comme processus de décision Markovien

Un processus de décision markovien [Puterman, 1990] est une structure de décision définie par quatre ensembles :

— S : L'ensemble des états possibles de l'environnement.

— A : L'ensemble des actions possibles afin d'interagir avec l'environnement.

— T : La matrice de transition $T : S \times A \to S$ qui représente les dynamiques de l'environnement : $T(s, a, s')$ est la probabilité d'arriver dans l'état s' en prenant l'action a depuis l'état s.

— R : La matrice de récompense $R : S \times A \times S \to \mathbb{R} : R(s, a, s')$ qui représente la récompense associée à l'action a prise depuis l'état s menant à l'état s'.

Ainsi on peut formuler le problème des enchères en temps réel sous la forme d'un processus de décision markovien tel que : les actions permettent de définir l'enchère et les récompenses sont les clics obtenus ou non. Comme souligné précédemment, la représentation de l'environnement ne peut qu'être incomplète par la nature même du système d'enchères du RTB puisque nous n'avons pas accès à l'ensemble des informations. Par exemple, l'accès au prix de vente d'un objet que l'agent n'a pas remporté n'est pas disponible. De même nous n'avons pas accès aux stratégies d'enchères des concurrents ni même de leur nombre. Il faut donc définir la formulation des états utilisée pour construire le processus de décision markovien. Dans un but d'optimisation, la formulation doit être choisie de manière réfléchie : d'une part elle doit fournir assez d'informations pour que l'agent puisse apprendre ce qu'est un bon ou un mauvais choix en fonction d'un état donné de l'environnement. D'autre part, elle ne doit pas être trop complexe : cela augmenterait la taille de l'espace d'état, donc diminuerait la probabilité de retrouver un état déjà visité et empêcherait un agent de déduire une bonne stratégie de son expérience accumulée. Cet équilibre est en pratique très compliqué à trouver.

Dans la suite de ce chapitre nous présentons les stratégies que nous avons utilisées dans nos études dont seule l'approche par renforcement dépend de la formulation des états. Nous reviendrons plus en détail sur le choix de formulation dans la section consacrée à cette approche.

4.2 Stratégies d'enchères

Nous présentons dans cette section, les trois stratégies d'enchères que nous incluons dans nos études. Tout d'abord, la stratégie d'enchères constantes est l'approche la plus naïve consistant à parier constamment le même prix quelle que soit la valeur, réelle ou estimée, de l'affichage. Cette stratégie nous donne une indication sur les performances minimales que l'on peut attendre d'un algorithme d'enchères. Comme deuxième approche, nous utilisons la stratégie de enchères linéaire (LinBid), qui, quant à elle, adapte le montant de l'enchère à sa valeur estimée en faisant intervenir le pCTR dans la fonction d'enchères. Cette approche est répandue dans l'industrie de l'affichage publicitaire en ligne et permet une meilleure allocation du budget à condition que le calcul du pCTR ne soit pas biaisé.

Cependant, comme souligné dans l'introduction de ce chapitre, cette stratégie d'enchères échoue à gérer la vitesse de dépense du budget. C'est pour résoudre ce problème que nous incluons également une stratégie dérivée des enchères linéaires faisant intervenir un terme de contrôle de la dépense du budget dans la fonction d'enchères désignée ici par LBBP (Linear bidding with budget pacing). Ce terme de contrôle a pour rôle de lisser la dépense du budget au cours de la campagne en influençant à la baisse les enchères si le rythme de dépense est trop élevé ou au contraire à la hausse s'il est faible. La dernière stratégie que nous utilisons est basée sur la même idée de gérer la dépense et les montants enchéris au cours de la campagne mais cette fois, de manière plus "intelligente". Comme indiqué en introduction, cette approche, que nous appelons λ-adjustments, est basée sur l'apprentissage par renforcement. Elle est en fait un dérivé de la stratégie linéaire, à la différence que le facteur multiplicatif de la valeur estimée de l'enchère est adapté périodiquement par l'agent au cours de la campagne. Cette adaptation de facteur permet ainsi de miser plus ou moins en fonction des conditions actuelles de la campagne. L'agent est ici entraîné à apprendre quels ajustements de ce facteur prendre selon l'état de la campagne afin de maximiser son revenu.

4.2.1 Déroulement d'une campagne RTB

Nous fournissons dans cette section les informations concernant le déroulement d'une campagne de RTB du point de vue d'un enchérisseur. Une campagne se déroule sur une période donnée (en général quelques jours). Celles-ci sont découpées en épisodes de taille fixe $e \in E$, une journée par exemple, et ces épisodes sont eux-mêmes divisés en *timesteps* $t \in T$ et le budget total pour la campagne est distribué uniformément pour chaque épisode B_e. Ce découpage des campagnes permet d'abord de suivre les performances d'enchères de manière plus précise mais surtout de mettre à jour la stratégie d'enchères à différentes échelles de temps. Cela pose cependant des questions sur la taille optimale des pas de temps ainsi que celle des épisodes. Cette question reste très peu discutée et il n'y a, à notre connaissance, pas de cadre permettant l'optimisation de ces tailles autrement que par recherche empirique, comme nous l'avons fait dans les travaux présentés ici.

4.2.2 Enchères constantes

La stratégie des enchères constantes est l'approche la plus naïve possible consistant à parier toujours le même montant :

$$b_{cst} = c$$

Dans nos travaux, ce montant d'enchère constant est fixé empiriquement à la valeur optimale par campagne calculée sur les données passées. Ces valeurs sont présentées dans le tableau 4.1.

TABLE 4.1 – Valeurs d'enchères optimales pour chaque campagne pour la stratégie d'enchères constantes

Campagne	Valeur
1458	33.70
2259	36.73
2261	26.22
2821	25.74
2997	48.12
3358	15.03
3386	27.66
3427	30.25
3476	37.24

4.2.3 Enchères linéaires avec contrôle du rythme de dépense (Linear Bidding with budget pacing, LBBP)

La stratégie des enchères linéaires ne prend pas en compte le taux de dépense du budget et ne parvient donc pas à adapter les offres aux conditions budgétaires. Par conséquent, comme Lee et al. [2013b], nous ajoutons, aux enchères linéaires, un terme de contrôle de la dépense (budget pacing) dépendant du ratio entre le budget restant $\frac{B_i}{B_e}$ et le ratio de temps restant dans l'épisode $\frac{t}{T}$. Ces deux termes vont augmenter les ordres si les conditions budgétaires sont élevées et au contraire les diminuer si elles sont faibles :

$$b_i = \alpha * \left(\frac{pCTR_i}{CTR_{avg}}\right) * \left(\frac{B_i}{B_e}\right) * \left(\frac{t}{T}\right) \tag{4.1}$$

4.2.4 Enchères linéaires (Linear Bidding)

Nous avons choisi l'algorithme d'enchères linéaires [Perlich et al., 2012b] car il est le plus utilisé dans le secteur du RTB :

$$b_i = \alpha * pCTR_i \tag{4.2}$$

où b_i est le montant proposé pour l'enchère i et α une constante. Dans (4.2), il est important de noter que puisque la $pCTR$ est très faible, elle implique un très grand α qui n'est pas très intuitif à définir. Pour résoudre ce problème, nous suivons Cai et al. [2017] et utilisons une fonction d'enchères linéaire normalisée définie comme suit :

$$b_{i,norm} = \alpha * \left(\frac{pCTR_i}{CTR_{avg}} \right) \tag{4.3}$$

où CTR_{avg} est le taux de clics moyen calculé à partir des données historiques. Le terme α est également choisi de manière empirique sur les données passées et sont présentées dans le tableau 4.2. Nous avons ainsi défini ces valeurs optimales en développant un algorithme de recherche dichotomique que nous ne présentons pas dans ce document.

TABLE 4.2 – α optimaux pour chaque campagne

Campagne	LR	wLR	DFM
1458	269.7	269.7	236.0
2259	69.1	57.6	69.1
2261	76.9	65.9	274.5
2821	91.7	103.2	91.7
2997	67.4	78.6	67.4
3358	211.7	185.2	211.7
3386	123.8	99.1	123.8
3427	190.2	190.2	570.5
3476	98.4	131.2	98.4

4.2.5 Optimisation des enchères par apprentissage par renforcement

Comme présenté précédemment, l'apprentissage par renforcement (RL) [Sutton and Barto, 1998] est capable de résoudre des problèmes complexes tels que les célèbres jeux Atari [Mnih et al., 2013b] et a été appliqué dans de nombreux domaines, les enchères en temps réel étant l'un d'entre eux [Cai et al., 2017, Wu et al., 2018, Zhao et al., 2018]. Nous incluons donc des algorithmes d'apprentissage par renforcement populaires dans cette étude.

Les algorithmes basés sur un modèle (RL) apprennent une représentation de l'environnement comme un processus de décision Markovien (MDP) $[S, A, P, R]$ en modélisant à la fois les probabilités de transition d'état P et la fonction de récompense R associée aux états $s \in S$ et aux actions $a \in A$. Nous fournissons une représentation du RTB comme MDP en figure 4.1 . Dans le cas d'enchères en temps réel, la formulation des états doit refléter les conditions de l'environnement telles que le budget restant, le nombre d'enchères restantes dans l'épisode ou le taux de consommation du budget, tandis que les actions correspondent aux prix des enchères. Il convient de noter qu'il existe un compromis entre la complexité de l'espace d'état pour modéliser l'environnement aussi précisément que possible et la variance des chemins possibles dans le MDP.

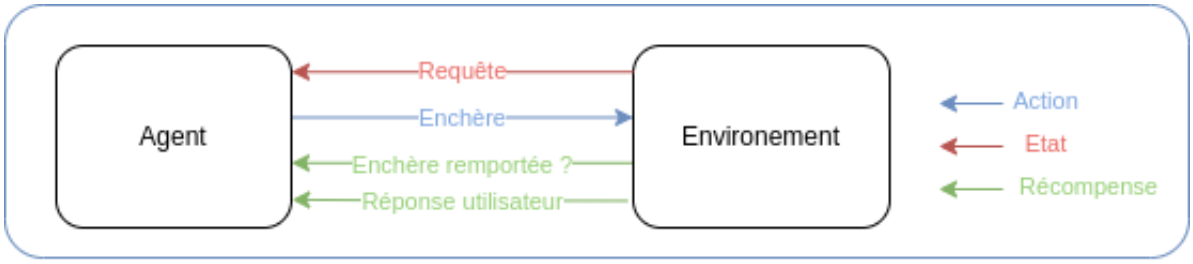


FIGURE 4.1 – Représentation processus Markovien du RTB

4.2.5.1 Apprentissage par renforcement

L'apprentissage par renforcement consiste à faire interagir un agent avec un environnement. Le but est de réussir à faire apprendre les dynamiques de l'environnement à l'agent de manière à maximiser la somme des récompenses. A chaque pas de temps t, l'agent reçoit la représentation de l'environnement appelée état $s_t \in S$ et doit choisir parmi des actions prédéfinies $a_t \in A$. Pour chaque action prise, l'agent reçoit la récompense r_t correspondant à la paire (a_t, s_t) ainsi que s_{t+1} correspondant au nouvel état dans lequel se trouve l'environnement après l'application de l'action a_t depuis l'état s_t. Ainsi on appelle épisode l'ensemble des étapes de $t = 0$ jusqu'à un état terminal, signalé par l'environnement $s_{t=terminal}$. L'agent est donc entraîné à maximiser la somme des récompenses obtenues durant un épisode.

Plus formellement, on appelle trajectoire notée $\tau = \{s_0, a_0, s_1, a_1, ..., s_t\}$ une suite d'états et d'actions. Le but est de trouver une politique (policy) π^* de paramètres θ^*, qui si suivie, donne une trajectoire τ^* qui rapporte la meilleure somme de récompense $R(\tau^*)$. On note $Q^*(s, a)$ la fonction ac-

tion/valeur qui depuis un état s et donné une action a, donne l'espérance de la somme des récompenses en suivant la politique optimale π^* telle que :

$$Q^*(s,a) = \max_{\pi} \mathop{\mathbb{E}}_{\tau\sim\pi}[R(\tau)|s,a] \tag{4.4}$$

Ainsi, la meilleure action à prendre depuis l'état s est définie par :

$$a^* = \arg\max_{a} Q^*(s,a)$$

De plus, on note $V^*(s) = \max_\pi \mathop{\mathbb{E}}_{\tau\sim\pi}[R(\tau)|s]$, la fonction de valeur qui donne la somme des récompenses attendue en suivant la stratégie optimale depuis l'état s.

A partir de ces équations nous pouvons dériver les équations de Bellman qui stipulent que la valeur associée à un état est la somme de la récompense immédiate $r(s_t, a_t)$ et de l'espérance de gain :

$$V^\pi(s_t) = \mathop{\mathbb{E}}_{a_t\sim\pi}[r(s_t,a_t) + \gamma V^\pi(s_{t+1})] \tag{4.5}$$

$$Q^\pi(s_t,a_t) = r(s_t,a_t) + \gamma \mathop{\mathbb{E}}_{a_{t+1}\sim\pi}[Q^\pi(s_{t+1},a_{t+1})] \tag{4.6}$$

De ces équations est né l'algorithme du Q-learning [Watkins and Dayan, 1992] qui repose sur l'apprentissage des Q-Values calculées grâce à la Q-function et l'équation de Bellman donnée dans eq. (4.7) pour chacun des états du problème.

$$Q^{new}(s_t,a_t) \leftarrow (1-\alpha)\cdot \underbrace{Q(s_t,a_t)}_{\text{ancienne valeur}} + \underbrace{\alpha}_{\text{taux d'apprentissage}} \cdot \overbrace{\Big(\underbrace{r_t}_{\text{récompense}} + \underbrace{\gamma}_{\text{facteur de remise}} \cdot \underbrace{\max_a Q(s_{t+1},a)}_{\text{estimation de la valeur future}} \Big)}^{\text{valeur apprise}} \tag{4.7}$$

Le terme γ est utilisé afin de spécifier le compromis entre la valeur d'une récompense immédiate comparée à une récompense dans le futur : plus γ se rapproche de 1, moins l'algorithme sera poussé à choisir une récompense immédiate contre une récompense future [François-Lavet et al., 2015].

L'apprentissage profond par renforcement introduit pour la première fois en 2013 [Mnih et al., 2013a] repose sur l'idée de faire estimer cette Q-function à un réseau de neurones artificiels, appelé Deep Q-Network (DQN) afin de ne pas avoir à calculer les nouvelles Q-value $Q^{new}(s_t,a_t)$ mais plutôt de les faire approximer par ce réseau de neurones $Q^{net}(s_t,a_t)$. Cela permet de généraliser l'apprentissage par renforcement à des problèmes plus complexes de manière efficace puisqu'il n'est plus nécessaire de recalculer les Q-values pour l'ensemble des paires états/actions à chaque étape comme c'est le cas pour le Q-learning.

En 2015, les équipes de *DeepMind* ont présenté dans une amélioration des DQN appelée Double DQN (DDQN) [Van Hasselt et al., 2016] consistant à utiliser une copie Q' du DQN Q afin de ne pas faire les prédictions à partir d'un réseau en cours d'apprentissage. Ainsi les poids θ' de Q' qui sert à faire les prédictions de valeur sont mis à jour de manière différée vers θ les poids de Q, le réseau servant à l'apprentissage tel que : $\theta \leftarrow \alpha_{DDQN}\theta + (1 - \alpha_{DDQN})\theta'$. Ce mécanisme permet de faire converger le modèle malgré l'évolution des "targets" au cours de l'entraînement en contrôlant la vitesse d'évolution de celles-ci. L'équation 4.7 se réécrit :

$$Q^{new}(s_t, a_t) \leftarrow (1 - \alpha) \cdot Q(s_t, a_t) + \alpha \cdot \left(r_t + \gamma \cdot \max_a Q'(s_{t+1}, a) \right) \tag{4.8}$$

Ce type d'algorithme est dit Off-policy (ou Value-based), cela signifie que la fonction état/valeur donnée dans l'équation 4.8 est mise à jour grâce à $\max_a Q'(s_{t+1}, a)$ où l'action a est choisie de manière "greedy". Ce choix d'action de la part de l'agent le pousse à n'apprendre que des actions qu'il a déjà effectuées dans une situation similaire, cela conduit à de mauvaises performances. Pour corriger cela, on ajoute aux algorithmes de type Q-learning, une dose d'aléatoire contrôlée dans le choix des actions, on appelle cela le compromis exploration/exploitation. Une des approches de contrôle de ce compromis est l'approche dite ϵ-greedy. Elle consiste à beaucoup explorer en début de jeu afin d'ajuster les valeurs pour différentes paires état/action et au fur et à mesure du jeu, de faire baisser la dose d'exploration au profit de l'exploitation.

Plus formellement, on note ϵ le paramètre contrôlant l'exploration initialisé à 1. A chaque état s_t, l'action choisie sera définie par :

$$a_t = \begin{cases} \max_a Q(s_t, a), & \text{avec une probabilité } 1 - \epsilon \\ random(a), & \text{avec avec une probabilité } \epsilon \end{cases} \tag{4.9}$$

L'algorithme commence donc par prendre des actions aléatoires pour explorer et petit à petit, il choisira de plus en plus les actions jugées les meilleures grâce à la fonction Q. On contrôle le taux de baisse de epsilon (epsilon decay rate) grâce à un paramètre α_{decay} à estimer empiriquement.

4.2.5.2 Apprendre les ajustements de λ (Learning λ-Adjustments, λ-Adj)

Dans l'article de Wu et al. [2018], les auteurs présentent un algorithme d'apprentissage par renforcement sans modèle qui permet de faire face à la grande volatilité du marché RTB, qui rend difficile le calcul ou la prédiction de la dynamique de transition à partir d'un modèle. Au lieu de cela, ils

formulent le problème comme un problème d'apprentissage par renforcement des ajustements de λ et le résolvent en utilisant l'algorithme Double Deep Q-network (DDQN). La fonction d'enchères est donc presque identique à celle des enchères linéaires :

$$b_i = \lambda_i * pCTR_i \tag{4.10}$$

Au lieu d'être réglé manuellement, le terme de contrôle des enchères λ_i est ici mis à jour à chaque nouveau pas de temps par un DDQN [Mnih et al., 2013b] ainsi que d'un compromis exploration-exploitation ϵ-greedy. Nous définissons sept actions possibles correspondant à un ajustement de la valeur de λ_i : $a_i = [-8\%, -3\%, -1\%, 0\%, +1\%, +3\%, +8\%]$ avec la même valeur initiale que pour les enchères linéaires. Cet espace d'action est consistant avec l'approche présentée par Wu et al. [2018]. A chaque nouvelle requête d'enchère, l'agent sélectionne une action qui augmente ou baisse l'enchère selon :

$$\lambda_i = \lambda_{init} * (1 + a_i)$$

Ainsi, notre implémentation de cet algorithme diffère de celle présentée par Wu et al. [2018] car nous utilisons une fonction de récompense classique au lieu de la fonction de récompense approximée par un réseau de neurones introduite par ces auteurs. Cette simplification permet une convergence plus rapide de l'algorithme.

Comme souligné précédemment dans la description de la formulation du problème sous forme de MDP, nous n'avons pas accès à la totalité de l'information concernant l'état de l'environnement. Il est donc nécessaire de faire un choix sur la formulation des états. Cette formulation doit donner à l'agent une représentation assez complète de l'état actuel de la campagne tout en restant suffisamment compacte afin de limiter la taille de l'espace des états. En effet, bien que l'approximation de la fonction de valeur par un réseau de neurones profond dans l'apprentissage par renforcement permette l'optimisation de problèmes à l'espace d'état plus complexe, il est toujours nécessaire de contrôler la taille de cet espace. Un espace d'état trop complexe diminue la probabilité pour l'agent de visiter des états qu'il aurait déjà visités et sur lesquels il aurait une bonne estimation de valeur. Plus la formulation des états sera complexe plus le besoin d'exploration sera grand et la partie exploitation s'en trouvera diminuée conduisant à une baisse de la somme des récompenses.

Dans les travaux présentés dans cette section, la définition de la formulation des états est choisie de manière empirique. Il ressort que pour notre implémentation, la formulation des états permettant les

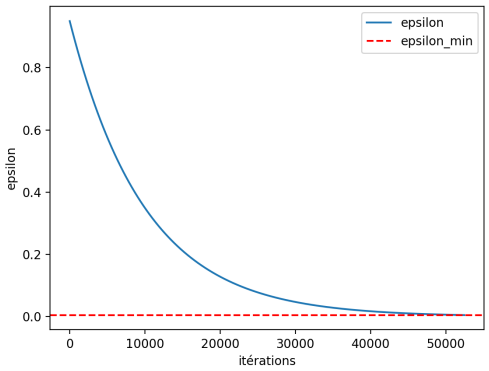


FIGURE 4.2 – Graphique montrant la courbe de la valeur de ϵ contrôlant le compromis entre exploration et exploitation. La valeur initiale est $\epsilon = 0.95$, le facteur de diminution est $\epsilon - decay = 0.0001$ et la valeur minimale est $\epsilon - min = 0.005$. Cette valeur minimale est atteinte au bout de 52 469 itérations.

meilleurs performances de l'algorithme λ-ajustements est un triplet $s =< Budget\_ratio, Bids\_left, pCTR >$. Sur la base de ces informations, l'agent prendra la décision d'augmenter ou de diminuer la mise pour les enchères courantes, selon la probabilité estimée du clic, la vitesse de consommation du budget ainsi que le nombre estimé d'opportunités restantes. Cette formulation est en plus cohérente avec les travaux sur le même sujet [Jin et al., 2018a, Liu et al., 2019, Wu et al., 2018]. Nous soulignons cependant qu'il n'existe pas à notre connaissance de travaux sur la formulation des états pour le RTB. Nous présentons nos travaux sur l'apprentissage automatique de la formulation des états d'un MDP pour l'apprentissage par renforcement dans le chapitre 5.

4.2.6 Résultats et discussion

Dans nos études sur le DDQN, chaque campagne est divisée en épisodes de 10000 enchères. Les valeurs de départ de λ optimales ont été calculées pour chaque campagne et sont présentées dans le tableau 4.2. Le taux d'apprentissage est fixé à 0,001 et les poids du réseau cible sont mis à jour toutes les 150 itérations. Le compromis exploration-exploitation est réalisé avec ϵ-greedy dont la courbe est montrée en figure 4.2. Le modèle d'inférence est entraîné toutes les 1000 enchères et les poids de ce modèle sont transférés au modèle cible (*target network*) toutes les 5 enchères.

Ces valeurs ont été définies de manière empirique (grid search).

Nous présentons dans le tableau 4.3 les performances des approches étudiées dans cette étude exprimées sous différentes fonctions d'évaluation. On peut y voir que ces dernières se contredisent

TABLE 4.3 – Tableau de performance des algorithmes de prédiction du CTR sous différentes mesures.

Campagne	Algorithme CTR	AUC	logloss	Precision Recall
1458	LR	0.96	**0.04**	**0.23**
	wLR	**0.97**	0.21	0.21
	DFM	0.64	0.11	0.00
2259	LR	0.77	**0.10**	0.00
	wLR	**0.78**	0.58	0.00
	DFM	0.64	0.12	0.00
2261	LR	0.79	0.11	0.00
	wLR	**0.82**	0.47	0.00
	DFM	0.60	**0.08**	0.00
2821	LR	0.70	**0.10**	0.00
	wLR	**0.71**	0.61	0.00
	DFM	0.61	**0.10**	0.00
2997	LR	0.68	**0.10**	**0.01**
	wLR	**0.69**	0.63	**0.01**
	DFM	0.51	0.11	0.00
3358	LR	0.95	**0.05**	**0.11**
	wLR	**0.96**	0.26	0.08
	DFM	0.71	0.10	0.00
3386	LR	0.75	**0.09**	**0.01**
	wLR	**0.76**	0.57	**0.01**
	DFM	0.65	0.10	0.00
3427	LR	0.93	**0.05**	**0.17**
	wLR	**0.94**	0.32	0.06
	DFM	0.63	0.11	0.00

dans la plupart des cas. L'AUC donne toujours wLR comme meilleur modèle alors que d'après la logloss, ce serait LR. La mesure Precision/Recall, quant à elle a tendance à aller dans le sens de la logloss mais ne crédite les modèles que de performances extrêmement basses quand l'AUC est, en général, très proche de 1, créditant les modèles de très bonnes performances. Ces observations sur les différences entre les différentes métriques d'évaluation confirment qu'elles ne sont pas suffisantes pour la sélection de modèle de prédiction du clic pour le RTB.

Afin de pouvoir mieux choisir le modèle de prédiction, nous fournissons dans le tableau 4.4 les performances croisées entres les algorithmes de CTR et les différentes stratégies d'enchères. Par souci de clarté dans ce manuscrit, nous ne fournissons ici que la somme des clics totaux sur les neuf campagnes, le

détail des clics par campagne est laissé en annexe C. En observant les résultats sur les clics, on remarque que l'approche par DDQN est l'approche qui obtient les meilleurs résultats, à part sur les probabilités de clics calculées par l'algorithme DFM. Cela peut être expliqué par les faibles performances de prédiction du clic obtenues par cet algorithme. Sans surprise l'algorithme d'enchères constantes est loin derrière les deux autres approches. De même, l'approche LBBP obtient des résultats médiocres, parfois encore moins bons que ceux de l'approche constante dans le cas de l'approche wLR comme prédicteur CTR. Cela s'explique par le fait que le terme de lissage de la dépense du budget prend trop d'importance sur la probabilité du clic dans la formulation du montant de l'enchère. L'approche classique de régression logistique donne de bien meilleurs résultats en termes de nombre de clics (près de deux fois plus) que l'approche par régression logistique pondérée. Les trois approches d'optimisation des enchères donnent également de bien meilleurs résultats avec la prédiction du clic basé sur DFM par rapport à wLR.

Ces résultats vont à l'encontre des ceux obtenus dans le chapitre précédent où la version pondérée de la régression logistique donnait les meilleurs résultats. Nous expliquons ce retournement dans les résultats par le fait que dans le chapitre dédié à la prédiction du clic, nous utilisons un algorithme d'enchères parfait afin de calculer notre fonction de coût spécifique au RTB. En pratique, avec un algorithme d'enchères imparfait, comme c'est le cas ici, la régression logistique qui polarise moins les prédictions permet des montants d'enchères également moins polarisés et donc de remporter des clics peu chers qu'un algorithme d'enchères se basant sur la régression pondérée aurait ignoré.

La figure 4.3 présente les valeurs de V sous différentes valeurs de v. On observe sur ces figures que V augmente linéairement avec v. Une valeur négative de V peut être interprétée comme une perte, ainsi, l'ordonnée à l'origine de chaque droite peut être considéré comme l'équilibre coût/récompense. De manière générale, pour toutes les paires d'algorithmes d'enchères/algorithmes CTR, la fonction de valeur V n'est négative que pour des valeurs très basses de v. Les campagnes deviennent donc assez rapidement rentables. Ces graphes montrent la supériorité de l'approche de prédiction de la probabilité de clic par régression logistique classique, allant à l'encontre des résultats obtenus avec un algorithme d'enchères parfait. Ce résultat est cependant consistant avec ce que l'on observe dans le tableau 4.4. On observe que, excepté pour le cas particulier de l'approche des enchères constantes, la régression logistique classique est supérieure aux deux autres approches en termes de fonction de coût V. On observe également que les fonctions de coûts obtenues dans chacun des algorithmes d'enchères en temps réel sont très similaires, ce qui signifie qu'aucune de ces approches ne permet d'incrément

TABLE 4.4 – Résultats sur la prédiction du CTR et sur l'optimisation du bid.

Algorithme CTR	Algorithme d'enchères	Clics
LR	Constant	794
	LBBP	2354
	Linéaire	6864
	DDQN	**6867**
wLR	Constant	794
	LBBP	711
	Linéaire	3446
	DDQN	**3498**
DFM	Constant	794
	LBBP	819
	Linéaire	**4312**
	DDQN	4267
	Total de clics potentiels	**11557**

substantiel de performance pour les campagnes de RTB étudiées ici.

Le tableau 4.5 donne pour chacun des algorithme d'enchères et des algorithmes de CTR la valeur de la fonction de coût V pour une valeur de clic associée $v = 13333$ correspondant au coût moyen d'un clic sur le marché [1]. Ce tableau nous montre ainsi les différents gains rapportés par chacune des paires d'approches étudiées ici, pour valeur de clic fixée au prix moyen du marché. On y observe logiquement la même hiérarchie que sur la figure 4.3, mais avec tableau nous permet de comparer numériquement les approches et de voir que pour une probabilité de clic prédite par régression logistique pondérée, le meilleur modèle d'enchères est l'approche par renforcement DDQN ce qui n'était pas directement visible sur la figure. Même l'écart avec l'approche linéaire reste très faible, confirmant le peu de gain supplémentaire apporté par l'approche DDQN. Cela dit, le cas intéressant des performances obtenues par la paire DDQN/wLR montre tout de même la capacité de l'apprentissage par renforcement à optimiser sa stratégie au cours de la campagne, permettant de surpasser les trois autres approches.

1. `https://www.wordstream.com/cost-per-click`

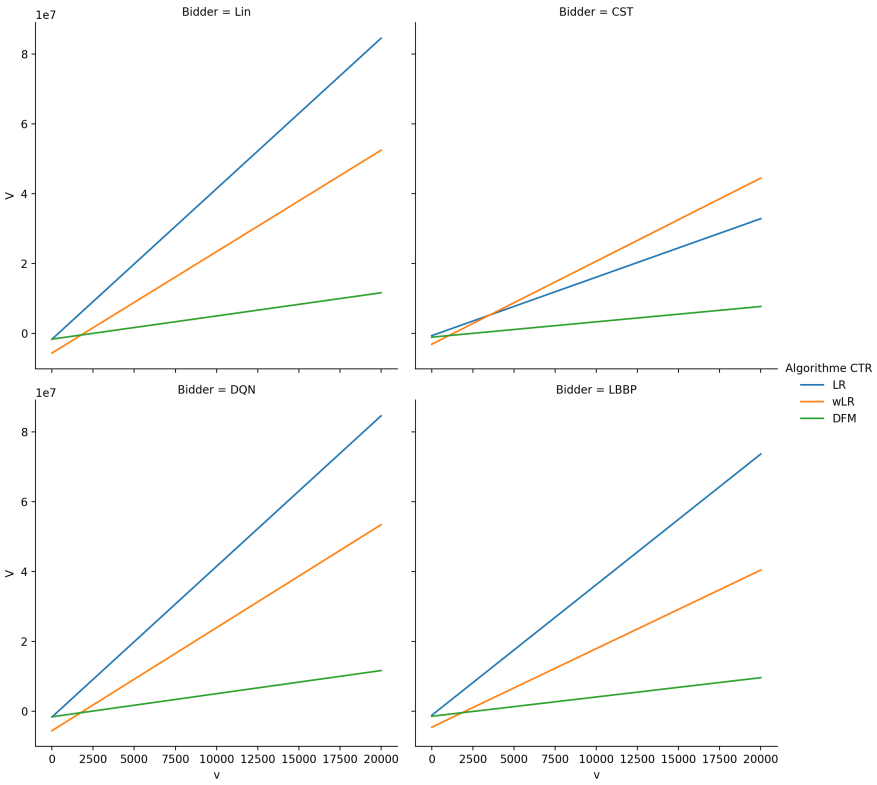


FIGURE 4.3 – Fonction de valeur V pour chaque algorithme d'enchères et chaque modèle de prédiction du CTR. En Abscisse, v, est la valeur associée à un clic. Les fonctions tracées sur le graphe prennent en compte l'ensemble des données d'enchères des neuf campagnes réunies.

TABLE 4.5 – Valeur de $V_{v=13333}$ pour chacune des paires algorithme d'enchères/algorithme CTR

Algorithme CTR	algorithme d'enchères	V
LR	CST	2.17e+07
	DDQN	**5.58e+07**
	LBBP	4.87e+07
	Linéaire	**5.58e+07**
wLR	CST	2.86e+07
	DDQN	**3.37e+07**
	LBBP	2.54e+07
	Linéaire	3.31e+07
DFM	CST	4.77e+06
	DDQN	7.18e+06
	LBBP	5.87e+06
	Linéaire	**7.20e+06**

4.3 Passage des enchères du second au premier prix (first to second price auctions)

En mars 2019, Google a annoncé le passage de son système d'enchères Google Ad Manager des enchères du second au premier prix. Avant eux, certains des principales ad exchanges du marché avaient déjà basculé sur les enchères au premier prix.

Ce changement est motivé par des raisons de simplicité et de transparence lors du processus d'enchères en temps réel. Ainsi, auparavant, dans les enchères en second prix, les intermédiaires comme les DSP (Demand Side Platforms) et les SSP (Supply Side Platforms) étaient incités à tricher. Par exemple, une DSP pouvait participer à des enchères au nom d'un annonceur et modifier artificiellement le prix du marché en augmentant le coût pour son client.

En outre, Google a annoncé que ce changement lui permettait de fournir beaucoup plus d'informations aux acheteurs et aux vendeurs, comme les logs des enchères pour les vendeurs qui pourraient ainsi évaluer leur inventaire avec plus de précision ou le prix du marché après la clôture des enchères pour aider les acheteurs à affiner leurs stratégies d'enchères.

Bien que de nombreuses personnes affirment que le passage des enchères au second prix aux enchères au premier prix a pour but d'accroître la transparence, Despotakis et al. [2019] proposent un modèle d'enchères théorique et soutient que le passage est davantage lié au passage du waterfalling aux enchères par en-tête (header bidding) qu'à un besoin de transparence. Le waterfalling consiste pour les éditeurs à avoir une hiérarchie dans les DSP auxquelles ils proposent leur inventaire. La DSP qui a payé le contrat le plus cher avec le fournisseur est invitée à faire une offre en premier lieu, puis si cette DSP ne fournit aucune offre ou si le prix de réserve n'est pas atteint, le fournisseur en propose un deuxième et ainsi de suite jusqu'à ce que l'offre ait trouvé preneur. Récemment, avec la croissance de la publicité mobile, les enchères par en-tête sont apparues comme une alternative crédible au waterfalling : une demande d'offre est générée directement dans l'en-tête HTML et présentée publiquement sans hiérarchie. D'après Despotakis et al. [2019], ce changement est à l'origine du passage des enchères au second prix aux enchères au premier prix.

Nous présentons dans cette section une étude sur l'adaptation des algorithmes de RTB à ce changement.

4.3.1 Résultats

Nous simulons des campagnes d'enchères en temps réel avec différentes conditions de budget pour les deux systèmes : premier et second prix afin de comparer les capacités d'adaptation des algorithmes d'enchères linéaires, (non-apprenant) et du DDQN (apprenant). Les différentes conditions de budget correspondent à une fraction de la somme des prix de l'ensemble des enchères dans le jeu de données et permettent d'étudier la capacité de gestion des algorithmes sous des conditions de budget plus ou moins tendues.

La version du dataset utilisé dans ces études est celle disponible en accès libre [2] avec le $pCTR$ pré-calculé et les résultats présentés dans le tableau 4.6 sont obtenus en testant les modèles sur 3 campagnes contenant au total 1 050 000 demandes d'enchères avec un maximum de 706 clics. Concernant l'algorithme entraînable DDQN, sur le premier et le second prix, un modèle est entraîné pour chaque condition de budget.

TABLE 4.6 – Somme des clics obtenus en premier et second prix

Budget	**Algorithme**	premier prix		second prix	
		Clics	**Budget consommé**	**Clics**	**Budget consommé**
1/4	LinBid	**299**	34.2%	209 (-30%)	69.0% (+101.7%)
	LBBP	288	63.6%	239 (-64%)	82.9% (+30.2%)
	DDQN	292	32.3%	**220** (-24%)	67.6% (+109.0%)
1/8	LinBid	**181**	52.4%	93 (-48%)	76.5% (+45.8%)
	LBBP	179	67.9%	136 (-24%)	78.6% (+15.6%)
	DDQN	175	53.7%	**98** (-44%)	76.2% (+41.8%)
1/16	LinBid	**85**	65.6%	51 (-40%)	80.9% (+23.2%)
	LBBP	**113**	69.8%	75 (-33%)	78.7% (+12.7%)
	DDQN	82	66.0%	52 (-36%)	80.8% (+22.3%)
1/32	LinBid	39	72.8%	**27** (-30%)	83.2% (+14.2%)
	LBBP	**51**	71.1%	34 (-33%)	78.7% (+10.7%)
	DDQN	**40**	72.9%	**27** (-32%)	83.1% (+13.9%)

4.3.2 Discussion

Comme on pouvait s'y attendre, nous observons une chute sévère des performances sur toutes les caractéristiques cibles. Nous observons dans les performances de λ-adj qu'il ne parvient pas à gagner autant que les enchères linéaires. Nous pouvons expliquer cela de plusieurs façons.

Le CTR prédit peut souvent être trompeur : un CTR faible menant à un clic et vice-versa génère des valeurs aberrantes dans les données d'entraînement et pourrait empêcher les modèles de converger

2. `https://github.com/han-cai/rlb-dp`

efficacement. La convergence dans l'apprentissage par renforcement n'est pas triviale bien qu'elle soit théoriquement fiable [Jacot et al., 2018, Lillicrap et al., 2016, Mnih et al., 2013b]. Les besoins de grosses quantités de données d'entraînement et l'instabilité des algorithmes qui sont très sensibles à la formulation du problème et au réglage des paramètres peuvent être difficiles à surmonter dans un contexte réel tel que les enchères en temps réel. D'une manière générale, la notion de robustesse dans les réseaux de neurones reste un sujet ouvert [Buşoniu et al., 2018, Carlini and Wagner, 2017] et représente une voie intéressante pour de futurs travaux La formulation de l'espace d'état telle que nous l'avons choisie est peut-être trop partielle pour que les modèles puissent apprendre correctement. Wu et al. [2018] utilisent une formulation d'état plus complexe qui peut aider le modèle à converger. La formulation de l'espace d'état dans les problèmes d'apprentissage par renforcement est cruciale et reste un sujet ouvert [Carrara et al., 2019, Doya, 2000, Herrmann and Der, 1999, Moore, 1993].

De plus, l'extrême rareté des clics rend l'amélioration des actions plus difficile pour les modèles [Agarwal et al., 2019, Riedmiller et al., 2018b]. Nous questionnons également la formulation de la fonction de récompense [Cai et al., 2017, Wu et al., 2018] dont nous savons qu'elle est d'une grande importance pour les problèmes d'apprentissage par renforcement et développons ce point dans la suite. Cette difficulté pour les algorithmes de renforcement à converger est une réelle préoccupation car le temps d'apprentissage requis dans un contexte réel est une pure perte de budget pour les annonceurs. Cela pose le problème du compromis entre la performance et la capacité à converger plus rapidement.

Globalement, les performances de tous les algorithmes que nous avons inclus dans cette étude chutent de manière significative lors du passage du deuxième au premier prix. Nous constatons également une augmentation spectaculaire du coût par clic pour tous les algorithmes, ce qui signifie qu'ils ont tendance à dépenser davantage dans des offres peu rentables. Cela était attendu pour les algorithmes sans apprentissage puisqu'avec ce changement de système, le prix du marché augmente mais on pouvait s'attendre à ce que les algorithmes d'apprentissage par renforcement s'adaptent mieux en trouvant par exemple la stratégie connue sous le nom de bid shading consistant à baisser les offres afin de maintenir le prix du marché dans les enchères de premier prix proche de celui en deuxième prix. La conclusion selon laquelle les algorithmes étudiés ne parviennent pas à découvrir le bid shading est confirmée par les données de consommation budgétaire. Nous voyons dans 4.6 que les algorithmes ont tendance à dépenser plus dans les enchères en premier prix tout en obtenant moins de clics. Ainsi, d'un point de vue économique, le changement aura des effets négatifs sur les revenus d'affichage en

ligne des annonceurs dans un avenir proche et des travaux doivent être effectués pour améliorer le comportement des algorithmes.

4.4 Conclusions

Nous avons présenté dans ce chapitre la synthèse des travaux effectués sur la prédiction du clic couplée aux travaux sur l'optimisation des enchères pour les campagnes publicitaires en ligne. Nous avons présenté les différentes approches étudiées, de la stratégie naïve des enchères constantes à l'utilisation d'algorithmes plus élaborés comme l'approche apprenante du double DQN conçue pour optimiser ses décisions dans le sens de la maximisation du nombre de clics obtenus. Nous avons comparé les performances de ces approches sur les probabilités de clic prédites par les différents algorithmes étudiés dans le chapitre 3. Ce croisement des travaux dans les deux principales tâches des enchères en temps réel montre que la prédiction du clic par l'approche de régression logistique pondérée ne convient pas pour une utilisation dans le contexte de l'optimisation des enchères, obtenant de moins bons résultats pour les trois approches classiques testées. Les prédictions ayant tendance à être mécaniquement sur-polarisées par cette approche, cela met en difficulté ces approches basées essentiellement sur la prédiction du clic sans capacité d'adaptation. Cette conclusion se vérifie également lorsque l'on regarde les performances mesurées par la fonction d'évaluation sensible aux coûts introduite dans le chapitre 3. L'apprentissage par renforcement produit quant à lui, de meilleurs résultats avec l'approche wLR montrant une certaine capacité d'apprentissage et d'adaptation. Nous avons également présenté une étude sur l'adaptation des algorithmes d'enchères d'un passage du premier au second prix. Celle-ci a montré qu'une nouvelle fois, l'algorithme d'apprentissage du DDQN permet une meilleure adaptation. Ces deux études nous permettent de conclure que l'apprentissage par renforcement est une approche prometteuse pour les enchères en temps réel en garantissant une bonne résilience ainsi qu'une adaptation certaine au contexte et à l'évolution des conditions. Le caractère hautement dynamique des enchères en temps réel dans le monde réel fait pourra être adressé par ce type d'algorithme. Nos expériences montrent que le gain est toutefois assez limité par rapport à des approches plus classiques et que la difficulté de formulation du problème ainsi que les problèmes de convergences et de très grands espaces d'hyper-paramètres en fait des approches complexes à mettre en place. Une des pistes clés vers l'amélioration des algorithmes d'apprentissage par renforcement consiste à redéfinir la fonction de récompense afin de pénaliser l'épuisement trop rapide du budget. Une des solutions proposée par Wu

et al. [2018] serait d'entraîner un réseau neuronal en parallèle du DQN qui aura pour but d'apprendre à dévaluer les récompenses renvoyées par l'environnement dans le cas où celles-ci pourraient conduire à un épuisement prématuré du budget. L'article introduit également une piste de réflexion quant à la formulation même de la récompense. L'introduction d'un facteur de pénalisation intégrant le coût dans la fonction de récompense $r_t^{'} = r_t + \alpha c_t$ pourrait empêcher la surévaluation par le DQN des actions menant à des récompenses directes élevées au détriment de potentielles meilleures enchères. Cette solution soulève toutefois un problème d'optimisation du paramètre α qui contrôle l'importance accordée au coût dans cette fonction.

Chapitre 5

Formulation des états et convergence de l'apprentissage par renforcement : étude de cas sur Atari-Breakout

L'importance de la formulation du problème à optimiser sous forme de Processus de Décision Markovien (MDP) revêt une importance cruciale afin de pouvoir lui appliquer un algorithme d'apprentissage par renforcement. Cette formulation, dans le cadre du RTB résulte d'une réflexion de la part du développeur exigeant un travail important et de nombreux essais et calibrations. Nous proposons une étude sur l'apprentissage automatique de la formulation de ces états afin de palier cela. Ces travaux ne sont cependant pas directement effectués sur le RTB. Nos études sur l'optimisation des campagnes d'affichage publicitaire en ligne nous ont montré à quel point ce problème est complexe et difficile à appréhender par les algorithmes d'apprentissage par renforcement. Afin d'échapper à cette complexité et pouvoir étudier l'apprentissage automatique de la représentation des états en détail, nous utilisons le jeu Atari-Breakout comme une abstraction simplifiée du problème du RTB.

Le jeu de casse-brique de Atari disponible dans la librairie *AiGym*[1] est un jeu où le joueur contrôle une raquette en bas de l'écran (c.f. figure 5.1) que l'on peut déplacer horizontalement afin de rattraper une balle que l'on renvoie en haut de l'écran afin de casser des briques. La partie démarre avec un total de cinq vies, que le joueur perd à chaque fois qu'il n'arrive pas à rattraper la balle. Au contraire, le joueur gagne des points à chaque brique cassée, le but du jeu étant de réussir à toutes les casser.

Ce jeu se prête très bien à l'apprentissage par renforcement. Les actions sont les déplacements : ne

1. `https://www.gymlibrary.ml/`

pas bouger, aller à droite ou aller à gauche. Les récompenses correspondent aux points engrangés. Les états peuvent ici prendre deux définitions différentes selon la version du jeu utilisée : dans *breakout-v0*, la dimension des états est de (210, 160, 3), correspondant à une image de 210 par 160 pixels en RGB. Dans *breakout-ram-v0*, la dimension des états est de 128 correspondants à la taille de la RAM de la console qui accueillait le jeu à l'origine. L'ensemble du jeu est donc contenu dans ces 128 octets, ce qui signifie qu'ils contiennent l'entièreté de l'information de l'état du jeu à chaque instant. En utilisant ces 128 octets, l'agent devrait donc être en mesure de déduire les corrélations entre les états lui permettant de prendre les meilleures décisions. Les travaux présentés ici sont réalisés avec la version RAM afin d'éviter les convolutions et d'alléger les calculs.

FIGURE 5.1 – Capture d'écran du jeu Breakout

Nous présentons dans cette section les études menées sur l'étude de la convergence de l'algorithme du Double DQN (DDQN) [Van Hasselt et al., 2016] ainsi que l'influence des hyperparamètres. Nous proposons également d'étudier plusieurs espaces d'action : de base, la version du jeu propose quatre actions les déplacements plus une touche "tirer", qui n'a pas d'effet sur le jeu. Nous avons étudié les performances de l'algorithme avec les quatre actions et les comparons avec un espace d'action de dimension trois. Transposée au RTB où les actions disponibles sont définies à priori, cela nous donne des indications supplémentaires sur la construction de la formulation du problème afin de faciliter la convergence de l'apprentissage par renforcement.

Enfin, nous proposons une étude sur l'apprentissage automatique de la formulation des états. Comme souligné plus tôt dans ce chapitre, concernant le RTB, le choix de la formulation des états n'est pas trivial. Nous avons donc mené une étude sur l'apprentissage automatique de cette formulation profitant de la relative simplicité du jeu *breakout-ram*. Nous proposons deux approches d'apprentissage de représentation par autoencoder : avec et sans contrainte sur la formulation apprise. La contrainte que nous proposons consiste à traduire le fait que deux états successifs devraient être proches dans l'espace des états. La contrainte agit donc comme un terme de pénalisation de la distance entre deux états successifs dans la fonction objective de l'encodage afin de forcer l'autoencoder à produire un encodage respectant cette contrainte.

Dans la suite de nos travaux, le score présenté est la moyenne des scores obtenus sur les 100 dernières vies. Ce choix est justifié par l'instabilité de l'apprentissage par renforcement qui peut passer d'une très bonne partie à une très mauvaise notamment à cause de la part d'actions aléatoires induites par le mécanisme d'exploration.

5.1 Étude des hyper-paramètres et de la convergence

Nos études sur l'application de l'apprentissage par renforcement au RTB nous ont appris l'importance des hyper-paramètres sur la convergence de ce type d'algorithme. C'est pour cela que nous proposons dans cette section, une analyse de l'influence de ces hyperparamètres sur la convergence du DDQN appliqué au jeu *Atari Breakout-ram*. Nous utilisons ce jeu comme une abstraction simplifiée afin d'échapper à la complexité et aux temps de calculs d'études menées directement sur les données de RTB Ipinyou.

L'apprentissage par renforcement est extrêmement sensible aux hyperparamètres et ceux-ci doivent être choisis par recherche en grille (gridsearch). Le tableau 5.1 liste les hyperparamètres dont nous avons étudié l'influence ainsi que les valeurs testées. Pour des raisons de clarté, nous ne présentons ici que les paramètres correspondants aux 50 meilleurs essais. Les intervalles de valeurs testés ont été fixés empiriquement. Nous avons d'abord lancé des recherches en grille avec des valeurs très larges puis avons recentré les recherches autour des meilleures valeurs. Nous présentons ici les derniers intervalles des recherches en grilles auxquels nous sommes arrivés. Dans nos études, un épisode correspond à une partie, à savoir les cinq vies. La récompense pour un épisode est donc la somme des récompenses

TABLE 5.1 – Hyperparamètres et les valeurs testées pour les 50 meilleurs essais.

Hyper-paramètre	Description	Valeurs testées
Batch_size	Taille des batch a sampler dans la replay memory	32, 128, 256
Discount_factor	Difference entre reward immediate ou future	0.99
Eps_decay	Taux de décroissance de ϵ	0.99999, 0.999999
Eps_min	Valeur minimale de ϵ	0.001, 0.005, 0.01
Update_rate	Fréquence d'entrainement du réseau	1, 2, 4, 5, 6, 10
Replay_mem_size	Taille du buffer de memoire	2 000 000, 5 000 000, 10 000 000
Soft_up	Contrôle la vitesse de mise à jour du target network	0.0001, 0.001, 0.002, 0.003 , 0.005 , 0.009
LR	Taux d'apprentissage du DQN	0.00025
RL_Network_shape	Structure du DQN	[128, 100, 50] [128, 512, 512, 100] [128, 1024, 1024, 100] [128, 50, 50, 20, 10] [128, 50, 50, 50, 50] [128, 50, 50, 50] [128, 100, 100, 100]

obtenues au cours de la partie. Chaque *Timestep* correspond ici à une nouvelle image correspond à un nouvel état s_t dans lequel l'agent doit choisir une action a_t.

Nous présentons la courbe des récompenses cumulées de l'agent ayant obtenu le meilleur score (voir tableau 5.2 pour les paramètres utilisés) en figure 5.2.

5.2 Influence de la taille de l'espace d'action sur la convergence du DDQN

Comme souligné dans l'introduction de ce chapitre, l'action du jeu correspondant au tir ne s'applique pas dans notre cas. Cette action est donc ici équivalente à l'action de ne rien faire. Cela augmente la dimension de l'espace d'action et ajoute donc en complexité dans l'apprentissage de la meilleure politique par l'agent. Nous proposons de comparer les performances d'un agent entraîné sur le jeu d'origine à quatre actions et celles d'un autre entraîné uniquement sur trois afin de mettre en évidence le gain potentiel de la réduction de dimension de l'état d'action.

Le tableau 5.2 présente les meilleurs paramètres et le meilleur score des agents pour les deux espaces d'actions. Les paramètres permettant d'obtenir les meilleurs résultats sont exactement les mêmes dans les deux cas et les meilleurs scores sont assez proches, l'agent évoluant avec quatre actions n'obtenant le meilleur score qu'à 1.09 (+2.5%) près. La figure 5.3 qui représente les courbes d'évolution du score des agents pour les meilleurs essais et pour les deux espaces d'actions. L'approche à trois actions permet de converger et d'atteindre le score maximum bien avant celle à quatre actions (2906 parties avant) mais diverge ensuite et chute très bas perdant la stratégie précédemment apprise.

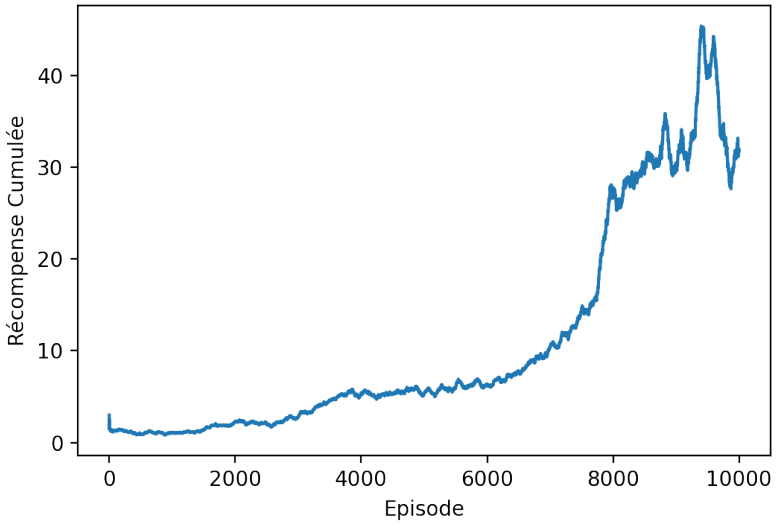


FIGURE 5.2 – Courbe des récompenses pendant l'entraînement de l'agent avec les meilleurs paramètres pour 10 000 épisodes.

TABLE 5.2 – Liste des paramètres donnant les meilleurs résultats pour action_space=3 et action_space=4

Paramètre	Valeur 3 actions	Valeur 4 actions
Batch_size	128	128
discount_factor	0.99	0.99
eps_decay	0.999999	0.999999
eps_min	0.001	0.001
update_rate	6	6
replay_mem_size	5 000 000	5 000 000
soft_up	0.001	0.001
Learning rate	0.00025	0.00025
RL_Network_shape	[128, 512, 512, 100]	[128, 512, 512, 100]
Récompense cumulée maximale	44.27	**45.36**

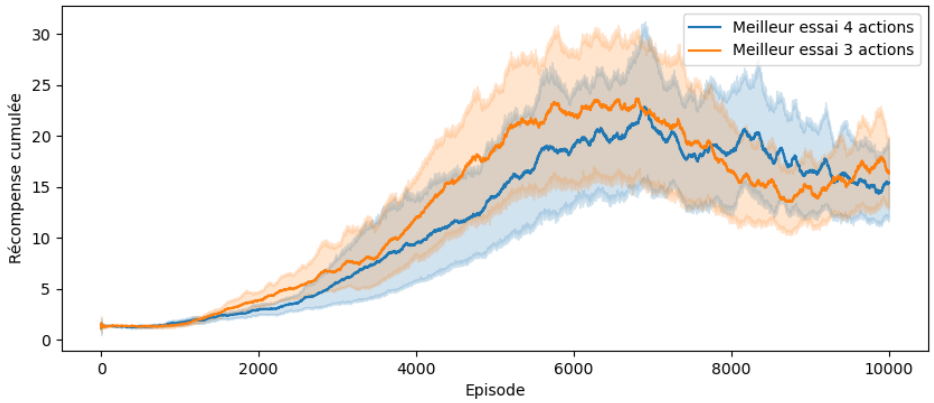


FIGURE 5.3 – Évolution du score cumulé entre les versions à 4 et à 3 actions.

5.3 Apprentissage de la représentation des états

Nous avons vu que les algorithmes d'apprentissage par renforcement sont basés sur des paires états/actions. Les états sont des représentations de l'environnement sensés permettre à un agent d'optimiser ses actions afin de maximiser la somme des récompenses obtenues. Lorsque le problème peut être formalisé sous la forme d'un processus de décision Markovien, ses états reflètent l'entièreté de l'information concernant l'environnement comme par exemple les états représentants un plateau d'échec contiendront l'entièreté des informations de la partie. Dans les études présentées jusqu'ici ainsi que dans de nombreux travaux d'application du RL aux enchères en temps réel, le problème est amalgamé à un MDP et la formulation des états est choisie de manière à donner à l'agent la meilleure représentation de la campagne en cours et ainsi lui permettre d'optimiser ses actions toujours dans le but de la maximisation des récompenses. Cependant, trouver une bonne formulation des états est une tâche complexe et requiert beaucoup de travail d'ingénierie et d'essais. Böhmer et al. [2015] donnent une définition de ce que devrait être une bonne formulation d'état pour l'apprentissage par renforcement :

— Les états doivent être Markoviens, c'est-à-dire qu'ils résument toutes les informations nécessaires pour pouvoir choisir une action selon la politique, en ne regardant que l'état actuel (et donc ne doivent pas être partiellement observables).

— La représentation des états doit être assez proche de la vraie valeur de l'état actuel de l'environnement pour permettre d'améliorer la politique.

— La représentation doit permettre la généralisation de la fonction de valeur apprise à des états inconnus ayant un avenir similaire.

— La représentation ne doit pas avoir une trop grande dimension afin de permettre l'apprentissage.

Afin d'éviter le travail manuel d'élaboration de la représentation, plusieurs approches regroupées sous le terme d'apprentissage de la représentation des états (State Representation Learning, SRL) ont été développées pour apprendre une formulation automatiquement. Appliqué à l'apprentissage par renforcement, cela consiste à optimiser la formulation conjointement au processus d'apprentissage de l'agent. Lesort et al. [2018] présentent une vue d'ensemble des approches permettant l'apprentissage de la représentation des états pour l'apprentissage par renforcement. Nous mettons ici en application l'une de ces approches : celle par autoencoders. La reconstruction des observations basées sur les autoencoders permet d'apprendre une représentation plus compacte des états avec un minimum de perte d'information. L'état s_t est donné en entrée à un autoencoder entraîné qui produit la représentation compacte s'_t dans sa couche cachée, qui est utilisée afin de reconstruire s_t en couche de sortie. La représentation s'_t permettant la reconstruction de s_t, contient donc l'information de l'état de l'environnement et permet à un agent d'apprentissage par renforcement de converger vers une bonne politique. Cette approche par autoencoders peut être complétée par des contraintes sur la dynamique des transitions d'un état s_t au suivant s_{t+1}, par exemple Goroshin et al. [2015] utilisent des autoencoders siamois permettant de contraindre les transitions entre les représentations compactes s'_t et s'_{t+1} à être linéaire. Ce type de contraintes rejoint l'idée des approches consistant à affirmer qu'une bonne représentation compacte s'_t doit permettre la prédiction de l'état suivant s_{t+1}.

Nous proposons d'étudier le comportement de l'algorithme du Double DQN couplé à un autoencoder appliqué au jeu *Atari Breakout-ram*.

Un autoencoder est une structure de réseau neuronal non supervisée qui tente de reconstruire la donnée d'entrée en sortie. Les couches cachées opèrent une extraction des caractéristiques de la donnée d'entrée en essayant soit d'augmenter la dimensionnalité si la couche cachée comporte plus de nœuds que l'entrée et la sortie, soit de la réduire s'il y a moins de nœuds. La figure 5.4 présente un autoencoder simple avec une seule couche cachée. La première étape de propagation vers l'avant est appelée la phase d'"encodage". Pendant cette phase, la dimensionnalité des données est réduite en caractéristiques qui résument les variables de la donnée d'entrée. La phase de décodage prend ces caractéristiques et les reconstruit dans la couche de sortie. La fonction d'erreur utilise le vecteur d'entrée comme cible, et

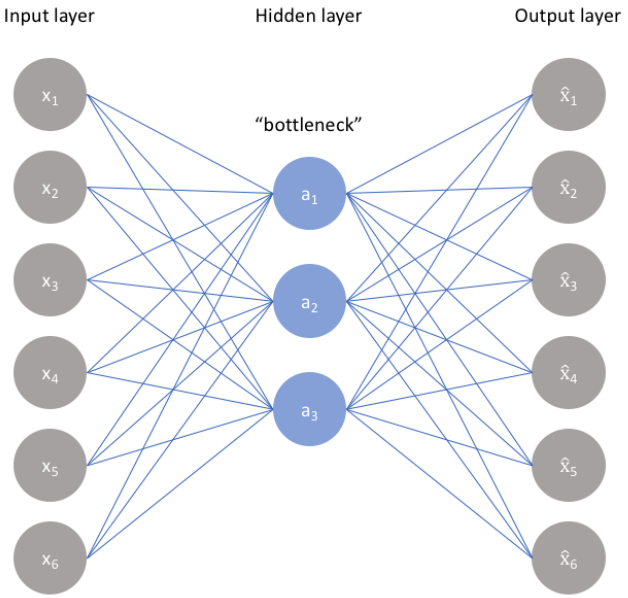


FIGURE 5.4 – Vanilla Autoencoder

rétropropage l'erreur de manière classique.

L'approche de réduction de dimension par autoencoder est voisine de l'analyse en composante principale (ACP). On peut même montrer qu'un autoencoder à une seule couche cachée (comme celui en figure 5.4) avec une fonction d'activation linéaire est quasi-équivalent à la projection en ACP à la différence près que les composants de l'ACP sont orthogonaux et donc non-corrélés alors que les éléments de la représentation apprise par l'autoencoder peuvent l'être [Plaut, 2018].

Ainsi on définit W le vecteur de poids de la couche d'encodage et V celui de la couche de décodage. Ainsi $z = f(Wx)$ est la représentation compacte de x et $\hat{x} = g(Vz)$ est la reconstruction de x. Le but du processus d'entraînement de l'autoencoder est de minimiser l'erreur de reconstruction définie par :

$$L_{reco} = \frac{1}{2N} \sum_{i=1}^{N} ||x_i - \hat{x_i}||^2$$

La principale différence entre l'ACP et les autoencoders intervient pour les autoencoders profonds et ceux ayant des fonctions d'activation non-linéaires. Dans ce cas, l'autoencoder devient capable d'opérer des transformations non-linéaires.

L'autoencoder est entraîné à apprendre une représentation compacte de l'état de la RAM. Cette représentation est fournie en entrée de l'algorithme d'apprentissage par renforcement présenté dans la

TABLE 5.3 – Meilleures valeurs pour l'apprentissage des états par autoencoders avec et sans contrainte sur la proximité des états successifs et pour différentes tailles d'encodage.

Paramètre	Valeurs optimales avec contrainte		Valeurs optimales sans contrainte	
Bottleneck_dim	128	50	128	50
Discount_factor	0.99	0.99	0.99	0.99
epsilon_decay	0.999999	0.999999	0.999999	0.999999
esilon_min	0.005	0.001	0.005	0.001
learning rate	0.0001	0.001	0.0001	0.0001
soft_update factor	0.0001	0.0001	0.0001	0.0001
batch_size	32	32	128	32
lambda	1	1	non-applicable	non-applicable
replay_mem	500 000	5 000 000	500 000	5 000 000
update_rate	1	5	1	5
Network_shape	[128, 50, 20]	[50, 256, 256, 50]	[128, 50, 20]	[50, 256, 256, 50]
cumulative_reward	**23.62**	9.8	17.57	10.63

section 5.1. Au début de l'entraînement, la représentation cachée donnée par l'autoencoder initialisé aléatoirement sera loin de pouvoir représenter l'état réel donné par l'environnement. Cette représentation sera améliorée au fur et à mesure de l'entraînement jusqu'à ce qu'elle synthétise assez bien l'information de l'état pour qu'il soit correctement reconstruit en sortie. Nous avons expérimenté deux approches d'apprentissage de la représentation des états par autoencoder. La première est une approche naïve consistant simplement à l'encodage des états réels par l'autoencoder sans aucune contrainte sur la représentation apprise. La seconde pénalise la distance entre deux états successifs en suivant l'idée que la représentation de deux états successifs devrait être proche. Ainsi La fonction d'erreur quadratique moyenne (MSE) utilisée dans la première approche devient :

$$E(s, s', s_{t+1}, s'_{t+1}) = MSE(s, s') + \lambda MSE(s', s'_{t+1})$$

Où s et s'_{t+1} sont respectivement les encodages de s et s_{t+1}. Le paramètre λ contrôle l'importance donnée à la contrainte et est intégré à la recherche en grille.

Dans le tableau 5.3, nous présentons les résultats obtenus pour différentes tailles d'encodage. Nous incluons également un encodage de la même taille que l'état réel de l'environnement à savoir 128 à des fins de comparaison.

Ce tableau nous permet de constater une dégradation des performances de l'agent par rapport à l'approche sans apprentissage de la représentation même lorsque la représentation apprise est de

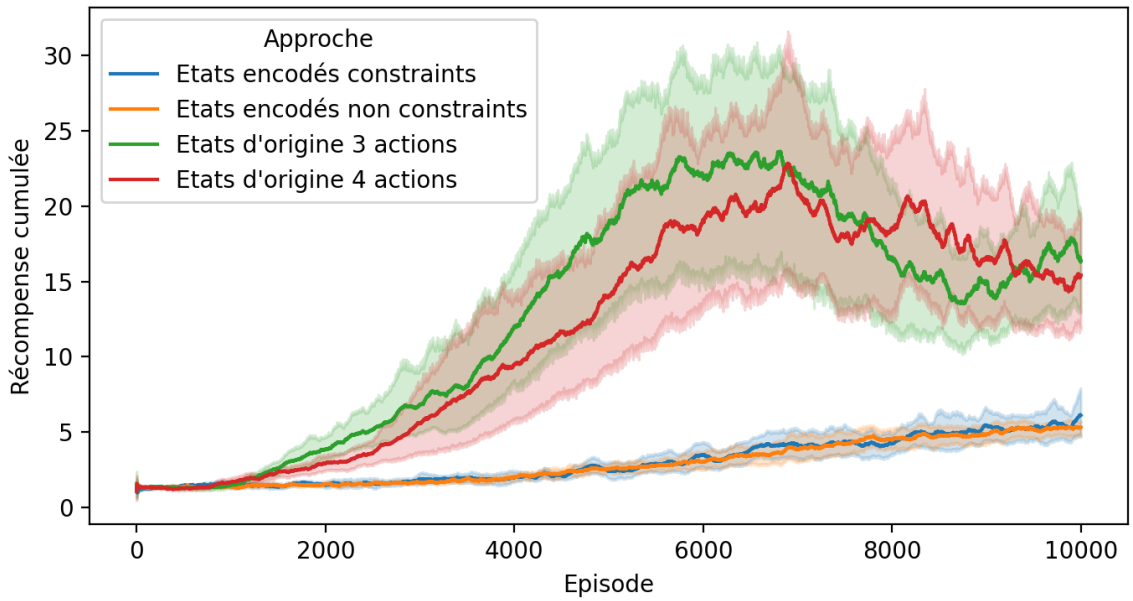


FIGURE 5.5 – Moyennes des récompenses cumulées sur les 100 derniers essais pour le meilleur épisode de chaque approches.

même taille que les états originaux. Lorsque la taille de la représentation est réduite, les performances se dégradent encore traduisant la difficulté de l'agent à identifier les bonnes actions à partir de cette représentation. On observe cependant que l'approche avec contrainte obtient de meilleurs résultats que l'approche non contrainte sur la représentation de même taille que les états originaux et des résultats très proches sur la représentation réduite.

La figure 5.5 nous permet de comparer les approches classiques avec les approches d'apprentissage de la représentation et de voir que ces dernières sont effectivement peu efficaces et peinent à converger. Ainsi là où pour les approches classiques on trouve un point de bascule où l'agent progresse très rapidement améliorant rapidement son score, les approches par autoencoders n'augmentent que linéairement et lentement. De plus, la version avec contrainte sur l'autoencoder est beaucoup plus instable que les autres bien qu'elle atteigne un meilleur score que la version non contrainte comme le montre la figure 5.6.

Cette figure montre (fig. 5.6) l'évolution des récompenses cumulées mais également, l'évolution des distances entre états successifs s_t et s_{t+1}. Elle permet de constater une corrélation très nette entre les performances du modèle et les distances. Lorsque les performances chutent, les distances

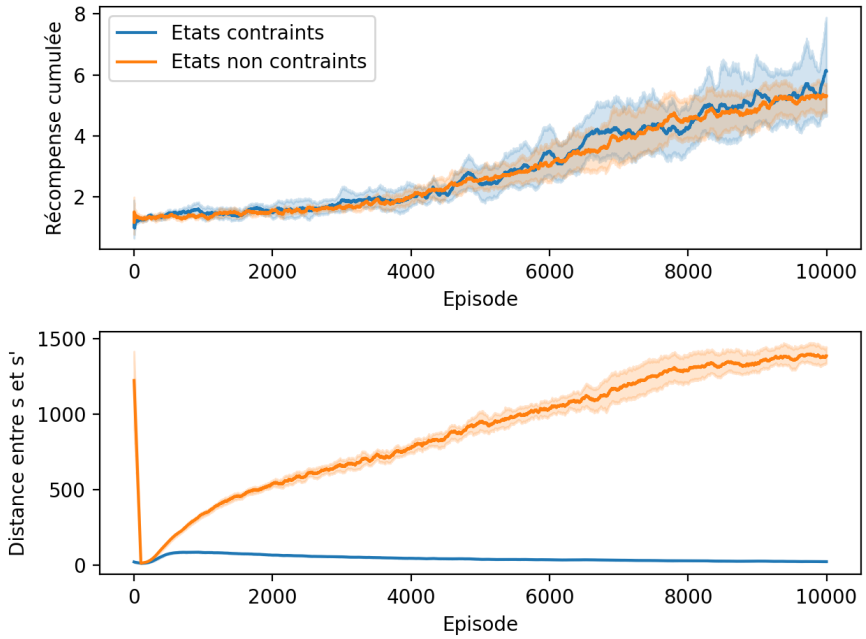
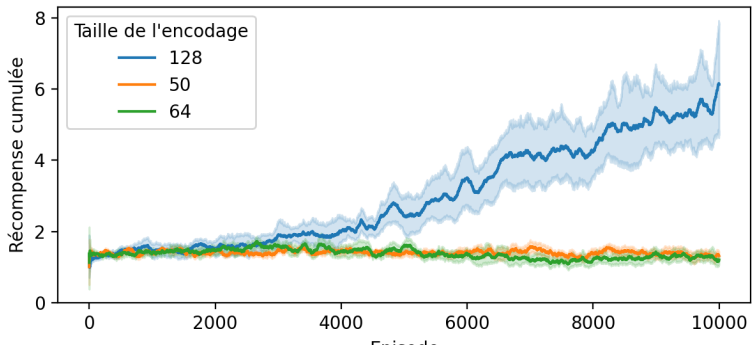


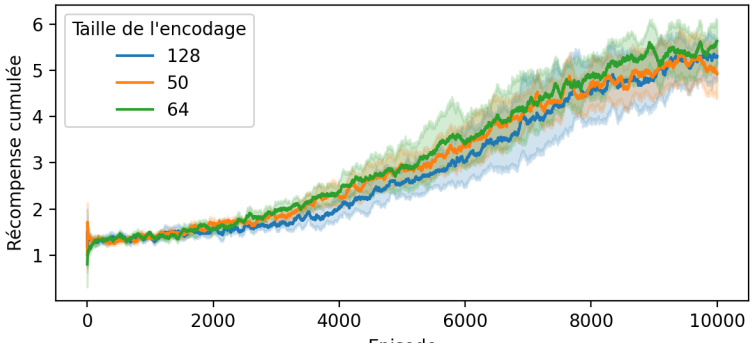
FIGURE 5.6 – Graphique de la courbe des récompenses cumulées (moyenne glissante sur les 100 derniers essais) en haut et les distances entre encodage d'états successifs $(s_t^{'}, s_{t+1}^{'})$

chutent aussi. Cela est plutôt contre intuitif et contredit notre hypothèse sur la mise sous contrainte des autoencoders. Il nous sera nécessaire d'analyser plus en détail cette contradiction mais également, ici encore, de consolider nos résultats avec plus d'essais avec les mêmes paramètres ainsi que de lancer les algorithmes avec plus d'épisodes.

Les résultats présentés jusqu'ici le sont pour la meilleure combinaison de paramètres trouvée par recherche en grille et donc avec une représentation latente de même dimension que les états d'origine (128). Notre but étant de trouver une approche permettant l'apprentissage d'une représentation compacte, nous présentons une comparaison des performances avec une taille de représentation réduite. Les figures 5.7a et 5.7b montrent les résultats obtenus avec une représentation de taille 50 comparée à la taille d'origine (128) pour les approches contraintes et non contraintes. On remarque que la représentation compacte est moins variante et semble apporter de la stabilité à l'agent. Au niveau des performances, cette représentation ne permet pas de surpasser la représentation plus grande mais elle semble tout de même être suffisante pour que l'agent progresse.



(a) Aucoencoder contraint : Moyennes glissantes des récompenses sur les 100 derniers essais pour les deux tailles d'encodage.



(b) Aucoencoder non contraint : Moyennes glissantes des récompenses sur les 100 derniers essais pour les deux tailles d'encodage.

5.4 Conclusion

Nous avons présenté dans ce chapitre les travaux menés sur l'étude de la convergence de l'algorithme d'apprentissage automatique DDQN sous plusieurs aspects. Pour cela, nous avons utilisé le jeu Atari-Breakout comme une abstraction du RTB afin de simplifier le problème et les calculs. Nous avons tout d'abord effectué un travail sur les valeurs des nombreux hyper-paramètres intervenant dans cet algorithme.

Nous avons ensuite effectué une comparaison des performances obtenues avec différentes tailles de l'espace d'action disponible pour l'agent (trois et quatre actions). Ce travail a montré que la simplification de l'espace d'action permet en effet une convergence plus rapide mais n'apporte pas de réel gain de performance en valeur.

Nous avons également présenté nos travaux sur l'apprentissage automatique de la formulation des états qui est à notre connaissance encore un sujet peu traité dans le domaine de l'apprentissage par renforcement. Nous proposons deux méthodes d'apprentissage de représentation par autoencoders, avec et sans contraintes sur la proximité spatiale de deux représentations d'états consécutifs. Les résultats obtenus sur cette partie ne sont encore qu'un premier pas dans l'apprentissage de la représentation automatique. L'approche avec contrainte n'est pas efficace et doit encore être modifiée afin que la représentation apprise permette à l'agent d'optimiser ses décisions.

Nous laissons donc comme perspective de retravailler ces approches d'apprentissage automatiques de la formulation des états. Il sera ensuite question d'appliquer ces méthodes directement sur les données de RTB afin d'en mesurer l'intérêt en application réelle. L'élaboration d'une approche efficace de définition automatique des espaces d'états pour l'apprentissage profond par renforcement est des conditions d'application de ce type d'algorithme aux enchères en temps réel dans le sens où cela permettra d'en améliorer grandement la convergence mais également de réduire le travail fastidieux de définition et de test des formulations des états construites à la main.

Chapitre 6

Conclusion et Perspectives

6.1 Conclusion

Avec les travaux présentés dans ce manuscrit, nous apportons une contribution à la recherche sur l'optimisation des enchères pour l'affichage publicitaire en ligne. Pour cela, il est nécessaire d'estimer l'utilité d'un affichage si remporté : c'est la prédiction de la probabilité du clic. Il faut ensuite, au cours de la campagne d'affichage, apprendre à gérer le budget, à enchérir et remporter les bonnes opportunités afin de maximiser le revenu.

Cette maximisation repose en partie sur la prédiction de la probabilité qu'un visiteur clique sur la publicité si elle lui est affichée. Ces clics n'arrivent que très peu souvent par rapport au nombre de publicités affichées et cela nous a amenés à travailler sur la prédiction d'événements rares. Ce type de prédiction entraîne des biais de prédiction sur les modèles de classification binaire classique. Notamment, nous avons montré que dans le cas de la régression logistique, il est possible de corriger une partie de ce biais grâce à la pondération de la fonction d'erreur. Le déséquilibre dans les données n'influe pas seulement sur les modèles mais aussi sur la manière de les évaluer. Ainsi nous avons étudié les biais d'une des principales mesures d'évaluation de classification binaire : l'AUC_{ROC}. Nous contribuons à la correction de ces biais en proposant une mesure de performance spécifique à la prédiction de la probabilité de clic pour les enchères en temps réel. Cette mesure intègre les coûts réels de ces prédictions et pénalise les cas défavorables comme celui de prédire un clic qui n'arrive pas et d'acheter un emplacement qui n'apportera pas de revenu. La mesure proposée intègre aussi la valeur que l'annonceur associe aux clics et permet donc la projection de la rentabilité d'une campagne. Dans cette partie, nous avons aussi présenté une approche de clustering ascendant de modalités pour la réduction de dimension et montré pour des données avec beaucoup de modalités, que les regroupements successifs ne dégradent que peu les performances de prédiction de la régression logistique. Ceci dit, les regroupements observés avec l'approche proposée n'ont pas permis de mettre en évidence de structure dans les données, ce qui aurait été une propriétés souhaitable dans l'objectif de créer une approche de pré-traitement automatique et explicable de données catégorielles de haute dimensionalité.

Nous avons dans une deuxième partie, nous avons présenté nos travaux sur l'apprentissage de la stratégie d'enchères. Nous avons mis en comparaison différentes stratégies d'enchères, plus ou moins élaborées : des stratégies statiques et dynamiques, dont une reposant sur un algorithme d'apprentissage automatique. Nos résultats ont montré l'instabilité de ce type d'approches sur un problème aussi

complexe que le RTB. Nous avons mis en application ces approches sur les probabilités de clics calculées grâce aux différents algorithmes utilisés dans la partie sur la prédiction d'événements rares réunissant les deux grandes parties de nos travaux, notamment sur la fonction d'évaluation spécifique au RTB. Les résultats obtenus montrent que s'agissant d'un enchérisseur imparfait et contrairement aux résultats obtenus dans la section consacrée à la prédiction de la probabilité de clic, la régression logistique classique permet de meilleurs résultats en terme de clics mais également en terme de gains, exprimés par la fonction de coût proposée.

Nous avons enfin étudié l'influence qu'aurait un changement de système d'enchères du second au premier prix et montré que cela entraîne une baisse des performances logique. Ces travaux devront être approfondis afin de mieux cerner l'étendue des changements à opérer sur les modèles pour qu'ils s'adaptent aux enchères en premier prix.

Dans une dernière partie, nous avons étudié la convergence de l'algorithme Double Deep Q-Network (DDQN) et sa sensibilité aux différents hyperparamètres sur le jeu Atari-Breakout. Nous avons notamment pu observer les différences de performance et de convergence selon la taille de l'espace d'action disponible à l'agent. Nos résultats sur ce point ont montré que la baisse du nombre d'actions disponibles permet bien à l'agent de converger et de déduire une stratégie plus rapidement mais que cela ne permet pas de dégager une meilleure stratégie à terme.

Nous avons prolongé cette étude en appliquant une approche d'apprentissage automatique de la représentation des états par autoencoder. Nous avons ainsi proposé deux approches. Une première sans contrainte, et la seconde avec une contrainte pénalisant la distance de représentation entre deux états successifs. Ces travaux n'ont malheureusement pas conduit à des résultats satisfaisants et les recherches sur ce sujet devront être poursuivies afin de pouvoir construire une approche fonctionnelle qui serait ensuite appliquée aux données du RTB afin d'automatiser le travail de définition des états et ainsi permettre une meilleure application de l'apprentissage par renforcement au RTB et plus généralement aux problèmes analogues pour lesquels l'environnement n'est pas directement observable.

6.2 Perspectives

Nous présentons dans cette section les perspectives et pistes de recherche que nous avons identifiées mais que nous n'avons pu explorer.

À court terme, il serait intéressant d'effectuer une étude comparative sur les performances des différents type d'algorithmes d'apprentissage par renforcement afin de mieux comprendre leurs propriétés de convergence sur le problème du RTB. La prise en compte du caractère temporel des campagnes d'enchères nous parait également constituer une piste intéressante pour des travaux futurs [Kulkarni et al., 2016]. Les approches présentées dans nos travaux ne prennent en compte que l'état de la campagne à un moment donnée, même si les termes de budget pacing donnent une indication sur les enchères passées. La prise en compte du rapport temporel entre les différentes actions prises par l'agent grâce, par exemple, à l'utilisation de réseaux récurrents comme les LSTM dans le processus d'apprentissage par renforcement [Bakker, 2001] nous parait être une piste prometteuse.

Un travail est nécessaire sur la fonction définition de fonctions de récompenses dans les cas où celles ci sont rares comme c'est le cas avec les clics dans notre cas. Ce problème à déjà été étudié notamment par Wu et al. [2018] qui utilisent un réseau de neurones afin d'estimer la fonction d'erreur. D'une manière plus générale, la question de la rareté de la récompense est un sujet actif dans le domaine de l'apprentissage par renforcement appliqué à la robotique [Nair et al., 2018, Riedmiller et al., 2018a, Vecerik et al., 2017]

L'amélioration de l'approche de regroupement de modalités pour le pré-traitement automatique est à moyen terme une piste intéressante. L'approfondissement des tests de notre approche sur l'ensemble des variables ainsi que sur les différentes campagnes devra être effectué afin de consolider les résultats et de valider l'efficacité de cette approche. Également, un travail sur l'ajout d'une contrainte d'explicabilité sur les regroupements peut être une piste prometteuse dans l'optique de la création d'approche de pré-traitement automatique des données.

Du coté de l'apprentissage de la représentation des états, nous n'avons pu l'appliquer au RTB. Nous pensons que cela serait d'un grand intérêt à court/moyen terme. En effet, les données de RTB sont de haute dimensionnalité et requirent une étape de prédiction du clic. Nous pensons qu'il peut être possible de regrouper l'ensemble du processus : pré-traitement du jeu de données, prédiction du clic et optimisation des enchères grâce à l'apprentissage de la représentation. Nous laissons comme piste à explorer l'étude de l'apprentissage automatique de la représentation avec d'autres algorithmes d'apprentissage par renforcement afin de voir si certaines approches sont plus à même de se prêter à cette tâche que d'autres.

Enfin et toujours dans le but de réunifier les différentes étapes de l'apprentissage de la stratégie

d'enchères, nous proposons comme piste l'utilisation d'une fonction d'évaluation spécifique comme celle que nous avons proposée comme fonction objective à optimiser dans le processus d'apprentissage. Étant donné que cette fonction permet une meilleure représentation des performances de l'algorithme de la prédiction de la probabilité de clic, nous pensons que son utilisation comme fonction objective pourrait aider à une meilleure convergence globale.

Bibliographie

Hisham Abdallah and Gilbert Saporta. Classification d'un ensemble de variables qualitatives. *Revue de Statistique Appliquée*, 46(4) :5–26, 1998. URL `https://hal-cnam.archives-ouvertes.fr/hal-02507938`.

Rishabh Agarwal, Chen Liang, Dale Schuurmans, and Mohammad Norouzi. Learning to generalize from sparse and underspecified rewards. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, Proceedings of Machine Learning Research, pages 130–140. PMLR, 2019. URL `http://proceedings.mlr.press/v97/agarwal19e.html`.

Bram Bakker. Reinforcement learning with long short-term memory. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001. URL `https://proceedings.neurips.cc/paper/2001/file/a38b16173474ba8b1a95bcbc30d3b8a5-Paper.pdf`.

Mohamed Bekkar, Hassiba Djema, and T.A. Alitouche. Evaluation measures for models assessment over imbalanced data sets. *Journal of Information Engineering and Applications*, 3 :27–38, 01 2013.

Wendelin Böhmer, Jost Tobias Springenberg, Joschka Boedecker, Martin Riedmiller, and Klaus Obermayer. Autonomous learning of state representations for control : An emerging field aims to autonomously learn state representations for reinforcement learning agents from their real-world sensor observations. *KI-Künstliche Intelligenz*, 29(4) :353–362, 2015.

Lucian Buşoniu, Tim de Bruin, Domagoj Tolić, Jens Kober, and Ivana Palunko. Reinforcement learning for control : Performance, stability, and deep approximators. *Annual Reviews in Control*, 46 : 8–28, 2018. ISSN 1367-5788. doi:https ://doi.org/10.1016/j.arcontrol.2018.09.005. URL `https://www.sciencedirect.com/science/article/pii/S1367578818301184`.

Han Cai, Kan Ren, Weinan Zhang, Kleanthis Malialis, Jun Wang, Yong Yu, and Defeng Guo. Real-time bidding by reinforcement learning in display advertising. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, page 661–670, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450346757. doi:10.1145/3018661.3018702. URL https://doi.org/10.1145/3018661.3018702.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017. doi:10.1109/SP.2017.49.

Nicolas Carrara, Edouard Leurent, Romain Laroche, Tanguy Urvoy, Odalric-Ambrym Maillard, and Olivier Pietquin. Budgeted reinforcement learning in continuous state space. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32 : Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 9295–9305, 2019. URL http://papers.nips.cc/paper/9128-budgeted-reinforcement-learning-in-continuous-state-space.

Olivier Chapelle, Eren Manavoglu, and Romer Rosales. Simple and scalable response prediction for display advertising. *ACM Trans. Intell. Syst. Technol.*, 5(4), dec 2015. ISSN 2157-6904. doi:10.1145/2532128. URL https://doi.org/10.1145/2532128.

Marie Chavent, Christiane Guinot, Yves Lechevallier, and Michel Tenenhaus. Méthodes divisives de classification et segmentation non supervisée : recherche d'une typologie de la peau humaine saine. *Revue de statistique appliquée*, 47(4) :87–99, 1999.

Julien Chiquet, Pierre Gutierrez, and Guillem Rigaill. Fast tree inference with weighted fusion penalties. *Journal of Computational and Graphical Statistics*, 26(1) :205–216, 2017. doi:10.1080/10618600.2015.1096789. URL https://doi.org/10.1080/10618600.2015.1096789.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 233–240, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. doi:10.1145/1143844.1143874. URL https://doi.org/10.1145/1143844.1143874.

Stylianos Despotakis, Ramamoorthi Ravi, and Amin Sayedi. First-price auctions in online display advertising. *SSRN Electronic Journal*, 01 2019. doi:10.2139/ssrn.3485410.

Kenji Doya. Reinforcement learning in continuous time and space. *Neural Computation*, 12(1) :219–245, 2000. doi:10.1162/089976600300015961. URL `https://doi.org/10.1162/089976600300015961`.

Chris Drummond and Robert C Holte. What roc curves can't do (and cost curves can). In *ROCAI*, pages 19–26. Citeseer, 2004.

Vincent François-Lavet, Raphaël Fonteneau, and Damien Ernst. How to discount deep reinforcement learning : Towards new dynamic strategies. *ArXiv*, abs/1512.02011, 2015.

Langche Zeng Gary King. Logistic regression in rare events data. *Political Analysis*, 9 :137–163, Spring 2001.

Ross Goroshin, Michael F Mathieu, and Yann LeCun. Learning to linearize under uncertainty. *Advances in neural information processing systems*, 28, 2015.

Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm : A factorization-machine based neural network for CTR prediction. *CoRR*, abs/1703.04247, 2017. URL `http://arxiv.org/abs/1703.04247`.

Michael Herrmann and Ralf Der. Efficient state-space representation by neural maps for reinforcement learning. In Wolfgang Gaul and Hermann Locarek-Junge, editors, *Classification in the Information Age*, pages 302–309, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5) :359–366, 1989.

Mohammad Hossin and Md Nasir Sulaiman. A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, 5(2) :1, 2015.

Guojing Huang, Qingliang Chen, and Congjian Deng. A new click-through rates prediction model based on deep&amp ;cross network. *Algorithms*, 13(12), 2020. ISSN 1999-4893. doi:10.3390/a13120342. URL `https://www.mdpi.com/1999-4893/13/12/342`.

Interactive Advertising Bureau (IAB). Programmatic in-housing : Benefits, challenges and key steps to building internal capabilities. https://www.iab.com/wp-content/uploads/2018/05/IAB_Programmatic-In-Housing-Whitepaper_v5.pdf, 2018.

Matthew Jackson. Matching, auctions, and market design. *SSRN Electronic Journal*, 04 2013. doi:10.2139/ssrn.2263502.

Arthur Jacot, Clément Hongler, and Franck Gabriel. Neural tangent kernel : Convergence and generalization in neural networks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31 : Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 8580–8589, 2018. URL http://papers.nips.cc/paper/8076-neural-tangent-kernel-convergence-and-generalization-in-neural-networks.

Junqi Jin, Chengru Song, Han Li, Kun Gai, Jun Wang, and Weinan Zhang. Real-time bidding with multi-agent reinforcement learning in display advertising. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 2193–2201, New York, NY, USA, 2018a. Association for Computing Machinery. ISBN 9781450360142. doi:10.1145/3269206.3272021. URL https://doi.org/10.1145/3269206.3272021.

Junqi Jin, Chengru Song, Han Li, Kun Gai, Jun Wang, and Weinan Zhang. Real-time bidding with multi-agent reinforcement learning in display advertising. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 2193–2201, 2018b.

Gary King and Langche Zeng. Explaining rare events in international relations. *International Organization*, 55(3) :693–715, 2001. doi:10.1162/00208180152507597.

Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical deep reinforcement learning : Integrating temporal abstraction and intrinsic motivation. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/f442d33fa06832082290ad8544a8da27-Paper.pdf.

Kuang-Chih Lee, Ali Jalali, and Ali Dasdan. Real time bid optimization with smooth budget delivery in online advertising. ADKDD '13, New York, NY, USA, 2013a. Association for Computing Machinery.

ISBN 9781450323239. doi:10.1145/2501040.2501979. URL https://doi.org/10.1145/2501040.2501979.

Kuang-Chih Lee, Ali Jalali, and Ali Dasdan. Real time bid optimization with smooth budget delivery in online advertising. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*, ADKDD '13, New York, NY, USA, 2013b. Association for Computing Machinery. ISBN 9781450323239. doi:10.1145/2501040.2501979. URL https://doi.org/10.1145/2501040.2501979.

Timothée Lesort, Natalia Díaz-Rodríguez, Jean-Franois Goudou, and David Filliat. State representation learning for control : An overview. *Neural Networks*, 108 :379–392, 2018.

Hairen Liao, Lingxiao Peng, Zhenchuan Liu, and Xuehua Shen. ipinyou global rtb bidding algorithm competition dataset. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, pages 1–6, 2014.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1509.02971.

Chi-Chun Lin, Kun-Ta Chuang, Wush Chi-Hsuan Wu, and Ming-Syan Chen. Budget-constrained real-time bidding optimization : Multiple predictors make it better. *ACM Trans. Knowl. Discov. Data*, 14 (2), February 2020. ISSN 1556-4681. doi:10.1145/3375393. URL https://doi-org.proxybib-pp.cnam.fr/10.1145/3375393.

M. Liu, J. Li, W. Yue, L. Qiu, J. Liu, and Z. Qin. An intelligent bidding strategy based on model-free reinforcement learning for real-time bidding in display advertising. In *2019 Seventh International Conference on Advanced Cloud and Big Data (CBD)*, pages 240–245, 2019.

J. Lobo, A. Jiménez-Valverde, and R. Real. Auc : a misleading measure of the performance of predictive distribution models. *Global Ecology and Biogeography*, 17 :145–151, 2008.

Maher Maalouf and Mohammad Siddiqi. Weighted logistic regression for large-scale imbalanced and rare events data. *Knowledge-Based Systems*, 59 :142–148, 2014.

Maher Maalouf and Theodore B Trafalis. Robust weighted kernel logistic regression in imbalanced and rare events data. *Computational Statistics & Data Analysis*, 55(1) :168–183, 2011.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013a. URL http://arxiv.org/abs/1312.5602.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *ArXiv*, abs/1312.5602, 2013b.

Andrew W. Moore. The parti-game algorithm for variable resolution reinforcement learning in multidimensional state-spaces. pages 711–718, 1993. URL http://papers.nips.cc/paper/742-the-parti-game-algorithm-for-variable-resolution-reinforcement-learning-in-multidimensi

Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6292–6299, 2018. doi:10.1109/ICRA.2018.8463162.

Claudia Perlich, Brian Dalessandro, Rod Hook, Ori Stitelman, Troy Raeder, and Foster Provost. Bid optimizing and inventory scoring in targeted online advertising. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, page 804–812, New York, NY, USA, 2012a. Association for Computing Machinery. ISBN 9781450314626. doi:10.1145/2339530.2339655. URL https://doi.org/10.1145/2339530.2339655.

Claudia Perlich, Brian Dalessandro, Rod Hook, Ori Stitelman, Troy Raeder, and Foster J. Provost. Bid optimizing and inventory scoring in targeted online advertising. In Qiang Yang, Deepak Agarwal, and Jian Pei, editors, *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, pages 804–812. ACM, 2012b. doi:10.1145/2339530.2339655. URL https://doi.org/10.1145/2339530.2339655.

Elad Plaut. From principal subspaces to principal components with linear autoencoders, 2018. URL https://arxiv.org/abs/1804.10253.

Martin L Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2 :331–434, 1990.

Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. Product-based neural networks for user response prediction. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 1149–1154, 2016. doi:10.1109/ICDM.2016.0151.

Yanru Qu, Bohui Fang, Weinan Zhang, Ruiming Tang, Minzhe Niu, Huifeng Guo, Yong Yu, and Xiuqiang He. Product-based neural networks for user response prediction over multi-field categorical data. *ACM Transactions on Information Systems (TOIS)*, 37(1) :1–35, 2018.

Chitta Ranjan. *Understanding Deep Learning : Application in Rare Event Prediction*. Connaissance Publishing, Dec 2020. doi:10.13140/RG.2.2.34297.49765.

Kan Ren, Weinan Zhang, Ke Chang, Yifei Rong, Yong Yu, and Jun Wang. Bidding machine : Learning to bid for directly optimizing profits in display advertising. *IEEE Transactions on Knowledge and Data Engineering*, 30(4) :645–659, 2018. doi:10.1109/TKDE.2017.2775228.

Steffen Rendle. Factorization machines. In Geoffrey I. Webb, Bing Liu, Chengqi Zhang, Dimitrios Gunopulos, and Xindong Wu, editors, *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010*, pages 995–1000. IEEE Computer Society, 2010. doi:10.1109/ICDM.2010.127. URL `https://doi.org/10.1109/ICDM.2010.127`.

Martin Riedmiller, Roland Hafner, Thomas Lampe, Michael Neunert, Jonas Degrave, Tom van de Wiele, Vlad Mnih, Nicolas Heess, and Jost Tobias Springenberg. Learning by playing solving sparse reward tasks from scratch. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4344–4353. PMLR, 10–15 Jul 2018a. URL `https://proceedings.mlr.press/v80/riedmiller18a.html`.

Martin Riedmiller, Roland Hafner, Thomas Lampe, Michael Neunert, Jonas Degrave, Tom van de Wiele, Vlad Mnih, Nicolas Heess, and Jost Tobias Springenberg. Learning by playing solving sparse reward tasks from scratch. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4344–4353, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018b. PMLR. URL `http://proceedings.mlr.press/v80/riedmiller18a.html`.

Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3) :1–21, 03 2015. doi:10.1371/journal.pone.0118432. URL https://doi.org/10.1371/journal.pone.0118432.

Helen Sofaer, Jennifer Hoeting, and Catherine Jarnevich. The area under the precision-recall curve as a performance metric for rare binary events. *Methods in Ecology and Evolution*, 10, 12 2018. doi:10.1111/2041-210X.13140.

Helen R Sofaer, Jennifer A Hoeting, and Catherine S Jarnevich. The area under the precision-recall curve as a performance metric for rare binary events. *Methods in Ecology and Evolution*, 10(4) : 565–577, 2019.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout : A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56) :1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press, 1998. ISBN 978-0-262-19398-6. URL http://www.worldcat.org/oclc/37293240.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1) :267–288, 1996. ISSN 00359246. URL http://www.jstor.org/stable/2346178.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society Series B*, 67 :91–108, 02 2005. doi:10.1111/j.1467-9868.2005.00490.x.

Michael Tomz, Gary King, and Langche Zeng. Relogit : Rare events logistic regression. *Journal of statistical software*, 8(1) :1–27, 2003.

Gerhard Tutz and Jan Gertheiss. Regularized regression for categorical data. *Statistical Modelling*, 16(3) :161–200, 2016. doi:10.1177/1471082X16642560. URL https://doi.org/10.1177/1471082X16642560.

M. Van Den Eeckhaut, T. Vanwalleghem, J. Poesen, G. Govers, G. Verstraeten, and L. Vandekerckhove. Prediction of landslide susceptibility using rare events logistic regression : A case-study in the flemish ardennes (belgium). *Geomorphology*, 76(3) :392–410, 2006. ISSN 0169-555X. doi:https ://doi.org/10.1016/j.geomorph.2005.12.003. URL `https://www.sciencedirect.com/science/article/pii/S0169555X05003788`.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Mel Vecerik, Todd Hester, Jonathan Scholz, Fumin Wang, Olivier Pietquin, Bilal Piot, Nicolas Heess, Thomas Rothörl, Thomas Lampe, and Martin Riedmiller. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv :1707.08817*, 2017.

William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1) :8–37, 1961.

Haiying Wang. Logistic regression for massive data with rare events. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9829–9836. PMLR, 13–18 Jul 2020. URL `http://proceedings.mlr.press/v119/wang20a.html`.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3) :279–292, 1992.

Gary M Weiss and Haym Hirsh. Learning to predict rare events in event sequences. In *KDD*, volume 98, pages 359–363, 1998.

Di Wu, Xiujun Chen, Xun Yang, Hao Wang, Qing Tan, Xiaoxun Zhang, Jian Xu, and Kun Gai. Budget constrained bidding by model-free reinforcement learning in display advertising. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 1443–1451, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450360142. doi:10.1145/3269206.3271748. URL `https://doi.org/10.1145/3269206.3271748`.

Easton Li Xu, Xiaoning Qian, Tie Liu, and Shuguang Cui. Pairwise interaction analysis of logistic regression models. In *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 187–191, 2016. doi:10.1109/GlobalSIP.2016.7905829.

Weinan Zhang, Shuai Yuan, and Jun Wang. Real-time bidding benchmarking with ipinyou dataset. *CoRR*, abs/1407.7073, 2014a. URL http://arxiv.org/abs/1407.7073.

Weinan Zhang, Shuai Yuan, and Jun Wang. Optimal real-time bidding for display advertising. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1077–1086, 2014b.

Weinan Zhang, Tianming Du, and Jun Wang. Deep learning over multi-field categorical data - - A case study on user response prediction. In Nicola Ferro, Fabio Crestani, Marie-Francine Moens, Josiane Mothe, Fabrizio Silvestri, Giorgio Maria Di Nunzio, Claudia Hauff, and Gianmaria Silvello, editors, *Advances in Information Retrieval - 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings*, volume 9626 of *Lecture Notes in Computer Science*, pages 45–57. Springer, 2016. doi:10.1007/978-3-319-30671-1_4. URL https://doi.org/10.1007/978-3-319-30671-1_4.

Jun Zhao, Guang Qiu, Ziyu Guan, Wei Zhao, and Xiaofei He. Deep reinforcement learning for sponsored search real-time bidding. In Yike Guo and Faisal Farooq, editors, *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 1021–1030. ACM, 2018. doi:10.1145/3219819.3219918. URL https://doi.org/10.1145/3219819.3219918.

Annexe A

Dataset Ipinyou

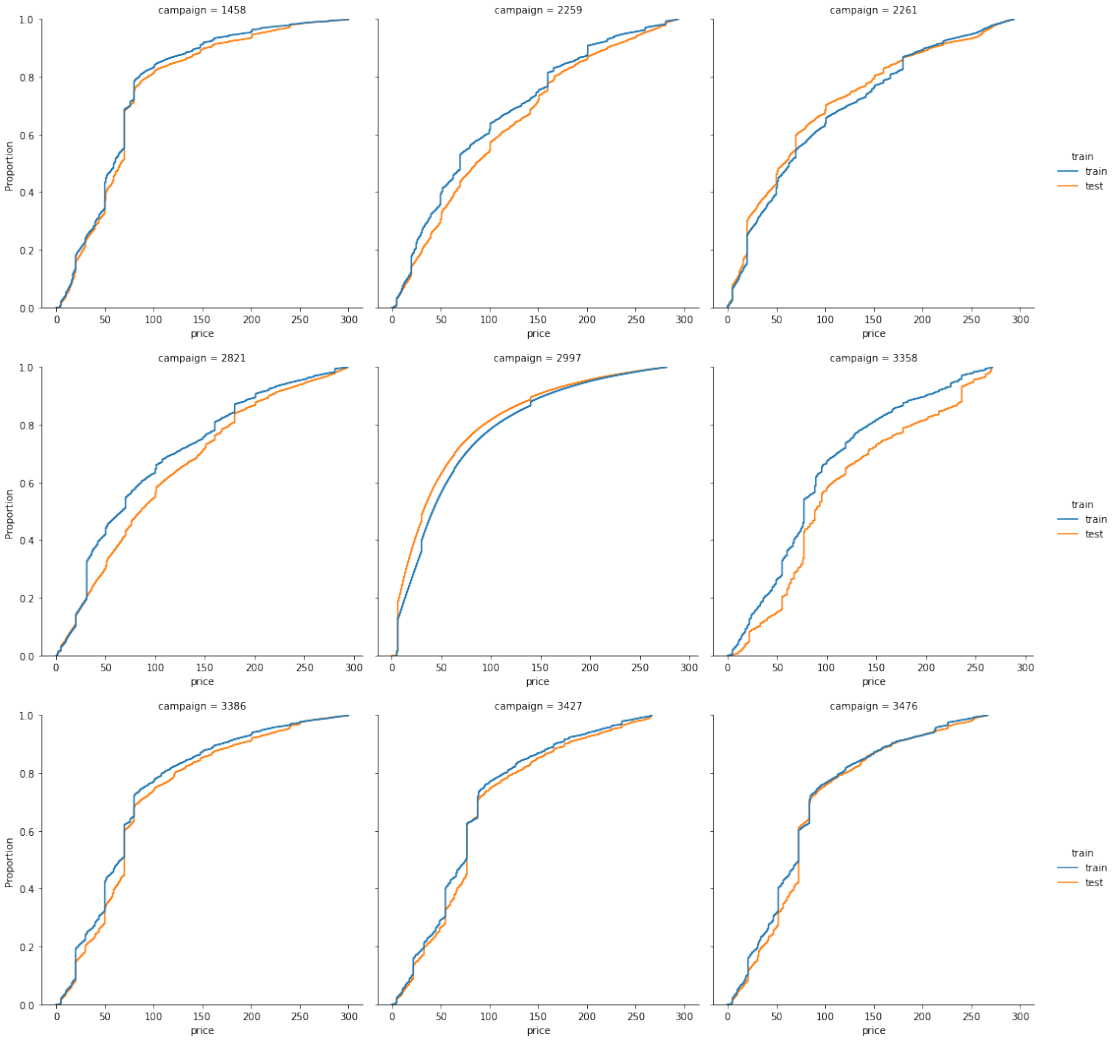


FIGURE A.1 – Distribution cumulative des prix gagnants dans les ensembles d'entraînement et de test

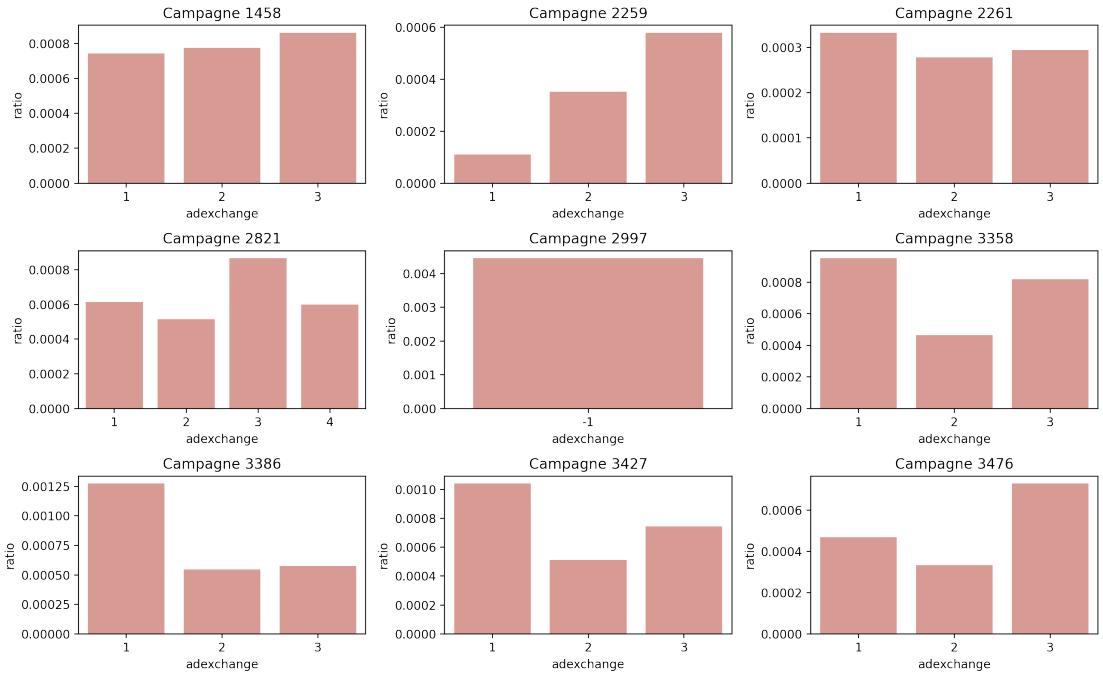


FIGURE A.2 – Graphique des ratio entre clic et non clics du nombre d'occurrences des modalités de la variable adexchange

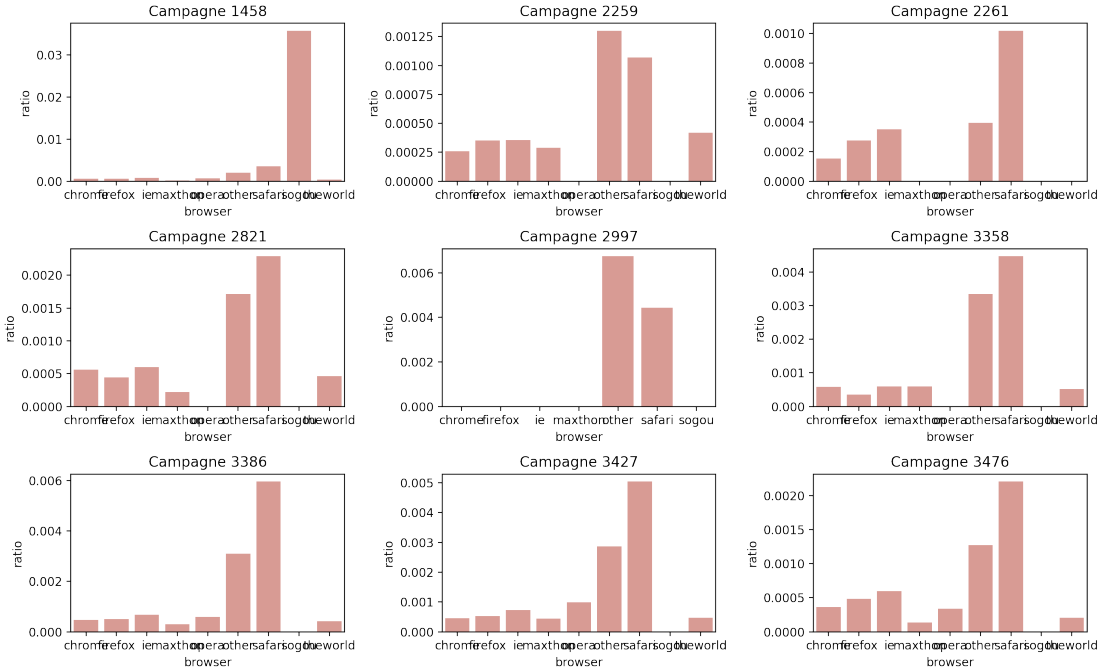


FIGURE A.3 – Graphique des ratio entre clic et non clics du nombre d'occurrences des modalités de la variable browser

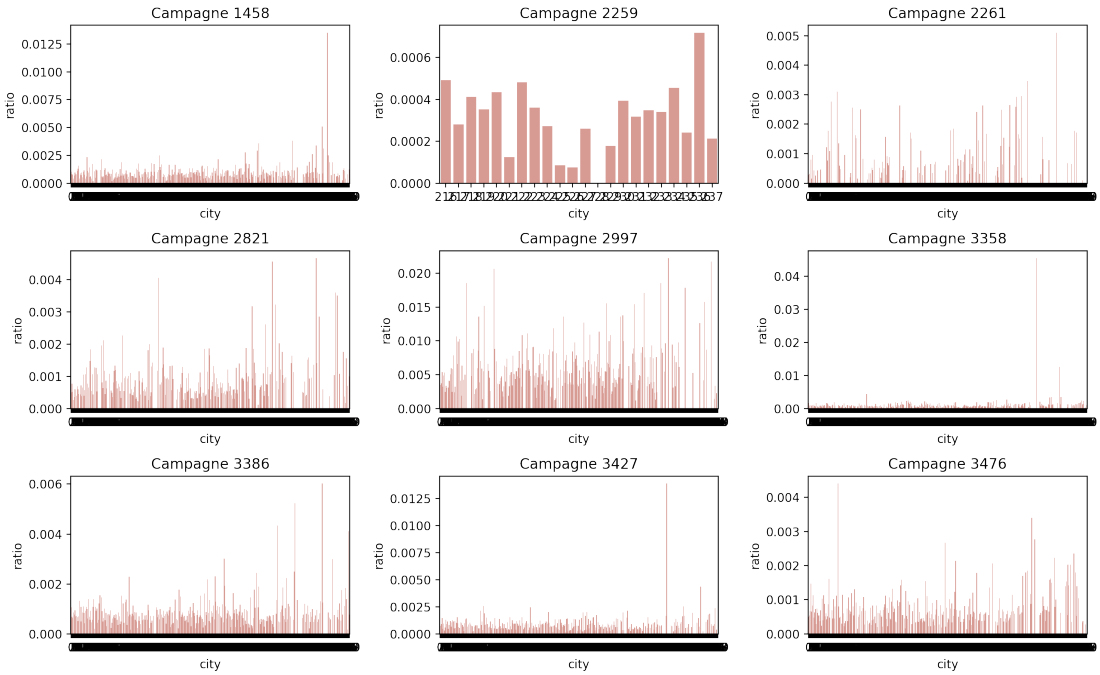


FIGURE A.4 – Graphique des ratio entre clic et non clics du nombre d'occurrences des modalités de la variable city

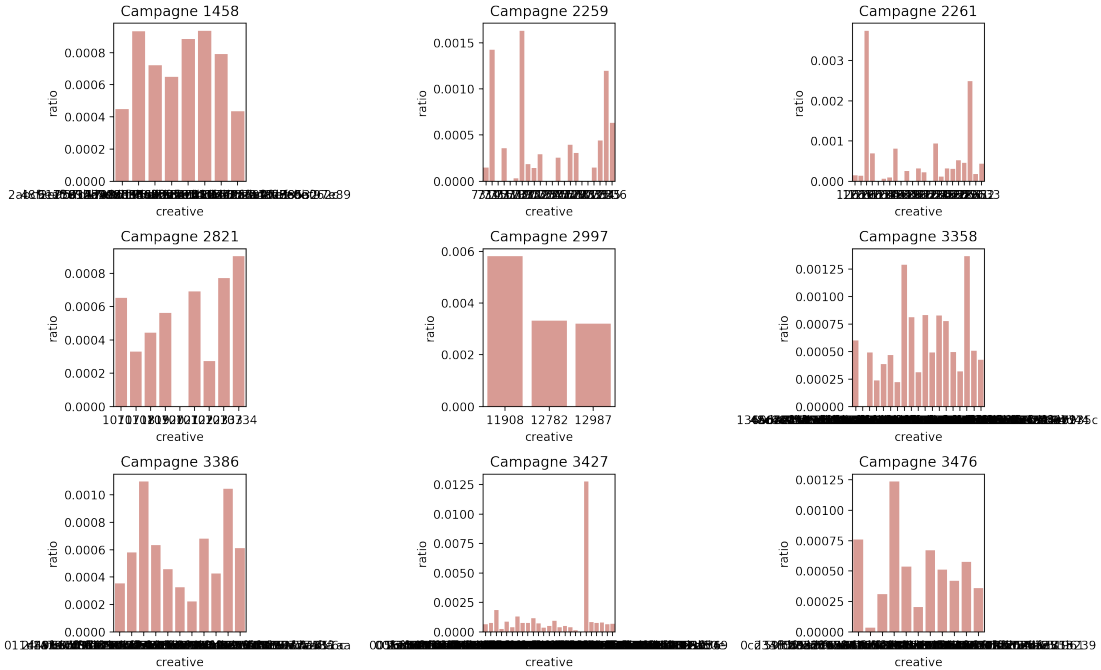


FIGURE A.5 – Graphique des ratio entre clic et non clics du nombre d'occurrences des modalités de la variable creative

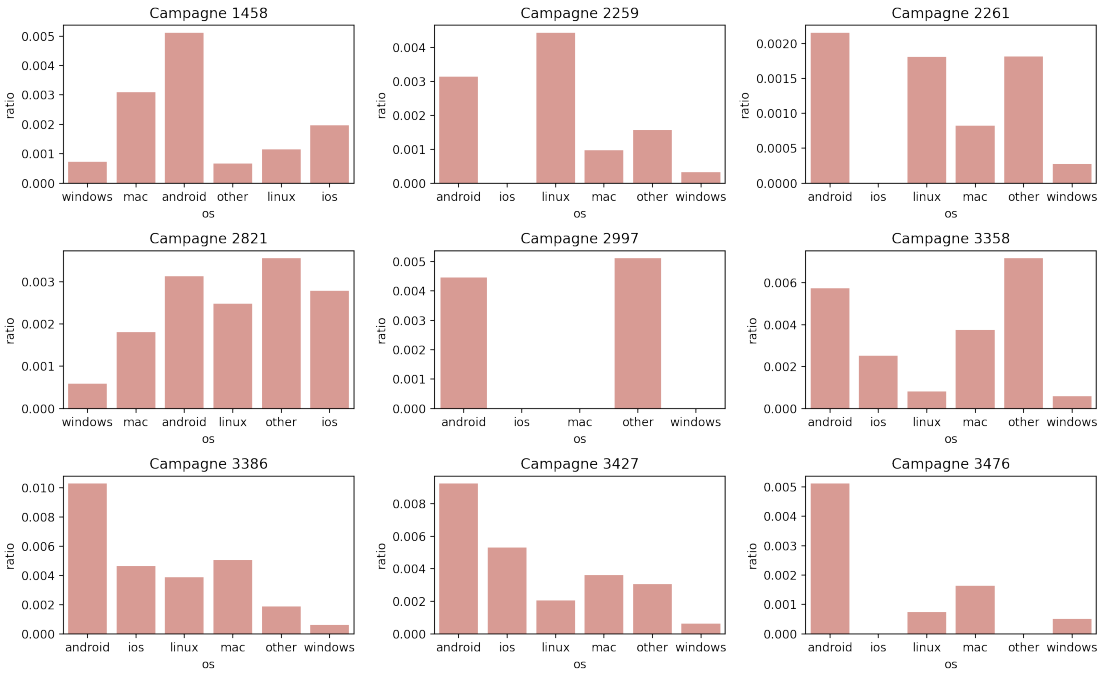


FIGURE A.6 – Graphique des ratio entre clic et non clics du nombre d'occurrences des modalités de la variable os

FIGURE A.7 – Graphique des ratio entre clic et non clics du nombre d'occurrences des modalités de la variable region

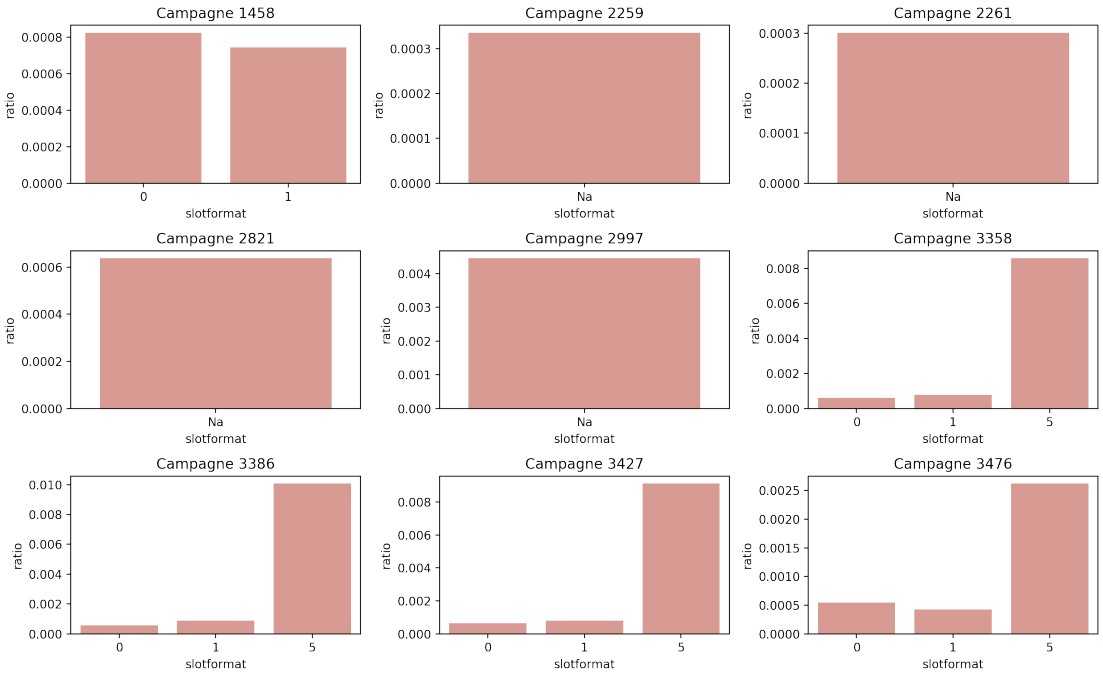


FIGURE A.8 – Graphique des ratio entre clic et non clics du nombre d'occurrences des modalités de la variable slotformat

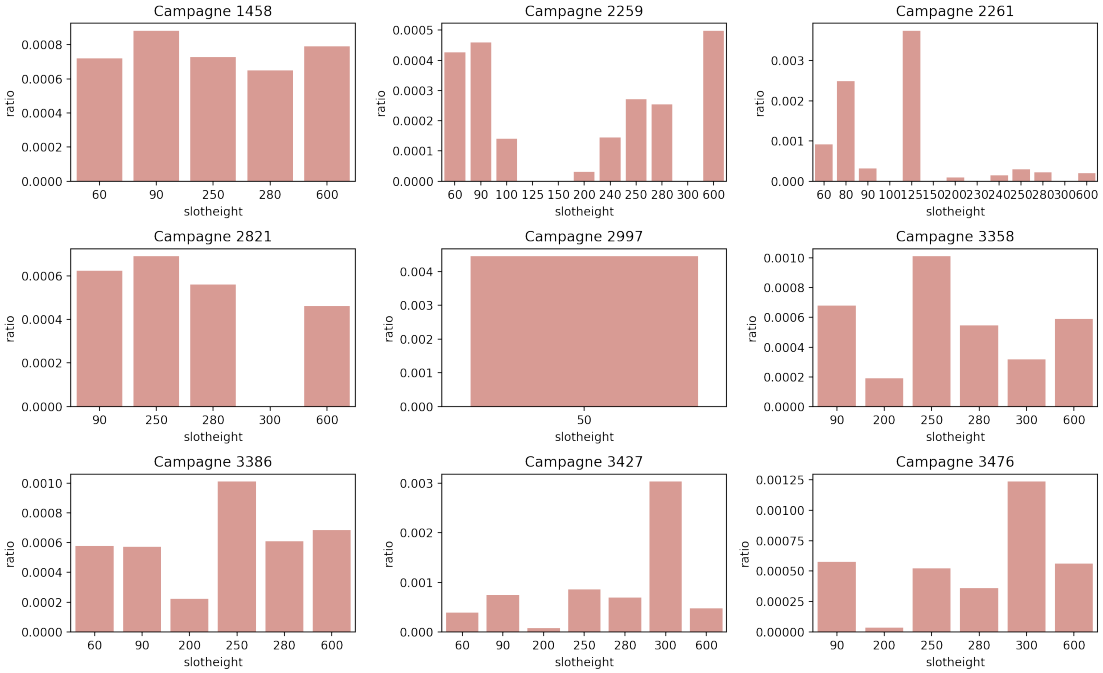


FIGURE A.9 – Graphique des ratio entre clic et non clics du nombre d'occurrences des modalités de la variable slotheight

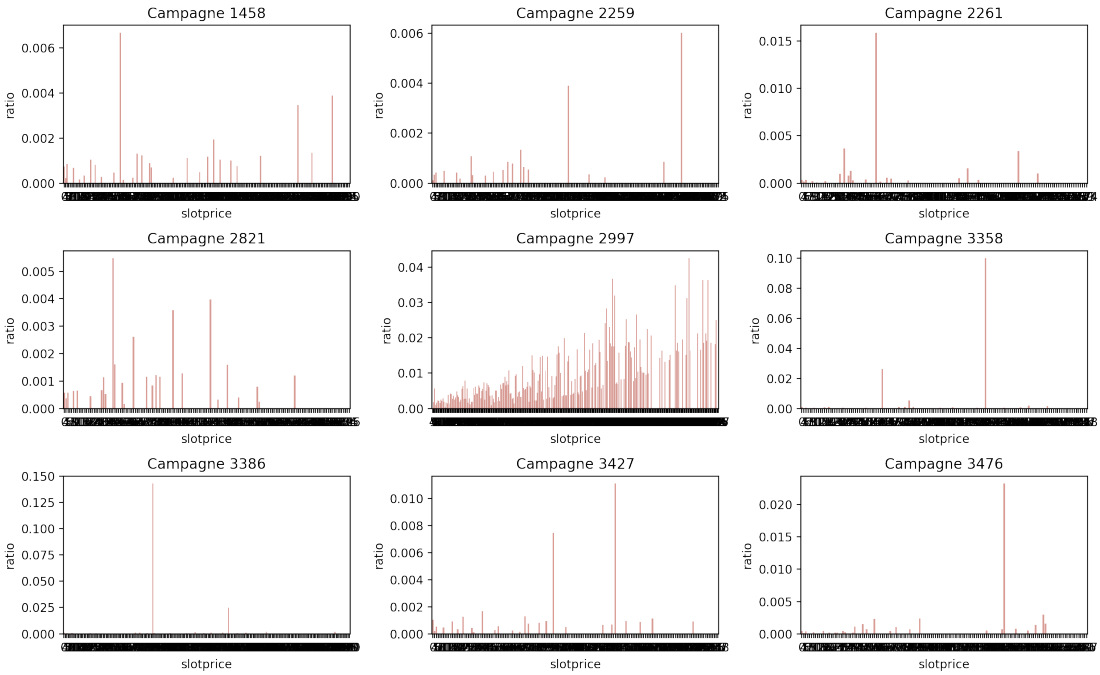


FIGURE A.10 – Graphique des ratio entre clic et non clics du nombre d'occurrences des modalités de la variable slotprice

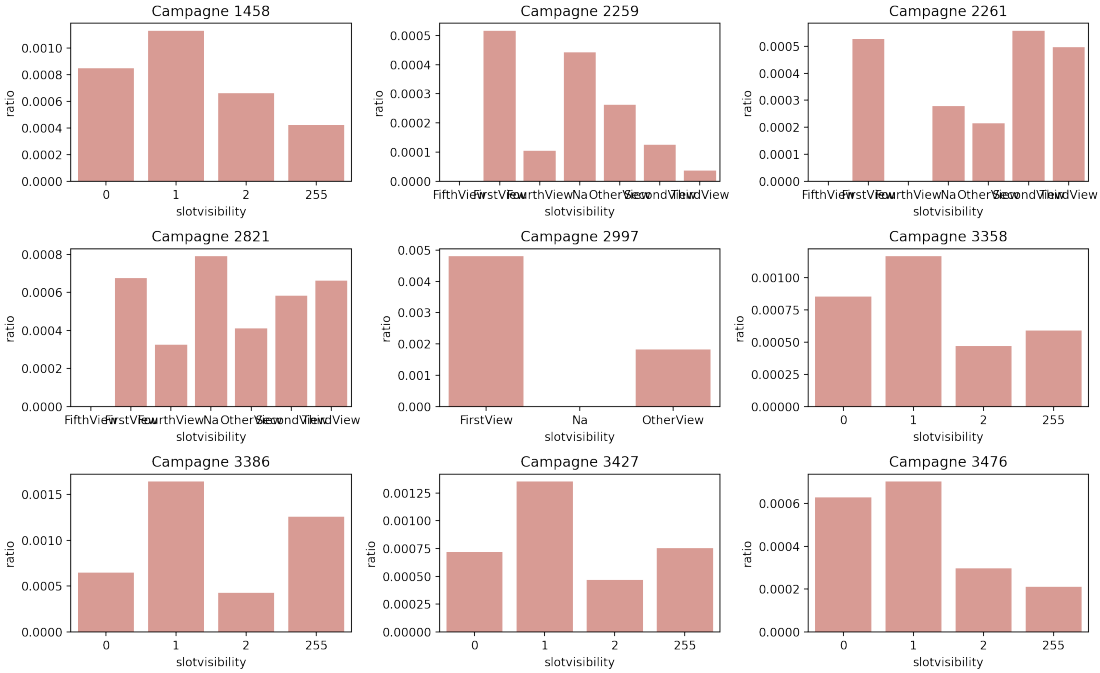


FIGURE A.11 – Graphique des ratio entre clic et non clics du nombre d'occurrences des modalités de la variable slotvisibility

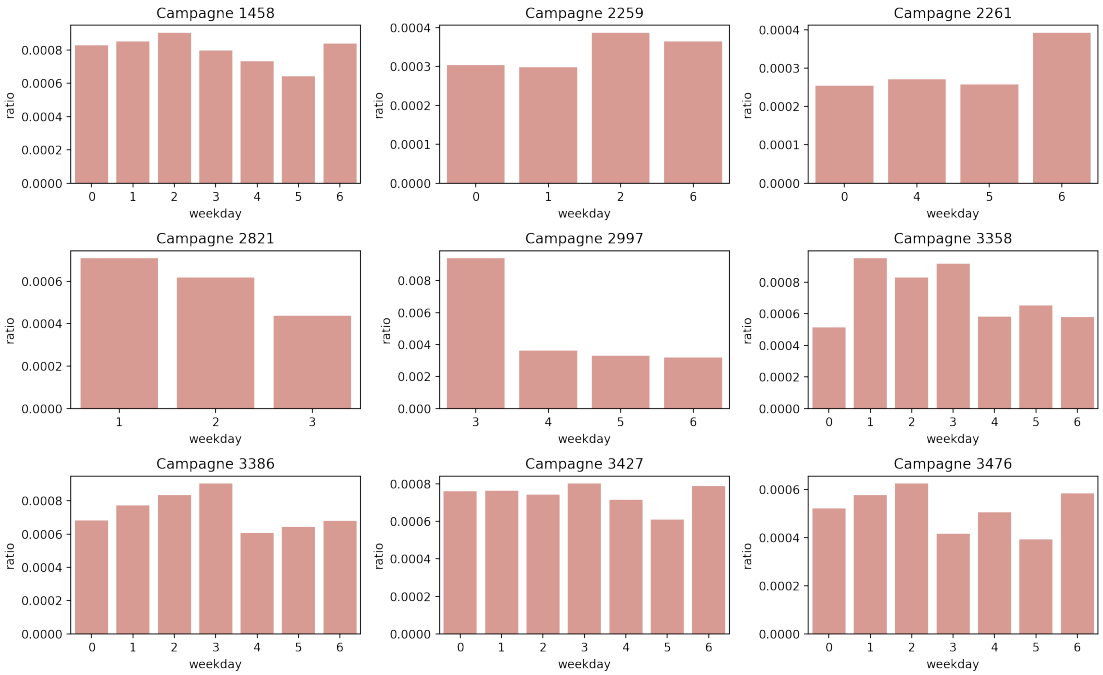


FIGURE A.12 – Graphique des ratio entre clic et non clics du nombre d'occurrences des modalités de la variable weekday

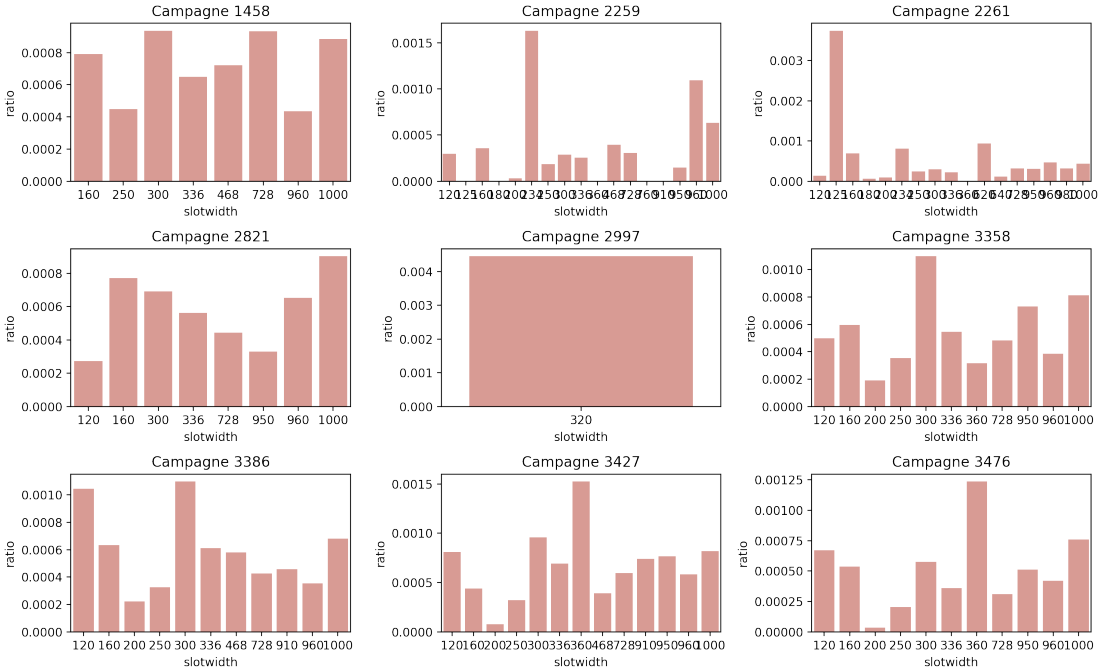


FIGURE A.13 – Graphique des ratio entre clic et non clics du nombre d'occurrences des modalités de la variable slotwidth

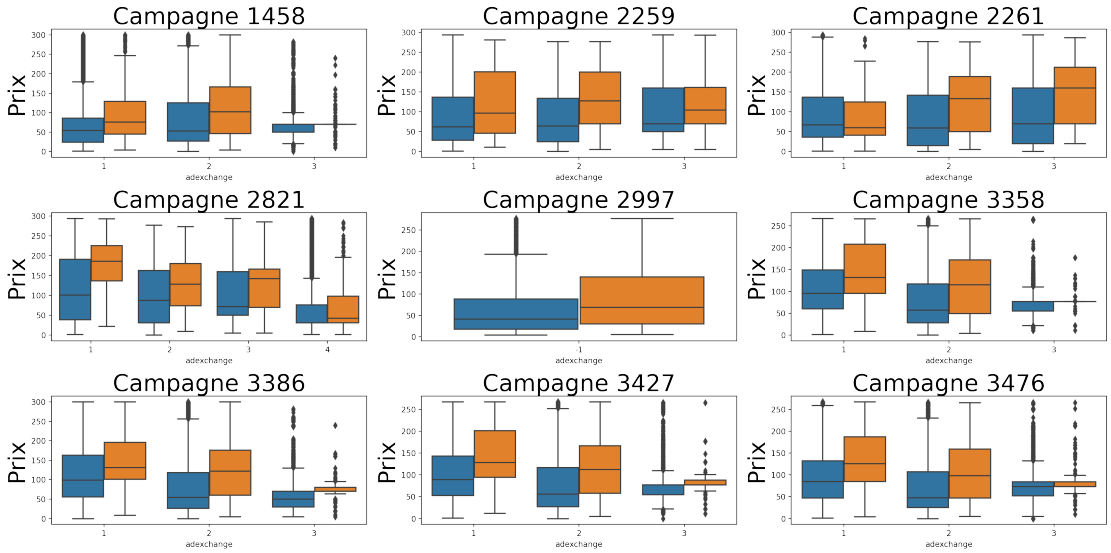


FIGURE A.14 – Distribution des prix pour les clics (orange) et non-clics (bleu) pour la variable adexchange

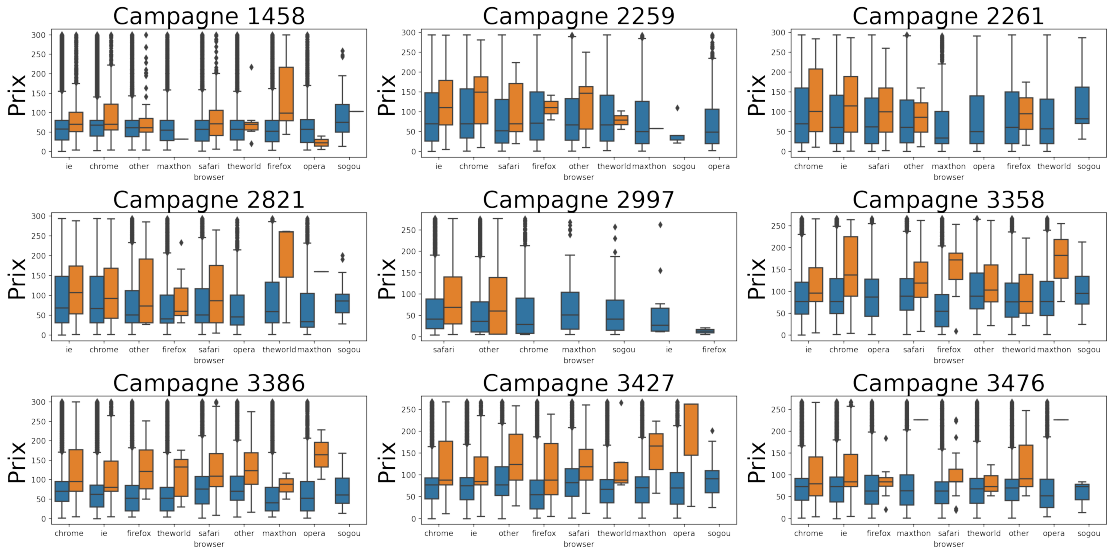


FIGURE A.15 – Distribution des prix pour les clics (orange) et non-clics (bleu) pour la variable browser

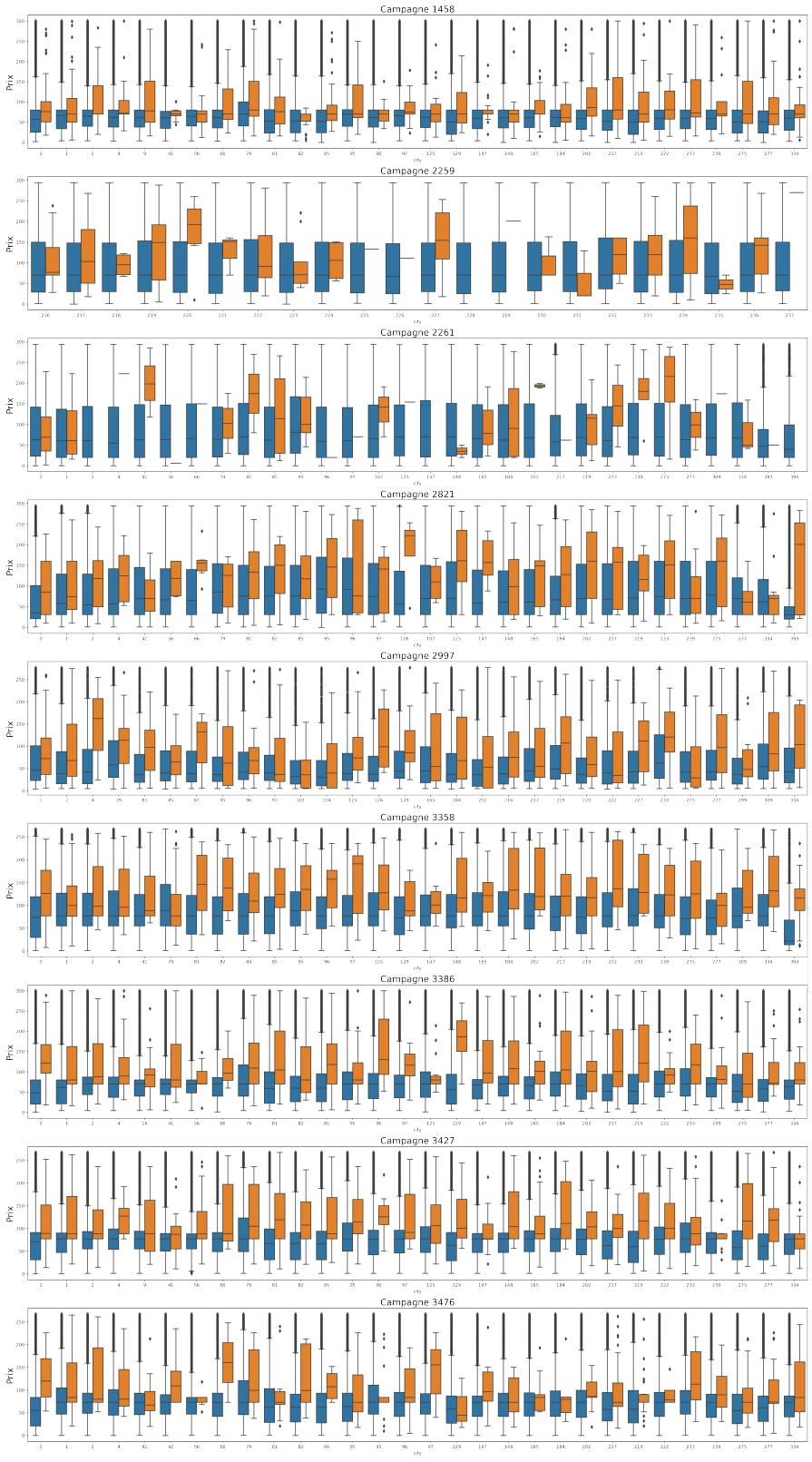


FIGURE A.16 – Distribution des prix pour les clics (orange) et non-clics (bleu) pour la variable city

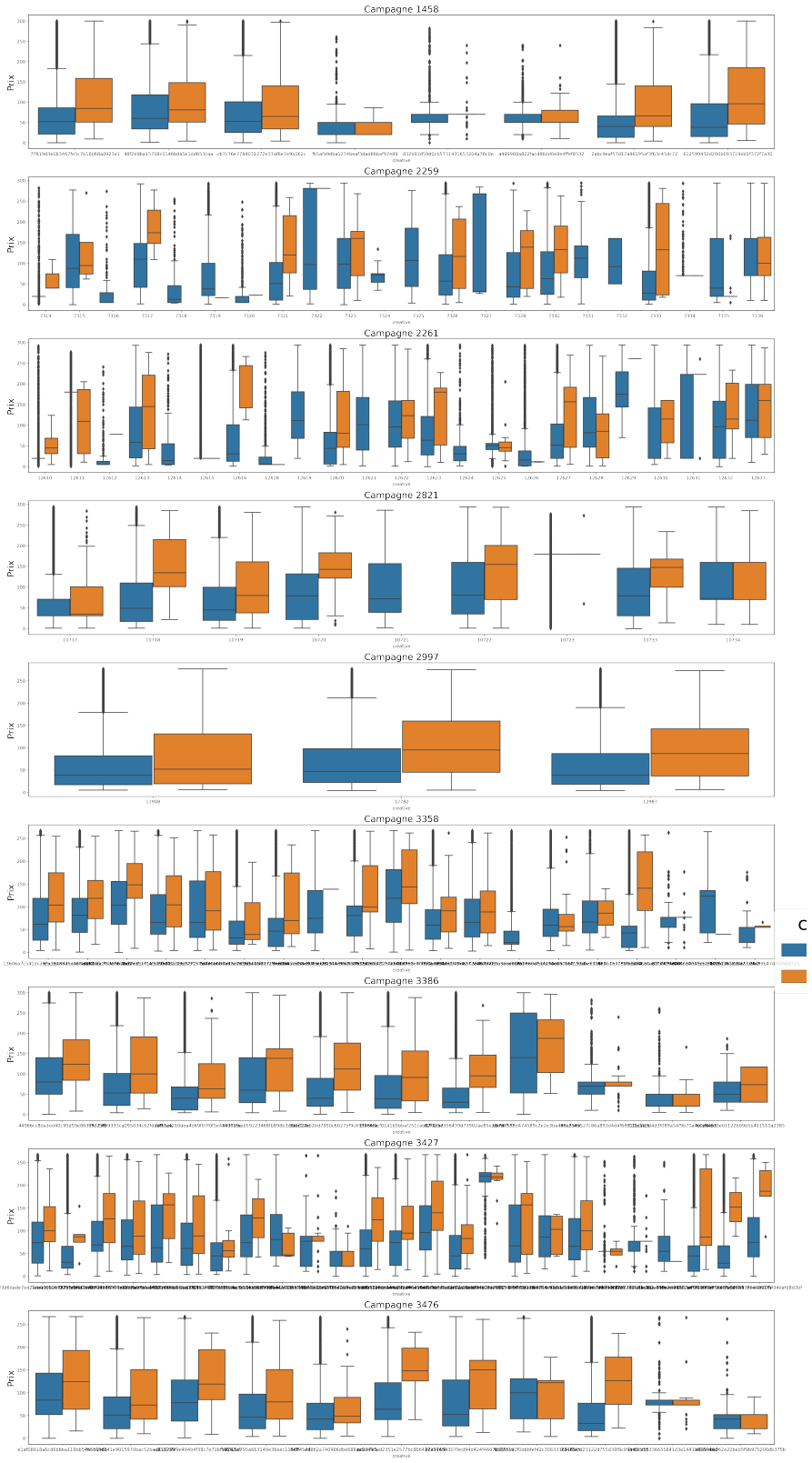


FIGURE A.17 – Distribution des prix pour les clics (orange) et non-clics (bleu) pour la variable creative

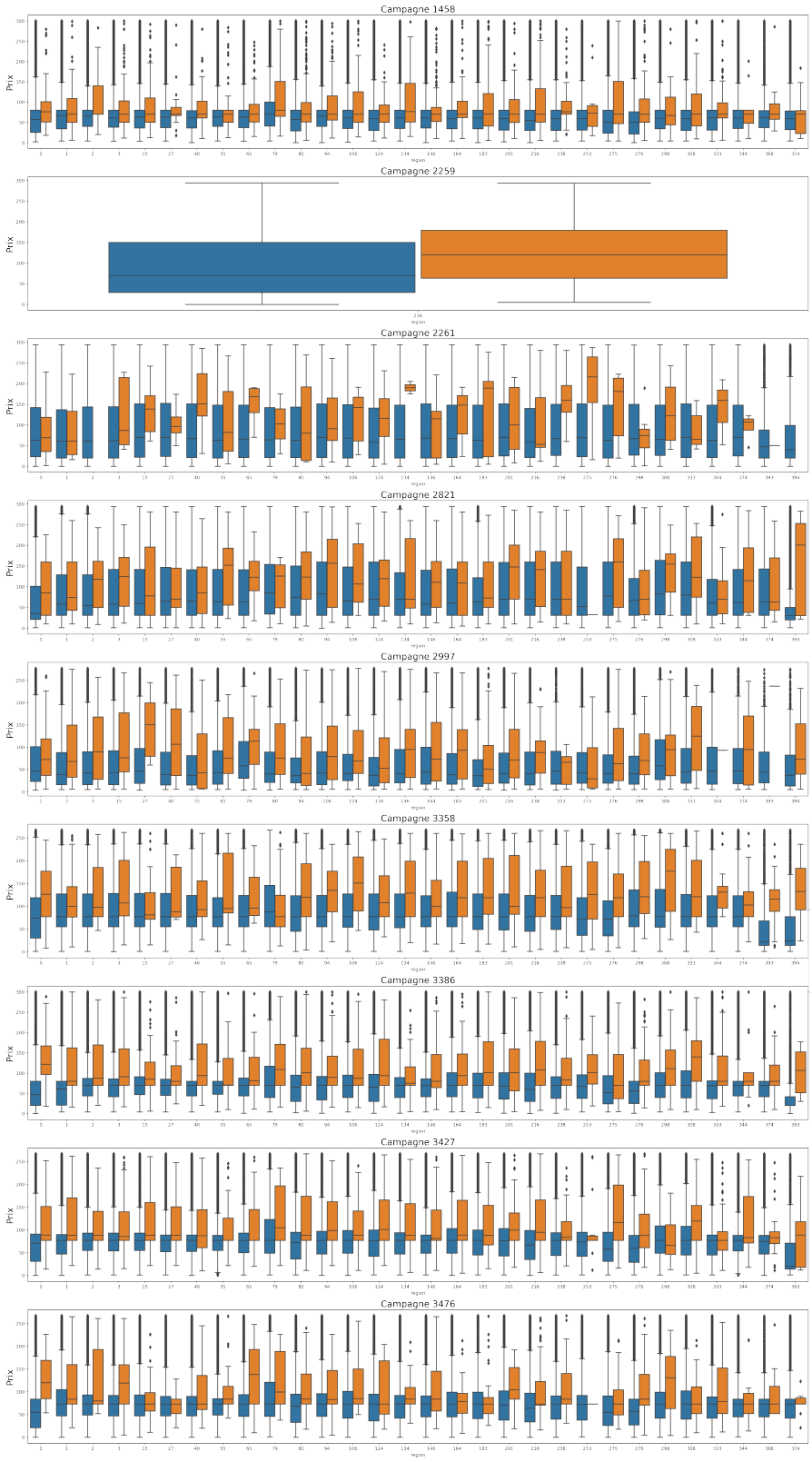


FIGURE A.18 – Distribution des prix pour les clics (orange) et non-clics (bleu) pour la variable region

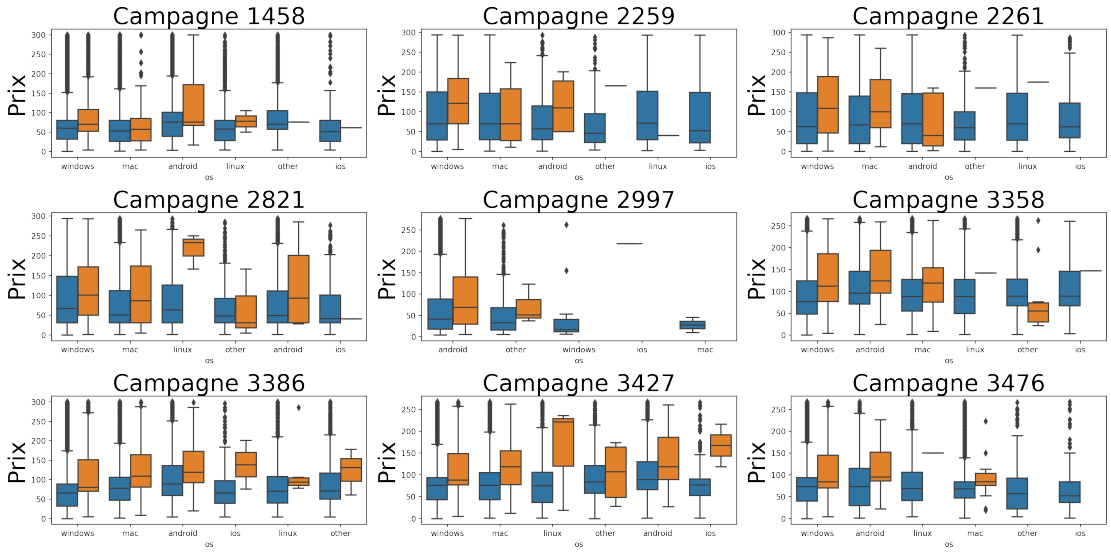


FIGURE A.19 – Distribution des prix pour les clics (orange) et non-clics (bleu) pour la variable os

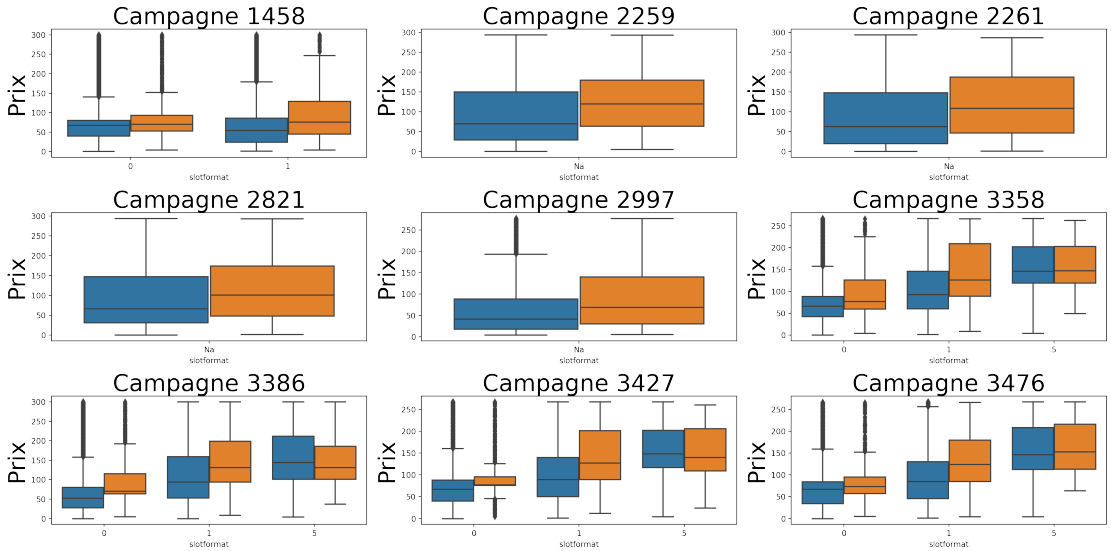


FIGURE A.20 – Distribution des prix pour les clics (orange) et non-clics (bleu) pour la variable slotformat

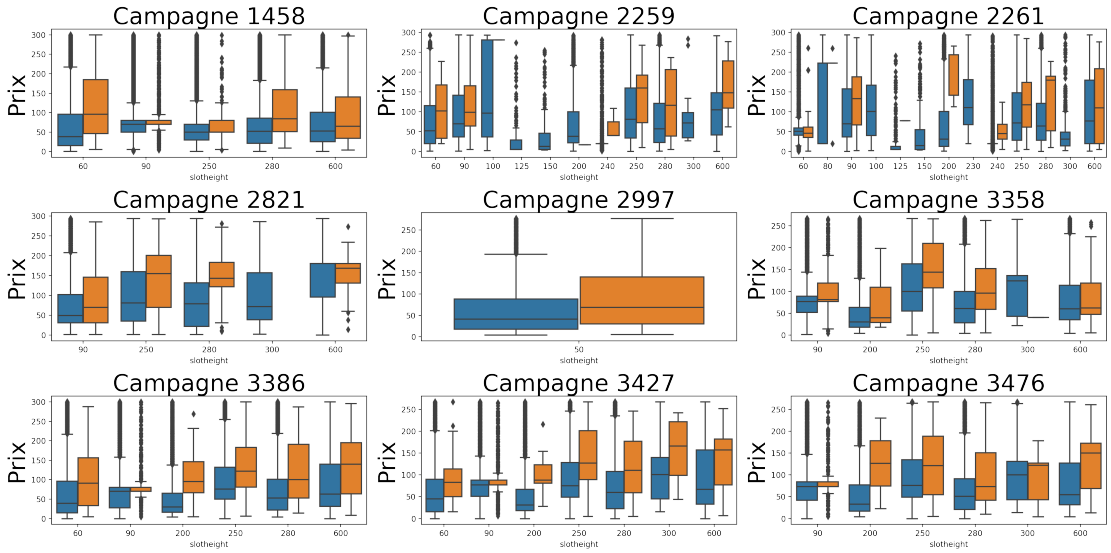


FIGURE A.21 – Distribution des prix pour les clics (orange) et non-clics (bleu) pour la variable slotheight

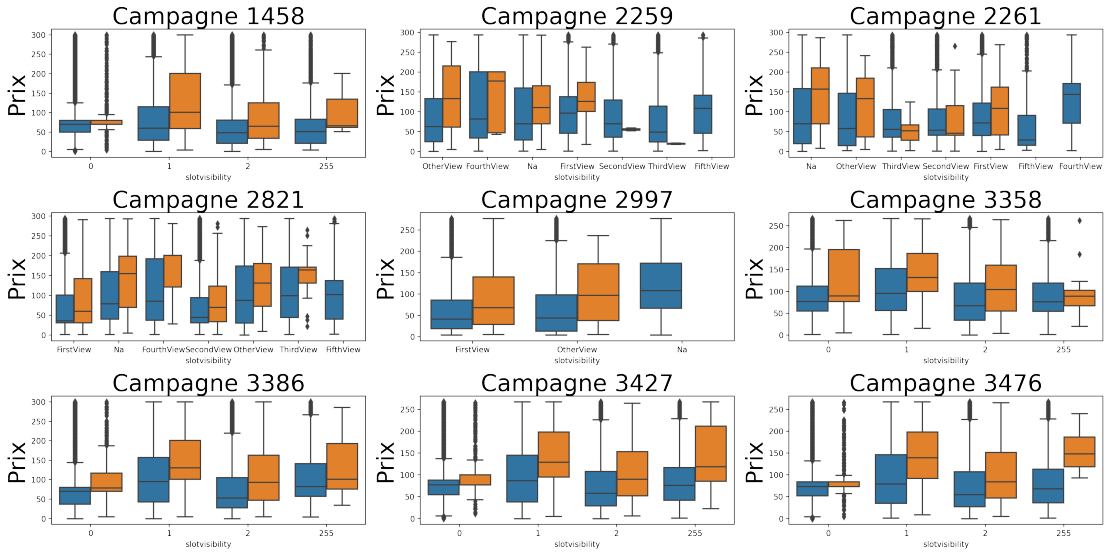


FIGURE A.22 – Distribution des prix pour les clics (orange) et non-clics (bleu) pour la variable slotvisibility

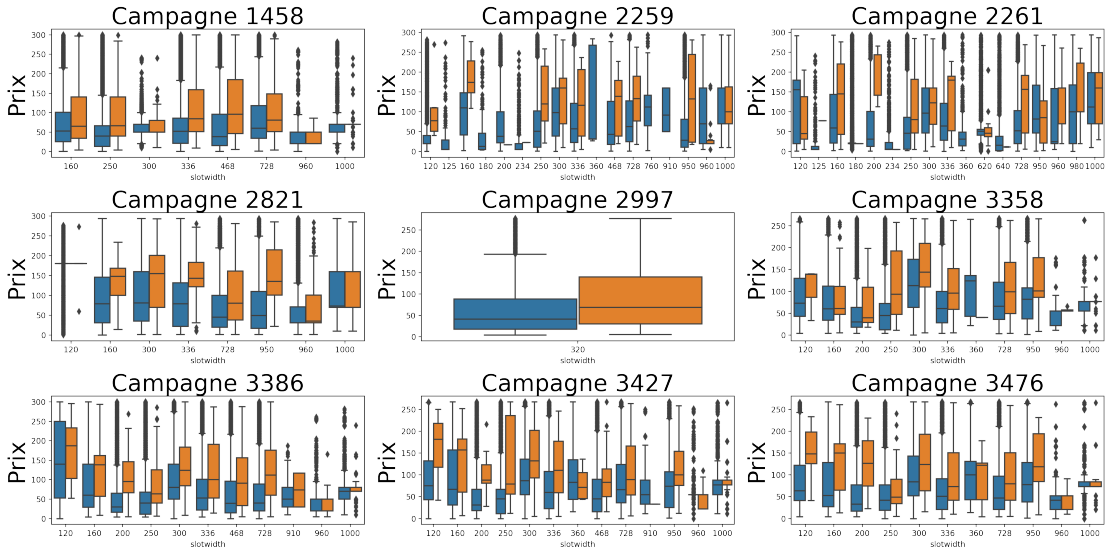


FIGURE A.23 – Distribution des prix pour les clics (orange) et non-clics (bleu) pour la variable slotwidth

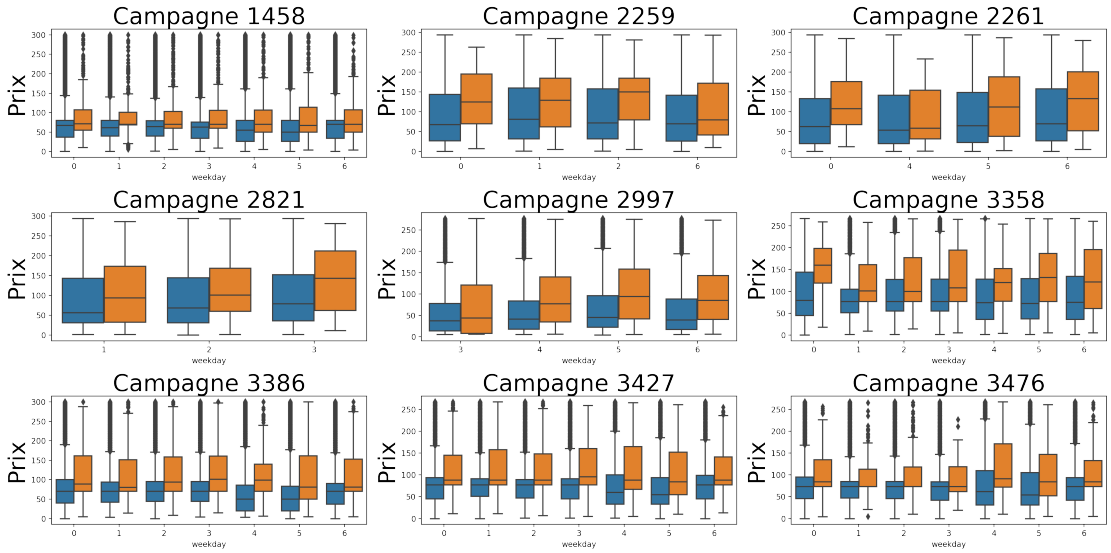


FIGURE A.24 – Distribution des prix pour les clics (orange) et non-clics (bleu) pour la variable weekday

Annexe B

Évenements rares et prédiction du clic

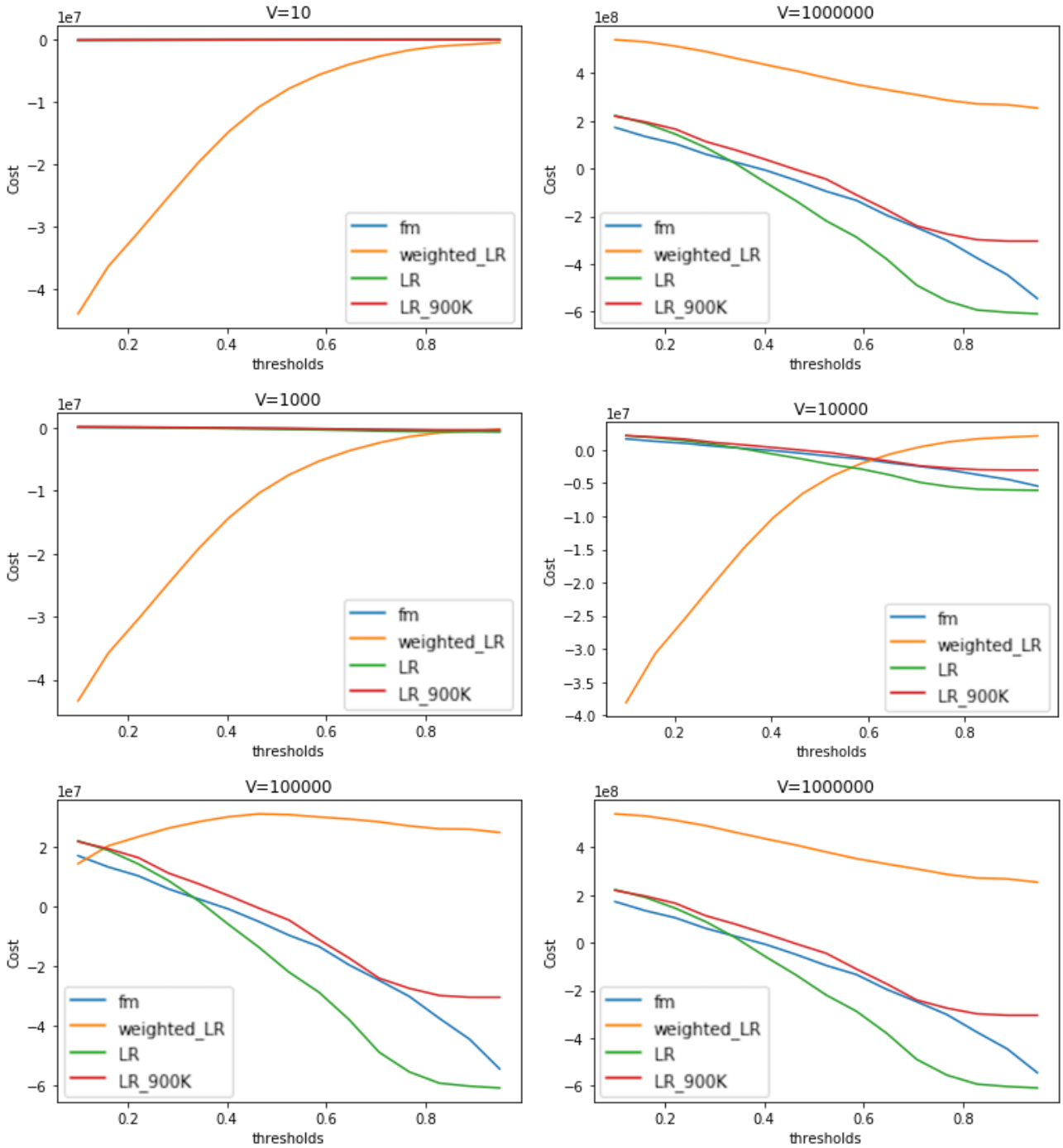


FIGURE B.1 – Value function value with false-negative penalization for different thresholds and different values of v for FM, weightedLogReg, LogReg and LR_900K (cost_ipy)

Annexe C

Śtratégies d'enchère

Stratégies d'enchère et apprentissage par renforcement

Table 1 – Performances croisées entre algorithmes de CTR et approche d'optimisation des enchères exprimées en termes de clics détaillées pour chaque campagnes.

Campagne	Algorithme CTR	Algorithme d'enchère	clics	clics max
1458	DFM	Constant	213	2454
		DDQN	**689**	
		LBBP	161	
		Linéaire	685	
	LR	Constant	213	
		DDQN	**1863**	
		LBBP	999	
		Linéaire	1858	
	wLR	Constant	213	
		DDQN	**1155**	
		LBBP	500	
		Linéaire	1121	
2259	DFM	Constant	23	280
		DDQN	101	
		LBBP	68	
		Linéaire	**103**	
	LR	Constant	23	
		DDQN	107	
		LBBP	69	
		Linéaire	**114**	
	wLR	Constant	23	
		DDQN	**34**	
		LBBP	14	

		Linéaire	31	
2261	DFM	Constant	11	207
		DDQN	117	
		LBBP	35	
		Linéaire	**118**	
	LR	Constant	11	
		DDQN	113	
		LBBP	30	
		Linéaire	**116**	
	wLR	Constant	11	
		DDQN	**39**	
		LBBP	10	
		Linéaire	38	
2821	DFM	Constant	60	843
		DDQN	**423**	
		LBBP	113	
		Linéaire	417	
	LR	Constant	60	
		DDQN	405	
		LBBP	77	
		Linéaire	**418**	
	wLR	Constant	60	
		DDQN	**74**	
		LBBP	21	
		Linéaire	**74**	
2997	DFM	Constant	108	1386
		DDQN	223	
		LBBP	220	
		Linéaire	**272**	
	LR	Constant	108	
		DDQN	**240**	
		LBBP	117	
		Linéaire	218	
	wLR	Constant	108	
		DDQN	138	
		LBBP	93	
		Linéaire	**140**	
3358	DFM	Constant	61	1358
		DDQN	**704**	
		LBBP	61	
		Linéaire	701	

	LR	Constant	61	
		DDQN	**1158**	
		LBBP	280	
		Linéaire	1150	
	wLR	Constant	61	
		DDQN	**735**	
		LBBP	59	
		Linéaire	728	
3386	DFM	Constant	115	2076
		DDQN	**982**	
		LBBP	182	
		Linéaire	976	
	LR	Constant	115	
		DDQN	1062	
		LBBP	179	
		Linéaire	**1065**	
	wLR	Constant	115	
		DDQN	**532**	
		LBBP	65	
		Linéaire	523	
3427	DFM	Constant	135	1926
		DDQN	695	
		LBBP	127	
		Linéaire	**707**	
	LR	Constant	135	
		DDQN	1397	
		LBBP	603	
		Linéaire	**1402**	
	wLR	Constant	135	
		DDQN	600	
		LBBP	131	
		Linéaire	**601**	
3476	DFM	Constant	68	1027
		DDQN	**333**	
		LBBP	136	
		Linéaire	**333**	
	LR	Constant	68	
		DDQN	522	
		LBBP	167	
		Linéaire	**523**	
	wLR	Constant	68	

DDQN	**191**
LBBP	81
Linéaire	190

le cnam

Slimane MAKHLOUF

Apprendre à enchérir : prédiction d'évènement rare et choix de stratégie.

HESAM UNIVERSITÉ

Résumé : Cette thèse porte sur l'amélioration des campagnes d'enchères pour l'affichage publicitaire en ligne. Nous considérons le problème à travers deux grandes questions : la prédiction de la probabilité de clic permettant d'obtenir une estimation de la valeur d'un affichage, et l'optimisation des enchères qui doit, en se basant sur cette estimation, gérer les montants des ordres et le budget afin d'obtenir le maximum de clics possibles. Les clics sur des publicités en ligne sont des événements rares. La prédiction de ce type d'événements requiert l'utilisation de modèles et de fonctions d'évaluation spécifiques. Nous étudions ces performances sous plusieurs fonctions d'évaluation nous permettant de montrer certains biais induits par les mesures de performances classiques. Nous présentons une mesure de performance spécifique aux enchères en temps réel (RTB) permettant de corriger les biais liés aux événements rares. Nous étudions les performances de plusieurs stratégies d'enchère et montrons que l'apprentissage par renforcement n'apporte pas d'amélioration significative par rapport aux autres approches entre autres à cause des problèmes de convergence de ce type d'approche notamment dus à la formulation des états du processus de décision markovien. Nous présentons une étude sur la convergence de l'apprentissage par renforcement et l'apprentissage de la formulation des états. Nous explorons l'utilisation d'autoencoders afin de synthétiser une formulation des états qui permettrait une meilleure convergence de l'apprentissage par renforcement.
Mots clés : Enchère en temps réel, Prédiction de clic, stratégie d'enchère, Prédiction d'événements rares, Apprentissage par renforcement, apprentissage de la représentation automatique.

Abstract : This thesis focuses on the improvement of real-time bidding for display advertising. We consider the problem through two main issues : the prediction of clicks probability and the bid optimization which, based on this estimate, should regulate the bids and the budget in order to maximize the number of clicks. Clicks on online ads are rare events. Predicting this type of event requires the use of specific models and evaluation functions. We study these performances under several evaluation functions allowing us to show some biases induced by classical performance measures and models. We present a performance measure specific to RTB that corrects the biases related to rare events. We study the performance of several auction strategies and show that reinforcement learning does not bring significant improvement over other approaches, among other reasons because of convergence problems of this type of approach, notably due to the formulation of the states of the Markov decision process. We present a study on the convergence of reinforcement learning and state formulation learning. We explore the use of autoencoders in order to synthesize a state formulation that would allow a better convergence of reinforcement learning.
Keywords :real-time auctions, click prediction, bidding strategy, rare event prediction, reinforcement learning, state representation learning.