Rapport de projet 8
Rossmann ventes

Mai 2019 - Xavier Montamat

Sommaire

1. Introduction

a. Prévision de série temporelle

Nous nous intéressons pour ce projet à la prédiction d'événements futurs. Et plus précisément les résultats futurs d'une série temporelle, en se basant sur un historique connu.

Ce type de problèmes a de nombreuses applications concrètes. On pense généralement à des événements météo tels que les chances de précipitation ou bien la température moyenne. Mais l'on trouve généralement des applications dans tous les domaines. On peut citer par exemple la prédiction des ventes d'un produit dans un magasin, de manière à anticiper les stocks. Ou bien la prédiction des logs d'une application, pour alerter sur une erreur ou un trafic anormal.

Certains événements ont plus aléatoires que d'autres. Mais si l'on arrive à identifier les facteurs de variations et à obtenir suffisamment d'historique de données, on pourra alors anticiper avec relative précision nos événements futurs. Si au contraire, peu de données sont disponibles, ou bien que les facteurs ne sont pas facilement identifiables, les tendances futures seront d'autant plus difficiles à prédire.

Un bon sujet aura donc des fluctuations diverses mais majoritairement explicables. Avec des événements aléatoires possibles mais ponctuels, ne constituants pas un facteur primaire de variation. Cela ne signifie pas qu'un environnement évolutif, par exemple un secteur en croissance, soit exclu. La croissance d'un marché, bien que complexe, reste prévisible dans une certaine mesure, et donc théoriquement possible à modéliser.

Pour ce qui est des données, il sera nécessaire d'avoir accumulé un historique au moins aussi long que la période que l'on souhaite prédire. Il est idéal d'avoir plusieurs répétitions de notre interval afin d'avoir davantage de chances d'identifier les tendances, et les différencier de fluctuations ponctuelles.

b. Sujet choisi

Pour ce projet j'ai choisi d'utiliser le dataset des magasins Rossmann, disponible sur Kaggle. Ce dataset date de 2016 et est un très bon exemple de mission professionnelle et concrète, s'adressant à un Data Scientist. Le but de la mission étant de prédire, avec un minimum d'erreurs, le montant des ventes de chaque magasin du groupe. Ce de manière journalière et jusqu'à 2 mois en avance.

Le dataset comporte plus d'un millier de magasins et environ 2 années et demi d'historique pour chacun d'entre eux. En plus de l'historique du montant des ventes par jour, nous disposons de divers données pour chaque magasin telles que sa structuration interne, ou encore les promotions effectuées sur telle ou telle période. Ces divers critères

en font un projet à la fois clair et complet. C'est un dataset qui m'offre la possibilité de travailler sur une prédiction temporelle réaliste et réalisable.

2. Analyse des données

a. Intemporelles

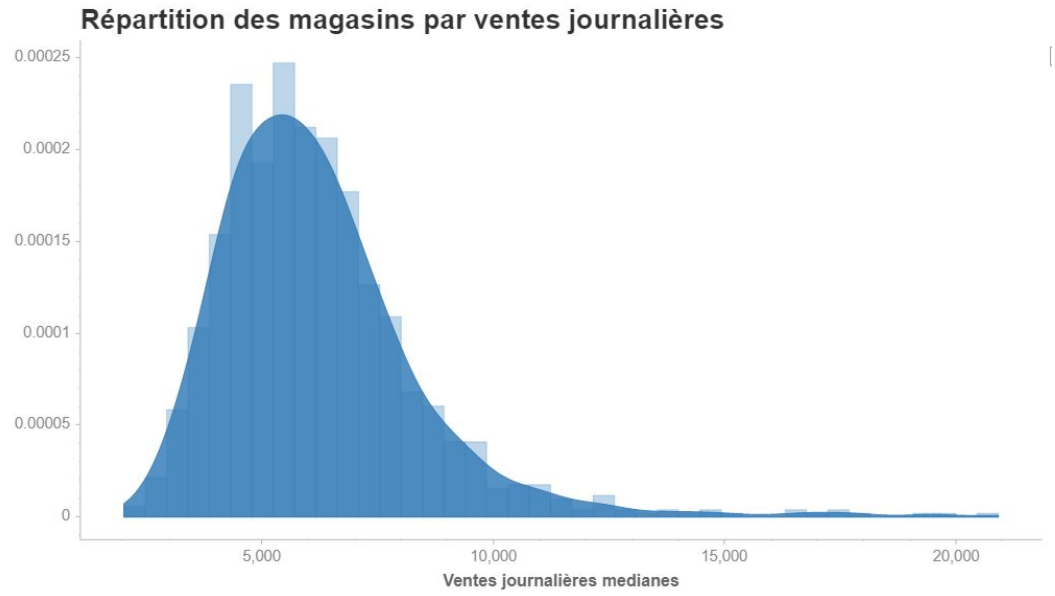
Comme pour tout projet, nous commencerons par observer les données pour obtenir une vue d'ensemble. Cela nous permettra de comprendre comment aborder le sujet par la suite.

Nous disposons de divers données que nous pouvons catégoriser en 5 groupes

- Spécificités intemporelles des magasins
- Periodes de promotions
- Periodes de vacances
- Compétiteurs
- Historique des ventes et nombre de clients

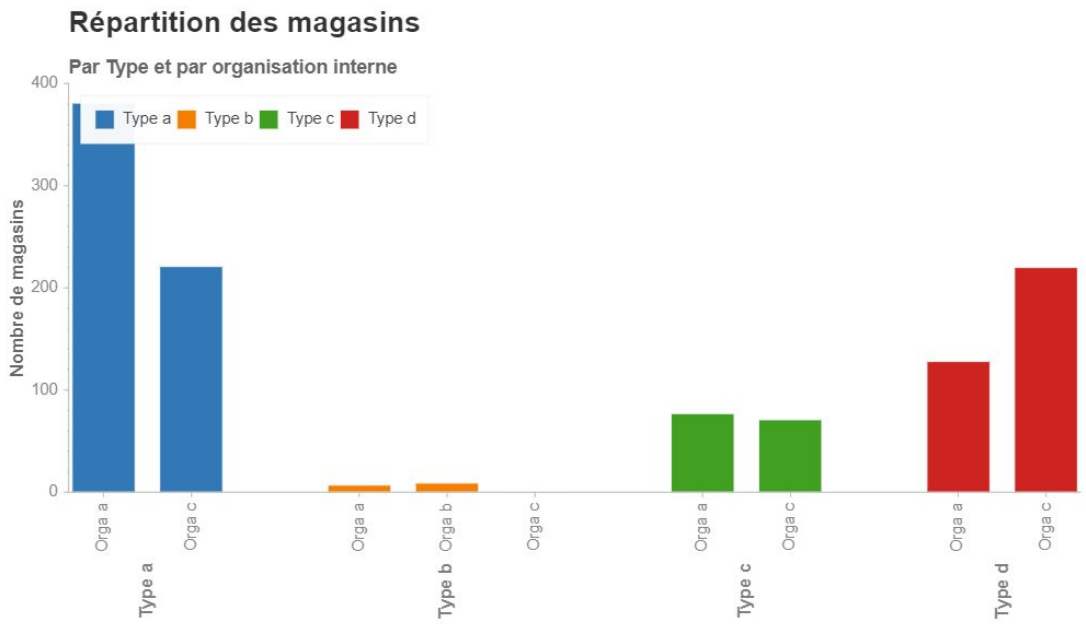
La majorité de nos observations se feront autour des 'ventes moyennes', qui est la métrique cible que nous souhaitons prédire.

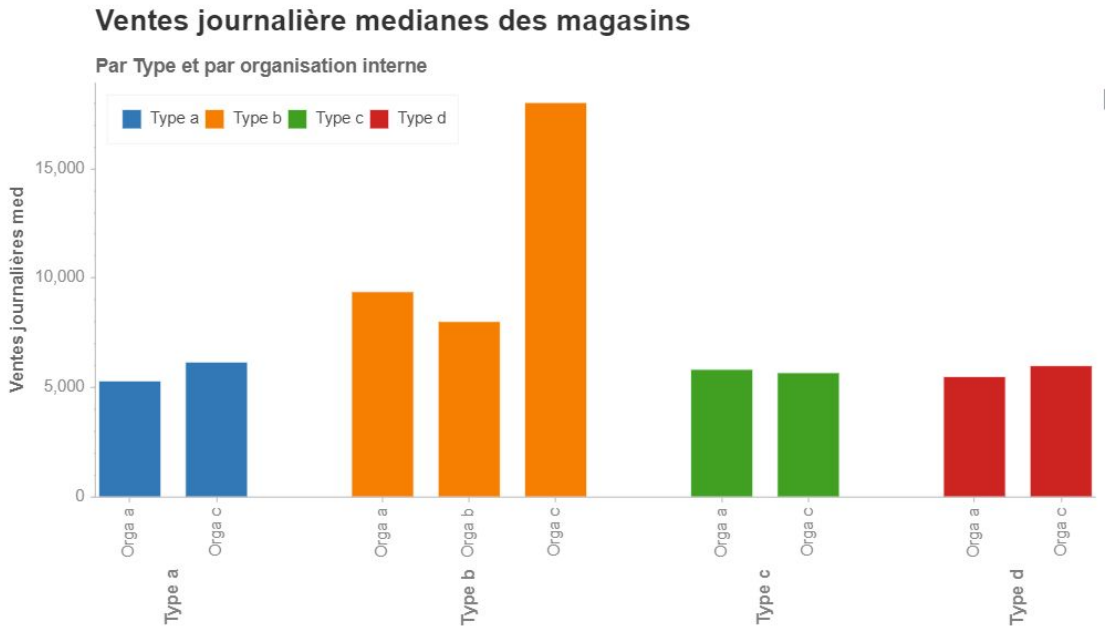
Regardons tout d'abord la répartition de nos 1115 magasins, en fonction de leur chiffre d'affaire.



La majorité des magasins réalisent environ 6000€ de chiffre d'affaire par jour. Avec certains d'entre eux dépassant les 20000€, mais aucun en dessous de 2000€. On peut imaginer que des magasins ayant réalisé des chiffres journaliers inférieurs sont fermés car déficitaires.

Regardons à présent les catégories de nos magasins. Chaque magasin est classé par 'Type', catégorisé de A à D. Ainsi que par organisation interne, catégorisé de A à C. Nous ne possédons pas davantage d'information sur la signification de ces groupes , mais l'on pourrait imaginer qu'ils aient une relation avec les ventes. Regardons donc la répartitions de ces catégories, ainsi que le chiffre d'affaire médian de chacune.

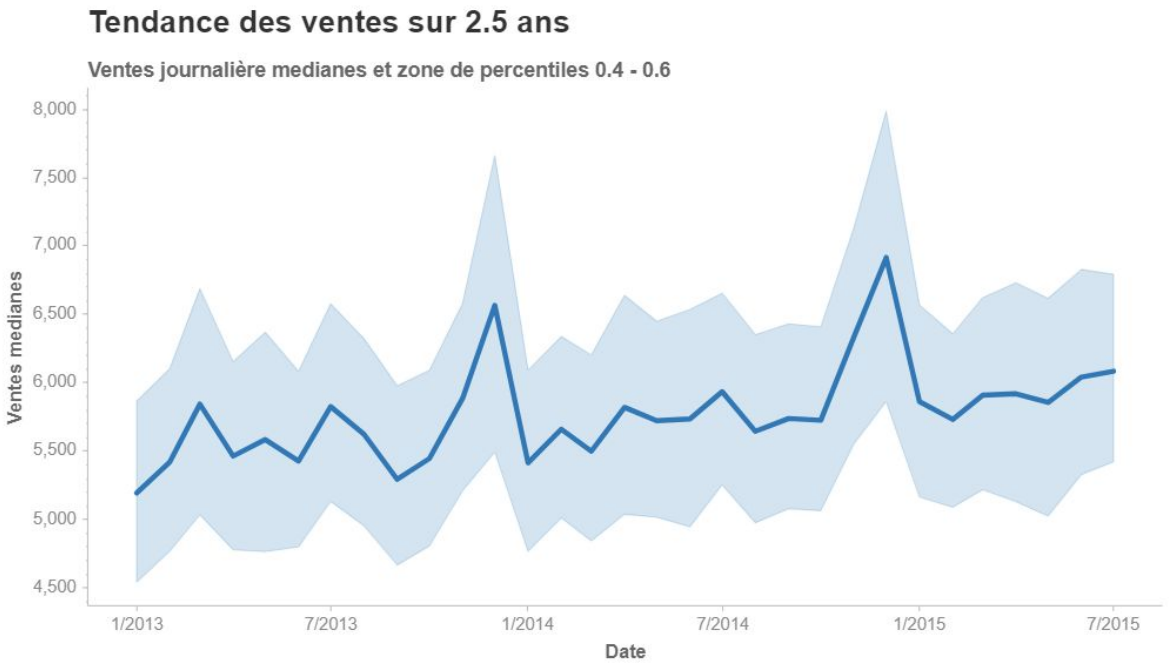




On observe que les magasins de type B ont tendance à réaliser de plus gros volumes de vente. Cependant ils sont aussi très peu nombreux (seulement 17 magasins de ce type). Les différences entre les types A, C et D , ainsi que de leurs organisations, sont moins évidentes pour ce qui est de leur effet sur le chiffre d'affaire.

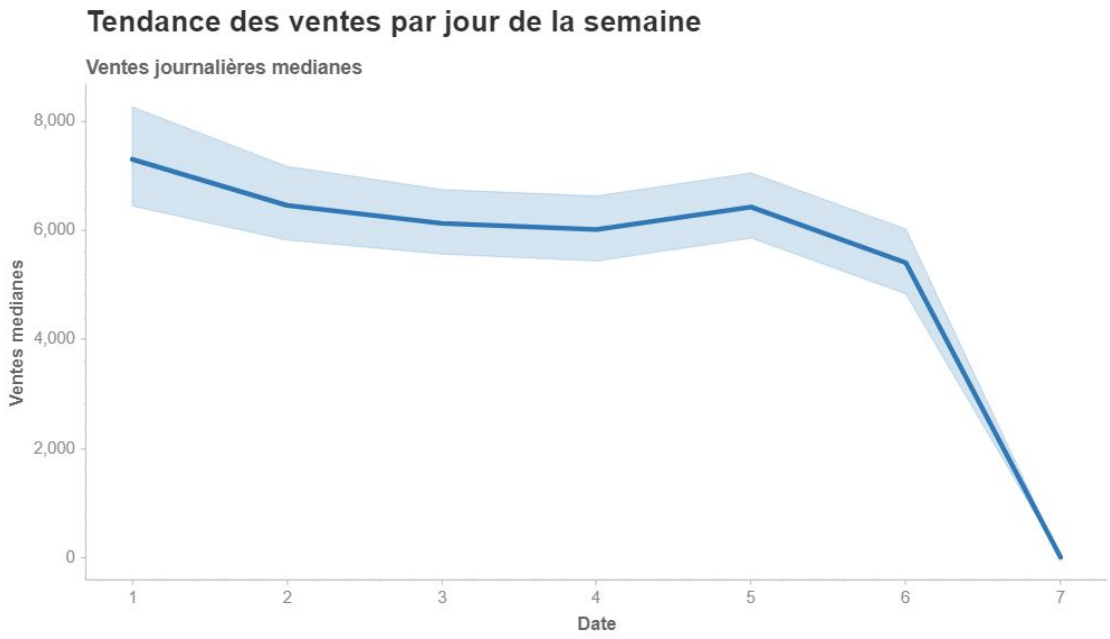
b. Temporelles

Observons à présent les courbes saisonnières des ventes de nos magasins. Cela nous permettra d'identifier des tendances telle qu'une croissance linéaire ou exponentielle, montante ou descendante, etc. Affichons la courbe globale sur l'ensemble du dataset. C'est le type de courbe est celle qui pourra être extrapolée par un modèle de prédiction temporel.



On note une légère croissance linéaire de nos ventes globales au cours de ces 2.5 ans. On peut aussi noter une tendance annuelle, où les meilleurs chiffres sont ceux de fin d'année. Probablement dû aux fêtes, ce pic est typique du marché de la vente. Les autres mois de l'année possèdent une tendance moins évidente, mais qui peut tout de même être utilisée par les algorithmes.

Nous allons cependant nous intéresser à un interval plus court, celui de la tendance hebdomadaire.



Ici également, la répartition n'est pas équilibrée. On observe de bonnes ventes le lundi et le vendredi, avec une diminution durant la semaine, et une fermeture majoritaire des enseignes le dimanche. Indiquer le jour de la semaine sera probablement très pertinent pour l'algorithme de prévision temporelle.

3. Modèlisation

a. Métrique d'évaluation

Pour ce projet la métrique choisie par Rossmann pour leur concours Kaggle est le Root Mean Square Percentage Error (RMSPE) .

Elle se calcule ainsi :

$$\mathrm{RMSPE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{y_i}\right)^2},$$

Elle présente l'avantage par rapport à RMSE d'être invariante de l'unité. Cependant elle ne peut pas être appliquée si certaines ventes sont à 0. Nous ignorerons donc les dimanches et jours fériés, comme précisé dans les informations du concours.

b. Transformations effectuées

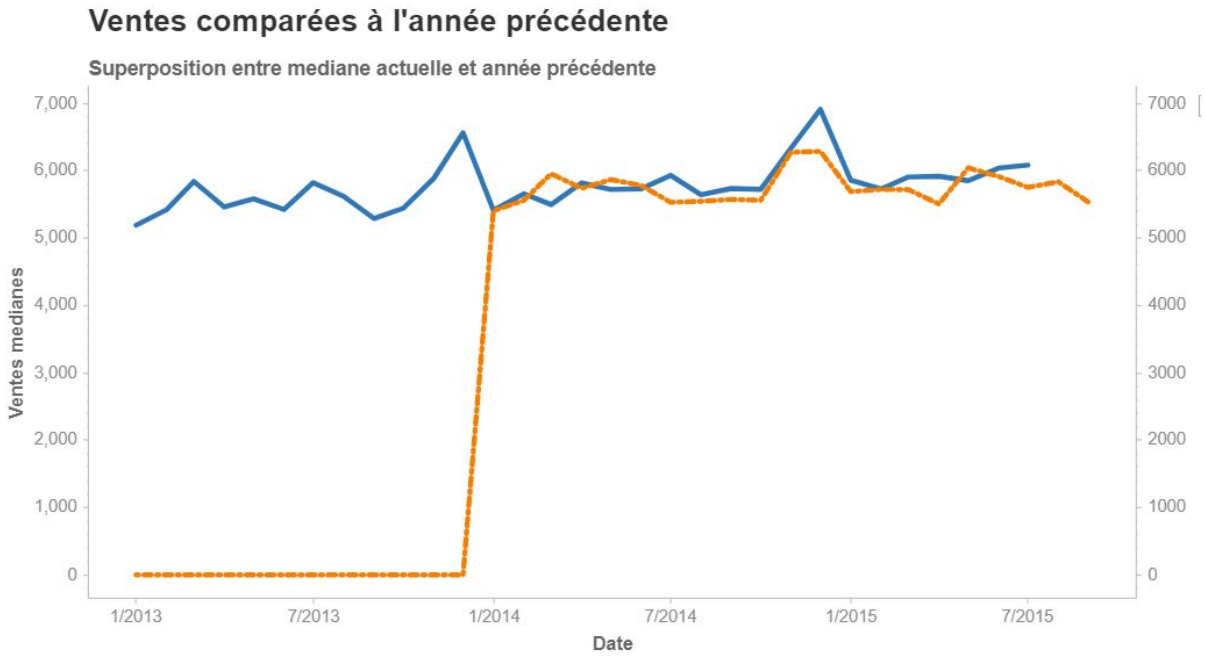
En fonction du modèle utilisé, il y a plus ou moins de transformations à réaliser avant de pouvoir entraîner notre modèle. Pour les modèles plus 'classiques', il y aura généralement d'avantage de transformations à effectuer manuellement.

Les catégories gagnerons généralement à être transformées par 'One Hot encoding'. Ce qui permet à l'algorithme de les traiter de manière distinctes, et non comme des variables continues. Pour ce projet, nos types et organisation interne sont des catégories, ainsi que nos numéros de magasins.

Il y a aussi pour ce projet de nombreuse variables de type 'date'. Celles ci nécessitent généralement d'être traitées au préalable afin d'être utiles à notre modèle. Par exemple, nous avons l'information de la date d'ouverture d'un magasin compétiteur dans le même secteur. Mais Il serait plus utile de traduire cela en nombre de jour avant et après son ouverture. Cela permettra à notre algorithme de repérer des changements tels qu'une baisse des ventes liée à l'ouverture d'un compétiteur, peut importe la date exacte. La même démarche peut s'appliquer aux promotions et aux vacances.

Enfin, pour ce qui est de l'historique de vente, généralement on peut ajouter l'information de la valeur médiane, sur une période glissante. Par exemple, sur le trimestre passé, le semestre, ou encore l'année. En faisant attention au data leak. J'ai tenté tenté plusieurs segments et laissé l'algorithme décider de ceux qui sont le plus pertinents. Une valeur qui semble assez efficace est celle des ventes de l'année précédente, pour la même semaine.

Nous pouvons observer sur le graph ci dessous une superposition entre les ventes actuelles et celles de la semaine correspondante sur l'année précédente.



c. Prophet

Prophet est une librairie 'State of the art', développée et utilisée en interne par Facebook, un des leaders du marché en Intelligence Artificielle. Apparue en 2017, elle possède plusieurs avantages qui la rende très intéressante. Elle est libre, simple, rapide, et plutôt efficace.

Par contre, il ne lui est pas possible d'exploiter des données autres qu'une série chronologique. Par exemple, nos données de types catégories et distance de la compétition ne seront pas exploitées. De plus, ne sachant pas différencier les magasins, il faudra créer autant de modèle que de magasins, et le modèle ne pourra pas extrapoler sur un magasin inconnu.

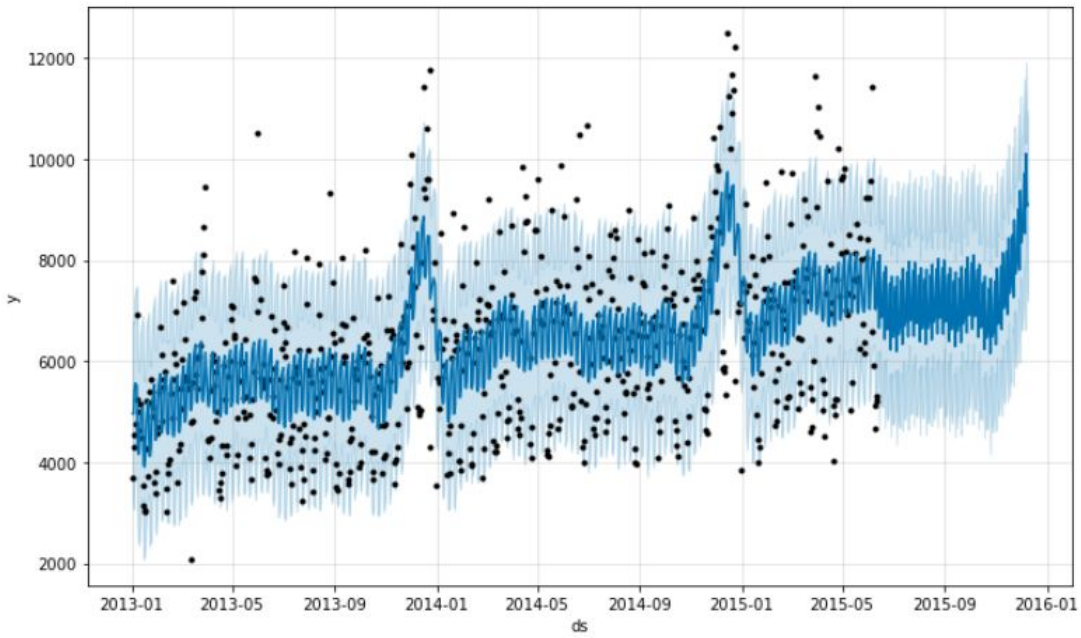
La librairie étant simple et rapide, cela reste possible de réaliser un modèle distinct pour chaque magasin.

Exemple des résultats pour un magasin

Prophet analyse les tendances de ventes sur ce magasin , comme nous l'avons fait sur l'ensemble du dataset dans la partie précédente.

En plus des tendances, prophet enregistre les minima et maxima par période, de manière à établir un interval de confiance. A partir de ces données, Prophet n'a plus qu'à extrapoler sur des dates fournies. Sans plus d'informations. Cela donne une courbe de prédiction qui semble réaliste avec un minimum d'informations (seulement l'historique des ventes).

Tendance et prévision des ventes par Prophet



En bleu nous avons les prédictions avec l'interval de confiance. Les points noirs sont les valeurs réelles journalières. On peut observer que la courbe suit relativement bien nos valeurs, mais ne parvient pas à anticiper les grosses irrégularités.

Les résultats obtenus constituent tout de même une bonne baseline pour une prédiction temporelle 'state-of-the-art'.

En créant un modèle pour chaque magasin, entraîné sur tout le dataset sauf les 48 derniers jours, nous obtenons un score global **RMSPE de 0.22**

Comparons ces résultat avec ceux d'un autre modèle, plus classique.

d. XGBoost

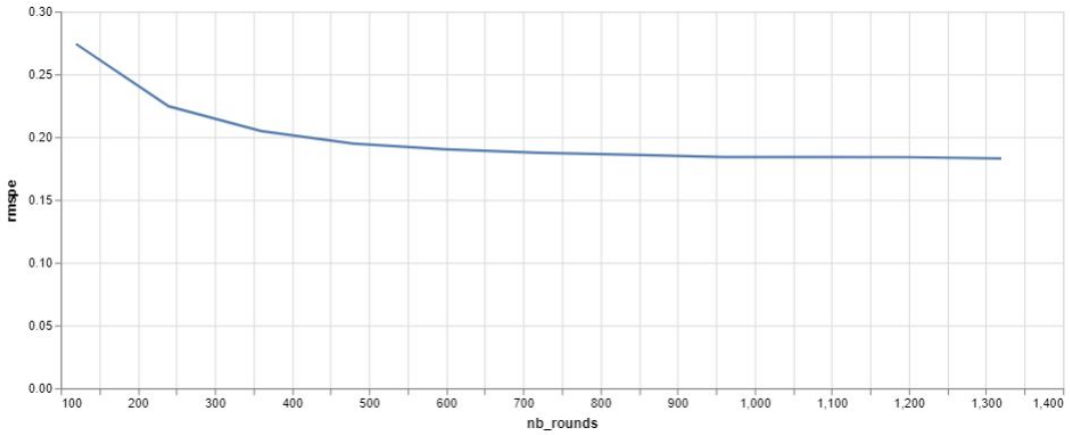
XGBoost est un algorithme très populaire pour son efficacité dans de nombreuses tâches de machine learning. C'est une évolution des Random Forests, qui y ajoute la notion de 'Gradient Boosting' pour accélérer le processus et éviter les mauvaises classifications. XGBoost est également capable de résoudre un problème de régression, ce qui est justement notre sujet.

L'avantage de XGBoost par rapport à Prophet est qu'il sera capable d'exploiter des informations autres que l'historique des ventes. Ce qui lui permettra d'exploiter d'autres variables et évènements influents, tels que l'ouverture d'un compétiteur, les périodes de promotions, etc.

L'algorithme possède de nombreux paramètres à configurer, principalement pour gérer l'overfitting. Après avoir mis au point ces paramètres, nous pouvons commencer à entraîner notre modèle final.

Plus notre modèle apprend, moins il fera d'erreur sur notre set d'entraînement. Cependant, un entraînement trop long aura tendance à overfitter. Nous obtiendrons donc un moins bon score sur notre set de validation.

Score RMSPE en fonction du nombre de rounds



Malgré un entraînement long, le modèle n'a pas overfitté.

Au bout de 1200 rounds, nous obtenons un score **RMSPE de 0.18**

Ce score est donc meilleur que les 0.22 obtenus avec Prophet.

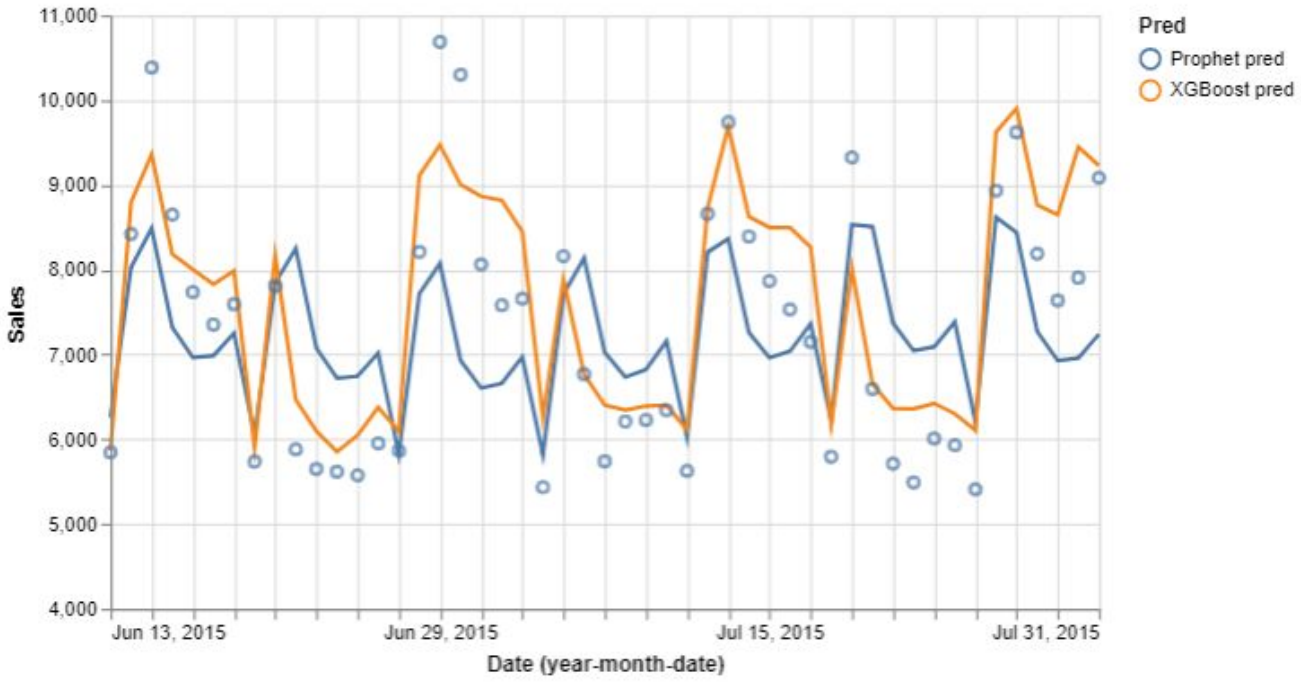
Nous avons également la possibilité de voir l'importance estimée des features de XGboost. Ce dernier affiche une préférence pour exploiter le numéro de magasin, la distance avec la compétition, ainsi que les derniers jours de ventes glissantes. D'autres informations jugées les moins utiles ont été retirées, afin de faciliter l'entraînement du modèle. Par exemple, l'information sur le type de jours fériés était superflue, et génèrent davantage de bruit qu'elle n'aidait à la précision.

4. Résultats

a. Analyse des résultats

Le score obtenu en utilisant Prophet a été inférieur à celui obtenu par XGBoost. (0.22 contre 0.18 d'erreur RMSPE). On peut supposer que les performances du dernier sont augmentées par son utilisation d'informations additionnelles par rapport à Prophet.

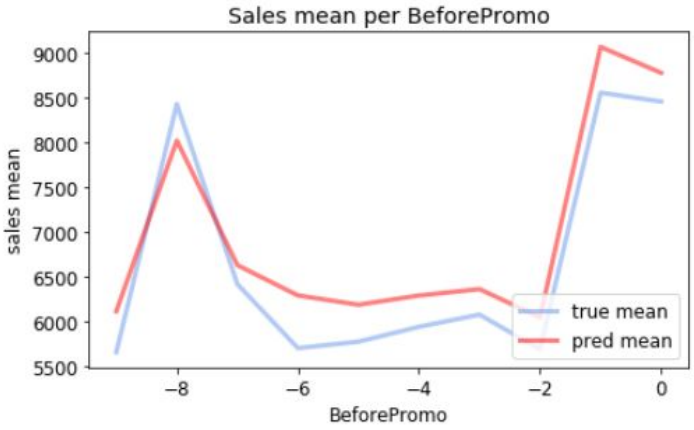
Tentons de comparer les modèle de manière visuelle en les observant sur un même graphique. Nous faisons une moyenne journalière des ventes de nos 1115 magasins que nous afficherons par des points. Sur le même graphique , nous affichons également les prédictions de chacun des algorithmes sur les 48 jours de test.



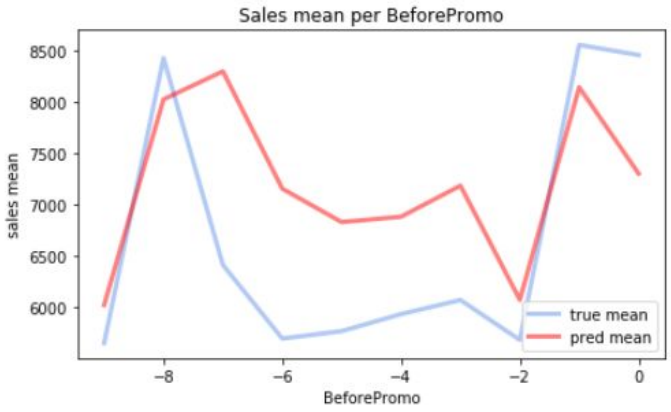
Nous avons en orange les prédictions XGBoost, et en bleu les prédictions Prophet. On peut observer comme prévu que la courbe XGBoost se rapproche davantage des points 'véritables'. Il y a une sorte de double tendance sur notre interval de test, où une semaine sur deux réalise davantage de ventes. La courbe orange suit correctement cette tendance et arrive à une prédiction plus réaliste que Prophet. La courbe bleu est beaucoup plus monotone et suit une répétition exclusivement hebdomadaire sur cet interval.

Nous pouvons également comparer les résultats des deux modèles en fonction de certains attributs. Par exemple en comparant les ventes avant une promotion

XGboost



Prophet



On observe un bien meilleur raccord côté XGboost pour les prévisions moyennes de cette métrique. En effet, Prophet n'a pas pu exploiter les dates de promotion lors de son entraînement, ou de sa prédiction.

Par curiosité, j'ai tenté de combiner les deux modèles, en prédisant tout d'abord les ventes de chaque magasin avec prophet. Puis j'ai entraîné un modèle XGboost en exploitant ces prédictions, ainsi que les autres colonnes non temporelles et non accessibles à Prophet. Cela m'a permis de réduire encore l'erreur RMSPE de 0.14. Cependant, j'ai préféré garder le modèle XGBoost classique comme modèle final. Ce dernier étant davantage exploitable et explicable dans un contexte réel d'entreprise.

b. Conclusion

Malgré les performances Etat de l'art de la libraire Prophet de Facebook, on se rend compte que son utilisation se limite à des cas peu complexes. Ainsi, pour le sujet choisi, XGboost a pu exploiter davantage d'informations et obtenir moins d'erreurs. Ce malgré qu'il n'ai pas été exclusivement conçu dans le but de réaliser des prédictions temporelles.

Il pourrait être intéressant de comparer ces résultats avec davantages de modèle tels que ceux en réseaux de neurones, qui sont également dans l'état de l'art des prédictions temporelles.

5. Etat de l'art

Recurrent Neural Networks

Les réseaux de neurones de type RNN sont spécialement adaptés aux prédictions temporelles, ou autres prédictions séquentielles. Ils sont particulièrement populaires actuellement à cause de leur performances comparées à des techniques plus anciennes.

De manière abstraite, ils sont capables de comprendre un contexte grâce à leur mémoire qui se souvient de l'historique des variables. En pratique, le réseau va boucler pour avancer séquentiellement dans son entraînement, tout en mémorisant l'entrée et sortie à chaque boucle. Cela lui permet au fur et à mesure de comprendre comment les évènements passés influencent les événements présents. Il sera ainsi capable d'extrapoler à partir d'une simple séquence, et même d'être combiné à d'autres paramètres intemporels.

Son inconvénient principal est qu'il n'oublie pas (du moins dans sa version classique), et donc va être de plus en plus gourmand à entraîner au fur et à mesure que l'historique grandit.

WaveNet

WaveNet est un modèle de réseau de neurones de type CNN, développé par Google, servant à générer des sons réalistes. Il a la particularité de pouvoir générer des sons sur une fréquence de 16 kHz, soit 16K prédictions sur une seule seconde à générer, avec une précision jusqu'à présent inégalée.

Cette technique, adaptée d'un réseau de neurones CNN pour images 2D, est ici adaptée pour prédire une onde sonore 1D.

En Mars 2018, les chercheurs Glib Kechyn, Lucius Yu, Yangguang Zang et Svyatoslav Kechyn ont tenté d'utiliser ce même algorithme pour prédire les ventes d'un magasin de grande surface. En effet un signal audio, tout comme une courbe de vente, sont des sinusoïdes ayant des fluctuations non aléatoires, et l'on peut s'imaginer que certaines règles leur sont communes.

L'équipe a choisi une compétition kaggle de Janvier 2018, et est arrivée seconde du concours en utilisant cette technique. Ce qui montre que leur solution fut une réussite sur le plan de la performance pure.

Le but de la compétition choisie était de prédire les ventes unitaires d'un certain nombre de produits, dans divers magasin. Cela basé sur un historique de ventes, et divers informations telles que la promotion d'un produit sur une période donnée.

Lien du PDF publié par les chercheurs :

https://arxiv.org/pdf/1803.04037.pdf