Projet Data Science

Ahmed Amine Majdoubi
Imad Al Moslli
Adnan Asadullah

April 2020

1 Introduction

L'entreprise qui nous emploie exploite plus de 3 000 magasins dans 7 pays européens. Actuellement, les directeurs de magasin sont chargés d'estimer leurs ventes quotidiennes jusqu'à six semaines à l'avance. Les ventes des magasins sont influencées par de nombreux facteurs, notamment les promotions, la concurrence, les vacances scolaires, la saisonnalité et la localité. Avec des milliers de gestionnaires individuels prédisant les ventes en fonction de leur situation particulière, la précision des résultats peut être très variable.

Analyse Statistique des données et validation des données

	Store	StoreType	Assortment	CompetitionDistance	CompetitionOpenSinceMonth	CompetitionOpenSinceYear	Promo2	Promo2SinceWeek	Promo2SinceYear	PromoInterval
0	1	c	a	1270.0	9.0	2008.0	0	NaN	NaN	NaN
1	2	a	a	570.0	11.0	2007.0	1	13.0	2010.0	Jan,Apr,Jul,Oct
2	3	a	a	14130.0	12.0	2006.0	1	14.0	2011.0	Jan,Apr,Jul,Oct
3	4	c	c	620.0	9.0	2009.0	0	NaN	NaN	NaN
4	5	a	a	29910.0	4.0	2015.0	0	NaN	NaN	NaN

FIGURE 1 – Store data

	Store	DayOfWeek	Date	Sales	Customers	Open	Promo	StateHoliday	SchoolHoliday
0	1	5	2015-07-31	5263	555	1	1	0.0	1.0
1	2	5	2015-07-31	6064	625	1	1	0.0	1.0
2	3	5	2015-07-31	8314	821	1	1	0.0	1.0
3	4	5	2015-07-31	13995	1498	1	1	0.0	1.0
4	5	5	2015-07-31	4822	559	1	1	0.0	1.0

FIGURE 2 – Training data

Nous avons plusieurs indcateurs concernant les 3 000 magasins que nous regroupons dans un dataframe avec les variables suivantes :

— StoreID
— StoreType
— Assortment
— CompetitionDistance
— CompetitionOpenSinceMonth
— CompetitionOpenSinceYear
— Promo2

— Promo2SinceWeek
— Promo2SinceYear
— PromoInterval
— DayOfWeek
— Date
— Sales (per day)
— Customers
— Open
— Promo
— StateHoliday
— SchoolHoliday

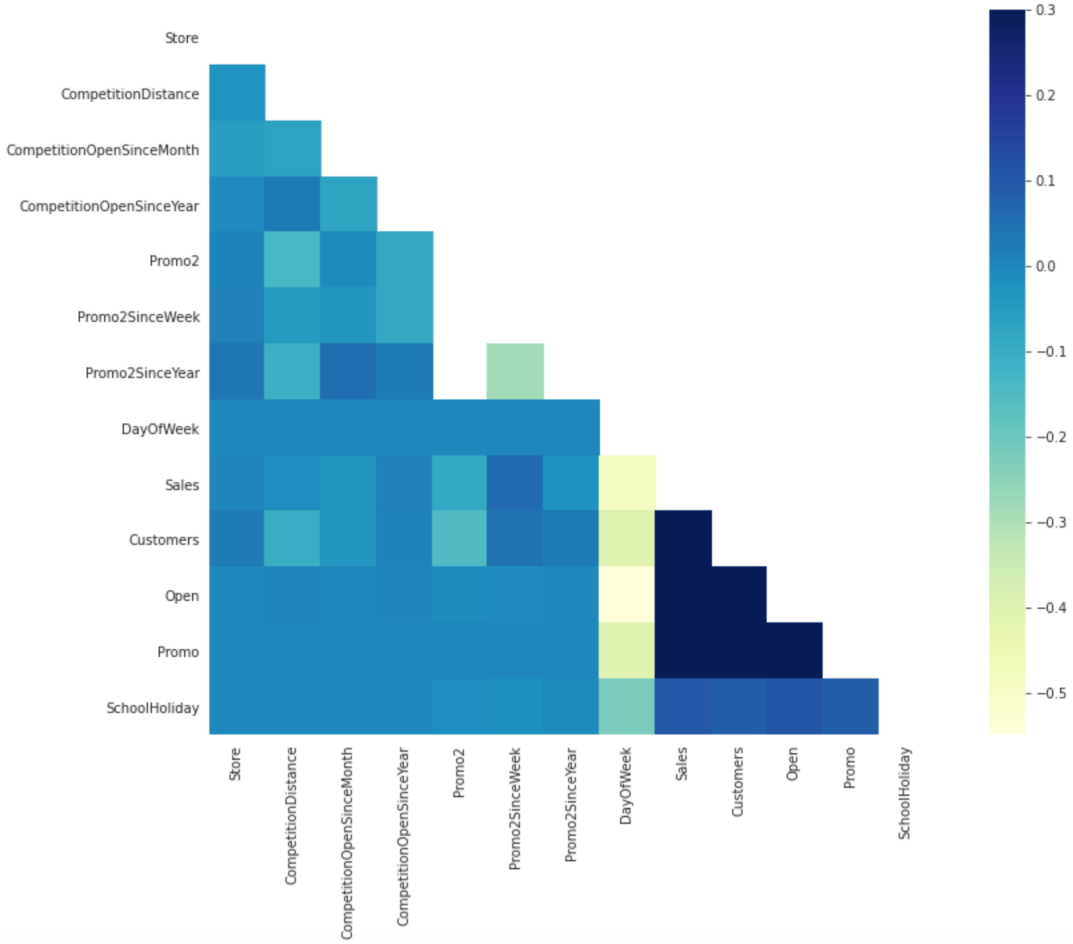

FIGURE 3 – Matrice de Correlation

A partir de la matrice de correlation, nous pouvons voir que la variable cible (Sales) est fortement correlée aux variables Customers, Open, Promo et DayOfWeek. En général, les autre variables ne sont pas très corrélées, ce qui est plutôt une bonne nouvelle car elles peuvent potentiellement toutes apporter une information différente, peut être utiles pour estimer les ventes.

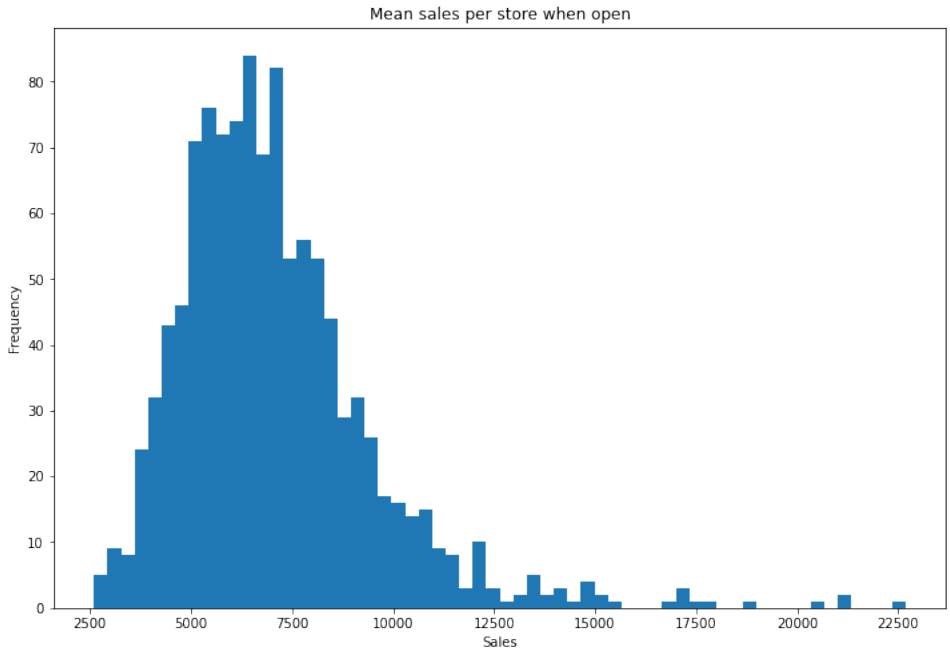

FIGURE 4 – Mean sale per day when store are open

Comme on peut voir, la majorité des magasins ont des ventes quotidiennes comprises entre 3500 et 9000. De plus en moyenne les ventes est de 6100, le minimum étant de 0 et le maximum de 41 550.

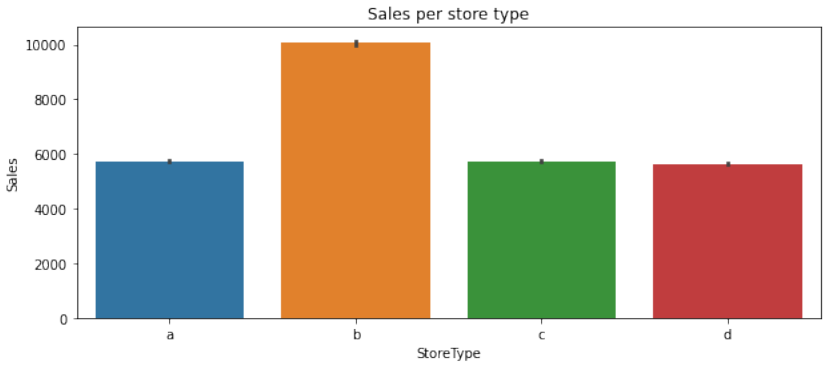

FIGURE 5 – Vente par type de magasin

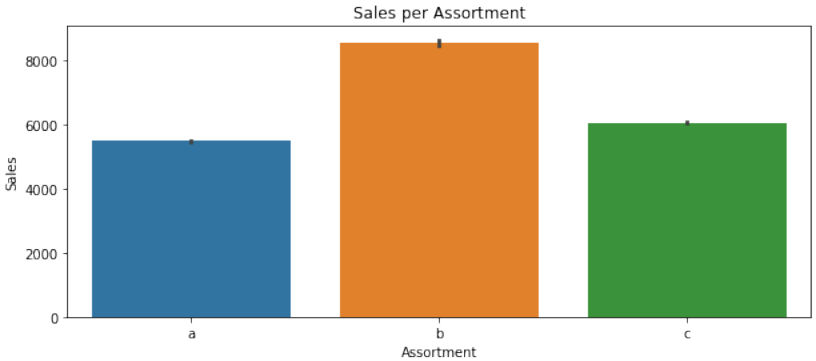

FIGURE 6 – Vente par Assortissement

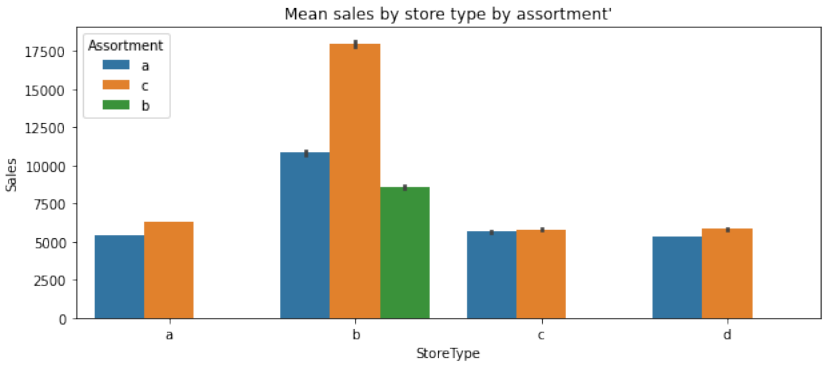

FIGURE 7 – Vente par Assortissement

En moyenne, les magasins de types A, C et D génèrent les mêmes revenus tandis que ceux de type B génèrent environs deux fois plus de revenus que les 3 autres. Cela peut être dû au fait que les magasins de types B sont les seuls types à vendre d'autres types d'assortiments de produits tandis que les autres ne vendent que des produits de type A et C. Enfin, nous pouvons voir que le type B est celui qui, globalement, génère le plus de revenus.

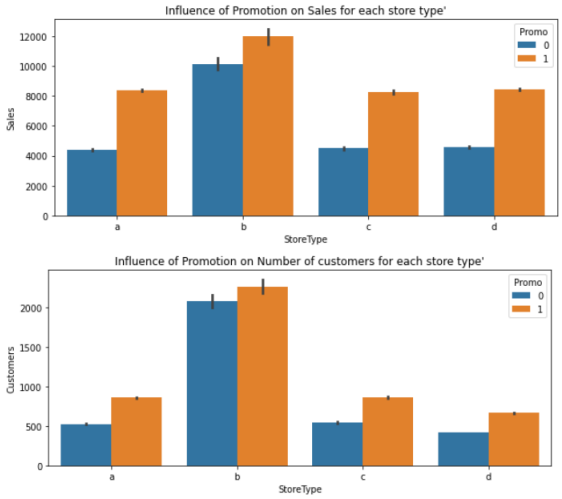

FIGURE 8 – Ventes et clients par promotion selon leurs types

Nous pouvon constater que lors de promotions, les ventes et le nombre de clients ont tendance à augmenter quelque soit le type de magasin. Cependant on remarquera que l'augmentation est plus faible dans le type B que dans les autres types.

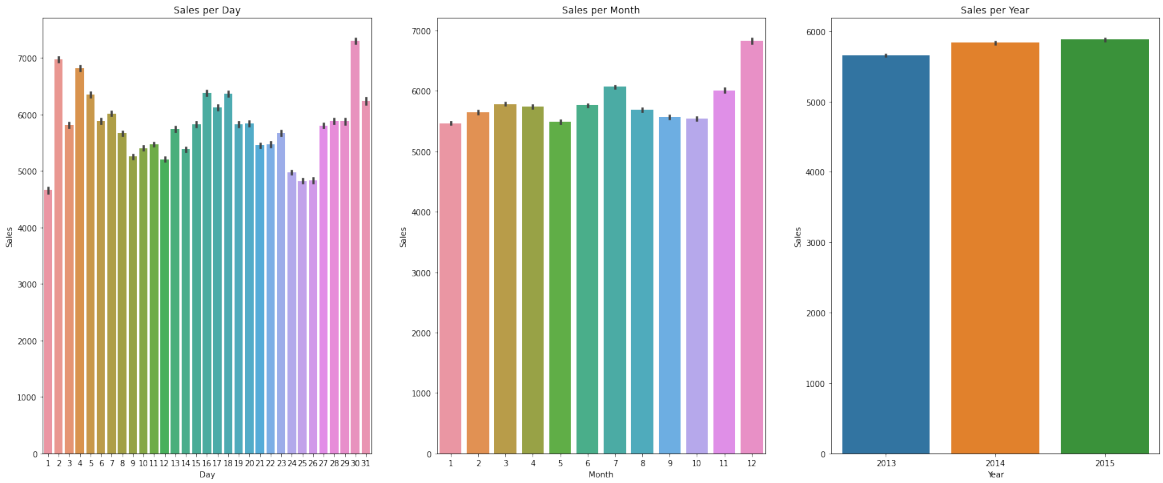

FIGURE 9 – Vente selon la periode

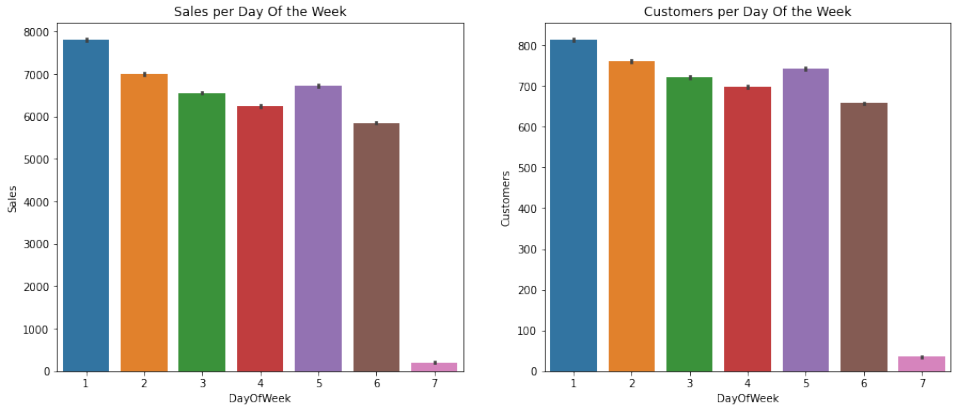

FIGURE 10 – Ventes selon le jour de la semaine

Le jour de la semaine a une importance au niveau des ventes, le 1er jour correspond au jour avec les meilleures ventes puis les ventes diminuent jusqu'au jour 5 où elles augmentent. Le jour 7 correspond au jour où les magasins sont généralement fermés. Nous pouvons voir le même profil durant les jours du mois où le motif se répète. Enfin au niveau des mois, il y a une grande augmentation du nombre de vente au mois de décembre.

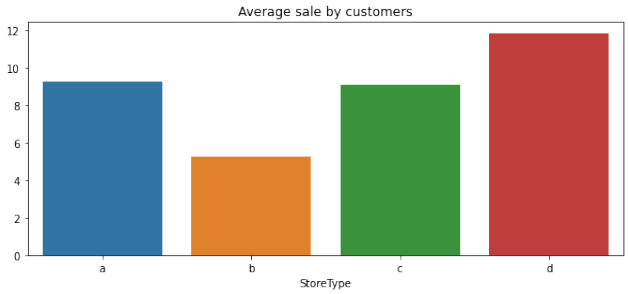

FIGURE 11 – Vente par client

Comme on a pu le voir précédemment, les magasins de types B réalise le plus de profit or nous pouvons voir sur ce graphique que les magasins de type B font moins de ventes par client. Ainsi, cela signifie qu'ils attirent beaucoup plus de client que les autres types, ce qui leur permet de générer le plus de ventes au global.

Au contraire, les magasins de types D génèrent plus de vente par client mais attirent moins de personnes, ce qui leur procure moins de revenues. Les deux types n'attirent pas les mêmes profils de client. Enfin les types A et C se ressemblent concernant les ventes par clients, entre les deux autres.

Création du jeu de données

Nous avons commencé par fusionner les tables "train" qui contenait des informations sur les magasins (nombre de clients, ventes, etc.) avec la table "store" qui contenait des informations sur la concurrence et les promotions.

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1017209 entries, 0 to 1017208
Data columns (total 9 columns):
Store            1017209 non-null int64
DayOfWeek        1017209 non-null int64
Date             1017209 non-null object
Sales            1017209 non-null int64
Customers        1017209 non-null int64
Open             1017209 non-null int64
Promo            1017209 non-null int64
StateHoliday     1017209 non-null object
SchoolHoliday    1017209 non-null int64
dtypes: int64(7), object(2)
memory usage: 69.8+ MB
```

FIGURE 12 – Colonnes dans le dataset train.csv

```
store.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1115 entries, 0 to 1114
Data columns (total 10 columns):
Store                        1115 non-null int64
StoreType                    1115 non-null object
Assortment                   1115 non-null object
CompetitionDistance          1112 non-null float64
CompetitionOpenSinceMonth    761 non-null float64
CompetitionOpenSinceYear     761 non-null float64
Promo2                       1115 non-null int64
Promo2SinceWeek              571 non-null float64
Promo2SinceYear              571 non-null float64
PromoInterval                571 non-null object
dtypes: float64(5), int64(2), object(3)
memory usage: 87.2+ KB
```

FIGURE 13 – Colonnes dans le dataset store.csv

2 Nettoyage des données

Pour le nettoyage des données, nous avons commencé par traiter la date. Celle-ci était de type texte, nous l'avons donc converti en date et extrait l'année, le mois et le jour du mois qui peuvent bien sûr influer sur les ventes, tandis qu'une date brut n'apporterais pas autant d'information et ne pourrait de toute façon pas être traitée par notre algorithme d'apprentissage..

	DayOfMonth	Month	Year
0	31	7	2015
1	31	7	2015
2	31	7	2015
3	31	7	2015
4	31	7	2015

FIGURE 14 – Nouvelles colonnes DayOfMonth, Month et Year

Nous avons également remplacé les valeurs manquantes dans la colonne CompetitionDistance par la moyenne, ce que l'on a jugé suffisant étant donnée le faible ratio de valeurs manquantes (0,002597).

```
Store                        0.000000
DayOfWeek                    0.000000
Date                         0.000000
Sales                        0.000000
Customers                    0.000000
Open                         0.000000
Promo                        0.000000
StateHoliday                 0.000000
SchoolHoliday                0.000000
StoreType                    0.000000
Assortment                   0.000000
CompetitionDistance          0.002597
CompetitionOpenSinceMonth    0.317878
CompetitionOpenSinceYear     0.317878
Promo2                       0.000000
Promo2SinceWeek              0.499436
Promo2SinceYear              0.499436
PromoInterval                0.499436
dtype: float64
```

FIGURE 15 – Pourcentages de valeurs manquantes des différentes colonnes

Pour les autre colonnes contenant des valeurs manquantes, nous les traitons dans la section suivante.

3 Création de nouvelles variables

3.1 Concurrence et promotion

Comme on peut le voir sur la figure précédente, les colonnes "CompetitionOpenSinceMonth", "CompetitionOpenSinceYear", "Promo2SinceWeek", "Promo2SinceYear" contiennent également des valeurs manquantes. Pour celles-ci, nous avons tous simplement fait le choix de les remplacer par la durée de la concurrence (en mois) et de promotion (en semaine) respectivement car une durée est plus explicative qu'une date dans ce cas. Nous avons donc ajouté les colonnes "MonthsSinceCompetition" et "WeeksSincePromo2" ;

	Year	Month	CompetitionOpenSinceYear	CompetitionOpenSinceMonth	MonthsSinceCompetition
0	2015	7	2008.0	9.0	82.0
1	2015	7	2007.0	11.0	92.0
2	2015	7	2006.0	12.0	103.0
3	2015	7	2009.0	9.0	70.0
4	2015	7	2015.0	4.0	3.0

FIGURE 16 – Aperçu de la nouvelle colonne "MonthsSinceCompetition"

	Year	Month	Promo2SinceYear	Promo2SinceWeek	WeeksSincePromo2
0	2015	7	NaN	NaN	0.0
1	2015	7	2010.0	13.0	234.0
2	2015	7	2011.0	14.0	185.0
3	2015	7	NaN	NaN	0.0
4	2015	7	NaN	NaN	0.0

FIGURE 17 – Aperçu de la nouvelle colonne "WeeksSincePromo2"

Lorsque les données étaients manquantes, nous avons mis une valeur nulle pour les nouvelles colonnes. En effet, nous avons pu vérifier que les valeurs dans les colonnes "Promo2SinceWeek" et "Promo2SinceYear" étaient manquantes tout simplement lorsque le magasin ne faisait pas de Promo2 (Promo2 = 0).

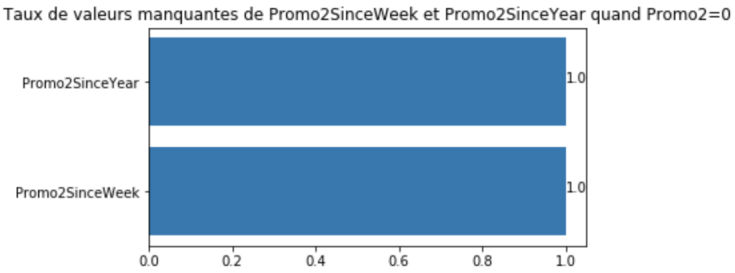

FIGURE 18 – Valeurs manquantes quand Promo2=0

```
Promo2SinceWeek    0.0
Promo2SinceYear    0.0
dtype: float64
```

FIGURE 19 – Valeurs manquantes quand Promo1=0

Une fois ces nouvelles colonnes ajoutées, nous n'avons pas besoin de conserver les colonnes initiales (complètement corrélées avec les nouvelles). Ainsi, nous n'avons plus de valeurs manquantes dans notre jeu de données.

```
]: train_df.isna().sum() / train_df.shape[0]

Store                      0.0
DayOfWeek                  0.0
Sales                      0.0
Customers                  0.0
Open                       0.0
Promo                      0.0
StateHoliday               0.0
SchoolHoliday              0.0
StoreType                  0.0
Assortment                 0.0
CompetitionDistance        0.0
Promo2                     0.0
DayOfMonth                 0.0
Month                      0.0
Year                       0.0
MonthsSinceCompetition     0.0
WeeksSincePromo2           0.0
dtype: float64
```

FIGURE 20 – Taux de valeurs manquantes après traitement

Nous avons également gérer les données catégorielles qui ne sont souvent pas supportées par les algorithmes d'apprentissage mais qui contiennent bien de l'information utiles pour la modélisation. Nous avons chosi de représenter les variables "StateHoliday", "StoreType" et "Assortment" sous forme de one-hot encoding :

StateHoliday_a	StateHoliday_b	StateHoliday_c	StoreType_a	StoreType_b	StoreType_c	StoreType_d	Assortment_a	Assortment_b	Assortment_c
0	0	0	0	0	1	0	1	0	0
0	0	0	1	0	0	0	1	0	0
0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	1	0	0

FIGURE 21 – Représentation One-hot encoding des variables "StateHoliday", "StoreType" et "Assortment"

3.2 Ventes et nombre de clients

Etant donné que nous ne disposons pas du nombre de client dans le test set, nous avons choisi de l'estimer dans un premier temps afin de l'intégrer comme variable du modèle de prédiction des ventes. En effet, puisque le modèle va apprendre sur des données disposant du nombre de clients pour estimer les ventes, il faut pouvoir ajouter le nombre de clients (estimé) durant la phase de test. Nous aurons donc deux modèles d'apprentissage : un pour estimer le nombre de clients et l'autre pour estimer les ventes.

Nous avons également pensé qu'il était pertinent d'ajouter les ventes ainsi que le nombre de clients à J-1 car comme on a pu le voir, leur évolution est périodique. Nous nous sommes limités à un jour avant car le temps de calcul était très long (près de 3h pour ajouter les ventes et le nombre de clients à J-1 sur tout le jeu de données). Ajouter cette variable était très utile pour prédire les ventes mais étant donné qu'on n'aura pas cette information lorsqu'on voudra estimer les ventes 6 semaines à l'avance, nous ne l'avons finalement pas conservée.

Toutefois, certains magasins performant mieux que d'autres, il semble tout de même utile d'ajouter les ventes et le nombre de clients moyens par magasin.

4 Sélection des variables les plus pertinentes

Pour la sélection des variables du modèle, nous avons adopté une approche "Backward" en entrainant le modèle sur le jeu de données complet avant de retirer les variables dont le score d'importance (F-score) était faible. Nous devons réaliser deux sélections de variable distinctes : une pour le modèle qui estime le nombre de clients et l'autre pour celui qui estime les ventes.

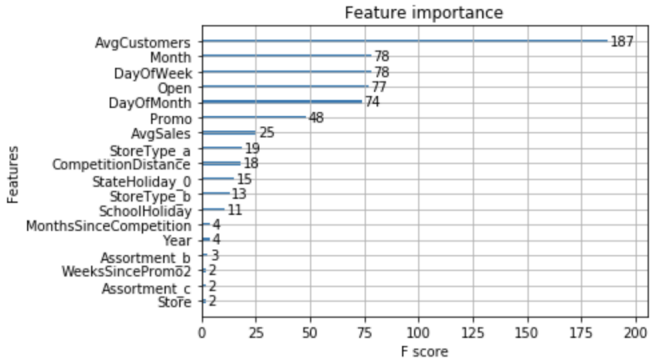

FIGURE 22 – Importance des variables dans le modèle d'estimation du nombre de clients à partir de leur F-score (avant sélection)

On peut voir que pour estimer le nombre de clients, les variables que nous avons créées sont très utiles, notamment la moyenne des clients pas magasin. Par contre, cela draine toute l'explicati-

vité d'une variable comme "Store" justement, avec laquelle la corrélation est très forte bien entendu.

Le modèle que nous avons choisi (voir partie suivante) nous permet d'avoir l'importance de chaque variable explicative pour estimer la variable à expliquer. Il nous suffit donc seulement de décider combien de variables nous souhaitons conserver puisque nous avons leur ordre d'importance. Pour cela, nous monitorons la performance du modèle en retirant petit à petit les variables les moins importantes.

On peut voir sur la figure suivantes que le coefficient de détermination (voir choix de la métrique d'évaluation) du modèle est proche de 0.90 lorsqu'on prend tous les variables. Ensuite, il reste assez stable lorsqu'on retire les variables les moins importantes. Cependant, il y a une chute brutale de la performance du modèle lorsque l'on passe de 6 à 5 variables explicatives, c'est-à-dire lorsque l'on retire le jour du mois de la liste des variables du modèle. Pour garder de bonnes performances avec un nombre de variables raisonnable, nous décidons de conserver les 10 variables les plus explicatives.

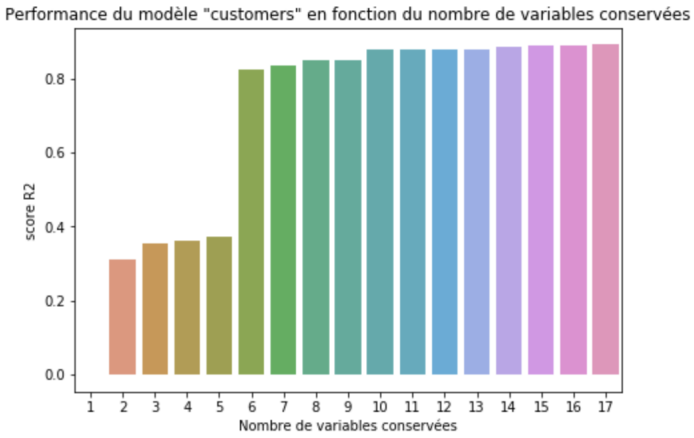

FIGURE 23 – Variation de la performance du modèle "customers" en fonction du nombre de variables conservées

Pour le modèle d'estimation des ventes, on peut voir que la variable "customers" que l'on a estimé et la moyenne des ventes que l'on a ajouté sont de loin les plus utiles pour l'estimation des ventes. Ensuite, on a d'autre variables qui sont assez significatives (Promo, date, ...) et d'autres qui ne le sont pas vraiment.

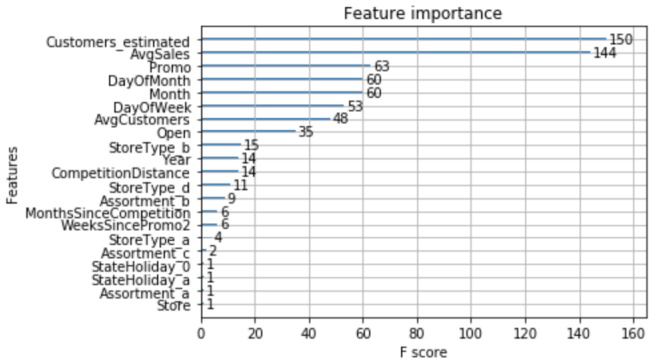

FIGURE 24 – Importance des variables dans le modèle d'estimation des ventes à partir de leur F-score (avant sélection)

Encore une fois, nous allons monitorer la performance du modèle pour déterminer le nombre de variables à conserver. On peut voir sur la figure suivante que pour le modèle d'estimation des

ventes, on peut retirer beaucoup de variables sans perdre en termes de performance. En effet, le coefficient de détermination reste égale à 0.90, que l'on ait 20 variables ou 7 variables.

Il commence à descendre ensuite si l'on réduit davantage le nombre de variables. On décide de conserver 8 variables explicatives (pour garder la variable "Open" qui est assez significative).

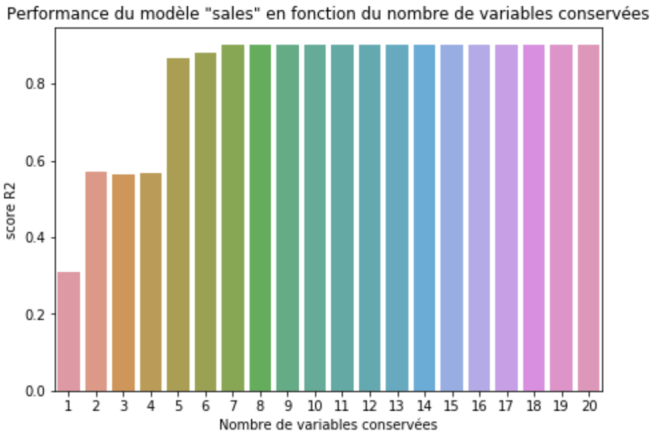

FIGURE 25 – Variation de la performance du modèle "sales" en fonction du nombre de variables conservées

Création du modèle

5 Choix de l'algorithme d'apprentissage

Le problème que nous cherchons à résoudre passe par de l'apprentissage supervisé. En effet, nous utilisons des variables explicatives x_i pour estimer une variable cible y_i.

Nous avons choisi XGBoost (eXtreme Gradient Boosting) pour entrainer le modèle. C'est une technique d'assemblage (qui combine plusieurs modèles) qui traite très bien les problèmes de régression comme le nôtre. En effet, le fait de combiner plusieurs modèles à partir d'échantillons Bootstrap augmente fortement les performances.

De plus, XGBoost nous a également permis de sélectionner très simplement les variables utiles pour le modèle (grâce à une fonction intégrée).

6 Choix de la stratégie de validation

Pour valider le modèle, nous avons réaliser un découpage temporel du jeu de données. En effet, le problème que nous cherchons à résoudre consiste à estimer les ventes jusqu'à 6 semaines à l'avance. Nous avons donc reproduit ce schéma pour l'évaluation du modèle en séparent les 6 dernières semaines (utilisées pour la phase de test) du reste des données (utilisées pour la phase d'entrainement) dans notre jeu de données initial.

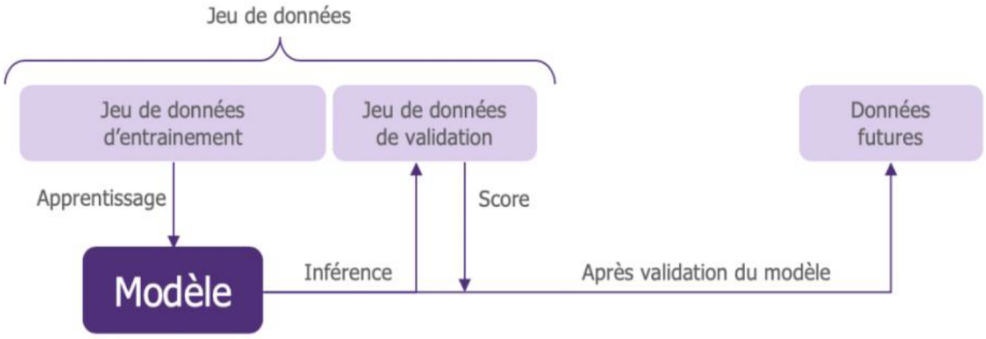

FIGURE 26 – Schéma de validation du modèle

7 Choix de la métrique d'évaluation

Pour la métrique d'évaluation, nous avons décider de nous baser sur le coefficient de détermination, défini comme : $R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y}_i)^2}$ où y_i est la valeur de la ième observation du jeu de données de validation, $\overline{y}$ est la moyenne des valeurs du jeu de données de validation et $\hat{y}_i$ est la valeur prédite pour la ième observation.

Cette métrique présente plusieurs avantages. Tout d'abord, elle permet d'évaluer la part de variation de la variables cible qui est expliquée par les variables explicatives. C'est donc une mesure d'adéquation du modèle adaptée pour sélectionner des variables.

Ensuite, elle a l'avantage d'être simple à interpréter et d'être naturellement normalisée, c'est-à-dire qu'elle ne présentera pas de biais provenant des différences d'ordre de grandeur (des ventes et du nombre de clients) qui peuvent exister sur différentes périodes, contraiment à d'autres métriques comme MSE ou RMSE.

FIGURE 27 – Résultats de la validation du modèle

On peut voir qu'au final, notre modèle permet d'expliquer 94% de la variation du nombre de clients et près de 90% de celle des ventes. Notre objectif n'a pas été d'améliorer les performances au maximum mais si on voulait aller plus loin, on pourrait par exemple optimiser le modèle pour choisir les meilleurs paramètres et avoir la meilleure performance possible.

Présentation des résultats

Après avoir créé notre modèle de prédiction, nous allons passer à la phase de présentation de résultats. Une bonne présentation des résultats est indispensable pour la réussite d'un projet de

conseil en data science. On ne peut pas présenter la structure du modèle et le code directement au client puisqu'on suppose toujours que ce dernier n'a pas des connaissances approfondies en data science. C'est pourquoi il est indispensable de présenter les résultats du projet sous forme d'interfaces simplifiées, intuitives et compréhensibles par tout le monde.

Afin de créer ces interfaces, nous avons utilisé le logiciel de visualisation de données "TABLEAU". Nous nous sommes servis de ce dernier pour créer deux tableaux de bords interactifs.

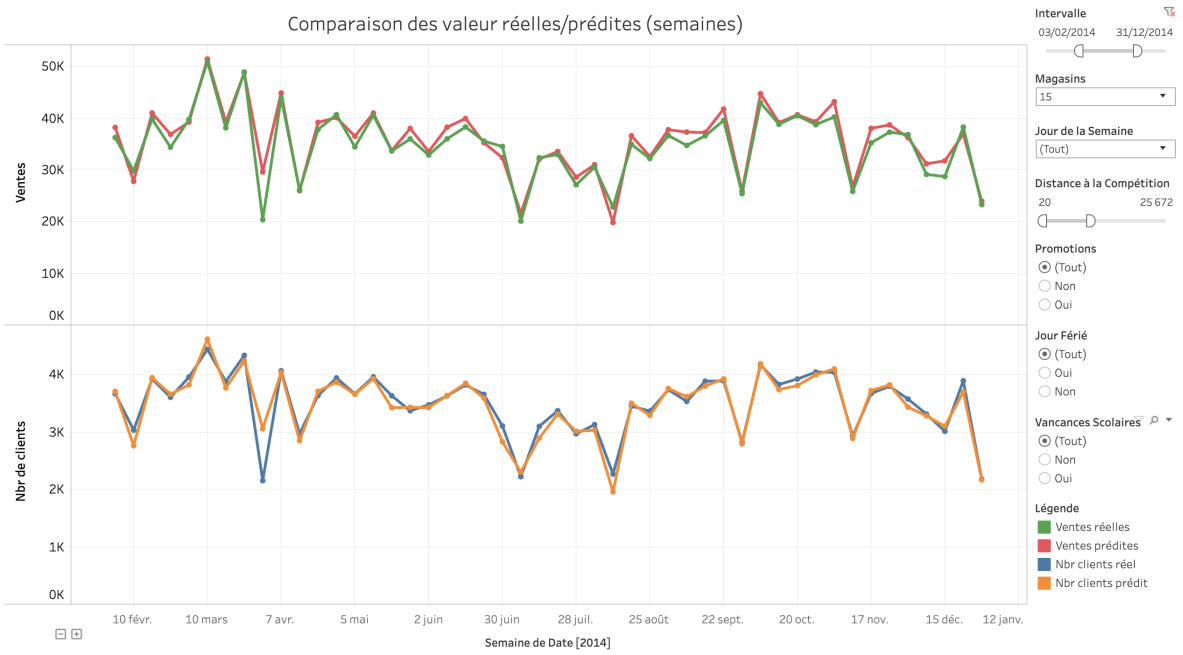

FIGURE 28 – Tableau de Bord 1 : Résultats du modèle

Le ***premier tableau de bord*** sert à comparer les données réelles avec les prédiction générée par notre modèle. les deux courbes en haut présentent les valeurs réelles et prédites des ventes du magasin, et les deux courbes en bas présentent les valeurs réelles et prédites du nombre de client qui ont fréquenté le magazin. L'intervalle de comparaison est en semaines, et peut être modifiée en haut à droite. l'interface donne aussi la possibilité de choisir le magazin à afficher (un ou plusieurs en même temps), et permet de filtrer sur les jours de la semaine, les jours avec promotions, les jours fériés et les vacances scolaires.

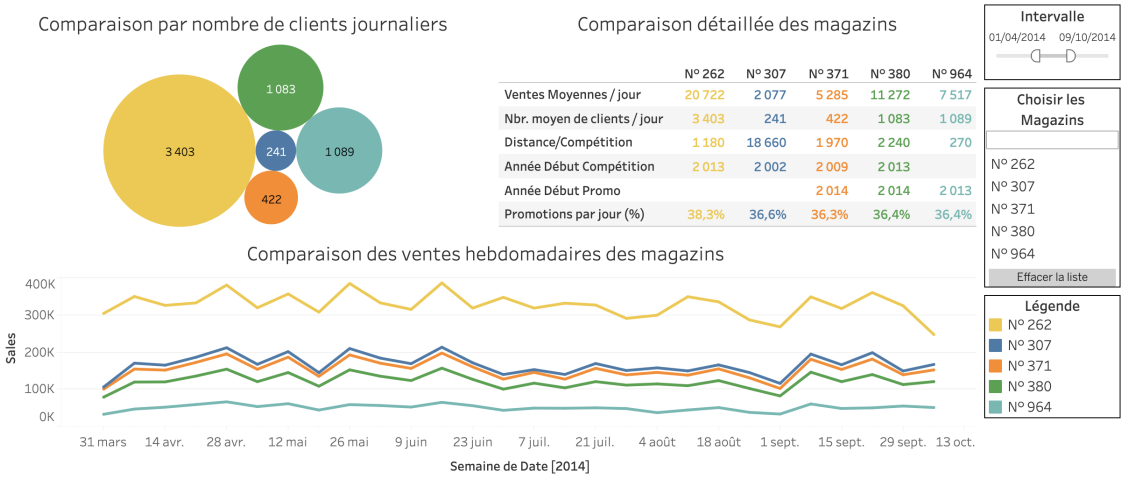

FIGURE 29 – Tableau de Bord 1 : Comparaison des magazins

Le ***deuxième tableau de bord*** sert à comparer les différents magasins de l'entreprise. On peut ajouter ou supprimer les magasins et définir l'intervalle du temps pour la comparaison. Les couleurs des trois graphiques affichés sont liées à la légende en bas à droite. Comme pour le premier tableau de bord. le graphique en dessous sert à comparer les ventes hebdomadaires dans une intervalle de temps définie, mais cette fois la comparaison est faite entre les magazins eux-memes. La taille des bulles dans le graphique en haut à gauche est définie en fonction du nombre moyen des clients journaliers, afin d'avoir une idée des tailles des magazins quand on compare leurs ventes hebdomadaires. le tableau en haut à droite affiche une description détaillée de chaque magasin afin de permettre à notre client de suivre leurs performances.

Industrialisation

8 Interprétabilité du modèle

Un des grands avantages de notre modèle est qu'il est interprétable. En effet, on a pu identifier les variables qui contribuent le plus à l'estimation et quantifier leur importance, ce qui nous a permis de ne conserver que les plus pertinentes.

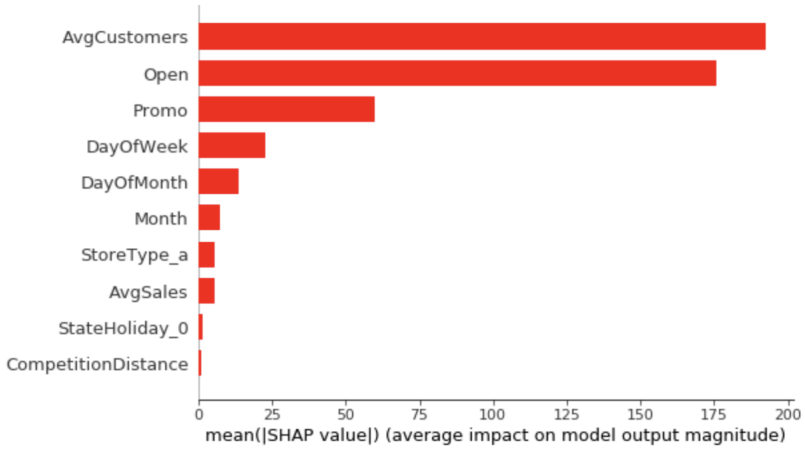

FIGURE 30 – Importance de chaque variable pour estimer le nombre de clients

On peut voir que globalement, les variables les plus influentes sont le nombre moyen de client du magasin, le fait qu'il soit ouvert ou non (évidemment), le fait qu'il y ait des promotions ou encore la période (jour de la semaine, du mois, de quel mois).

On peut voir ci-dessus comment certaines variables peuvent influencer l'estimation du nombre de clients. Dans cet exemple, le nombre de clients estimé est de 691. On voit que le fait que le magasin soit ouvert, qu'il y ait des promotions et que l'on soit lundi joue en faveur du magasin (rouge) tandis que le fait que le magasin ne soit pas de type a et que le magasin ait un nombre de client moyen de 468 impacte négativement (bleu) l'estimation.

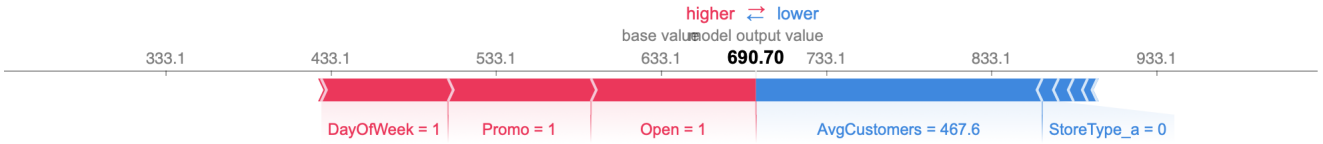

FIGURE 31 – Exemple d'influence des variables dans une estimation du nombre de client

On peut également voir l'influence individuelle de chaque variables selon les valeurs qu'elles prennent. Par exemple, on voit sur la figure suivante que lorsque la distance vis-à-vis de la concurrence est faible, cela impacte généralement négativement le nombre clients, étant donnée qu'ils peut être attirés par cette concuurence. Et à l'inverse, si la concurrence est loin, cela a tendance a amener plus de clients dans le magasin.

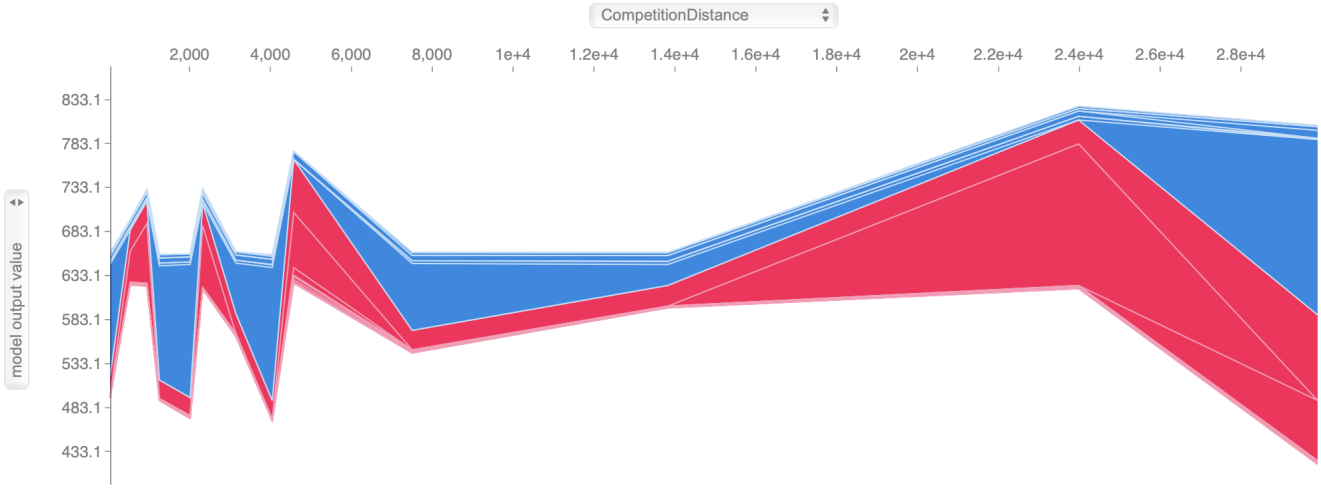

FIGURE 32 – Influence de la distance à la concurrence dans l'estimation du nombre de clients

Pour finir voici un résumé de l'influence des différentes variables dans les modèles d'estimation du nombre de clients :

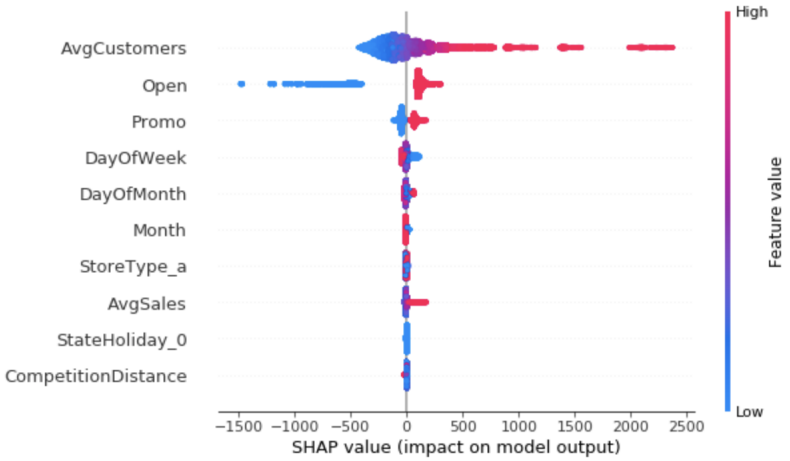

FIGURE 33 – Influence des différentes variables dans l'estimation du nombre de clients

Pour l'estimation des ventes, on peut voir que le nombre de clients estimé est très influent. Evidemment, plus le nombre de clients estimé est important, plus les ventes estimées le sont également. Il en est de même pour la moyenne des ventes. Ensuite, les autres variables influentes sont les mêmes que pour l'estimations du nombre de clients.

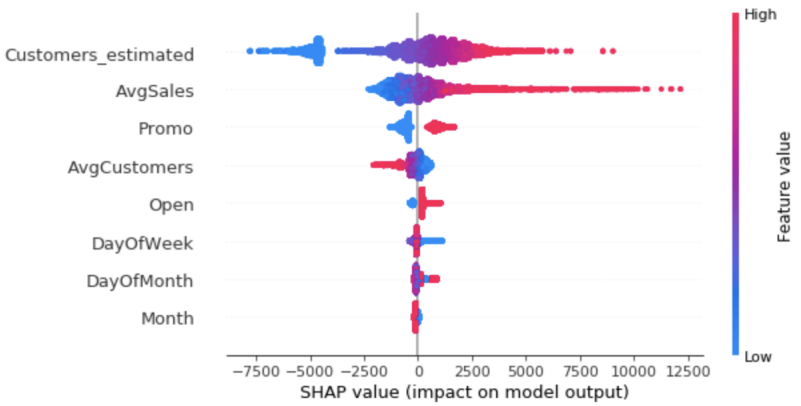

FIGURE 34 – Influence des différentes variables dans l'estimation des ventes

9 Mise en production et maintenance du modèle

Comme on peut le voir dans l'interface, les performances du modèle sont raisonablement bonnes pour le mettre en production. Toutefois, il est important de monitorer ses performances étant donné que la distribution des données peut changer d'une période à une autre. Dans ce contexte, la première interface est intéressante car elle permet de visualiser la différence entre les valeurs estimées par le modèle et les valeurs réelles.

Ainsi, il n'est pas nécessaire de réentrainer le modèle à chaque fois que l'on souhaite l'uiliser pour estimer le ventes. Cependant, notre modèle a été entrainé sur une période de 2 ans, il n'est donc pas étonnant qu'il puisse avoir une bonne performance sur les 6 semaines suivantes. Par contre, si on se rend compte que les performances se dégradent et que les estimations sont mauvaises, il faut le réentrainer pour qu'il s'ajuste à la nouvelle distribution des données.

10 Conclusion

Nous offrons une solution simple, intelligente et interprétable pour faciliter et harmoniser l'estimation des ventes des différents magasins. Ses performances sont bonnes et peuvent encore être améliorée davantage assez rapidement. Son utilisation est simple à travers l'interface que nous avons conçu pour l'accompagner.

L'objectif n'est pas de se reposer sur ces estimations comme si elles étaient infaillibles. On ne peut évidemment pas prévoir tous les évènements externes (comme un confinement par exemple) qui pourrraient survenir et affecter les ventes. Lorsque la situation change et que les ventes s'en retrouvent profondément affectées, il faut alimenter le modèle avec les nouvelles données pour qu'il s'adapte à la situation.

Pour finir, notre solution n'est pas seulement utile pour estimer les ventes. Elle donne une visibilité sur leurs évolutions et les méchanismes qui les régissent afin de comprendre quels sont les facteurs qui font certains magasins performent miuex que d'autres. Ainsi, il s'agit également d'un outil d'aide à la décision qui peut accompagner la politique de gestion et d'investissement de l'entreprise à court et moyen terme.