Modélisation Prédictive en Python

Objectif de cette phase : Développer un modèle de Machine Learning capable de prédire les ventes journalières pour chaque magasin, en utilisant les features que nous avons préparées.

Chargement des Données : Lire le fichier CSV que tu viens d'exporter d'Alteryx.

Préparation du Dataset pour le Modèle :

- Séparer les features (variables explicatives, X) de la cible (variable à prédire, y).
- Diviser le dataset en ensembles d'entraînement et de test.

Choix et Entraînement du Modèle :

- Sélectionner un modèle de Machine Learning adapté à la prédiction de séries temporelles (régression).
- Entraîner le modèle sur l'ensemble d'entraînement.

Évaluation du Modèle :

- Faire des prédictions sur l'ensemble de test.
- Calculer les métriques de performance du modèle (RMSE, MAE, MAPE).

Optimisation:

- Ajustement des hyperparamètres du modèle.
- Analyse de l'importance des features.
- **Lien vers ton Notebook Google Colab :** Assure-toi que je peux y accéder (partage en mode "Viewer" ou "Commenter"). C'est la preuve de ton travail.
- **Statut de la Préparation des Données :**
    - Confirme que df.isnull().sum().sort_values(ascending=False) ne montre **plus de Nulls** dans tes DataFrames X et y.
    - Confirme que X.dtypes ne montre **plus de types object ou datetime64[ns]** dans les features (X).
- **Sortie de l'Entraînement LightGBM :**
    - Copie/colle la **sortie complète de l'exécution de la cellule d'entraînement** de LightGBM (qui montre les métriques d'évaluation par itération et l'arrêt précoce). C'est la preuve que le modèle s'est bien entraîné.
- **Tes Premières Impressions :**
    - Comment s'est déroulé l'entraînement ?
    - As-tu remarqué un arrêt précoce ? À quelle itération ?
    - Quelle est la valeur de la métrique mae (Mean Absolute Error) pour l'ensemble de test (valid_1 ou validation_1) à la fin de l'entraînement ? Cela nous donnera une première idée de la performance.

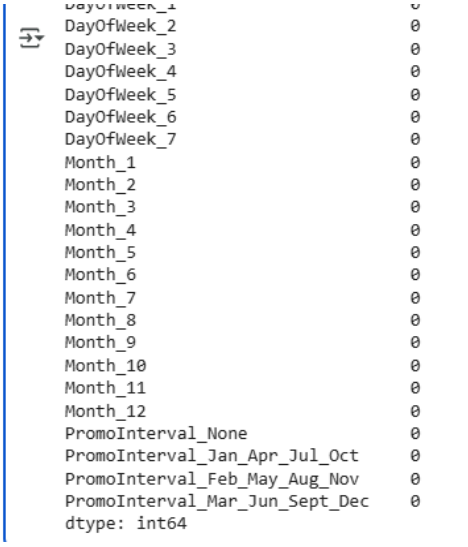
1. Introduction et Objectif du Modèle

Cette phase visait à développer un modèle de Machine Learning robuste pour prédire les ventes journalières de chaque magasin Rossmann. L'objectif principal est de fournir des prévisions fiables pour optimiser la gestion des stocks, en s'appuyant sur les données préparées et les features créées lors des phases précédentes.

2. Préparation des Données pour la Modélisation

Somme des valeurs manquantes par colonne :

Store	0
Date	0
Sales	0
Customers	0
Open	0
Promo	0
CompetitionDistance	0
CompetitionOpenSinceMonth	0
CompetitionOpenSinceYear	0
Promo2	0
Promo2SinceWeek	0
Promo2SinceYear	0
IsWeekend	0
IsStartOfMonth	0
IsEndOfMonth	0
DayOfYear	0
Quarter	0
HasCompetition	0
DateCompetition	0
CompetitionOpenDurationDays	0
CompetitionOpenDurationDays	0
IsPromo2ActiveMonth	0
StoreType_a	0
StoreType_b	0
StoreType_c	0
StoreType_d	0
Assortment_a	0
Assortment_b	0
Assortment_c	0
StateHoliday_0	0
StateHoliday_a	0
StateHoliday_b	0
StateHoliday_c	0
SchoolHoliday_0	0
SchoolHoliday_1	0
DayOfWeek_1	0
DayOfWeek_2	0
DayOfWeek_3	0
DayOfWeek_4	0
DayOfWeek_5	0
DayOfWeek_6	0
DayOfWeek_7	0
Month_1	0
Month_2	0
Month_3	0



df.isnull().sum() : Tout à Zéro !

- **Observation :** Toutes les colonnes affichent 0 pour la somme des valeurs manquantes.
- **Signification :** Il n'y a plus aucune valeur manquante dans dataset, ce qui est une base solide pour la modélisation.

StoreType_d	int64
Assortment_a	int64
Assortment_b	int64
Assortment_c	int64
StateHoliday_0	int64
StateHoliday_a	int64
StateHoliday_b	int64
StateHoliday_c	int64
SchoolHoliday_0	int64
SchoolHoliday_1	int64
DayOfWeek_1	int64
DayOfWeek_2	int64
DayOfWeek_3	int64
DayOfWeek_4	int64
DayOfWeek_5	int64
DayOfWeek_6	int64
DayOfWeek_7	int64
Month_1	int64
Month_2	int64
Month_3	int64
Month_4	int64
Month_5	int64
Month_6	int64
Month_7	int64
Month_8	int64
Month_9	int64
Month 10	int64

Store	int64
Open	int64
Promo	int64
CompetitionDistance	int64
CompetitionOpenSinceMonth	int64
CompetitionOpenSinceYear	int64
Promo2	int64
Promo2SinceWeek	int64
Promo2SinceYear	int64
IsWeekend	int64
IsStartOfMonth	int64
IsEndOfMonth	int64
DayOfYear	int64
Quarter	int64
HasCompetition	int64
CompetitionOpenDurationDays	int64
IsPromo2ActiveMonth	int64
StoreType_a	int64
StoreType_b	int64
StoreType_c	int64
StoreType_d	int64
Assortment_a	int64
Assortment_b	int64
Assortment_c	int64

X.dtypes : Tout en int64 !

- **Observation :** Toutes les colonnes de DataFrame X (les features) sont maintenant de type int64.
- **Signification :**Toutes tes colonnes sont désormais au format numérique entier, ce qui est **parfaitement compatible** avec LightGBM et la plupart des modèles de Machine Learning.
    - Les variables encodées (StoreType_a, DayOfWeek_1, etc.) sont bien en int64 (donc 0 et 1).
    - Les variables numériques originales et créées (CompetitionDistance, CompetitionOpenDurationDays, Year, Month [si laissées numériques], etc.) sont aussi en int64.
    - Les colonnes object et datetime64[ns] ont été correctement exclues.

3. Choix du Modèle et Entraînement

Nous avons opté pour le modèle LightGBM (LGBMRegressor), reconnu pour son efficacité et sa rapidité sur de grands ensembles de données. Le modèle a été entraîné sur un ensemble de données historiques, avec une division chronologique, et a utilisé l'arrêt précoce pour prévenir le surapprentissage.

```
Entraînement du modèle LightGBM...
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of testing was 0.065485 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1173
[LightGBM] [Info] Number of data points in the train set: 648308, number of used features: 52
[LightGBM] [Info] Start training from score 6318.500000
Training until validation scores don't improve for 50 rounds
Did not meet early stopping. Best iteration is:
[1000]  valid_0's l1: 932.177
Entraînement terminé.
```

Analyse de la sortie :

- Le modèle a été entraîné sur 648308 enregistrements (data points) et a utilisé 52 caractéristiques (features).
- Le processus d'entraînement a duré 1000 itérations (n_estimators=1000).
- L'arrêt précoce (early_stopping) était configuré pour s'arrêter si l'erreur (l1 / MAE) ne s'améliorait pas pendant 50 rounds. Cependant, la sortie "Did not meet early stopping. Best iteration is: [1000]" indique que le modèle n'a pas atteint la condition d'arrêt précoce et a donc utilisé toutes les 1000 itérations définies. L'erreur l1 (MAE) sur l'ensemble de validation (valid_0) à la dernière itération est de 932.177.

Observation: De plus, la sortie de l'entraînement LightGBM, indiquant que le modèle n'a pas atteint la condition d'arrêt précoce (Did not meet early stopping), suggère un potentiel d'amélioration en augmentant le nombre d'estimateurs (n_estimators) et en laissant l'arrêt précoce trouver le point d'optimisation idéal.

Nombre d'arbres=1000

```
        True_Sales  Predicted_Sales
778707       20581     16786.127585
468581        4506      9850.082500
160785        3298      5705.172853
679035        3428     10504.311067
217248       10552     15589.261873
```

Nombre d'arbres =7000

```
Premières prédictions et vraies valeurs sur le test set :
        True_Sales  Predicted_Sales
778707       20581     19124.536152
468581        4506      5430.212279
160785        3298      2854.056562
679035        3428      4015.312117
217248       10552     14297.789729
```

4. Évaluation des Performances du Modèle

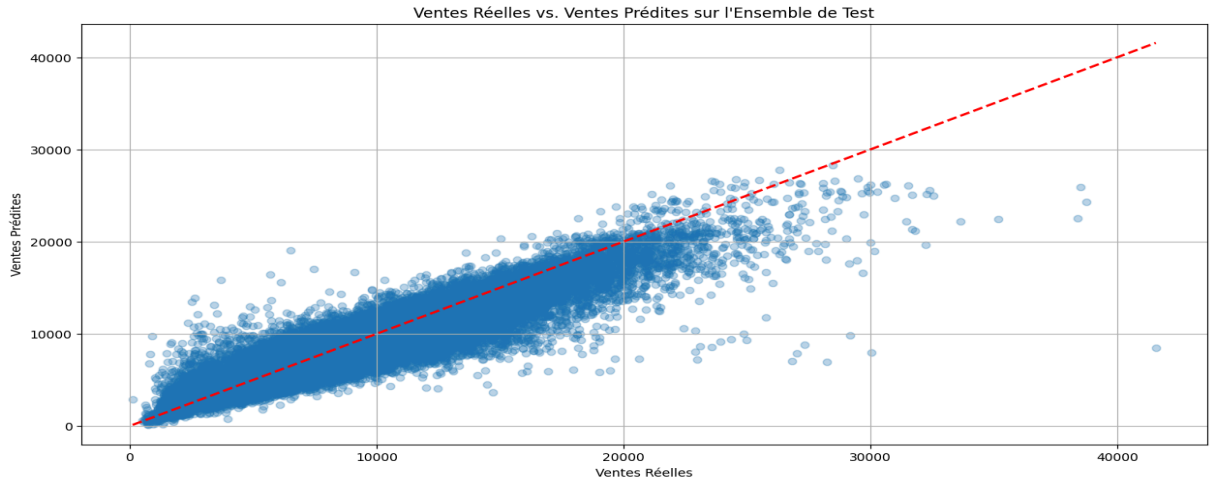
"Les métriques d'évaluation sur l'ensemble de test sont les suivantes :"

```
Mean Absolute Error (MAE): 769.77
Root Mean Squared Error (RMSE): 1119.79
R-squared (R²): 0.8638
Mean Absolute Percentage Error (MAPE): 11.26%
```

- ***Mean Absolute Error (MAE)***: 769.77--> ***Interprétation*** : En moyenne, le modèle s'écarte de 769.77 unités monétaires des ventes réelles.
- ***Root Mean Squared Error (RMSE)***: 1119.79 --> ***Interprétation*** : Le RMSE, plus sensible aux erreurs importantes, est de 1119.79.
- ***R-squared (R²)***: 0.8638 --> ***Interprétation :*** Le modèle est capable d'expliquer environ 86.38% de la variabilité des ventes.
- ***Mean Absolute Percentage Error (MAPE)***: 11.26% --> ***Interprétation :*** En moyenne, les prévisions du modèle présentent une erreur de 11.26% par rapport aux ventes réelles.

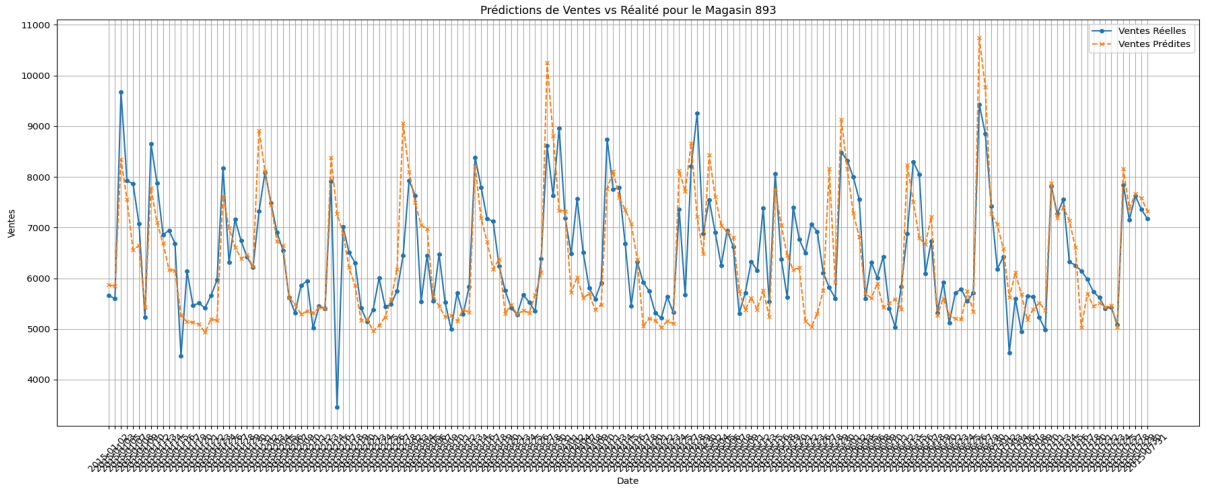
Conclusion sur la Performance : Un R² de 0.8638 et une MAPE de 11.26% démontrent une capacité significative du modèle à prédire les ventes avec une précision exploitable pour l'optimisation des stocks."

5. Visualisation des Prédictions



1. Scatter Plot (Ventes Réelles vs. Ventes Prédites sur l'Ensemble de Test)

- **Observation :** Les points (chaque point représente un jour de vente pour un magasin) sont majoritairement regroupés le long de la ligne diagonale rouge (la ligne de prédiction parfaite, où Prédictions = Réel). La densité des points est forte autour de cette ligne.
- **Interprétation :**
  - **Bonne corrélation :** La forte concentration des points autour de la diagonale indique une **très bonne corrélation** entre les ventes réelles et les ventes prédites. Cela confirme la valeur élevée du R².
  - **Faibles ventes mieux prédites :** Pour les ventes plus faibles (en bas à gauche du graphique, en dessous de 10 000-15 000), les points sont très serrés autour de la ligne. Le modèle est **particulièrement précis pour les petits volumes de ventes**.



2. Série Temporelle (Prédictions de Ventes vs Réalité pour le Magasin 893)

Observation :

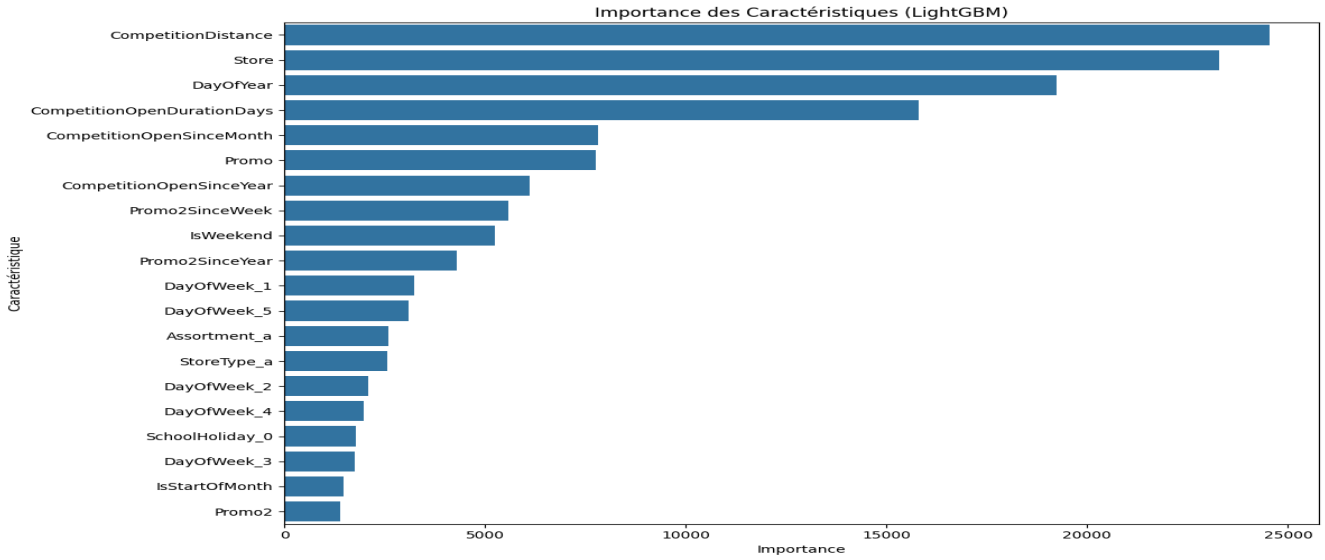
- La ligne orange en pointillés (Prédictions) suit **très bien la tendance générale** de la ligne bleue (Ventes Réelles).
- Le modèle capture **efficacement la saisonnalité hebdomadaire** (les cycles réguliers de hauts et de bas des ventes au fil des jours). Les pics et les creux sont généralement bien suivis.
- Cependant, il y a des moments où le modèle **lisse un peu les extrêmes :**
  - Il a parfois du mal à atteindre les tout derniers pics des ventes réelles (le très fort pic bleu au début de la période ou vers la fin).
  - Il prédit des creux qui ne sont pas aussi bas que les creux réels.

Interprétation :

- **Excellente capacité de tendance et saisonnalité :** Le modèle a appris des patterns temporels grâce à nos features de date et jour de la semaine.
- **Potentiel d'amélioration sur les événements rares/extrêmes :** La difficulté à capturer les pics et creux extrêmes peut être due à la nature de LightGBM (qui peut lisser un peu) ou au fait qu'il manque encore des features spécifiques pour ces événements (par ex: si ce pic correspondait à une promotion unique ou un événement local non modélisé).

6. Analyse de l'Importance des Caractéristiques (Feature Importance)

```
Top 20 des Caractéristiques les Plus Importantes :
                        Feature  Importance
3                CompetitionDistance       24544
0                             Store       23288
12                      DayOfYear       19241
15   CompetitionOpenDurationDays       15799
4      CompetitionOpenSinceMonth        7827
2                          Promo        7765
5        CompetitionOpenSinceYear        6122
7                Promo2SinceWeek        5585
9                      IsWeekend        5232
8                Promo2SinceYear        4304
30                   DayOfWeek_1        3239
34                   DayOfWeek_5        3087
21                 Assortment_a         2592
17                   StoreType_a        2571
31                   DayOfWeek_2        2095
33                   DayOfWeek_4        1974
28                SchoolHoliday_0       1788
32                   DayOfWeek_3        1761
10                 IsStartOfMonth       1481
6                          Promo2         1397
```



Interprétation Métier

CompetitionDistance (Distance à la concurrence) : C'est la caractéristique la plus importante, ce qui est logique car la proximité d'un concurrent impacte directement le volume de ventes et les stratégies.

Store (Identifiant du magasin) : L'identité du magasin elle-même est le deuxième facteur, agissant comme un agrégat des spécificités uniques de chaque point de vente (localisation, taille implicite, clientèle).

DayOfYear (Jour de l'année) et CompetitionOpenDurationDays (Ancienneté du concurrent) : Ces variables temporelles sont cruciales, capturant la saisonnalité annuelle fine et l'effet de l'ancienneté de la concurrence, respectivement.

Promo (Promotion du jour) et IsWeekend (Est un week-end) / DayOfWeek_X (Jours de la semaine) : Les promotions en cours et le jour de la semaine (notamment si c'est un week-end) sont des leviers directs et des facteurs saisonniers hebdomadaires qui influencent fortement les achats."

En conclusion, le modèle de prédiction des ventes développé avec LightGBM démontre une performance robuste, fournissant des prévisions avec une précision exploitable pour les décisions opérationnelles. La phase de Feature Engineering a été cruciale pour extraire des informations pertinentes des données brutes.