DEVOIR DE FIN DE MODULE

PROGRAMMATION DU MODELE ARIMA

SUJET

Prévision du traffic sur un site web

ETUDIANT : **NJEUNGA YOPA** Rémy

MD4 | 2022 - 2024

PROFESSEUR : **Hakim HORAIRY**

ECOLE HETIC

29 - 04 - 2023

SOMMAIRE

Contexte de l'étude

Nous sommes dans une entreprise qui possède un site internet et désire mieux comprendre le Traffic sur celui-ci, pour mieux servir sa clientèle. Les données des traffics ont été sauvegardées pendant une longue période.

Nous disposons donc d'un data Set avec le nombre de traffics sur une longue période divisé en 385 valeurs. Il est question de prédire le nombre de traffics sur les 30 prochaines périodes.

Pour cela il faudra modéliser un modèle de prédiction comme une modèle **d'auto régression**. A l'exemple d'ARIMA. Mais avant il faudra se rassurer que notre modèle est stationnaire.

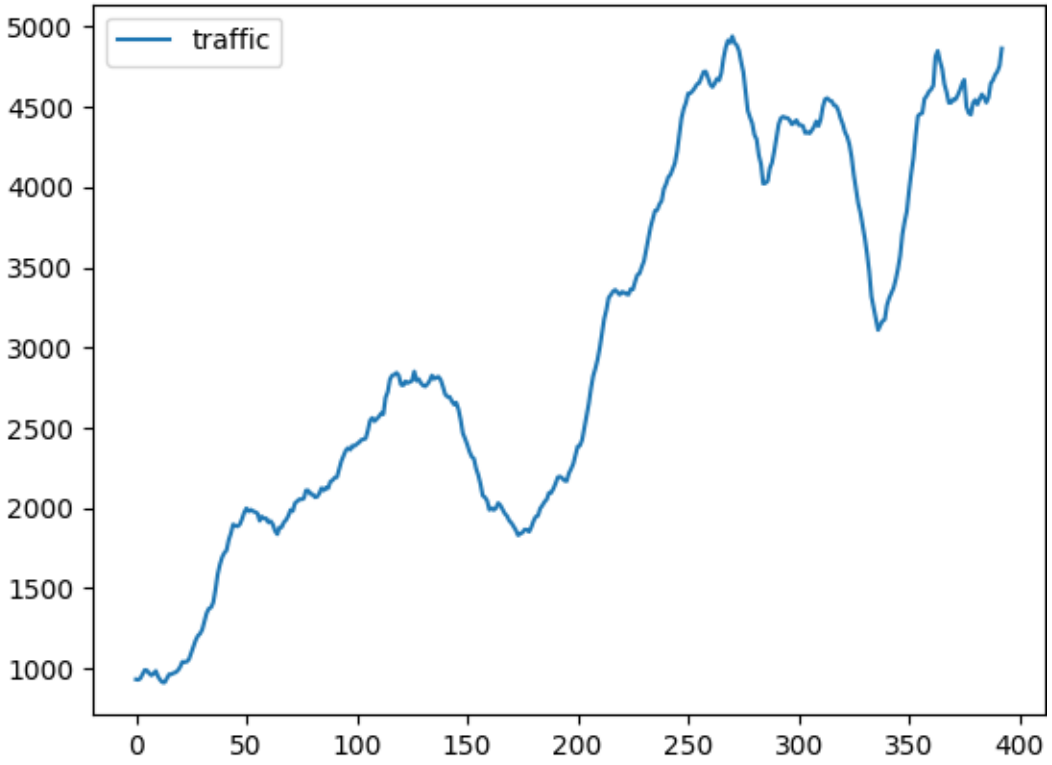
Notre travail a donc été préparé comme suit ; **premièrement** le traitement de la donnée, c'est-à-dire la visualision du jeu de donnée, le nettoyage, la normalisation, et le découpage en données d'entrainement et de test.

Deuxièmement vérifier la stationnarité de la série, et si elle est positive alors nous entrainons le modèle et réajustons le modèle avec les hyper-paramètres. **Enfin** nous utilisons notre modèle pour prédire sur les 30 prochaines périodes.

Traitement du DataSet

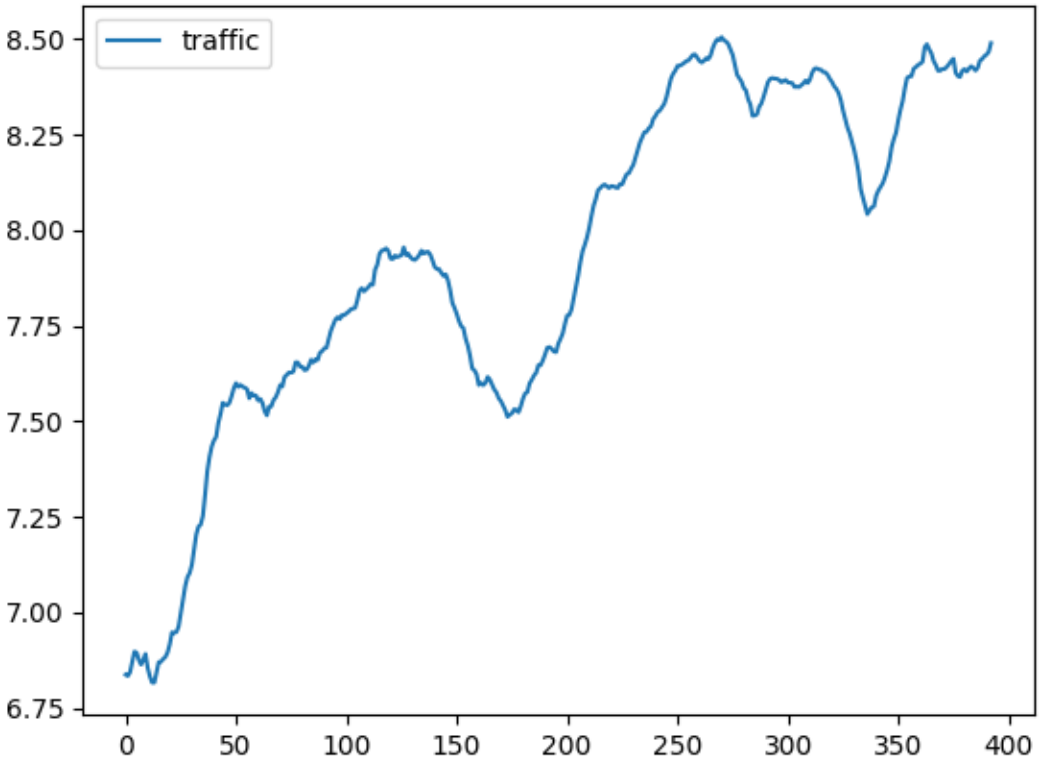
1. Visualisation du jeu de donnée

Le tracé de notre courbe nous montre une absence de régularité et beaucoup de variance. Déjà à première vue on ne constate pas une certaine stationnarité. Mais il y'a encore d'autres traitements afin de vérifier la temporalité de cette série.



2. Normalisation

La normalisation a pour but de réduire les écarts de grandeurs entre les données, elle aura aussi pour conséquence de lisser la courbe. Plus de détails dans le code source python rattaché à ce document.



3. Séparation du jeu de données

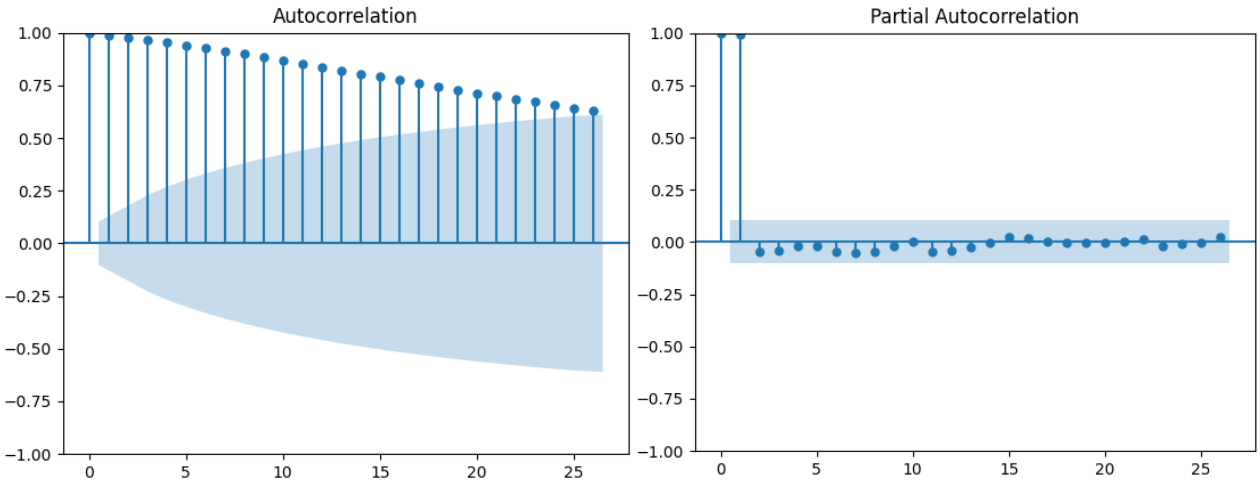
Notre jeu de données est découpé ainsi : **70%-**Jeu d'entrainement | Jeu de test-**30%**

4. Vérification de la stationnarité

Nous allons utiliser 2 fonctions : **ACF** et **PACF.**

Respectivement, l'**autocorrélation** mesure la corrélation entre une observation et une observation antérieure à un certain nombre de pas de temps, alors que **l'autocorrélation partielle** mesure la corrélation entre une observation et une observation antérieure après avoir tenu compte de toutes les observations intermédiaires.

Ces graphiques peuvent nous donner des indications sur le comportement de la série temporelle et nous aider **à sélectionner les paramètres du modèle ARIMA** approprié.



Dans notre cas:

L'autocorrélation diminue lentement et reste significative pour plusieurs retards, cela peut indiquer une tendance dans la **série temporelle qui nécessite une différenciation supplémentaire**.

L'autocorrélation partielle a une forte corrélation au premier délai et une corrélation faible pour les retards suivants, cela peut indiquer que nous avons besoin d'un **modèle AR(p), où p est le nombre de retards** avec une corrélation significative.

Il est important de noter que la visualisation des graphiques d'autocorrélation et d'autocorrélation partielle est une étape initiale importante dans le processus de modélisation et que d'autres tests statistiques et validations sont nécessaires pour confirmer les résultats et sélectionner le meilleur modèle.

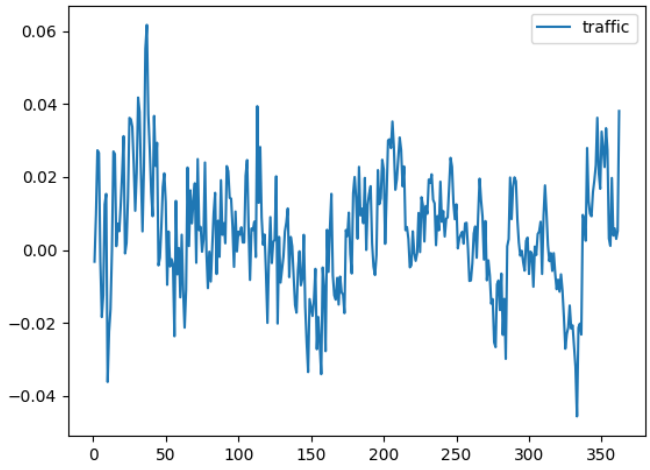
Le test de Dickey-Fuller sera utilisé pour vérifier si notre série temporelle est stationnaire ou non. Si **la p-valeur est inférieure à un certain seuil (généralement 0,05),** cela indique que la série temporelle est stationnaire avec une certaine confiance. Si la p-valeur est supérieure au seuil, cela indique que la série temporelle n'est pas stationnaire et nécessite peut-être une différenciation supplémentaire pour devenir stationnaire.

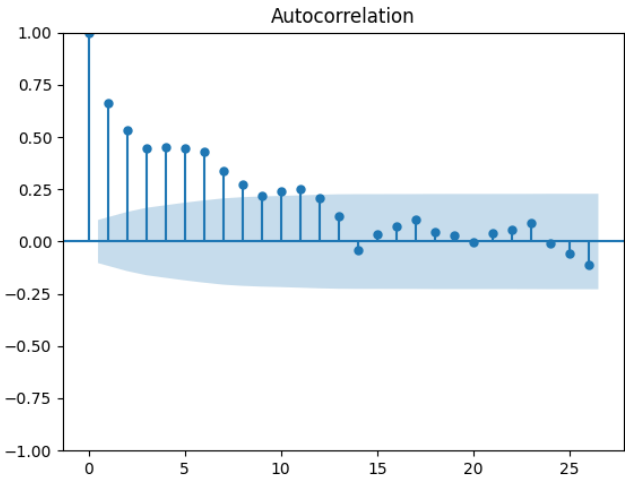
Notre P-value : 0.24126116082883653

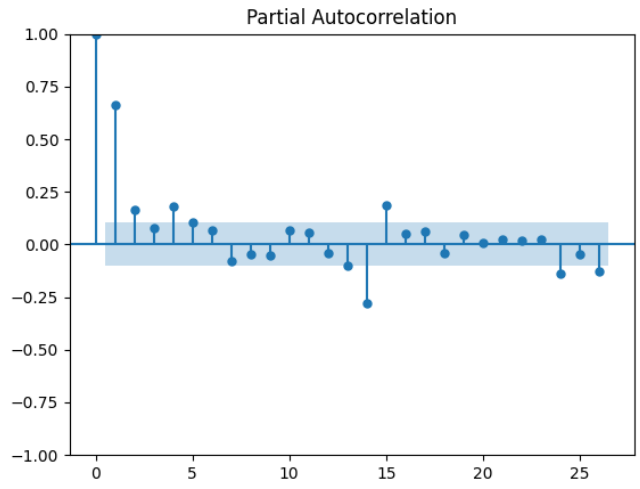
5. Transformation en série stationnaire

Nous allons utiliser la fonction **dropna() sur** notre jeu d'entrainement et nous obtenons nos 3 graphes avec de nouvelles allures.

La nouvelle **p-value** : 0.022059461239126298







Notre p-value est désormais inférieure à 0,05, ce qui signifie que notre série est désormais stationnaire.

6. Déterminons les paramètres P et Q du modèle ARIMA

Comme il nous a fallu une seule itération pour améliorer notre série, nous pouvons utiliser le modèle (2, 1, 0)

p = 2 : le nombre de termes autorégressifs (AR) à inclure dans le modèle est 2, ce qui signifie que la valeur actuelle de la série temporelle dépend des deux observations précédentes.

d = 1 : Cela signifie que la série nécessite une différenciation pour devenir stationnaire.

q = 0 : le nombre de termes de moyenne mobile (MA) à inclure dans le modèle est 0. Cela signifie que la valeur actuelle de la série temporelle ne dépend pas de l'erreur de prédiction précédente.

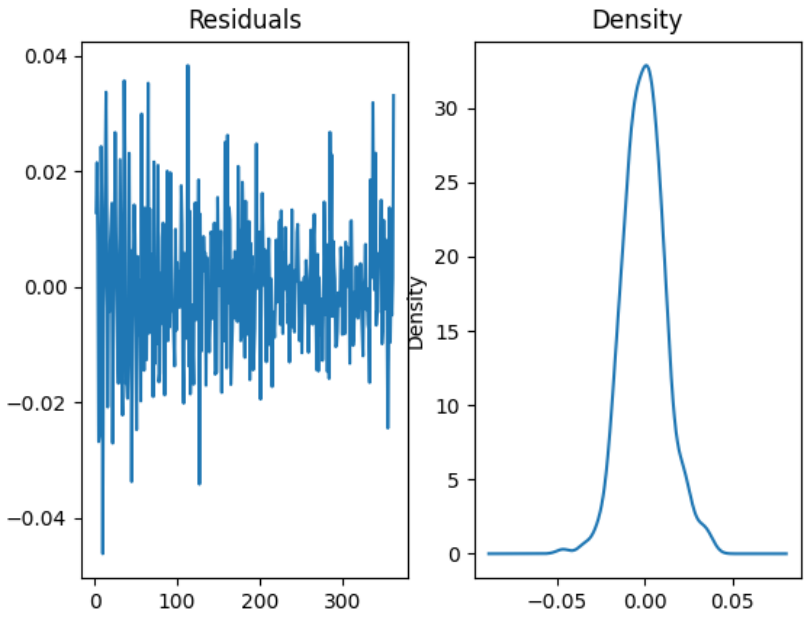
Prévisions et conclusion

1. Entrainement du modèle ARIMA

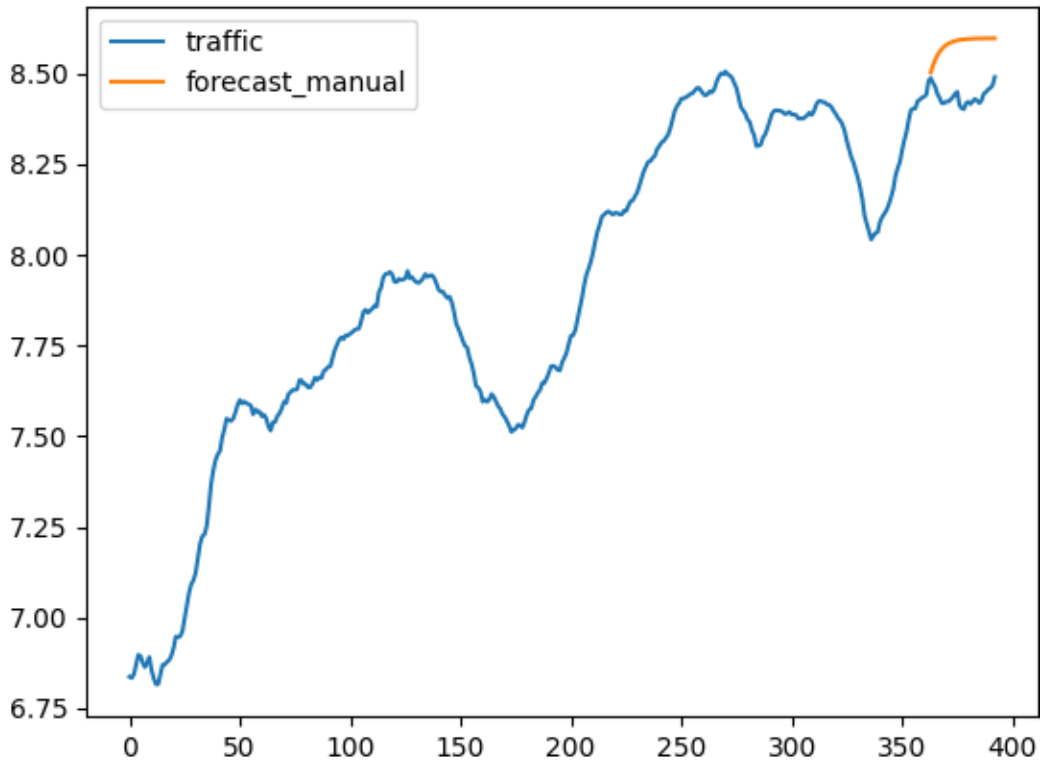
```
                               SARIMAX Results
==============================================================================
Dep. Variable:                traffic   No. Observations:                  362
Model:                 ARIMA(2, 1, 0)   Log Likelihood                1080.703
Date:                Thu, 27 Apr 2023   AIC                          -2155.406
Time:                        20:59:36   BIC                          -2143.739
Sample:                             0   HQIC                         -2150.768
                                  - 362
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.3499      0.050     -6.940      0.000      -0.449      -0.251
ar.L2         -0.1728      0.053     -3.274      0.001      -0.276      -0.069
sigma2         0.0001   9.53e-06     15.411      0.000       0.000       0.000
===================================================================================
Ljung-Box (L1) (Q):                   0.52   Jarque-Bera (JB):                 11.67
Prob(Q):                              0.47   Prob(JB):                          0.00
Heteroskedasticity (H):               0.39   Skew:                              0.17
Prob(H) (two-sided):                  0.00   Kurtosis:                          3.81
===================================================================================
```

2. Evaluation des résidus

On constate que les résidus sont très **proches du bruit** blanc. Il tourne entre autour de 0. Notre modèle est donc **prêt pour les prédictions**.



3. Prévision manuelle du modèle



4. Prévision automatique

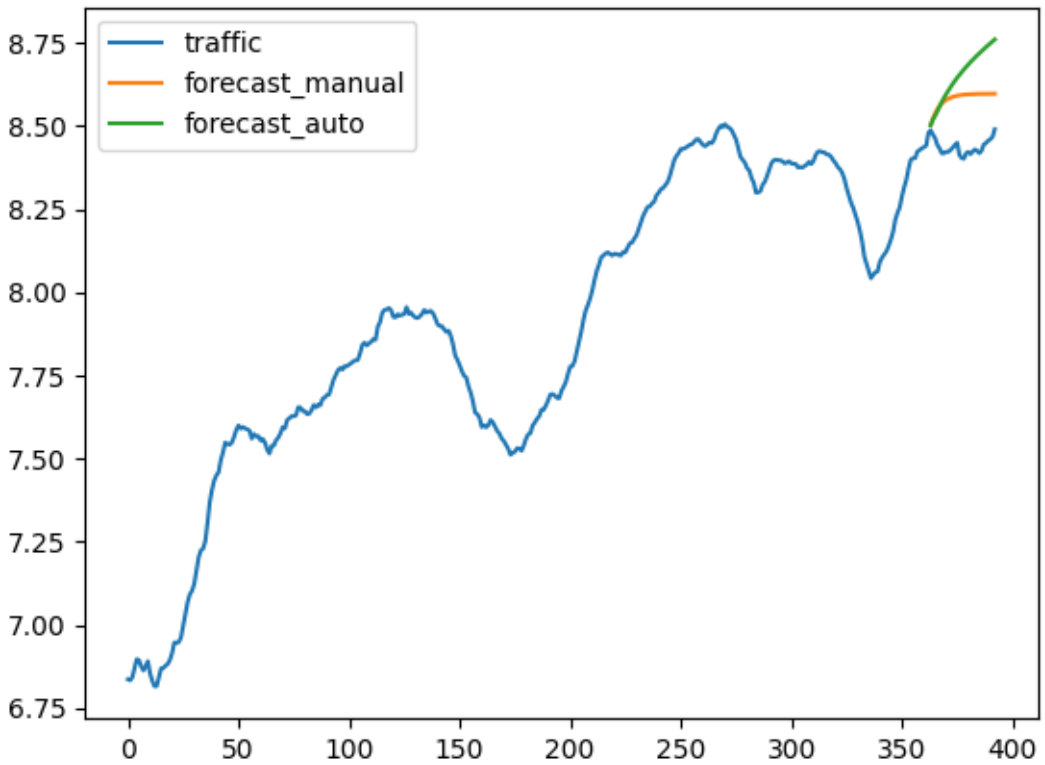
Dans le modèle précédent, nous avons-nous même trouvé les paramètres de notre modèle (2, 1, 0). Le résultat est satisfaisant, mais il peut être amélioré, nous proposons de laisser, une bibliothèque elle-même chercher les mêmes paramètres du modèle, à savoir **PMDARIMA.**

Ce module ajuste automatiquement un modèle ARIMA à la série temporelle df_train. La fonction **auto_arima** effectue une recherche exhaustive des modèles ARIMA les plus appropriés en fonction des paramètres fournis. Nous avons les résultats sur la figure ci-dessous.

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  363
Model:               SARIMAX(5, 1, 0)   Log Likelihood                1107.359
Date:                Thu, 27 Apr 2023   AIC                          -2200.719
Time:                        21:53:08   BIC                          -2173.477
Sample:                             0   HQIC                         -2189.889
                                  - 363
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
intercept      0.0009      0.001      1.511      0.131      -0.000       0.002
ar.L1          0.5213      0.052      9.979      0.000       0.419       0.624
ar.L2          0.0934      0.065      1.434      0.152      -0.034       0.221
===================================================================================
Prob(Q):                              0.91   Prob(JB):                         0.00
Heteroskedasticity (H):               0.54   Skew:                             0.23
Prob(H) (two-sided):                  0.00   Kurtosis:                         3.76
===================================================================================
```

Le module nous suggère donc le modèle **SARIMAX (5, 1, 0)**

Nous allons donc essayer de prédire avec ce modèle



5. Conclusion

Nous observons que les prédictions automatiques en vert du modèle (5, 1, 0) sont plus éloignées de la réalité que notre modèle manuel (2, 1, 0).

En plus de cela, vous pouvez aussi évaluer l'erreur avec la **mae**, **mse** et autres…

Il faudrait, aussi penser à la saisonnalité, parce que certains évènements peuvent avoir un impact sur le site web, par exemple, la publication de nouveaux articles.

Il est aussi important de penser comme amélioration de mélanger notre jeu de données.

Bibliographie

Documents	YouTube	Autres
Cours de M. HAKIM	Lianne and Justin	ChatGPT