Rapport du POM : Prédire les prochaines valeurs d'une série temporelle

Walid Serti, Grégory LARUE

Mai 2021

Résumé : Ce document présente l'étude de la prédiction de valeurs à partir de séries temporelles et de différentes méthodes statistiques, plus spécifiquement d'ARMA. Il présente également l'utilisation, la mise en place et l'évaluation de la précision de différents outils de prédictions. Une approche théorique de la prédiction de données sera présentée ainsi qu'une approche pratique à travers l'utilisation expérimentale d'outils prédictifs.

Mots-clés : prédiction, statistique, série temporelle, ARMA, précision, outil prédictif.

1 Introduction :

Les grandes entreprises s'intéressent depuis de nombreuses années à la prédiction de données. En effet, quel grand groupe ou entreprise ne rêverait pas de pouvoir moduler sa production ou ses investissements à l'avance ? Connaître les attentes et le futur du marché est donc devenu une grande préoccupation et un sujet de recherche qui attire l'attention encore aujourd'hui. De ce fait Amazon avec Amazon Forecast et DeepAR, ou encore Facebook avec FB Prophet se sont lancés dans le domaine de la prédiction de données à partir de séries temporelles. Mais ce ne sont pas les premiers, un modèle statistique apparu dans les années 80-90 est devenu un pilier de la prédiction de données à partir de séries temporelles : ARMA.

Notre projet est encadré par M Thomas Begin et il s'intègre dans une proposition de projet de recherche sur des données autonomes. Le postulat est : comment des données pourraient-elles s'auto-gérer dans le futur en fonction des prévisions statistiques ? Notre projet se base principalement autour de l'étude d'outils permettant de prédire la popularité d'une donnée en se basant sur sa popularité passée. L'étude des séries temporelles et des modèles statistiques tels qu'ARMA et ARIMA est le centre névralgique du sujet. Notre projet s'articule principalement autour de la découverte de ces modèles, comment les paramétrer et pouvoir étudier les prédictions qu'ils nous fournissent.

2- Quelques définitions :

2.1 - Série temporelle :

Une série temporelle ou série chronologique est une série de valeurs numériques qui représentent le changement d'une quantité spécifique au fil du temps.

Voici 2 exemples de séries temporelles :

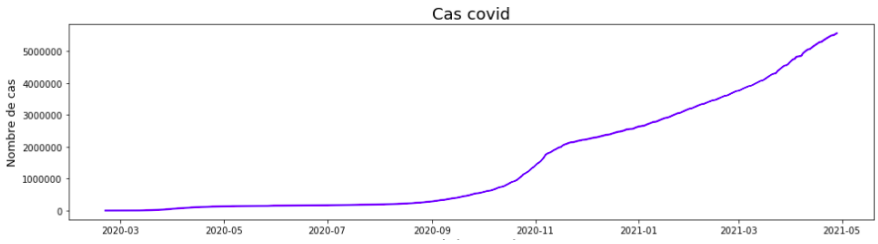


Figure 1.a: Série temporelle représentant l'évolution du nombre de cas cumulés total de Covid entre le 21 février 2020 et le 28 avril 2021.

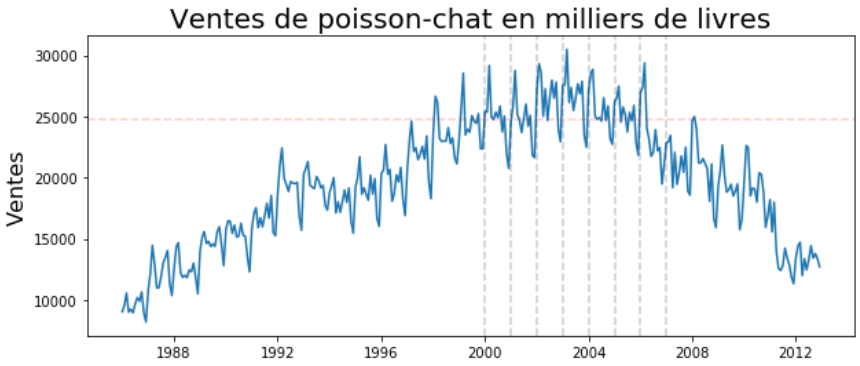


Figure1.b: Série temporelle représentant les ventes de poisson-chat en millier de livres par mois entre 1986 et 2012

2.2- Stationnarité de la série :

La stationnarité est une caractéristique très intéressante dans l'analyse des séries temporelles. Elle reflète en quelque sorte la stabilité des observations par rapport au temps. Un processus stochastique est stationnaire, si ses propriétés statistiques (espérance, variance, autocorrélation) ne changent pas dans le temps.

Une méthode pour rendre les séries temporelles stationnaires lorsqu'elles ne le sont pas consiste à les différencier. Supposons une série univariée {X,t ∈ [1,n] }. La différentiation de X d'ordre 1 est définie comme suit :

$$\Delta X(t) = X(t) - X(t-1)$$

Et la différentiation de X d'ordre d est définie comme suit :

$$\Delta^d X(t) = \Delta(\Delta^{d-1} X(t))$$

3-Les modèles de prédictions :

Les modèles de prédiction des séries temporelles diffèrent des modèles de prédiction classiques, car ils permettent d'utiliser l'historique des séries pour estimer les valeurs futures, en utilisant des variables retardées. Autrement dit, pour faire des prévisions à l'instant t, on suppose qu'on connait juste les valeurs passées(t−1,t-2...) Les modèles les plus populaires de ce type sont ; AR, MA, ARMA et ARIMA.

3.1- Le modèle AR :

Un processus X à temps discret est un processus AR d'ordre p ("Autoregressive" Process) s'il vérifie la relation de récurrence suivante :

AR (p) $x_t = \emptyset_1 x_{t-1} + \ldots + \emptyset_p x_{t-p} + a_t$:

a_t (un résidus) + la somme des valeurs précédentes (Xt-i) pondérées par un coef Φ_i.

3.2-Le modèle MA :

Le modèle Moyennes Mobiles (Moving Average) a la même structure que le modèle AR, mais en considérant les termes d'erreurs au lieu des valeurs précédentes de la série. Le modèle MA(q) peut être exprimé comme suit :

MA (q) : $x_t = a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q}$

a_t (un résidus) + la somme des résidus des valeurs précédentes pondérées par un coef Θ_i.

3.3-Le modèle ARMA/ARIMA :

Avant de voir le modèle ARIMA, il est important de comprendre le modèle ARMA. Le modèle ARMA(p,q) combine les deux processus AR(p)et MA(q) en considérant à la fois les termes d'erreurs et les valeurs précédentes de la série :

ARMA (p,q) : qui est la combinaison de AR et MA

$$x_t = \emptyset_1 x_{t-1} + \dots + \emptyset_p x_{t-p} + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q}$$

Petit exemple : ARMA (2,3) :

$$x_t = \emptyset_1 x_{t-1} + \emptyset_2 x_{t-2} + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \theta_3 a_{t-3}$$

Le modèle ARIMA(p,d,q) est plus complet. Il consiste à appliquer le modèle ARMA(p,q) sur la série transformée par différenciation d'ordre d, c'est-à-dire, en calculant d fois les différences entre les observations consécutives. Il existe un dernier modèle de ce type, SARIMA qui reprend le modèle ARIMA en y ajoutant cette fois une composante de saisonnalité (La saisonnalité est une variation cyclique dépendant de la période de l'année).

4-Méthode de Box & Jenkins :

L'objectif de cette méthodologie est la modélisation d'une série temporelle en fonction de ses valeurs passées et présentes afin de déterminer le processus ARIMA approprié par principe de parcimonie. Cela correspond au meilleur rapport coût/efficacité, car quoi qu'il arrive plus les ordres seront haut plus notre prédiction sera précise mais il faut savoir déterminer les ordres les plus faibles tout en restant le plus précis possible.

Cette méthodologie suggère une identification du modèle avec une procédure en quelques étapes : (Figure 2)

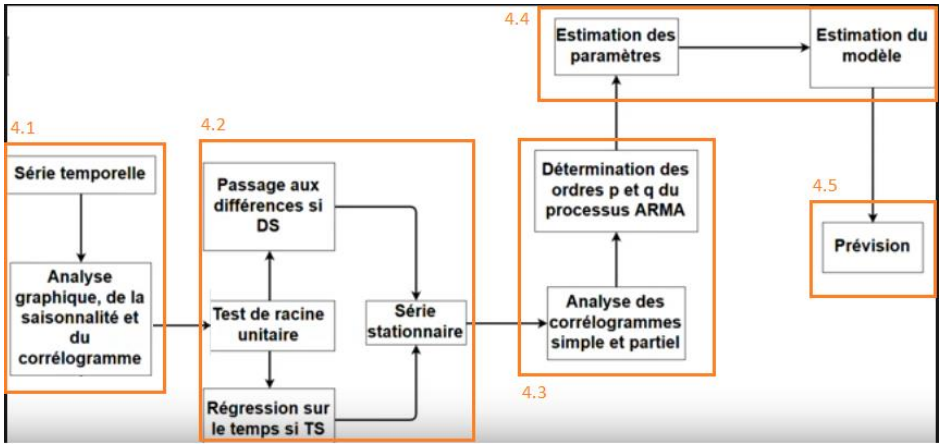


Figure 2 : Les étapes de la méthode Box & Jenkins, en orange la correspondance à chaque sous partie.

4.1- Identification et analyse du modèle :

La première étape consiste à identifier la série temporelle a étudier (prédire), lors de cet exemple on étudiera la série temporelle sur les ventes de poissons-chats : (voir Figure 1 bas)

Analyse de la saisonnalité : Lorsque cette composante existe, il convient de l'isoler afin de pouvoir analyser les autres caractéristiques, ce qui se fait à l'aide de la méthode de désaisonnalisation. Sans tester, l'existence de cette composante peut créer un « bruit » parasite nuisible à l'analyse de la série et donc dégrader la qualité de la prévision.

4.2 - Test de racine unitaire (Dickey-Fuller) et stationnarisation de la série :

Test de racine unitaire de Dickey-Fuller est un test statistique qui vise à savoir si une série temporelle est stationnaire. On estime que la série est stationnaire lorsque la p-value du test se rapproche de 0 (ou < à 0.05). Sur la figure suivante on peut voir le test de racine unitaire sur la série temporelle avant différenciation (figure 3 à gauche) et après une 1ère différenciation (figure 3 à droite). On remarque que la p-value devient inférieure à 0.05, il n'est donc pas nécessaire d'augmenter l'ordre de différenciation au-dessus de 1, la série est devenue stationnaire (figure 4).

```
perform_adf_test(first_diff)
```

```
ADF Statistic: -4.310935
p-value: 0.000425
```

```
perform_adf_test(catfish_sales)
```

```
ADF Statistic: -1.589903
p-value: 0.488664
```

Figure 3 : Test de racine unitaire Dickey Fuller sur Python

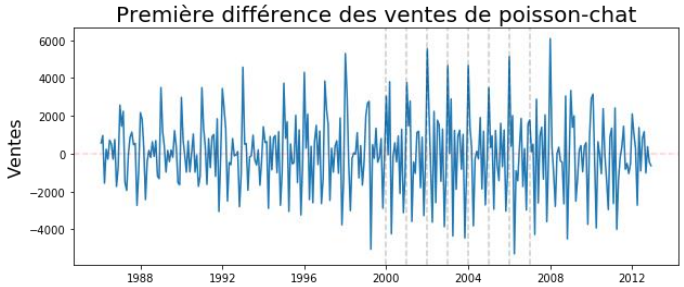


Figure 4 : Différenciation à l'ordre 1, stationnarisation de la série

4.3-Analyse des corrélogrammes :

La corrélation (Coefficient) est une grandeur qui mesure la force de la relation entre deux variables, elle est toujours comprise entre -1 et 1.

Si ce coef est proche de -1 : les variables sont inversement corrélées.

Si ce coef est proche de 0 : les variables ne sont pas liées.

Si ce coef est proche de 1 : les variables sont corrélées

Finalement calculer l'autocorrélation c'est calculer le lien que la série a avec elle-même avec un décalage temporel, le tout dans le but de déterminer les ordres p et q du processus ARIMA. Le corrélogramme est quant à lui la représentation graphique de la fonction de corrélation ou d'autocorrélation appliquée à la série. Lors de l'analyse des corrélogrammes, les ordres candidats sont ceux qui sont en dehors de la zone de confiance (figure 5, en bleu clair). On déterminera les ordres exacts lors de l'estimation du modèle, en calculant les AIC et BIC qui sont des mesures de la qualité d'un modèle statistique.

4.3.1 - corrélogramme simple :

Il permet de visualiser l'influence de la valeur précédente sur une valeur actuelle mais aussi de l'influence cumulée des valeurs intermédiaires. Ce corrélogramme nous servira à déterminer les ordres q qui sont candidats.

4.3.2 - corrélogramme partiel :

Permet de visualiser l'influence de la valeur précédente sans parasite des valeurs intermédiaires. Ce corrélogramme nous servira à déterminer les ordres p qui sont candidats.

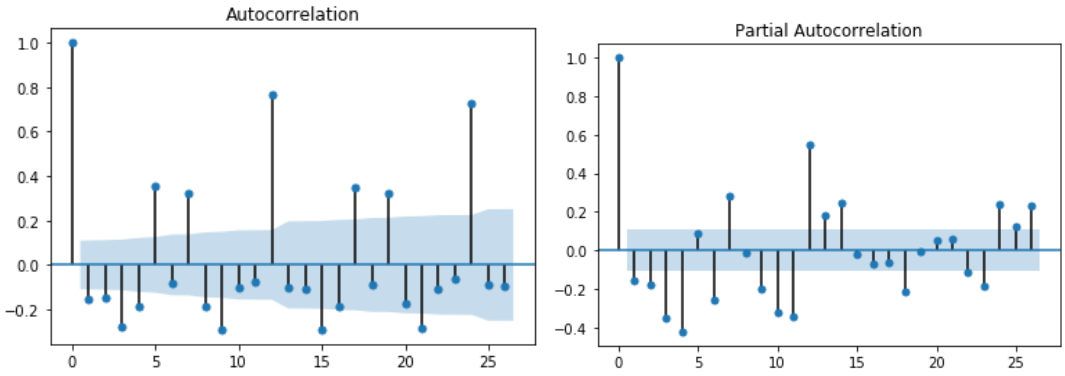


Figure 5 : corrélogrammes simple et partiel de notre série

4.4- Estimations des paramètres et du modèle :

Φ et θ (cf. 3.3 formule de ARMA(p,q)) représentent des coefficients (nombre réels) qui vont nous indiquer avec quel poids une mesure du passé va influer sur la valeur à déterminer. Pour les estimer plusieurs méthodes existent voici les deux principales :

- La méthode des moindres carrés : qui est une approche algébrique par résolution d'équation matricielle. (Celle utilisée par la méthode en python cf. Annexe 1)
- La méthode de vraisemblance : qui est une approche statistique en considérant que la valeur des paramètres suit une voie statistique.

Après avoir déterminé les ordres de notre modèle nous récupérons un résumé de notre modèle ARMA et notons les coefficients afin d'obtenir l'équation de notre modèle :

Le modèle ARMA(10,4) est:

$$\hat{x}_t = 0.54x_{t-1} - 1.38x_{t-2} + 0.27x_{t-3} - 0.73x_{t-4} - 0.07x_{t-5} - 0.67x_{t-6} + 0.37x_{t-7} - 0.79x_{t-8} + 0.17x_{t-9} - 0.45x_{t-10} - 1.02a_{t-1} + 1.68a_{t-2} - 0.97a_{t-3} + 0.76a_{t-4}$$

Figure 6 : Equation du modèle ARMA obtenue sous python

Par la suite on vérifie la concordance entre le modèle et les données (figure 7)

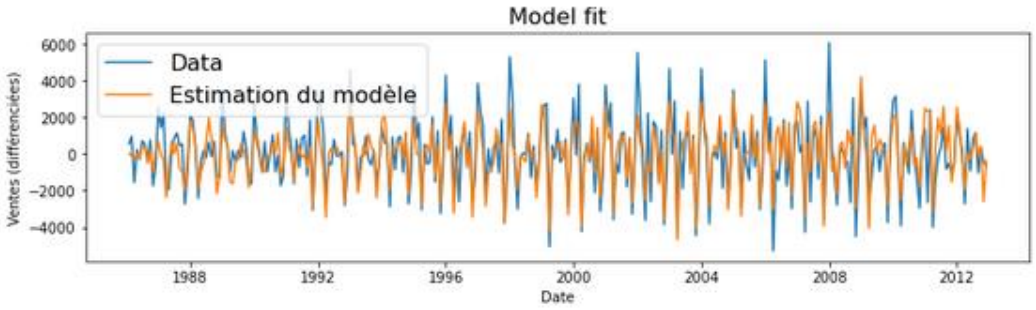


Figure 7 : Estimation du modèle via ARMA

4.5- Prédictions :

On passe ensuite une série de données en guise de données d'apprentissages à notre modèle. Par la suite, ce dernier prédira des valeurs par lui-même. Dans notre cas les données d'apprentissage étaient de 1986 à 2005 et nous avons ensuite prédit à l'aide de notre modèle les valeurs de 2005 à 2013 (figure 8).

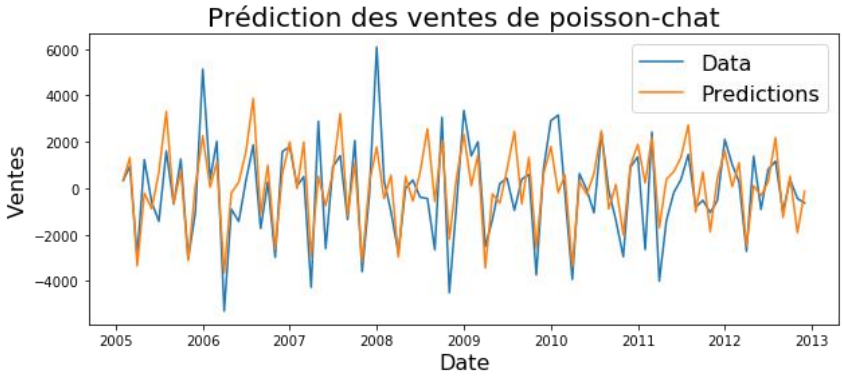


Figure 8 : Résultat de la prédiction (ARMA)

5-Les outils disponibles et nos choix d'implémentations :

Il existe différents outils permettant de réaliser de la prédiction de données à partir de séries temporelles. Prenons simplement l'exemple d'ARMA/ARIMA. Il est possible d'utiliser ces modèles statistiques sur différentes plateformes ou avec différents langages. On notera R, Matlab/Octave, Python comme les principaux médiums d'utilisation. Dans le cadre de notre projet nous avions commencé par utiliser Octave et la bibliothèque tsa (Schloegl, 2019). Après quelques tests des différentes fonctions, nous avons remarqué que ce n'était le plus adéquat pour commencer à appréhender le fonctionnement du modèle (fonctions trop complètes et denses). Nous nous sommes rapidement dirigés vers Python qui nous a permis de mieux comprendre le modèle ARMA et son fonctionnement à travers le découpage en plusieurs étapes de la mise en place du modèle (cf. parties 3 & 4).

Après avoir découvert le fonctionnement étape par étape d'ARMA sous python nous nous sommes renseignés sur les autres méthodes et modèles de prédiction de données. Le Machine Learning (MASINI et al.) est une solution qui se développe de plus en plus afin de faire de la prédiction de données sur les séries

temporelles (Amazon Forecast). Un autre outil prédictif a aussi attiré plus particulièrement notre attention. FBProphet, une solution openSource développée par Facebook utilisable sous python. Nous avons donc décidé de la tester rapidement et ensuite de la comparer à (S)ARIMA sur un même jeu de données (cf. partie 7).

6-Méthodes d'évaluation de la précision des prédictions :

6.1-Erreurs de prévision

Une "erreur" de prévision est la différence entre une valeur observée et sa prévision. Ici, "erreur" ne signifie pas une faute, mais la partie imprévisible d'une observation. Elle peut s'écrire comme suit :

$$e_{T+h} = y_{T+h} - \hat{y}_{T+h|T},$$

Où les données d'apprentissage sont données par {y1,..., yT}et les données de test sont données par {yT+1, yT+2,...}.

Notez que les erreurs de prévision diffèrent des résidus de deux façons. Premièrement, les résidus sont calculés sur l'ensemble d'apprentissage alors que les erreurs de prévision sont calculées sur l'ensemble de test. Deuxièmement, les résidus sont basés sur des prévisions à une étape, tandis que les erreurs de prévision peuvent impliquer des prévisions à plusieurs étapes.

6.2-Erreurs dépendantes de l'échelle

Les erreurs de prévision sont sur la même échelle que les données. Les mesures de précision qui se basent uniquement sur e_t et sont donc dépendantes de l'échelle et ne peuvent pas être utilisées pour faire des comparaisons entre des séries qui impliquent des unités différentes.

Les deux mesures dépendantes de l'échelle les plus couramment utilisées sont basées sur les erreurs absolues ou les erreurs au carré :

$$\text{Mean absolute error: MAE} = \text{mean}(|e_t|),$$
$$\text{Root mean squared error: RMSE} = \sqrt{\text{mean}(e_t^2)}.$$

Lors de la comparaison de méthodes de prévision appliquées à une seule série temporelle, ou à plusieurs séries temporelles avec les mêmes unités, la MAE est populaire car elle est facile à comprendre et à calculer. Une méthode de prévision qui minimise la MAE conduira à des prévisions de la médiane, tandis que la minimisation de la RMSE conduira à des prévisions de la moyenne. Par conséquent, la RMSE est également largement utilisée.

6.3-Erreurs relatives

L'erreur en pourcentage est donnée par : pt=100et/yt. Les erreurs relatives ont l'avantage d'être sans unité, et sont donc fréquemment utilisées pour comparer les performances de prévision entre les ensembles de données.

La mesure la plus couramment utilisée est la suivante :

$$\text{Mean absolute percentage error: MAPE} = \text{mean}(|p_t|).$$

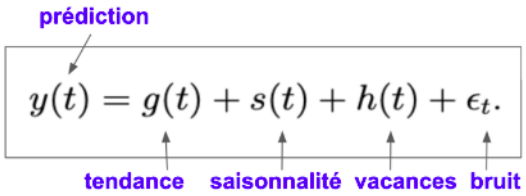
Les mesures basées sur les erreurs relatives ont l'inconvénient d'être infinies ou indéfinies si yt=0 pour tout t dans la période d'intérêt, et d'avoir des valeurs extrêmes si tout yt est proche de zéro. Un autre problème des erreurs relatives, souvent négligé, est qu'elles supposent que l'unité de mesure a un zéro significatif. Par exemple, un pourcentage d'erreurs n'a aucun sens lorsqu'on mesure la précision des prévisions de température sur les échelles Fahrenheit ou Celsius, car la température a un point zéro arbitraire.

7- Autre outil de prédictions :

7.1- Facebook prophet :

Prophet est un outil développé par Facebook dans le but de démocratiser les prévisions des séries temporelles sur un modèle additif et de les simplifier. Ce modèle convient particulièrement à des séries temporelles de type 'business' affectées par des événements ou des saisonnalités liées à l'activité humaine (exemple : fêtes de fin d'année, soldes, saisons, vacances, etc.). Le modèle est implémenté dans un langage d'inférence statistique en C++ qui s'appelle Stan et est disponible en open source depuis 2017 en R et en Python

Le modèle de Facebook Prophet est un modèle additionnant 3 éléments : la tendance, la saisonnalité, l'effet de l'événement / vacances (auxquels s'ajoute le bruit) :



où :
Tendance g (t) : modélise les changements non périodiques.
Saisonnalité s (t) : représente les changements périodiques.
Composant vacances h (t) : fournit des informations sur les vacances et les événements.

7.2 Pourquoi Prophet ?

Prophet vient répondre à 3 problématiques de scalabilité :

1. Le grand nombre de personnes qui travaillent sur les prévisions temporelles et qui ont une faible connaissance mathématique du domaine.
2. Le grand nombre de problèmes de prévisions qui peuvent parfois venir d'un caractère idiomatique.
3. Le grand nombre de sujets sur lesquels il est possible d'appliquer des prévisions temporelles qui demandent des configurations différentes.

7.3- Facebook Prophet vs ARIMA :

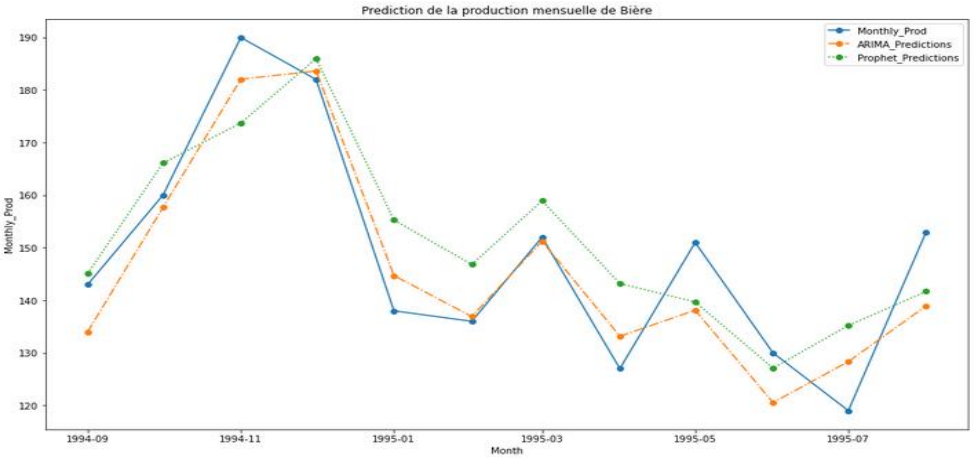


Figure 9 : Prévision des 12 mois correspondants sur la production mensuelle de bières selon ARIMA et Prophet

Comparaison des prédictions :

	Models	RMSE Errors	MSE Errors
0	ARIMA	8.107055	65.724344
1	Prophet	11.434214	130.741242

En étudiant ici les erreurs, ARIMA semble plus efficace et précis dans ses prédictions que Prophet qui reste tout de même très correct. Ainsi on déduit que Prophet est très pratique par sa facilité d'utilisation mais ARIMA, lorsqu'il est bien paramétré est un outil prédictif excellent.

8- Conclusion :

La prévision des séries temporelles peut s'avérer complexe, mais de nombreuses techniques, telles que le modèle ARIMA ou l'API Facebook Prophet, peuvent offrir l'avantage de bons résultats pour un faible coût en efforts et complexité. Lors de notre projet nous avons pu appréhender le fonctionnement étape par étape d'ARMA/ARIMA via la méthode Box & Jenkins (nom parfois aussi employé pour désigner ARMA). De plus, nous avons pu découvrir le domaine de la prédiction de données et de l'étude des séries temporelles. Enfin nous nous sommes ouverts à des outils différents, notamment en manipulant brièvement fbProphet.

Malgré tout nous avons remarqué lors de ce projet que pour utiliser ARMA de façon optimale, un background mathématique important était nécessaire.

ARMA reste ainsi un très bon modèle quand on appréhende bien la série étudiée d'un point de vue statistique. Dans les cas où il n'est pas évident de faire ressortir les propriétés statistiques, d'autres méthodes telles que le Deep Learning ou d'autres modèles statistiques (fbProphet...) peuvent être intéressants à utiliser.

Références

Schloegl, A. (2019, 10 24). Retrieved from Octave Forge: https://octave.sourceforge.io/tsa/overview.html

MASINI, Ricardo P, et al. *Machine Learning Advances for Time Series Forecasting*. arXivLabs, 2012.

ANNEXES :

```
best_aic = np.inf
best_order = None
best_mdl = None

for i in [1,3,4,6,7,9,10]:
    for j in [1,3,4,5,7]:
        try:
            tmp_mdl = smt.ARMA(first_diff, order=(i,j)).fit()
            tmp_aic = tmp_mdl.aic
            if tmp_aic < best_aic:
                best_aic = tmp_aic
                best_order = (i, j)
                best_mdl = tmp_mdl
        except: continue


p('aic: {:6.5f} | order: {}'.format(best_aic, best_order))
```

Annexe 1 : Détermination des ordres p et q

ARMA Model Results

Dep. Variable:	Total	No. Observations:	323
Model:	ARMA(10, 4)	Log Likelihood	-2741.064
Method:	css-mle	S.D. of innovations	1157.573
Date:	Mon, 17 May 2021	AIC	5514.128
Time:	14:41:54	BIC	5574.570
Sample:	02-01-1986 - 12-01-2012	HQIC	5538.256

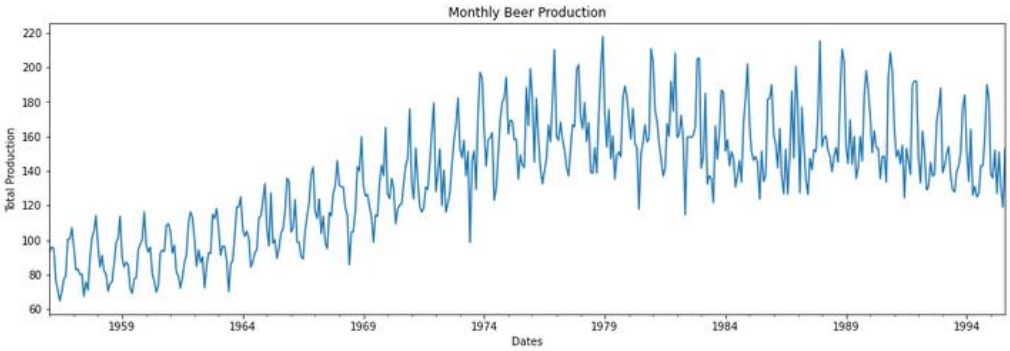
	coef	std err	z	P>|z|	[0.025	0.975]
const	14.3199	25.428	0.563	0.573	-35.519	64.159
ar.L1.Total	0.5371	0.073	7.344	0.000	0.394	0.680
ar.L2.Total	-1.3826	0.074	-18.619	0.000	-1.528	-1.237
ar.L3.Total	0.2729	0.097	2.816	0.005	0.083	0.463
ar.L4.Total	-0.7307	0.094	-7.751	0.000	-0.916	-0.546
ar.L5.Total	-0.0655	0.088	-0.744	0.457	-0.238	0.107
ar.L6.Total	-0.6703	0.085	-7.871	0.000	-0.837	-0.503
ar.L7.Total	0.3702	0.087	4.250	0.000	0.199	0.541
ar.L8.Total	-0.7865	0.088	-8.989	0.000	-0.958	-0.615
ar.L9.Total	0.1721	0.065	2.640	0.008	0.044	0.300
ar.L10.Total	-0.4519	0.056	-8.028	0.000	-0.562	-0.342
ma.L1.Total	-1.0153	0.067	-15.129	0.000	-1.147	-0.884
ma.L2.Total	1.6838	0.069	24.284	0.000	1.548	1.820
ma.L3.Total	-0.9654	0.073	-13.139	0.000	-1.109	-0.821
ma.L4.Total	0.7605	0.051	14.807	0.000	0.660	0.861

Roots

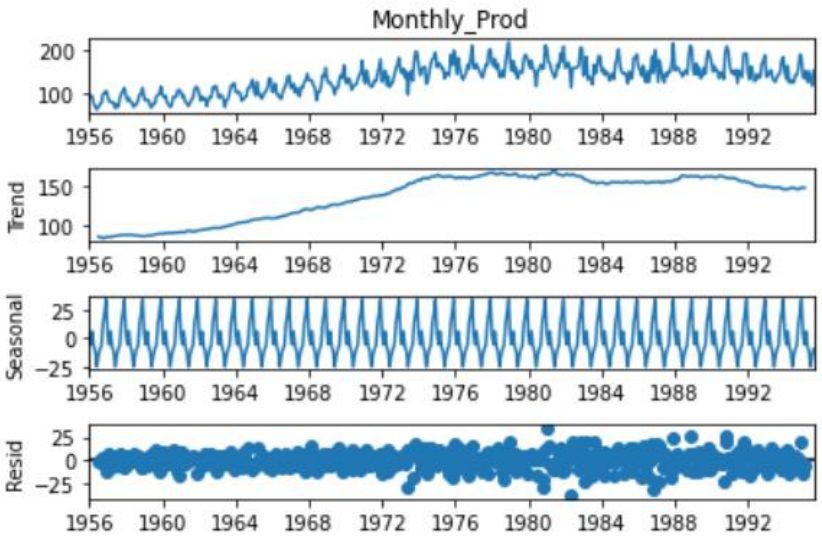
	Real	Imaginary	Modulus	Frequency
AR.1	0.9467	-0.5620j	1.1009	-0.0853
AR.2	0.9467	+0.5620j	1.1009	0.0853
AR.3	0.5024	-0.8714j	1.0059	-0.1668
AR.4	0.5024	+0.8714j	1.0059	0.1668
AR.5	-0.9392	-0.5426j	1.0847	-0.4166
AR.6	-0.9392	+0.5426j	1.0847	0.4166
AR.7	-0.0027	-1.0038j	1.0038	-0.2504
AR.8	-0.0027	+1.0038j	1.0038	0.2504
AR.9	-0.3168	-1.1924j	1.2337	-0.2913
AR.10	-0.3168	+1.1924j	1.2337	0.2913
MA.1	-0.0324	-1.0294j	1.0300	-0.2550
MA.2	-0.0324	+1.0294j	1.0300	0.2550
MA.3	0.6671	-0.8913j	1.1133	-0.1477
MA.4	0.6671	+0.8913j	1.1133	0.1477

Annexe 2 : Bilan du modèle ARMA(10,4)

Annexe 3 : Principe de jeu de données d'entraînement pour les prédictions



Annexe 4 : Série Temporelle représentant la production de Bière mensuelle en Australie (utilisée dans la comparaison ARIMA/FBProphet)



Annexe 5 : Décomposition de la série sur la production de bière selon FBProphet