



Rapport P7 : Méthodes de prévision des séries chronologiques

LO OUSMANE

PARCOURS MACHINE LEARNING ENGINEER

Novembre 2022

Mentor :
NICOLAS TISSERAND
ntisserand@gmail.com

Etudiant :
OUSMANE LO
ousmanelo78@gmail.com

Résumé

For many years, time series forecasting techniques have been in strong development, in particular thanks to the constant increase in available computing power. This has in particular allowed to develop reliable prediction models on large databases important, responding to complex issues. Classically used models currently are based on different methods : Auto ARIMA, PROPHET, TBATS etc. None of these methods cannot be considered better than another, since the performance of each will strongly depend on the type and number of data and the complexity of the problem. other newer methods exist in order to improve performance.[1] ,[3] These articles are published from 2019, so this is a new approach. The principle consists in applying methods to produce accurate probabilistic forecasts, based on the training of an auto-regressive recurrent network model on a large number of related time series. My project would therefore be to use this method on more real, less structured data in order to be able to analyze the performance on this type of data. To use the most real data possible, several databases are available on the datasets of the darts package. The theme chosen is the forecast of the current price of Energy in Spain in several cities. Over four years of energy and weather data are provided in 1 csv files. The productions are in megawatts MW and the prices are in Euros. The idea is to use the information provided to answer certain questions, essentially to see if it is possible to improve the results obtained with the classical models. I will be using three different time series libraries to try and accomplish this task. The data is available in the DARTS/datasets library.

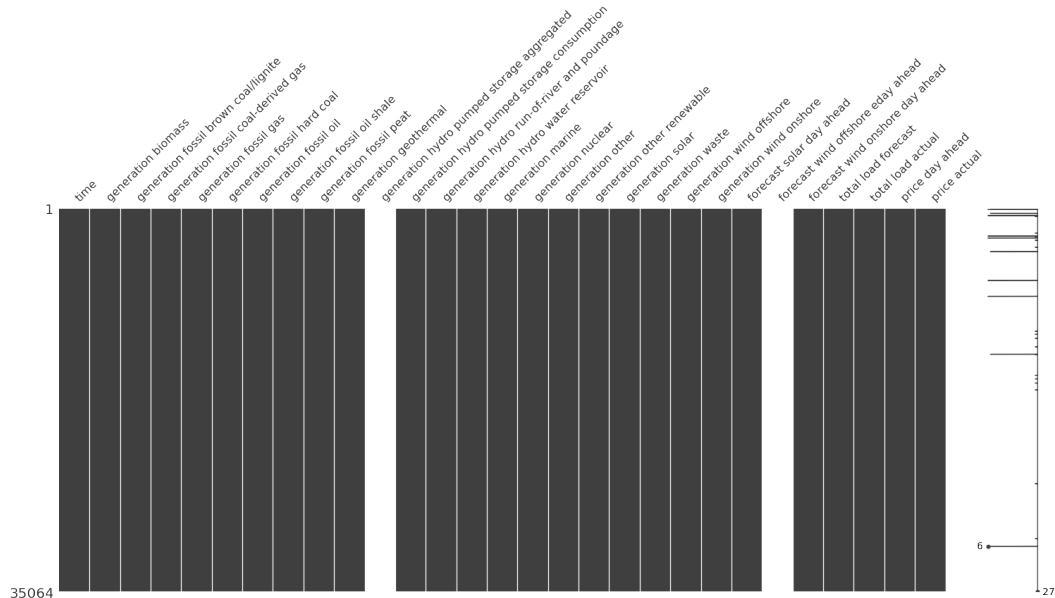


FIGURE 2 – Dataset's features

Table des figures

2	Dataset's features	1
3	Plot dataset's train and test	3
4	Summary of Auto arima model	4
5	Predictions by Auto arima model	5
6	Plot Prophet model	6
7	Predictions by Prophet model	6
8	Predictions by Xgboost model	7
9	Predictions by Tbats model	8
10	Predictions by Nbeats model	9
11	Predictions by Tft model	10
12	Predictions by DeepAR model	11
13	plot models with rmse/training time	12

Table des matières

1	Introduction	3
2	Prévision à l'aide de modèles classiques	4
2.1	Auto ARIMA	4
2.1.1	Resultats obtenus avec Auto ARIMA	4
2.2	Prophet	5
2.2.1	Resultats obtenus avec Prophet	5
2.3	Xgboost	6
2.3.1	Resultats obtenus avec Xgboost	6
2.4	Tbeats	7
2.4.1	Resultats obtenus avec Tbats	7
3	Prévision à l'aide de modèles récents	8
3.1	Nbeats models	8
3.1.1	Resultats obtenus avec Nbeats	8
3.2	Tft models	9
3.2.1	Resultats obtenus avec Tft	9
3.3	DeepAR models	10
3.3.1	Resultats obtenus avec DeepAR	10
4	Comparaison de Modèles	11
5	Conclusion	12
	Références	13

1 Introduction

Depuis de nombreuses années, les techniques de prévision des séries temporelles sont en fort développement, notamment grâce à l'augmentation constante de la puissance de calcul disponible. Cela a notamment permis de développer des modèles de prédictions fiables sur des bases de données de tailles importantes, répondant à des problématiques complexes. Les modèles classiquement utilisés actuellement sont basés sur différentes méthodes : les méthodes Auto ARIMA, PROPHET, TBATS etc. Aucune de ces méthodes ne peut être considérée meilleure qu'une autre, puisque les performances de chacune vont fortement dépendre du type et du nombre de données et de la complexité du problème. d'autres méthodes plus récentes existent afin d'améliorer les performances.[1], [3] Ces articles sont publiés à partir de 2019, il s'agit donc d'une nouvelle approche. Le principe consiste à appliquer des méthodes pour produire des prévisions probabilistes précises, basée sur la formation d'un modèle de réseau récurrent auto-régressif sur un grand nombre de séries temporelles connexes. Mon projet serait donc d'utiliser cette méthode sur des données plus réelles, moins structurées afin de pouvoir analyser la performance sur ce type de données. Pour utiliser des données les plus réelles possibles, plusieurs bases de données sont disponibles sur le datasets du package darts. Le thème choisi est la prévision du prix actuel de l'Energie en Espagne sur plusieurs villes. Plus de quatre années de données énergétiques et météorologiques sont fournies dans 1 fichiers csv. Les productions sont en mégawatts MW et les prix sont en Euros. L'idée est d'utiliser les informations fournies pour répondre à certaines questions, essentiellement pour voir s'il est possible d'améliorer les résultats obtenus avec les modèles classiques. J'utiliserai trois bibliothèques de séries chronologiques différentes pour essayer d'accomplir cette tâche. Les données sont disponibles dans la bibliothèque DARTS/datasets.

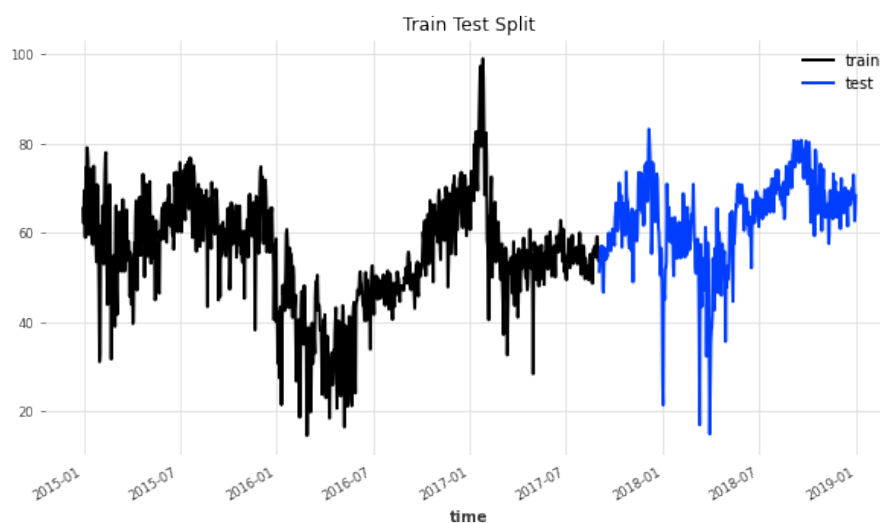


FIGURE 3 – Plot dataset's train and test

2 Prédiction à l'aide de modèles classiques

2.1 Auto ARIMA

Habituellement, dans le modèle ARIMA de base, nous devons fournir les valeurs p , d et q qui sont essentielles. Nous utilisons des techniques statistiques pour générer ces valeurs en effectuant la différence pour éliminer la non-stationnarité et en traçant les graphiques ACF et PACF. Dans Auto ARIMA, le modèle lui-même générera les valeurs p , d et q optimales qui conviendraient à l'ensemble de données pour fournir une meilleure prédiction.

2.1.1 Resultats obtenus avec Auto ARIMA

Le modèle auto arima nous donne SARIMAX(1,0,2) (Figure 4) comme modèle optimale pour fournir une meilleur prédiction. Ce dernier fournit une erreur RMSE de 12.10% (Figure 5). Cependant ce modèle ne semble pas mieux fonctionner correctement sur nos données car les valeurs sont légèrement constantes.

Dep. Variable:	y	No. Observations:	976			
Model:	SARIMAX(1, 0, 2)	Log Likelihood	-3090.560			
Date:	Mon, 21 Nov 2022	AIC	6191.121			
Time:	04:43:01	BIC	6215.538			
Sample:	12-31-2014	HQIC	6200.412			
	- 09-01-2017					
Covariance Type:	opg					
=====						
	coef	std err	z	P> z	[0.025	0.975]

intercept	0.7172	0.282	2.540	0.011	0.164	1.271
ar.L1	0.9872	0.005	203.591	0.000	0.978	0.997
ma.L1	-0.2624	0.025	-10.483	0.000	-0.311	-0.213
ma.L2	-0.4415	0.025	-17.925	0.000	-0.490	-0.393
sigma2	32.8882	1.212	27.129	0.000	30.512	35.264
=====						
Ljung-Box (L1) (Q):	0.15	Jarque-Bera (JB):	126.74			
Prob(Q):	0.70	Prob(JB):	0.00			
Heteroskedasticity (H):	0.58	Skew:	-0.56			
Prob(H) (two-sided):	0.00	Kurtosis:	4.37			

FIGURE 4 – Summary of Auto arima model

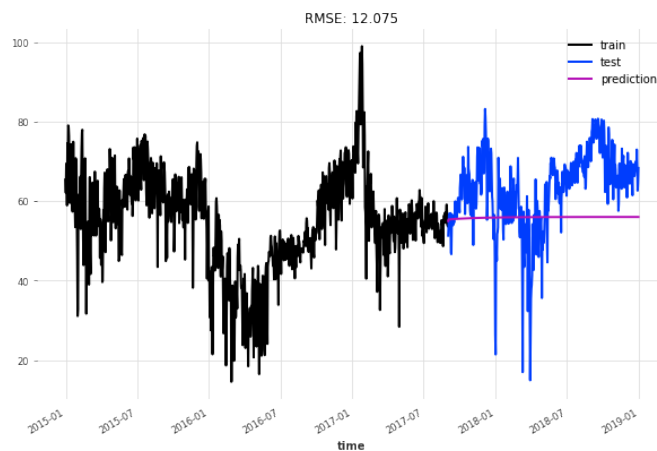


FIGURE 5 – Predictions by Auto arima model

2.2 Prophet

Facebook [2], a publié un modèle étonnant appelé Facebook Prophet. Prophet permet à presque tout le monde de prédire les valeurs de séries chronologiques, même si vous avez très peu ou pas d'expérience dans ce domaine. Dans la plupart des cas, cela fonctionne correctement et votre analyste de données sera en mesure de raconter des histoires assez précises avec la sortie. Mais, certains paramètres nécessitent un réglage car nous utiliserons une simple trame de données y et ds (le format utilisé par Prophet).

2.2.1 Resultats obtenus avec Prophet

Les predictions obtenus avec le modèle prophet sont un peu éloignés du jeu de données de validation ce qui donne une erreur RMSE très élevée de 30.51% comparé au modèle Auto arima qui est de 12.10%.

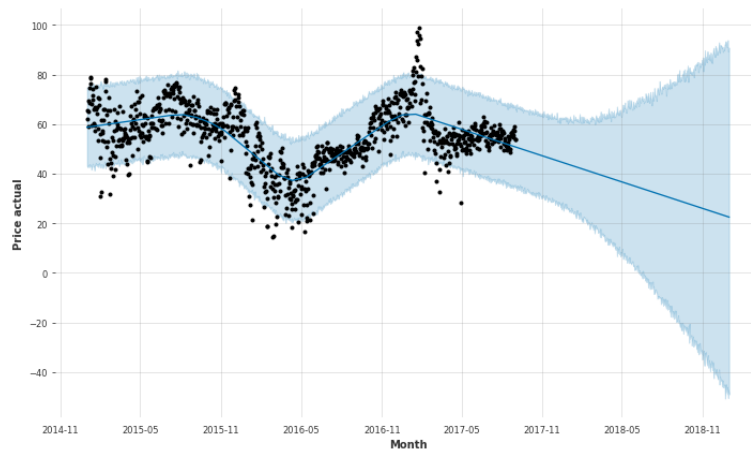


FIGURE 6 – Plot Prophet model

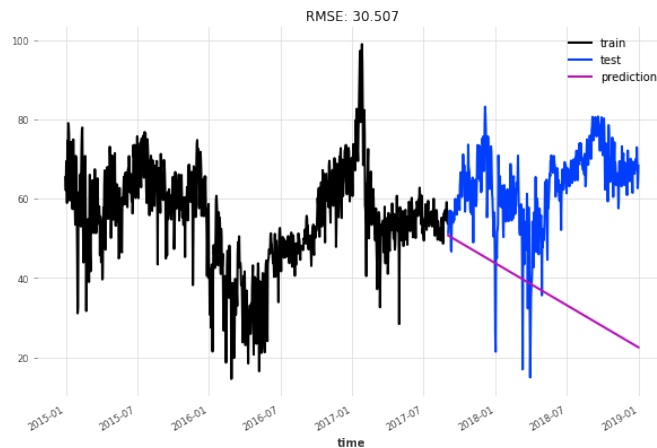


FIGURE 7 – Predictions by Prophet model

2.3 Xgboost

XGBoost (Extreme Gradient Boosting)[2] est une implémentation de gradient boosting pour les problèmes de classification et de régression. Cela peut être utilisé pour la prévision de séries chronologiques en restructurant l'ensemble de données d'entrée pour ressembler à un problème d'apprentissage supervisé.

2.3.1 Resultats obtenus avec Xgboost

La prévision avec le modèle Xgboost fournit de très bon résultat comparé avec une erreur de prévision de 6.42% (Figure 8) aux modèles Auto arima et prophet

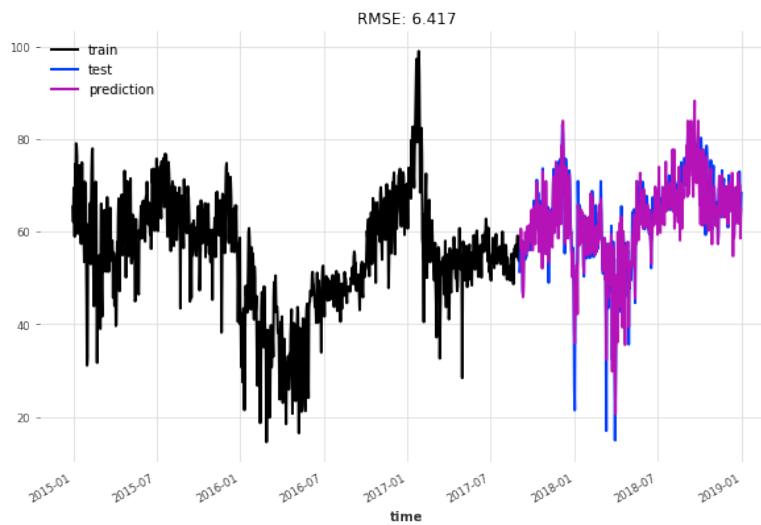


FIGURE 8 – Predictions by Xgboost model

2.4 Tbeats

BATS et TBATS[2] sont des algorithmes de prévision de séries chronologiques et fonctionnent bien avec plusieurs périodes saisonnières. Le TBATS est préféré lorsque la saisonnalité est complexe. TBATS[5] est l'acronyme de :

- Saisonnalité trigonométrique
- Transformation de Box-Cox
- Une erreur RMSE
- Tendence
- Composantes saisonnières

2.4.1 Resultats obtenus avec Tbats

Les prédictions obtenus avec le modèle Tbats semble légèrement correctes comparé aux modèles Auto arima et prophet. Ce modèle fournit une erreur RMSE de prévision de 13.03 (Figure 9).

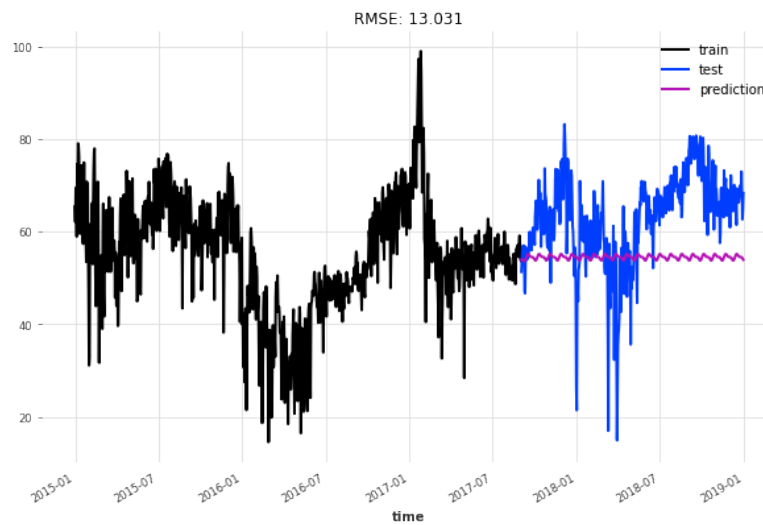


FIGURE 9 – Predictions by Tbats model

3 Prédiction à l'aide de modèles récents

3.1 Nbeats models

NBEATS [4], une architecture de réseau de neurones pour la prédiction de séries temporelles NBEATS est issu des recherches de Boris Oreshkin et de ses co-auteurs chez Element AI, malheureusement de courte durée. NBEATS est une étape intéressante dans l'application de l'apprentissage en profondeur aux séries chronologiques, car il crée une architecture dédiée aux séries chronologiques. NBEATS utilise une approche complètement différente : il prend une fenêtre entière de valeurs passées et calcule de nombreuses valeurs de points de temps de prédiction en une seule passe. Pour ce faire, il utilise largement des couches entièrement connectées.

3.1.1 Resultats obtenus avec Nbeats

Les prévisions obtenues avec ce modèle sont meilleurs par rapports aux modèles classiques sauf Xgboost. Nbeats fournit une erreur RMSE de 8.50% (Figure 10) qui est meilleur par rapport aux derniers.

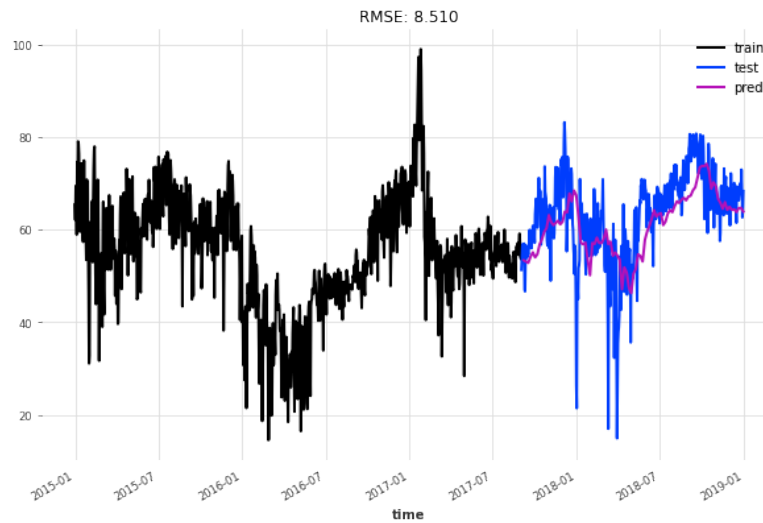


FIGURE 10 – Predictions by Nbeats model

3.2 Tft models

Le Temporal Fusion Transformer (TFT), une nouvelle architecture basée sur l'attention qui combine une prévision multi-horizon haute performance avec des informations interprétables sur la dynamique temporelle. Apprendre des relations temporelles à différentes échelles, le TFT utilise des couches récurrentes pour le traitement local et des couches d'auto-attention interprétables pour apprendre les dépendances à long terme. Le TFT utilise également des composants spécialisés pour la sélection judicieuse des fonctionnalités pertinentes et une série de couches de déclenchement pour supprimer les composants inutiles, permettant de donner des performances élevées dans une large gamme de régimes.

3.2.1 Resultats obtenus avec Tft

Les résultats obtenus avec le modèle Tft sont encore mieux avec une erreur de prévision RMSE de 7.12% (Figure 11) hormis Xgboost qui reste toujours plus performant. On voit qu'aussi Tft est un modèle qui controle bien les prévisions dans le temps avec des intervalles de confiance.

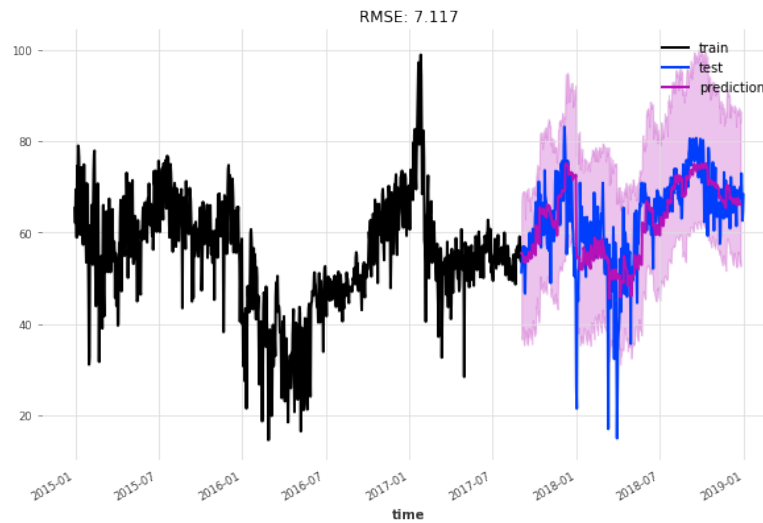


FIGURE 11 – Predictions by Tft model

3.3 DeepAR models

DeepAR [4] est un réseau neuronal récurrent (RNN) qui est un type de réseau neuronal bien adapté aux données de séries chronologiques. Les RNN traitent une série chronologique étape par étape, en maintenant un état interne d'un pas de temps à l'autre. DeepAR est entièrement récurrent dans le sens où, au moment de la prédiction, une sortie est calculée à l'aide de ces entrées :

- valeur cible précédente, qui sera définie sur la dernière valeur cible connue pour la première prédiction, et pour toutes les autres prédictions, elle sera définie sur la prédiction précédente (de manière auto-régressive),
- l'état caché précédent,
- les covariables au temps t pour prévoir la cible au temps t (si le modèle a été entraîné avec des covariables),

3.3.1 Resultats obtenus avec DeepAR

Le modèle fournit de très performances. Les prévisions sont contrôlées dans le temps avec des intervalles de confiances et que l'erreur RMSE obtenue avec ce modèle est 6.15% (Figure 12). C'est donc un intéressant modèle.

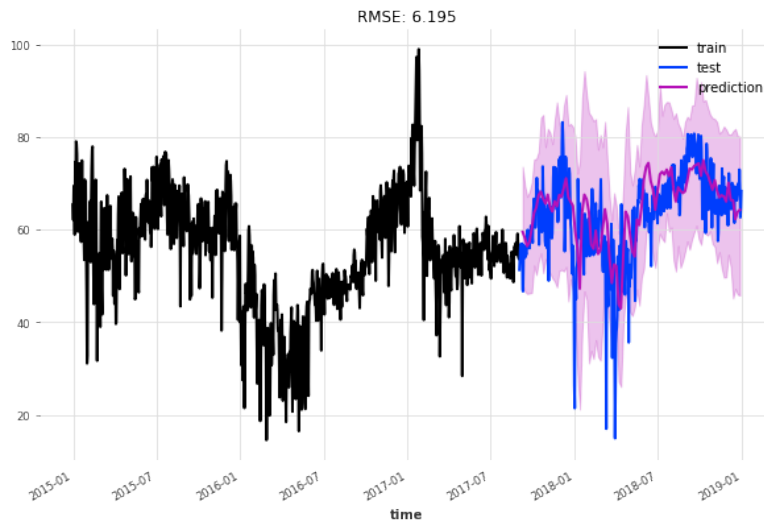


FIGURE 12 – Predictions by DeepAR model

4 Comparaison de Modèles

Les resultats de prévisions sur les modèles récents foournissent de meilleures per-
formances par rapport au modèles classiques. Et que DeepAR le modèle conçu sur
les rééseux recurents offre le meilleur resultat parmi tous modèles étudiés(voir Table
1 et Figure 13).

Modèles	Rmse	Time training(en seconde)
Auto arima	12.46	8.88
Fprophet	30.51	0.26
Xgboost	6.42	0.54
Tbats	13.03	42.68
Nbeats	8.51	21.74
Tft	7.11	23.74
DeepAR	6.20	15.74

TABLE 1 – Resultats metrics and times training

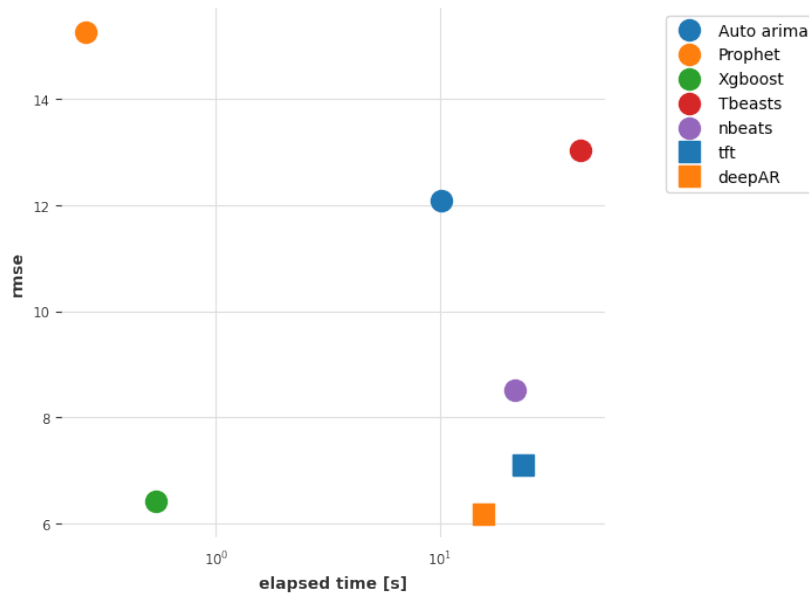


FIGURE 13 – plot models with rmse/training time

5 Conclusion

Les prévisions obtenues avec les méthodes récentes sont meilleures que celles des méthodes classiques. Cependant ces derniers sont moins couteuses et plus faciles à implémenter contrairement aux modèles récents qui sont très couteux et difficile à implémenter.

Maintenant, quelle méthode convient le mieux à votre cas ? Comme toujours, cela dépend. Si vous avez affaire principalement à des séries isolées qui ont un historique suffisant, les méthodes classiques comme Auto ARIMA vous permettront d'aller loin. Même sur des ensembles de données plus volumineux, si la puissance de calcul n'est pas trop un problème, ils peuvent représenter des options prêtes à l'emploi intéressantes. D'un autre côté, si vous avez affaire à un plus grand nombre de séries ou à des séries de dimensionnalités plus élevées, les méthodes ML et les modèles globaux seront souvent la voie à suivre. Ils peuvent capturer des modèles sur de larges plages de séries temporelles différentes et sont généralement plus rapides à exécuter. Si vous avez des raisons de croire que vous devez capturer des modèles plus complexes, ou si la vitesse d'inférence est vraiment importante pour vous, essayez les méthodes d'apprentissage en profondeur.

Références

- [1] Yoshua BORIS N. ORESHKIN DMITRI Carpov Nicolas Chapados. “NBEATS : Neural basis expansion analysis for interpretable time series forecasting”. In : *ICLR 2020 Conference Blind Submission* (2020). URL : <https://paperswithcode.com/paper/n-beats-neural-basis-expansion-analysis-for%7Chttps://openreview.net/forum?id=r1ecqn4YwB>.
- [2] Prosenjit CHAKRABORTY. *10 Time Series Forecasting Methods We Should Know*. Rapp. tech. 2021. URL : <https://cprosenjit.medium.com/10-time-series-forecasting-methods-we-should-know-291037d2e285>.
- [3] Jan Gasthaus DAVID SALINAS Valentin Flunkert. *DeepAR : Probabilistic Forecasting with Autoregressive Recurrent Networks*. Rapp. tech. 2019. URL : <https://paperswithcode.com/paper/deepar-probabilistic-forecasting-with%7Chttps://arxiv.org/abs/1704.04110v3>.
- [4] Julien HERZEN. *Entraînement de modèles de prévision sur plusieurs séries chronologiques avec des fléchettes*. URL : <https://unit8.com/resources/training-forecasting-models/>.
- [5] *Time Series Forecasting using TBATS Model*. 2021. URL : <https://medium.com/analytics-vidhya/time-series-forecasting-using-tbats-model-ce8c429442a9>.