Data challenge: Rapport final

Alexis Solar

10/11/2024

1 Introduction

Le marché SPOT est un marché boursier européen où l'électricité est achetée la veille pour le lendemain. Le marché Intraday, quant à lui, est également un marché boursier européen, mais où l'électricité est achetée le jour même, permettant d'ajuster l'offre en fonction des variations imprévues de la demande. Ce Data Challenge vise à modéliser, de façon supervisée, l'écart de prix entre ces deux marchés, par une régression ou une classification, car seul le sens de cet écart importe.

La métrique utilisée pour évaluer les performances est la "weighted accuracy" : il s'agit de la proportion des prédictions dont le sens (positif ou négatif) est correctement identifié, pondérée par la valeur absolue des écarts réellement observés. Cette métrique peut sembler moins intuitive qu'une simple accuracy au départ, mais elle permet de mettre l'accent sur l'importance de prédire correctement le sens de l'écart lorsque celui-ci est important. Cela est cohérent dans un contexte de passage d'ordres de marché, car il est bien plus crucial d'anticiper correctement les hausses ou baisses lorsque la volatilité est élevée, plutôt que lorsque celle-ci est faible.

2 Description et visualisation des données

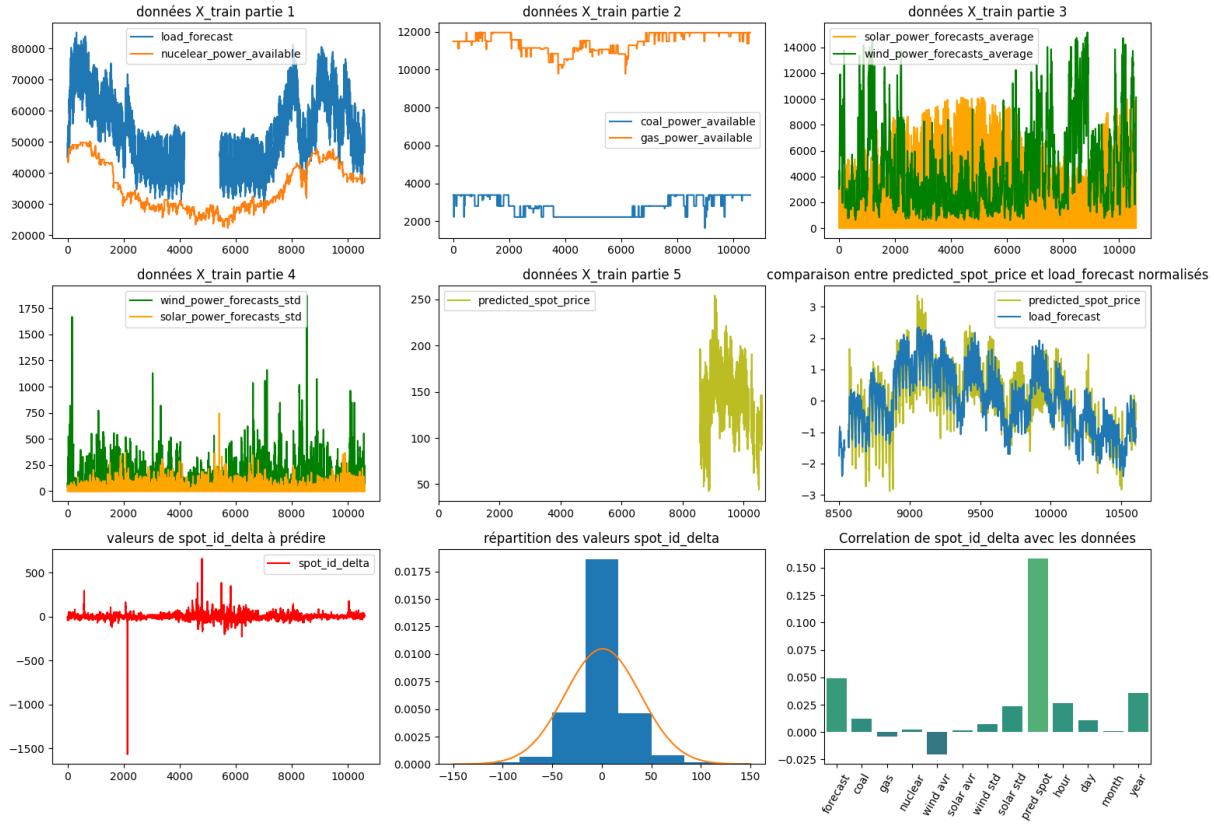

FIGURE 1 – Visualisation des données d'entrainement

Les données d'entrainement commencent le 01/01/2022 à 1h (heure UTC) et se terminent le 29/03/2023 à 21h. Afin de prédire correctement l'écart, on dispose de 11 variables explicatives : la prévision de consommation totale d'électricité en France, les capacités de production totale d'électricité des centrales à charbon, gaz et nucléaire, les moyennes de différentes prévisions de production totale d'électricité éolienne et solaire avec leur écart-type, la prévision du prix SPOT de l'électricité issues d'un modèle interne de Elmy, l'organisateur du Data Challenge et enfin les informations liées à la date : heure, jour, mois et année.

On observe que certaines variables sont liées les unes aux autres, qu'il y a également beaucoup de valeurs manquantes à traiter et que les corrélations sont assez faibles entre les variables explicatives et la variable à prédire. Les variables les mieux corrélés sont aussi celles qui possèdent le plus de valeur manquantes. Enfin les valeurs à prédire ont l'air d'être concentrées autour de 0 en général.

L'hypothèse de gaussianité n'étant pas valable pour les données de $X \mid sign(y)$ (test de Shapiro-Wilk), l'analyse linéaire ou quadritique discriminante sera écartée.

3 Evaluation des modèles

Évaluer correctement la performance d'un modèle est une tâche délicate car les données ne sont pas uniformément distribuées dans le cas d'une série temporelle. Les valeurs manquantes et la métrique de performance compliquent également l'évaluation. Plusieurs approches sont testées pour évaluer le modèle : une simple division du jeu d'entraînement en un ensemble d'entraînement et un ensemble de validation donnerait une idée trop biaisée de la performance du modèle selon les données sélectionnées. De plus, les données ne peuvent pas être sélectionnées aléatoirement, car il serait trop simple pour le modèle de prédire la valeur au temps n en connaissant les résultats aux temps $n-1$ et $n+1$, ce qui ne sera pas le cas dans la pratique.

Les données seront donc évaluées à l'aide de différentes divisions en K-fold, choisies de manière non aléatoire. Cela permet au modèle d'être évalué sur sa capacité à prédire certaines parties de la série temporelle, même si certaines données apprises peuvent se situer après la date à prédire. Pour pallier ce biais, une évaluation similaire sera réalisée en utilisant uniquement des K-folds basés sur les données passées. Cette approche n'est pas parfaite non plus, car la colonne "predicted spot price", la plus corrélée et essentielle à l'objectif, contient peu de données, principalement en fin de dataset. Il est donc difficile à la fois d'entraîner le modèle sur ces données et de l'évaluer correctement dessus.

4 Approche naïve

Ma première approche a été de commencer par tester les performances obtenables sans pré-traitement avec des modèles capables d'utiliser des données brutes afin de ne pas apporter de biais dans les données. De l'optimisation d'hyper-paramètre a aussi été réalisé ensuite pour améliorer les performances avec Grid Search ou optuna pour l'optimisation bayésienne.

J'ai ainsi testé trois méthodes de boosting : XGBoost Regressor, LightGBM Regressor, et CatBoost Regressor. Les méthodes de boosting sont assez robustes et obtiennent généralement de bons résultats en s'adaptant à une grande variété de problèmes de machine learning. C'est également le cas ici, car mes meilleures performances ont été obtenues avec XGBoost Regressor sans pré-traitement, surpassant même tous mes autres modèles avec pré-traitement. J'ai ensuite optimisé les hyperparamètres de ce modèle en utilisant les méthodes d'évaluation précédemment présentées. La recherche de la meilleure méthode d'optimisation a également été expérimentale, car le modèle finit par overfit le critère d'évaluation, réduisant ainsi sa capacité de généralisation sur X test.

5 Traiement des valeurs manquantes

Après avoir constaté de très bonnes performances de XGBoost sans pré-traitement, j'ai exploré plusieurs options de pré-traitement, en me concentrant principalement sur la gestion des valeurs manquantes :

```
X_train.isna().sum()
✓ 0.0s
load_forecast                  1287
coal_power_available              1
gas_power_available               1
nucelear_power_available          1
wind_power_forecasts_average     24
solar_power_forecasts_average    24
wind_power_forecasts_std         24
solar_power_forecasts_std        24
predicted_spot_price           8759
hour                              0
day                               0
month                             0
year                              0
```

```
X_test.isna().sum()
✓ 0.0s
load_forecast                     0
coal_power_available              0
gas_power_available               0
nucelear_power_available          0
wind_power_forecasts_average      0
solar_power_forecasts_average    24
wind_power_forecasts_std          0
solar_power_forecasts_std        24
predicted_spot_price           1536
hour                              0
day                               0
month                             0
year                              0
```

FIGURE 2 – Valeurs manquantes

- Une seule ligne du dataset contient des valeurs manquantes pour "coal power available", "gas power available" et "nuclear power available". La solution la plus simple est donc de supprimer cette ligne de X train.

- Les 24 valeurs manquantes pour "wind power" et "solar power" peuvent également être retirées, compte tenu de la taille du dataset, ou remplacées par des valeurs similaires. L'impact de ces valeurs manquantes reste relativement faible. Cependant, il est essentiel de traiter les valeurs manquantes dans X test, qui correspondent à des pics solaires, celles-ci peuvent donc être remplacées de manière cohérente avec les données disponibles.

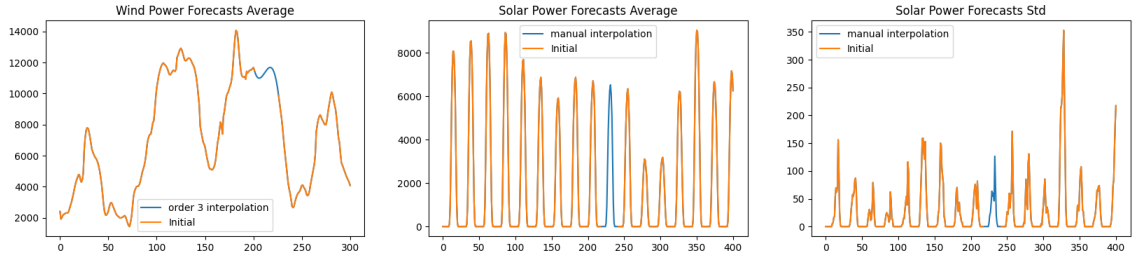

FIGURE 3 – remplacement des valeurs manquantes

Il reste donc à traiter les deux variables explicatives les plus importantes et comportant le plus de valeurs manquantes.

- Pour les 1287 valeurs manquantes de "load forecast", j'ai d'abord essayé d'interpoler les valeurs manquantes à partir des données réelles de consommation d'électricité renormalisées ces jours là, récupérées sur le site de RTE (figure 4). Cette approche n'a cependant pas donné de résultats satisfaisants. J'ai donc décidé de simplement retirer ces lignes afin de minimiser le biais dans les données, d'autant plus que ces valeurs ne sont pas manquantes dans le jeu de test.

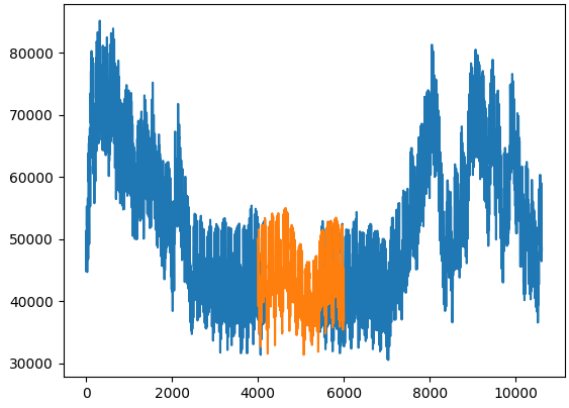

FIGURE 4 – tentative d'interpolation pour load forecast

- Enfin la partie la plus cruciale est "predicted spot price" avec encore 7471 valeurs manquantes à ce stade. J'ai donc décidé dans un premier temps de mettre les valeurs manquantes à 0 et d'ajouter une colonne "is predicted available" binaire pour que le modèle exploite cette information lui même et dans un second temps de complêtement retirer les 8500 premières valeurs, ce qui réduit considérablement le dataset mais introduit moins de biais.

J'ai testé plusieurs modèles avec ces deux approches de pré-traitement. Bien qu'ils constituent les meilleurs pré-traitements que j'ai pu trouver, ils ne permettent pas d'atteindre les performances du modèle XGBoost initial. Sur ces deux versions de X train, j'ai entraîné différents modèles de boosting, des Random Forest, Random Trees, SVR, ainsi que des modèles de régression tels que Ridge, Lasso, ElasticNet, BayesianRidge, et d'autres modèles comme BaggingRegressor et KNeighborsRegressor.

J'ai également essayé des approches de classification, car la plupart de ces modèles ont un équivalent pour la classification. Cependant, je n'ai pas obtenu de performances significativement meilleures avec les classifieurs par rapport aux régressions.

6 Conclusion

Après de nombreux tests de pré-traitement et d'essais de différents modèles, mon meilleur résultat est finalement obtenu avec XGBoost sans pré-traitement, en utilisant une optimisation bayésienne des hyperparamètres et une évaluation basée sur une validation croisée par séries temporelles à 10 parties ("Time Series Cross Validation") pour maximiser la performance. C'était déjà mon meilleur modèle a mi-parcours avec un score public de 0,5642 et privé de 0,5824, je l'ai ensuite encore amélioré en optimisant les hyper-paramètres jusqu'à obtenir un score public de 0,5873.

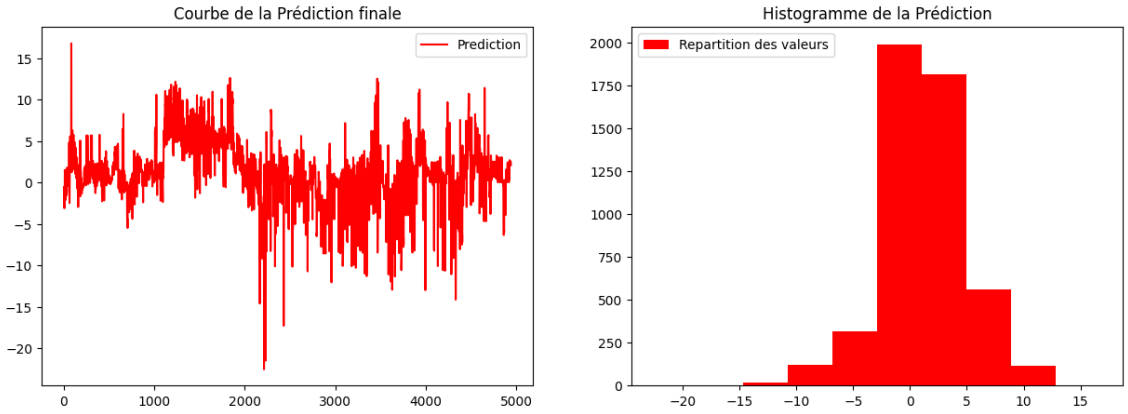

FIGURE 5 – Prédiction finale de y test

En revanche, ce modèle présente des limites. Les données sont faiblement corrélées et contiennent souvent des valeurs manquantes, ce qui complique l'apprentissage. Le modèle lui-même n'est probablement pas le plus adapté pour ce type de tâche complexe : il est difficile d'éviter le surajustement même en appliquant une régularisation. Par ailleurs, la métrique de performance, qui met l'accent sur les données bien classées pondérées par leur valeur, ajoute beaucoup de bruit dans les résultats. Certaines valeurs extrêmes influencent fortement la métrique, ce qui peut induire des confusions. De manière générale, il est difficile d'affirmer qu'un meilleur modèle produira une meilleure précision pondérée (weighted accuracy) et, inversement, qu'une meilleure précision pondérée est toujours le signe d'un modèle de meilleure qualité.