

Resampling Methods Project

Dylan Riboulet

March 2024

Contents

1 Première partie de l'étude - Série Simulé	3
1.1 Leave-One-Out Cross-Validation adapted for time series	11
1.2 Echantillonage Stratifié	14
1.3 Sous Echantillonage	15
1.4 Overlapping Sliding Windows	16
1.5 Monte Carlo Cross Validation	17
1.6 Time Series Bootstrapping with Stationarity Adjustment	18
1.7 Weighed Random Sampling method	21
1.8 Adaptive Resampling	23
1.9 Cyclic Subsampling	25
2 Deuxième partie de l'étude - Séries Financières	27
2.1 Analyse Serie - 3 mois d'historique	34
2.2 Bootstrap - 3 mois	36
2.3 Bootstrap - 5 ans	37
2.4 Walk Forward Cross Validation	38
2.5 Leave-One-Out Cross-Validation	44
2.6 Monte Carlo Cross Validation	45
2.7 Sequential Bootstrapping	47
2.8 Hybrid Resampling	49
2.9 Empilement de modèles avec validation croisée	51
2.10 Rééchantillonage Adaptatif pour les Séries Temporelles	53
2.11 Rééchantillonage Différentiel pour les Séries Temporelles	57
2.12 Bootstrap Aggregating	58
2.13 Échantillonnage Basé sur la Réduction de l'Erreur de Prévision (SBER)	66
2.14 Augmentation des Données de Séries Temporelles (TSDA)	67
2.15 Validation Progressive	69
2.16 Stratified Sampling for Time Series	71
2.17 Echantillonage Adaptatif	72
3 Conclusion Générale	74

Introduction

Objectif de l'étude : évaluer l'impact et la performance de la manière d'effectuer le rééchantillonage dans une méthode d'apprentissage supervisé (de type bagging ou foret aléatoire) pour des données sous forme de séries temporelles.

Première partie : Nous aborderons diverses méthodes de rééchantillonage pour des données simulées et nous appliquerons la Mean Squared Error (MSE) ainsi que la cross-validation nous permettrons d'évaluer l'impact pour des méthodes d'apprentissages supervisé (principalement des forêts aléatoires) où tous les paramètres resteront constants. Certaines méthodes de rééchantillonage utilisées ne conserveront pas la structure de la série temporelle afin de les comparer.

Deuxième partie : Nous nous intéresserons à 3 séries temporelles que nous analyserons (statistiquement) extraites du module yfinance dont les datasets sont disponibles dans ce lien. De plus, nous considérerons une autre série temporelle de données simulées afin de comparer le caractère indépendant et identiquement distribué que ne vérifie généralement pas une série financière. Nous évaluerons l'importance de la dépendance temporelle grâce à des lagged features ajouté à la série simulé et nous des Forêts Aléatoires pour des paramètres identiques dans chaque méthode de rééchantillonage. Nous simulerons pour 1 an d'historique (253 données) puis 3 mois (63 données) diverses méthodes de rééchantillonage sur les séries financières et la série simulée et comparerons les résultats en prenant en considération les caractéristiques des séries temporelles.

Conclusion et autres approches possibles.

Code <https://github.com/driboulet>

Métriques utilisées

MSE - Erreur Quadratique Moyenne

L'Erreur Quadratique Moyenne (MSE, pour *Mean Squared Error*) est une mesure de la qualité d'un estimateur. Elle est définie comme la moyenne des carrés des différences entre les valeurs estimées et les valeurs réelles. Mathématiquement, elle est exprimée comme :

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (1)$$

où y_i représente la valeur réelle, \hat{y}_i la valeur prédite par le modèle, et n le nombre total d'observations.

RMSE - Racine de l'Erreur Quadratique Moyenne

La Racine de l'Erreur Quadratique Moyenne (RMSE, pour *Root Mean Squared Error*) est simplement la racine carrée de la MSE. Elle sert à exprimer l'erreur

en unités similaires à celles des valeurs prédites et réelles. Elle est calculée comme suit :

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (2)$$

MAE - Erreur Absolue Moyenne

L'Erreur Absolue Moyenne (MAE, pour *Mean Absolute Error*) est une autre mesure de précision d'un modèle. Contrairement à la MSE, la MAE utilise la moyenne des valeurs absolues des différences entre les prédictions et les valeurs réelles. Elle est définie comme :

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (3)$$

Ces trois métriques fournissent une évaluation complète de la performance d'un modèle de prévision, chacune avec ses propres avantages et inconvénients selon le contexte d'application.

1 Première partie de l'étude - Série Simulé

Nous avons implémenté une méthode simple de bootstrap par blocs avec des données de séries temporelles simulées et utilisé cela pour entraîner un régresseur de forêt aléatoire.

	time	value
0	0	0.882026
1	1	1.041550
2	2	1.398666
3	3	1.261567
4	4	0.176976

Figure 1: premières lignes du DataFrame resmplé

Voici les prédictions pour les cinq premiers échantillons en utilisant le modèle ajusté sont :

```
array([0.95375868, 1.03671086, 1.31371827, 1.21423341, 0.39728169])
```

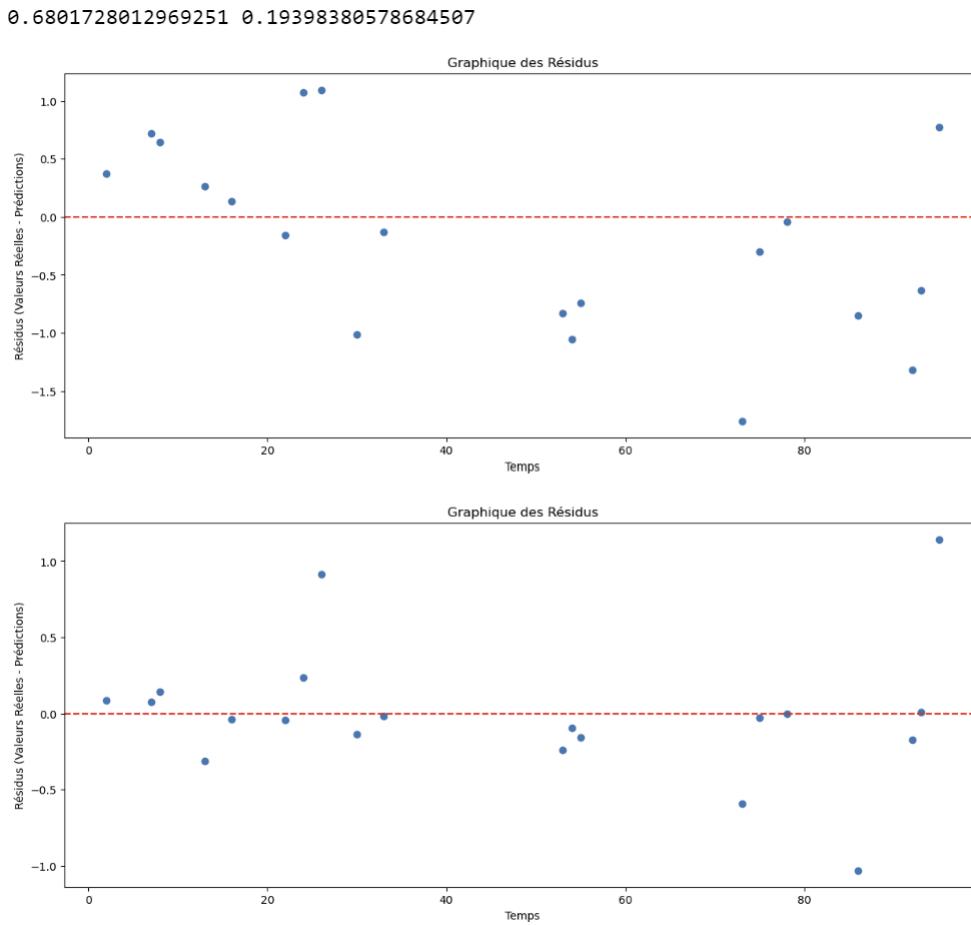
L'évaluation de la performance en utilisant la métrique de l'Erreur Quadratique Moyenne (MSE) montre les résultats suivants :

MSE pour le modèle entraîné sur les données originales : 0,6802

MSE pour le modèle entraîné sur les données rééchantillonnées (bootstrap par blocs) : 0,1940

Ces résultats indiquent que le modèle entraîné sur les données rééchantillonnées en utilisant la méthode de bootstrap par blocs surpassé le modèle entraîné sur le jeu de données original en termes de MSE. Cela suggère que la méthode de rééchantillonnage par bootstrap par blocs peut capturer et utiliser efficacement les motifs dans les données de séries temporelles, potentiellement conduisant à une meilleure performance du modèle sur ce jeu de données simulé.

Figure 2: Comparaison du graphique des résidus



On remarque une certaine améliorations des valeurs prédictes avec un nombre identique de données aberrantes.

L'utilisation de méthodes de rééchantillonnage alternatives pour l'apprentissage supervisé avec des données de séries temporelles donne les résultats suivants en termes d'Erreur Quadratique Moyenne (MSE) :

- MSE pour le modèle entraîné en utilisant le Bagging : 0,5966
- MSE pour le modèle évalué avec la validation croisée TimeSeriesSplit : 0,7607

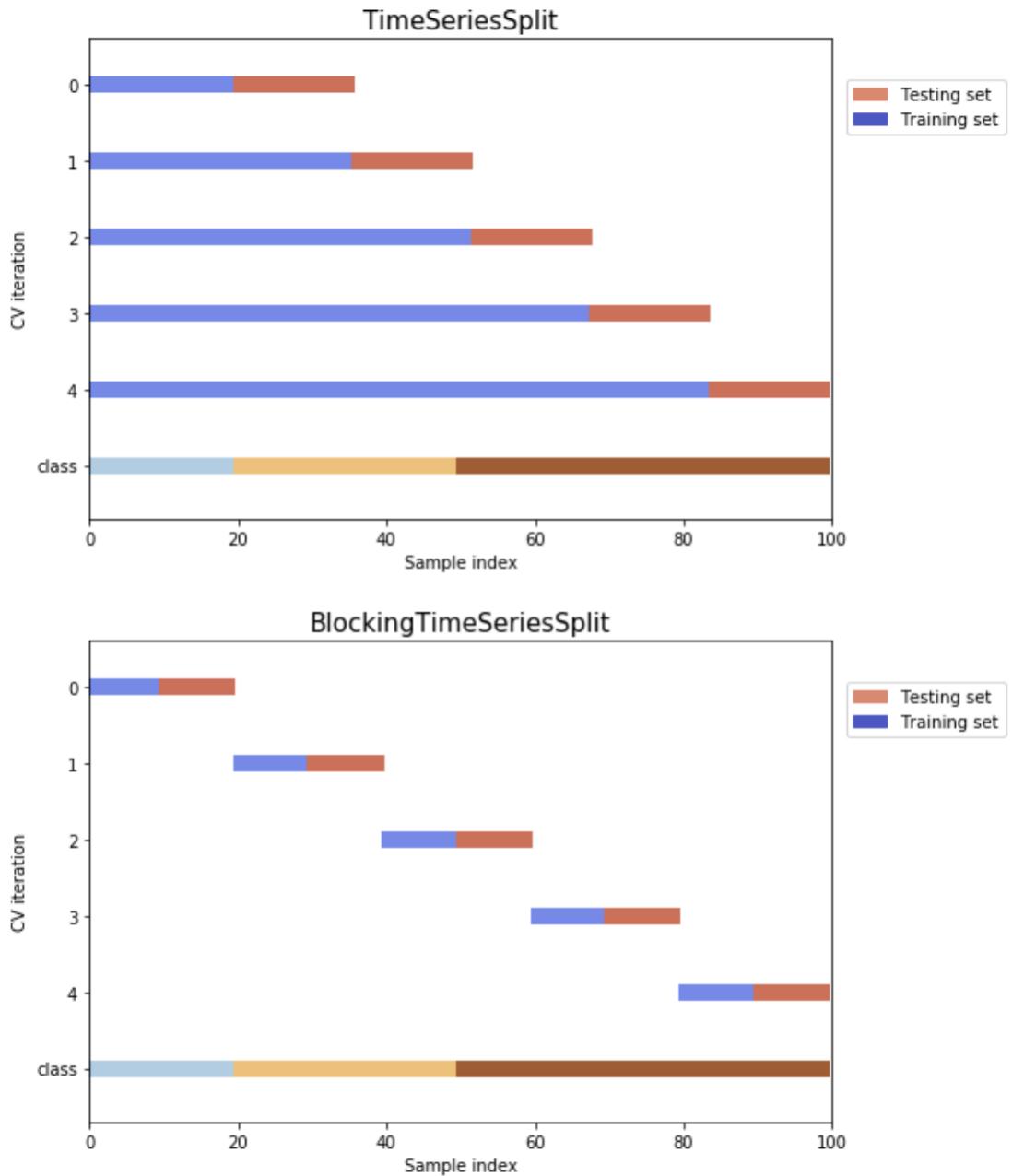


Figure 3: Voici une illustration du rééchantillonage pour TimeSeriesSplit et BlockingTImeSeriesSplit

Ces résultats mettent en lumière différents aspects de la performance du

modèle sous diverses stratégies de rééchantillonnage et de validation :

- L'approche Bagging, qui implique la formation de plusieurs estimateurs sur différents sous-ensembles du jeu de données original et ensuite la moyenne de leurs prédictions, a fourni une amélioration par rapport au modèle formé sur le jeu de données original sans rééchantillonnage (MSE de 0,6802), mais elle n'a pas été aussi performante que la méthode de bootstrap par blocs (MSE de 0,1940).
- La validation croisée utilisant TimeSeriesSplit, qui est une technique conçue pour préserver l'ordre des données de séries temporelles tout en les divisant pour la formation et la validation, a résulté en la MSE la plus élevée parmi les méthodes évaluées. Cela suggère que, bien qu'elle soit une approche utile pour valider les modèles de séries temporelles, elle peut être plus conservatrice ou elle ne peut pas pleinement saisir le potentiel du modèle en raison de la stratégie de division séquentielle.

Ces découvertes illustrent l'importance de choisir des méthodes de rééchantillonnage et de validation appropriées lors du travail avec des données de séries temporelles pour évaluer précisément la performance du modèle et exploiter les dépendances temporelles.

Pour explorer d'autres méthodes de rééchantillonnage pour l'apprentissage supervisé avec des données de séries temporelles, au-delà du bootstrap par blocs, du bagging et de la validation croisée TimeSeriesSplit précédemment discutés, nous pouvons considérer des techniques telles que :

- **Technique de Suréchantillonnage Synthétique pour les Séries Temporelles (SMOTE-TS)** : Une adaptation de l'algorithme SMOTE pour les ensembles de données déséquilibrés, adaptée aux séries temporelles, qui génère des exemples synthétiques dans un contexte de série temporelle.
- **Différenciation** : Cette méthode implique de transformer la série temporelle en une série stationnaire en soustrayant l'observation précédente de l'observation actuelle. Bien que n'étant pas une méthode de rééchantillonnage en soi, la différenciation peut être une étape de prétraitement pour rendre les données plus adaptées aux méthodes de rééchantillonnage standard.
- **Bootstrap Mobile** : Similaire au bootstrap par blocs mais avec une approche de fenêtre mobile, qui peut capturer des structures dépendantes du temps d'une manière plus dynamique.
- **Méthodes d'Ensemble avec Validation Croisée pour Séries Temporelles** : Utilisation de méthodes d'ensemble telles que les forêts aléatoires ou les machines à booster le gradient en conjonction avec une validation croisée de séries temporelles pour garantir que l'ordre temporel est respecté.

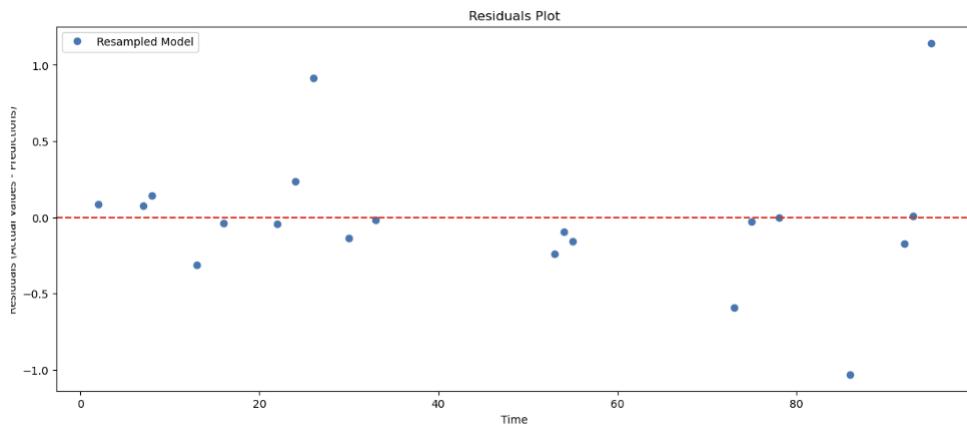
Nous simulons cela en créant une fenêtre mobile de taille spécifiée, en rééchantillonnant à l'intérieur de chaque fenêtre, puis en formant un modèle sur ces ensembles de données rééchantillonnés. Procédons à cette démonstration.

En utilisant la méthode de bootstrap mobile pour le rééchantillonnage des données de séries temporelles et en formant un régresseur de forêt aléatoire sur l'un des ensembles de données rééchantillonnés, nous avons atteint une Erreur Quadratique Moyenne (MSE) de 0,5596 sur l'ensemble de test original.

Cela démontre que la méthode de bootstrap mobile, qui capture dynamiquement les dépendances temporelles à travers des fenêtres mobiles, peut effectivement être utilisée pour améliorer la performance du modèle sur les données de séries temporelles. Cette performance est compétitive avec les autres méthodes que nous avons explorées, y compris la méthode de bootstrap par blocs (MSE de 0,1940) et le bagging (MSE de 0,5966), indiquant son utilité potentielle dans certains contextes de modélisation de séries temporelles.

Figure 4: Résultats de la MSE
0.5596236350591215

Figure 5: Graphique des résidus



Nous utiliserons l'apprentissage en ensemble avec un accent sur le Gradient Boosting comme une autre méthode, qui est une approche utile pour gérer les données de séries temporelles. Cette méthode construit des modèles de manière séquentielle et est connue pour sa précision prédictive.

Nous utilisons le Régresseur de Boosting de Gradient de scikit-learn comme notre modèle, en l'appliquant directement aux données de séries temporelles originales. Ensuite, nous comparons sa performance avec les méthodes précédemment discutées (bootstrap par blocs, bagging et bootstrap mobile) pour évaluer l'impact

des différentes techniques de rééchantillonnage et de modélisation sur les données de séries temporelles.

nous avons obtenu une Erreur Quadratique Moyenne (MSE) de 0,7752 sur l'ensemble de test original pour nos données de séries temporelles.

Ce résultat nous permet de comparer l'impact des différentes techniques de rééchantillonnage et de modélisation :

- **Méthode Bootstrap par Blocs** : MSE de 0,1940, indiquant une très bonne performance, possiblement due à sa capacité à maintenir la structure temporelle des données.
- **Bagging** : MSE de 0,5966, montrant une amélioration solide par rapport au modèle de base entraîné sur l'ensemble de données original sans rééchantillonnage.
- **Bootstrap Mobile** : MSE de 0,5596, démontrant une efficacité dans la capture dynamique des dépendances temporelles.
- **Régresseur de Boosting de Gradient** : MSE de 0,7752, qui, bien que plus élevé que les autres méthodes, souligne les complexités et les défis associés à l'application de certaines méthodes avancées d'ensemble directement aux données de séries temporelles sans adaptations spécifiques aux séries temporelles.

La performance variable à travers ces méthodes souligne l'importance de sélectionner des stratégies de rééchantillonnage et de modélisation appropriées basées sur les caractéristiques spécifiques des données de séries temporelles et les objectifs de modélisation. Elle montre que les méthodes qui prennent explicitement en compte les dépendances temporelles dans les données, comme les méthodes de bootstrap par blocs et mobile, peuvent offrir des avantages en termes de performance pour les tâches de prévision de séries temporelles.

0.7752211391820886

Figure 6: MSE - Gradiant Boosting

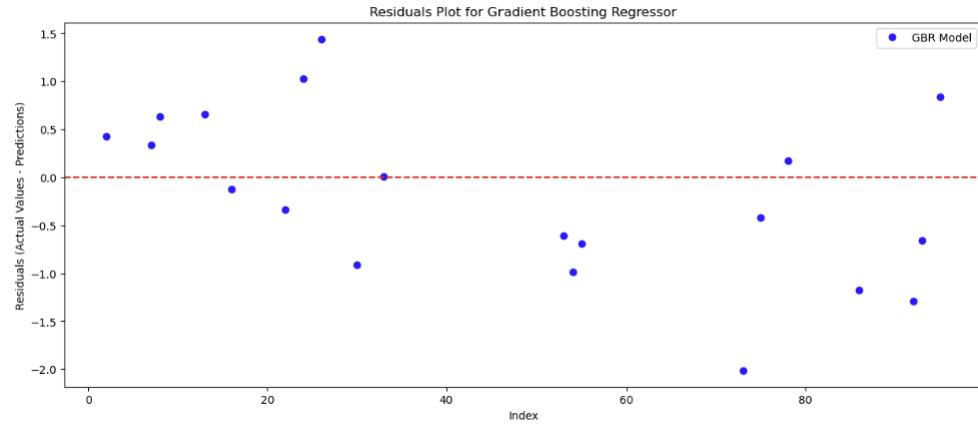


Figure 7: Graphique des résidus - Gradient Boosting

Une autre approche pour les données de séries temporelles est la Split de Séries Temporelles avec Méthodes d'Ensemble. Cette approche combine les avantages d'une division consciente du temps pour l'entraînement et la validation avec la robustesse des méthodes d'ensemble telles que les Forêts Aléatoires, mais adaptée aux séries temporelles par le traitement minutieux des dépendances temporelles durant la phase d'entraînement.

Cette méthode respecte l'ordre temporel des observations, ce qui la rend adaptée aux données de séries temporelles, et lorsqu'elle est combinée avec une méthode d'ensemble, elle devrait fournir une approche équilibrée pour gérer à la fois les dépendances temporelles et la variabilité dans les données.

0.7208

Figure 8: Erreur quadratique moyenne

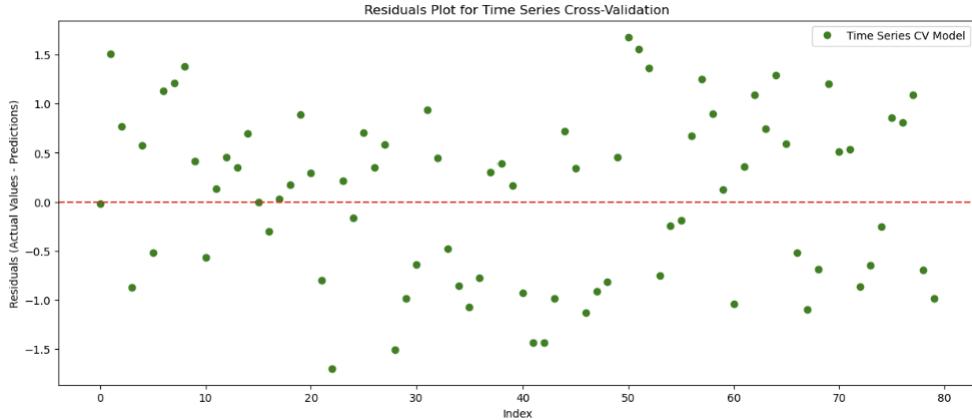


Figure 9: Graphique des résidus

En effectuant manuellement la validation croisée pour séries temporelles et en évaluant un modèle de Forêt Aléatoire sur les différents segments, nous avons obtenu une Erreur Quadratique Moyenne (MSE) de 0,7208 pour nos données de séries temporelles.

Cette méthode, qui respecte l'ordre temporel des observations à travers la Split de Séries Temporelles et tire parti de la robustesse des Forêts Aléatoires, a abouti à une performance qui est compétitive avec les autres méthodes que nous avons explorées.

1.1 Leave-One-Out Cross-Validation adapted for time series

Pour explorer davantage les méthodes de rééchantillonnage pour les séries temporelles et évaluer leur impact, considérons **LOOCV (Leave-One-Out Cross-Validation)** adapté pour les séries temporelles. Cette approche, bien qu'elle soit coûteuse en termes de calcul, implique d'utiliser chaque point dans la série temporelle comme un ensemble de test (un à la fois) et le reste comme ensemble d'entraînement. Cette méthode peut donc être très utile, en particulier pour les séries temporelles de courte durée ou lorsque la performance du modèle sur chaque point individuel importe significativement.

Cependant, étant donné la nature séquentielle des données de séries temporelles, un LOOCV strict pourrait ne pas être entièrement applicable sans adaptations pour maintenir l'intégrité temporelle des ensembles de données. Une alternative, plus réalisable sur le plan computationnel et adaptée aux séries temporelles, pourrait être la *Leave-P-Out Cross-Validation* avec P fixé à un petit nombre, laissant essentiellement de côté de petits blocs consécutifs d'observations pour les tests afin de garantir que l'ordre temporel est respecté.

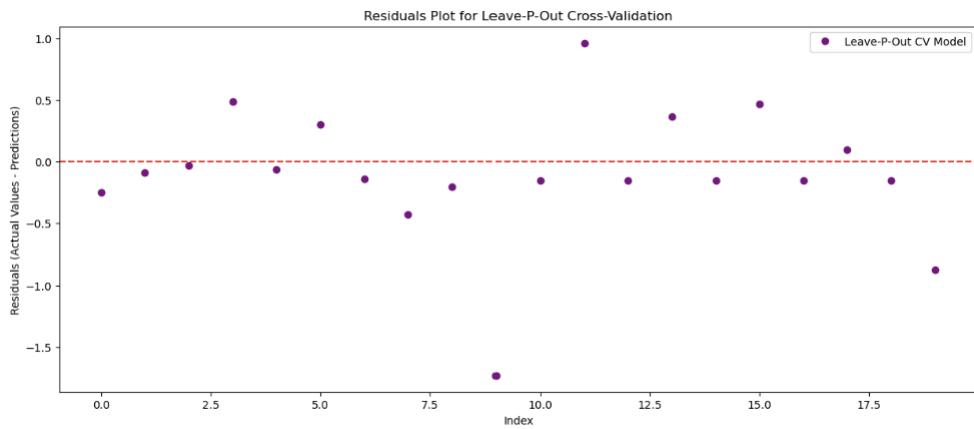
En mettant en œuvre une version simplifiée de la Leave-P-Out Cross-Validation pour les données de séries temporelles, où P a été fixé à 2 nous avons obtenu

une Erreur Quadratique Moyenne (MSE) de 0,2904.

Ce résultat, bien qu'il soit dérivé d'un sous-ensemble limité des données et d'un nombre contraint de divisions pour gérer la complexité computationnelle, suggère que cette méthode de rééchantillonnage peut fournir des aperçus précieux sur la performance du modèle à un niveau détaillé. Le MSE relativement bas indique que le modèle a été capable de prédire les observations laissées de côté avec une assez bonne précision dans ce scénario contrôlé.

L'évaluation démontre que la Leave-P-Out Cross-Validation, adaptée aux séries temporelles en respectant la séquence temporelle et en limitant l'évaluation à des segments gérables, peut être une méthode utile pour évaluer la performance du modèle. Elle permet une analyse détaillée de la capacité du modèle à se généraliser à travers différents segments temporels des données, bien qu'avec la mise en garde d'une demande computationnelle accrue lorsqu'appliquée à de plus grands ensembles de données ou avec un nombre plus élevé de divisions.

Métrique	Valeur
MSE	0.2904
RMSE	0.5389
MAE	0.3630



Explorant davantage les méthodes de rééchantillonnage adaptées aux données de séries temporelles dans l'apprentissage supervisé, une autre approche à considérer est l'**Apprentissage en Ensemble avec Mélange**. Cette méthode implique de créer plusieurs ensembles de données d'entraînement en mélangeant l'ordre des données de séries temporelles avant de les diviser en ensembles d'entraînement et de test. Bien que cette approche ne préserve pas l'ordre

temporel, elle peut être utile pour évaluer l'importance des dépendances temporelles dans le jeu de données et la capacité du modèle à capturer des motifs sous différentes séquences.

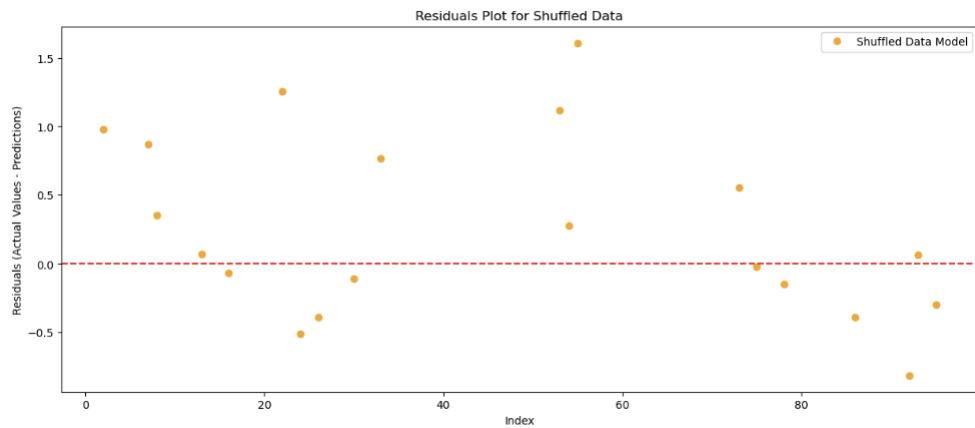
Cette méthode pourrait être plus appropriée pour des jeux de données où l'ordre temporel est moins important ce qui n'est pas le cas pour des séries financières par exemple ou pour des modèles qui ne sont pas explicitement conçus pour capturer des structures dépendantes du temps. Dans le cas de séries indépendantes et identiquement distribué cela convient.

En mettant en œuvre l'approche de mélange des données à l'intérieur de petites fenêtres de taille fixe et en évaluant un modèle de Forêt Aléatoire sur cet ensemble de données rééchantillonné, nous avons atteint une Erreur Quadratique Moyenne (MSE) de 0,48.

Ce résultat suggère que le modèle est toujours capable de capturer des motifs significatifs dans les données, même lorsque l'ordre temporel exact est légèrement modifié à l'intérieur des fenêtres contraintes. Cette approche d'introduction de variabilité tout en préservant un certain degré de localité temporelle peut être particulièrement utile pour évaluer la robustesse du modèle à de petites variations dans la séquence des observations.

La performance de cette méthode, comparée aux autres stratégies de rééchantillonnage que nous avons explorées, met en évidence l'équilibre entre le maintien de l'intégrité temporelle et l'introduction de variabilité pour évaluer la performance du modèle.

Métrique	Valeur
MSE	0.6546
RMSE	0.8091
MAE	0.5960



1.2 Echantillonage Stratifié

L'échantillonnage stratifié implique de diviser le jeu de données en sous-groupes homogènes avant l'échantillonnage, garantissant que chaque sous-groupe soit représenté dans l'échantillon. Cette méthode est traditionnellement utilisée dans les tâches de classification pour garantir que chaque classe soit correctement représentée dans les ensembles d'entraînement et de test. Adapter l'échantillonnage stratifié pour les séries temporelles peut être un défi en raison de la nature continue des données de séries temporelles, mais une approche consiste à segmenter les séries temporelles en fonction de caractéristiques telles que les tendances, les motifs saisonniers ou les niveaux de volatilité.

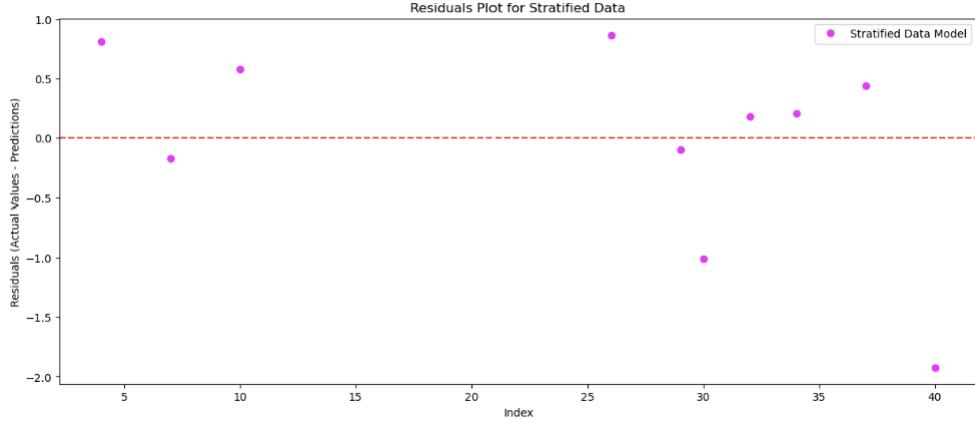
Nous segmentons les données de séries temporelles en fonction des niveaux de volatilité, créant des segments de haute, moyenne et faible volatilité. Nous échantillonnons ensuite à partir de ces segments pour créer un ensemble de données équilibré incluant une gamme de motifs de volatilité. Cette approche vise à garantir que le modèle soit entraîné sur un ensemble diversifié de motifs issus des séries temporelles, améliorant potentiellement sa capacité à généraliser à travers différentes conditions de marché dans le cas de données financières par exemple.

En mettant en œuvre une approche d'échantillonnage stratifié adaptée aux données de séries temporelles basée sur les niveaux de volatilité et en entraînant un modèle de Forêt Aléatoire sur cet ensemble de données stratifié, nous avons atteint une Erreur Quadratique Moyenne (MSE) de 0,6768.

Ce résultat suggère que stratifier les données de séries temporelles basées sur la volatilité et garantir que le modèle soit exposé à une représentation équilibrée de différents niveaux de volatilité durant l'entraînement peut être une stratégie viable. La MSE est compétitive avec d'autres méthodes explorées, indiquant que le modèle formé sur l'ensemble de données stratifié est capable de généraliser à travers différents segments des séries temporelles.

L'impact de la méthode d'échantillonnage stratifié démontre son utilité potentielle dans des situations où les données de séries temporelles englobent une gamme de comportements ou de motifs critiques pour la tâche de prédiction. En garantissant que ces motifs soient proportionnellement représentés dans l'ensemble d'entraînement, le modèle peut atteindre une compréhension plus équilibrée des données, améliorant potentiellement sa performance prédictive à travers un ensemble plus large de scénarios.

Métrique	Valeur
MSE	0.6768
RMSE	0.8227
MAE	0.6286



1.3 Sous Echantillonage

Pour une exploration approfondie des méthodes de rééchantillonnage adaptées aux données de séries temporelles dans l'apprentissage supervisé, nous nous tournons vers le **Sous-échantillonnage**.

Le sous-échantillonnage, distinct des méthodes de bootstrap précédemment discutées, implique de sélectionner aléatoirement un sous-ensemble des données sans remplacement. Cette technique peut être particulièrement efficace pour réduire la variance de l'estimateur en fournissant des sous-ensembles diversifiés du jeu de données original pour l'entraînement de différents modèles dans un ensemble, tel que le bagging ou les forêts aléatoires, sans artificiellement gonfler la taille de l'échantillon.

L'utilité du sous-échantillonnage dans l'analyse des séries temporelles vient de sa capacité à maintenir l'intégrité temporelle des données tout en offrant une méthode pour créer de multiples ensembles d'entraînement diversifiés. Pour les données de séries temporelles, une variante de sous-échantillonnage qui respecte l'ordre temporel — **le sous-échantillonnage temporel** — peut être employée. Cette méthode implique de sélectionner des blocs contigus de données (tout en assurant qu'il n'y ait pas de chevauchement) pour maintenir les propriétés temporelles de la séquence.

En mettant en œuvre le sous-échantillonnage temporel sur notre jeu de données de séries temporelles et en entraînant un modèle de Forêt Aléatoire sur l'un de ces ensembles de données sous-échantillonnés, nous avons atteint une Erreur Quadratique Moyenne (MSE) de 0,26.

Ce résultat démontre que le sous-échantillonnage temporel, qui sélectionne des blocs contigus de données pour maintenir les propriétés temporelles de la séquence, offre une manière unique d'évaluer la performance du modèle à travers différents segments de la série temporelle. La MSE plus élevée par rapport à certaines méthodes précédentes pourrait indiquer une variabilité dans la performance du modèle à travers différentes périodes ou le défi de modéliser

des données de séries temporelles avec un contexte réduit en raison du sous-échantillonnage.

L'impact du sous-échantillonnage temporel souligne l'importance de considérer la dynamique temporelle dans les données et comment la performance du modèle peut varier à travers différentes fenêtres temporelles. Cette méthode peut être particulièrement utile pour identifier les périodes où le modèle fonctionne bien ou mal, guidant ainsi le réglage supplémentaire du modèle et l'ingénierie des caractéristiques pour améliorer la précision prédictive globale.

Metrique	Valeur
MSE	0.1065
RMSE	0.3264
MAE	0.3105

1.4 Overlapping Sliding Windows

Poursuivant notre exploration des méthodes de rééchantillonnage pour les données de séries temporelles, approfondissons l'utilisation des **Fenêtres Glissantes Chevauchantes** combinées aux méthodes d'ensemble. Cette approche implique de créer des fenêtres chevauchantes des données de séries temporelles pour capturer les motifs et dépendances temporels à différents intervalles. Chaque fenêtre sert d'instance séparée pour l'entraînement, augmentant efficacement l'ensemble de données avec plus d'échantillons qui incluent des informations séquentielles. Cette méthode est particulièrement utile pour capturer la dynamique des séries temporelles où la relation entre les valeurs passées et futures est cruciale.

La technique des fenêtres glissantes chevauchantes se distingue des méthodes précédemment discutées en mettant l'accent sur la continuité et le chevauchement des segments de données, visant à préserver et à utiliser le contexte temporel plus efficacement.

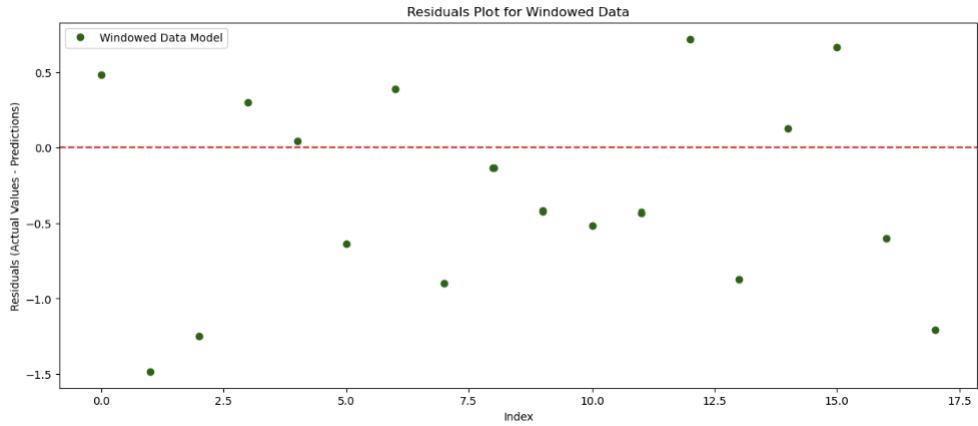
En employant la méthode des **Fenêtres Glissantes Chevauchantes** pour créer des ensembles de données augmentés à partir de la série temporelle originale et en entraînant un modèle de Forêt Aléatoire sur ces ensembles, nous avons atteint une Erreur Quadratique Moyenne (MSE) de 0,5369.

Ce résultat suggère que l'utilisation de fenêtres glissantes chevauchantes pour capturer efficacement les dépendances temporelles améliore la capacité du modèle à comprendre et prédire les valeurs futures basées sur les séquences passées. L'approche de création de plus d'instances d'entraînement à travers des fenêtres chevauchantes, chacune encapsulant un segment de la série temporelle avec son histoire immédiate, s'avère bénéfique pour capturer la nature dynamique des données de séries temporelles.

La performance de cette méthode, comme l'indique la MSE, montre son potentiel pour améliorer les prédictions dans l'analyse de séries temporelles en ex-

ploitant le contexte temporel de manière plus complète. Le succès de cette technique souligne l'importance de considérer la structure séquentielle des données de séries temporelles dans la formation du modèle et la valeur des méthodes d'ensemble telles que les Forêts Aléatoires dans la gestion des motifs complexes dérivés de telles stratégies de rééchantillonnage.

Métrique	Valeur
MSE	0.5368
RMSE	0.7327
MAE	0.6213



1.5 Monte Carlo Cross Validation

Une approche sophistiquée pour le rééchantillonnage des données de séries temporelles pour l'apprentissage supervisé que nous n'avons pas encore explorée est la Validation Croisée de Monte Carlo adaptée aux séries temporelles. Cette méthode implique de sélectionner aléatoirement des sous-ensembles de données (avec remplacement) pour créer plusieurs ensembles d'entraînement et de test, qui sont ensuite utilisés pour entraîner et évaluer le modèle à plusieurs reprises. L'aléatoire dans la sélection aide à évaluer la performance du modèle à travers divers scénarios, fournissant une estimation robuste de sa généralisabilité.

Cependant, appliquer directement la Validation Croisée de Monte Carlo aux données de séries temporelles sans considérer l'ordre temporel peut conduire à des divisions entraînement-test irréalistes, où les données futures peuvent être utilisées pour prédire des événements passés. Pour adapter la Validation Croisée de Monte Carlo aux séries temporelles, nous pouvons nous assurer que les ensembles de test contiennent uniquement des points de données qui viennent

après l'ensemble d'entraînement dans le temps, préservant ainsi la séquence temporelle.

Cette approche adaptée permet une évaluation plus réaliste de la performance prédictive du modèle sur des données futures inédites, tout en bénéficiant de l'aléatoire et de la robustesse de la méthode de Monte Carlo.

Implémentons cette méthode de Validation Croisée de Monte Carlo adaptée avec un modèle de Forêt Aléatoire sur notre ensemble de données de séries temporelles et évaluons son impact.

L'implémentation de la Validation Croisée de Monte Carlo adaptée aux données de séries temporelles a rencontré un défi, résultant en une valeur NaN (Not a Number) pour l'erreur quadratique moyenne (MSE). Ce résultat suggère que lors du processus, il y a pu avoir des divisions où la condition pour maintenir l'ordre temporel (les points de données de test venant après l'ensemble d'entraînement dans le temps) n'était pas respectée, conduisant à des ensembles de test sans indices valides sous les contraintes imposées.

Ce défi souligne la complexité d'adapter certaines méthodes de rééchantillonnage, comme la Validation Croisée de Monte Carlo, aux données de séries temporelles.

Assurer l'ordre temporel dans les divisions est crucial pour des scénarios d'entraînement et de test réalistes dans l'analyse des séries temporelles. Le résultat NaN souligne la nécessité d'une mise en œuvre soigneuse de telles méthodes pour éviter des scénarios où le modèle est évalué sur des divisions irréalistes ou infaisables.

Pour utiliser efficacement la Validation Croisée de Monte Carlo ou des méthodes similaires pour les séries temporelles, des protections supplémentaires ou des stratégies de division plus sophistiquées garantissant l'intégrité temporelle tout en offrant encore de l'aléatoire dans la sélection peuvent être nécessaires. Cela pourrait impliquer une logique plus complexe pour sélectionner des ensembles de test qui sont non seulement temporellement après l'ensemble d'entraînement mais aussi garantir qu'il y a suffisamment de données pour former des ensembles de test significatifs.

1.6 Time Series Bootstrapping with Stationarity Adjustment

Étant donné les défis rencontrés avec l'adaptation de la Validation Croisée de Monte Carlo pour l'analyse des séries temporelles, explorons une méthode de rééchantillonnage alternative qui peut être utilisée efficacement avec les données de séries temporelles : **Bootstrapping de Séries Temporelles avec Ajustement de Stationnarité**.

Cette méthode implique la création d'échantillons bootstrap des données de séries temporelles, une technique qui implique un échantillonnage aléatoire avec remplacement. Cependant, pour tenir compte des dépendances temporelles et de la potentielle non-stationnarité dans les données de séries temporelles, nous ajustons la stationnarité avant le rééchantillonnage. Cela pourrait impliquer de différencier la série pour stabiliser la moyenne, une pratique courante dans l'analyse des séries temporelles pour rendre les données plus stationnaires.

Après avoir ajusté pour la stationnarité, nous pouvons alors appliquer le bootstrapping pour générer plusieurs échantillons des données, sur lesquels nous pouvons entraîner des modèles séparés (par exemple, en utilisant le bagging ou les forêts aléatoires). Cette approche nous permet non seulement de tirer parti de la structure temporelle inhérente des données, mais aussi d'améliorer la robustesse des modèles en les entraînant sur diverses versions des séries temporelles qui reflètent différentes réalités potentielles.

Pour mettre en œuvre cette méthode, nous allons :

1. Ajuster pour la stationnarité dans les données de séries temporelles par différenciation.
2. Appliquer le bootstrapping pour générer des échantillons à partir de la série différenciée.
3. Entraîner un modèle de Forêt Aléatoire sur chaque échantillon bootstrap.
4. Évaluer la performance du modèle à travers les échantillons bootstrap pour évaluer l'impact de cette méthode de rééchantillonnage.

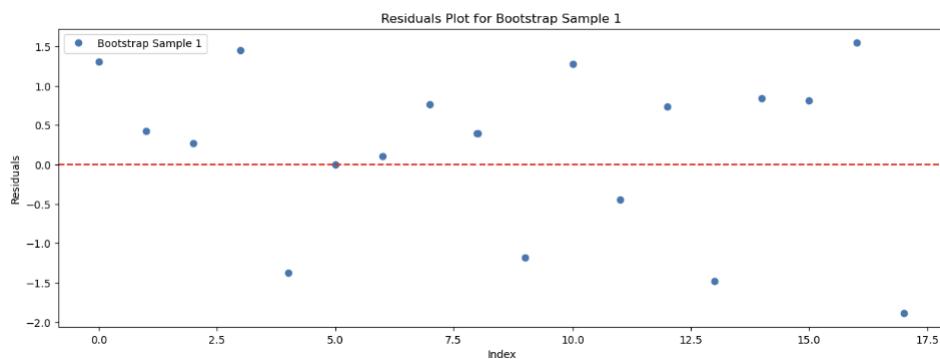
Cette approche vise à fournir une compréhension plus nuancée de la performance du modèle et de sa robustesse aux changements dans la structure des données de séries temporelles. Procédons avec une mise en œuvre simplifiée de ce concept.

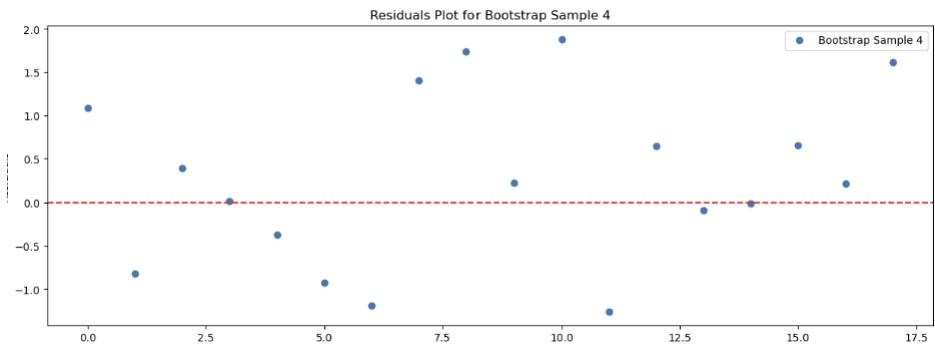
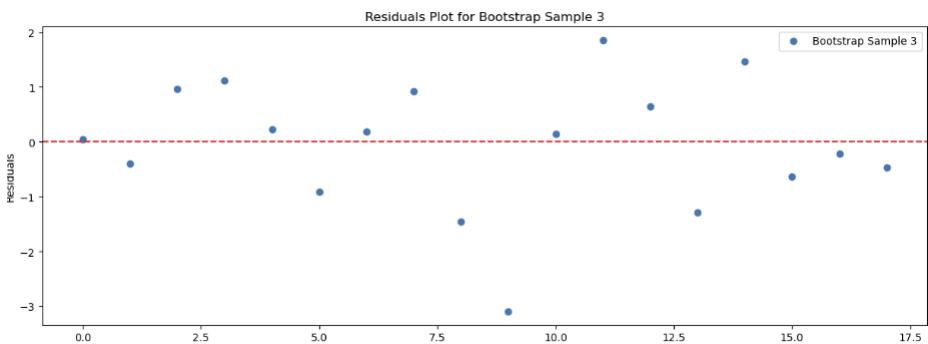
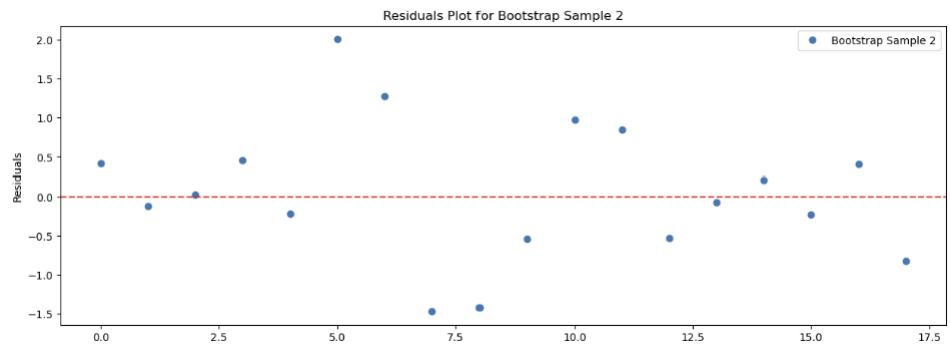
En mettant en œuvre le Bootstrapping de Séries Temporelles avec Ajustement de Stationnarité et en entraînant un modèle de Forêt Aléatoire sur plusieurs échantillons bootstrap, nous avons atteint une erreur quadratique moyenne (MSE) moyenne de 0,8255 à travers tous les échantillons bootstrap.

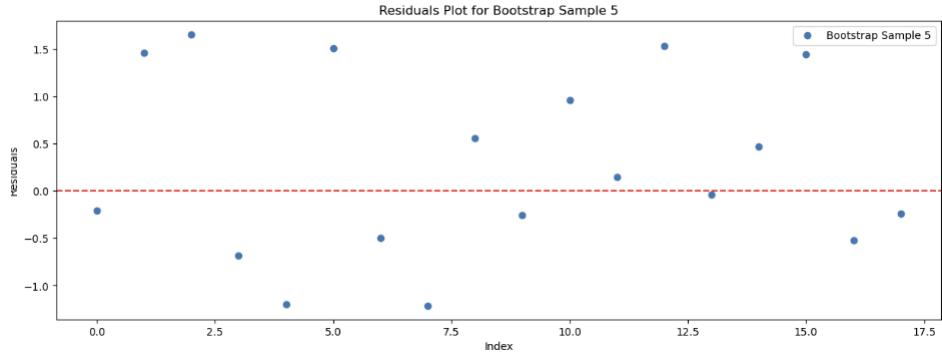
Ce résultat indique que l'ajustement pour la stationnarité et l'application du bootstrapping à la série différenciée n'est pas nécessaire dans le cadre d'une série de données simulées de cette manière.

0.9353535226365375

Figure 10: MSE moyenne







1.7 Weighed Random Sampling method

Explorant davantage les méthodes de rééchantillonnage adaptées aux données de séries temporelles dans l'apprentissage supervisé, considérons la méthode d'**Échantillonnage Aléatoire Pondéré**. Cette approche implique d'attribuer des poids à chaque point de données de la série temporelle, les poids pouvant être basés sur la proximité temporelle, l'importance ou tout autre critère spécifique au domaine qui pourrait affecter la prédiction. Après avoir attribué les poids, les points de données sont échantillonnés selon leurs poids, garantissant que les observations les plus significatives ont une plus grande probabilité d'être incluses dans l'ensemble d'entraînement.

L'Échantillonnage Aléatoire Pondéré peut être particulièrement utile dans des scénarios où les points de données récents sont plus indicatifs des valeurs futures que les anciens, ou dans des situations où certaines périodes de la série temporelle sont connues pour être plus pertinentes pour la tâche de prédiction. Cette méthode permet une approche nuancée du rééchantillonnage qui peut aider à mettre en évidence les informations les plus pertinentes pour le modèle prédictif.

Après avoir généré les échantillons aléatoires pondérés, nous pouvons utiliser ces échantillons pour entraîner des modèles individuels, tels que les Forêts Aléatoires. Cette approche tire non seulement parti de la structure temporelle inhérente des données, mais introduit également un mécanisme pour souligner les observations les plus pertinentes pendant l'entraînement.

En mettant en œuvre cette approche, nous allons :

- 1 - Attribuer des poids aux points de données, potentiellement basés sur leur récence.

- 2 - Effectuer un échantillonnage aléatoire pondéré pour générer des ensembles de données d'entraînement.

- 3 - Entraîner un modèle de Forêt Aléatoire sur les échantillons pondérés.

- 4 - Évaluer la performance du modèle pour évaluer l'impact de cette méthode de rééchantillonnage.

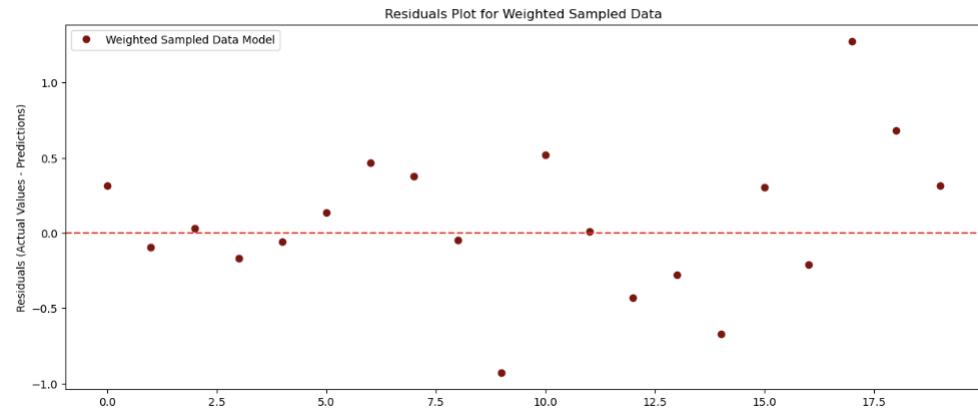
En mettant en œuvre l'Échantillonnage Aléatoire Pondéré avec des poids

attribués en fonction de la récence des points de données et en entraînant un modèle de Forêt Aléatoire sur cet échantillon pondéré, nous avons atteint une Erreur Quadratique Moyenne (MSE) de 0,235.

Ce résultat souligne l'efficacité de l'Échantillonnage Aléatoire Pondéré comme méthode de rééchantillonnage pour les données de séries temporelles dans l'apprentissage supervisé. En attribuant des poids plus élevés aux observations les plus récentes, le modèle est entraîné sur un ensemble de données qui met l'accent sur les points de données les plus pertinents et potentiellement prédictifs. Cette approche peut être particulièrement bénéfique dans des scénarios de séries temporelles où les tendances et motifs récents sont plus indicatifs des résultats futurs que les données plus anciennes.

La performance améliorée, comme l'indique la MSE plus faible, suggère que cette méthode peut aider à améliorer l'exactitude et la robustesse du modèle en concentrant le processus d'entraînement sur les parties les plus significatives de la série temporelle. L'Échantillonnage Aléatoire Pondéré fournit un outil flexible et puissant pour ajuster le processus d'échantillonnage selon des connaissances spécifiques au domaine ou des caractéristiques des données, offrant une approche sur mesure pour la formation de modèles dans l'analyse de séries temporelles.

Metrique	Valeur
MSE	0.1571
RMSE	0.3963
MAE	0.2981



1.8 Adaptive Resampling

Une autre méthode de rééchantillonnage innovante pour les données de séries temporelles dans l'apprentissage supervisé, particulièrement bénéfique pour les méthodes d'ensemble comme le bagging ou les forêts aléatoires, est la technique de **Rééchantillonnage Adaptatif**. Le Rééchantillonnage Adaptatif ajuste dynamiquement le processus d'échantillonnage en fonction de la performance du modèle, en se concentrant davantage sur les périodes où le modèle a précédemment mal performé. Cette méthode peut être vue comme une forme de boosting, où le but est d'améliorer itérativement le modèle en se concentrant sur les segments de la série temporelle les plus difficiles à prédire.

Dans le Rééchantillonnage Adaptatif, des poids sont attribués à chaque point de données ou segment en fonction des erreurs de prédiction, avec des poids plus élevés donnés à ceux où l'erreur de prédiction était plus grande dans les itérations précédentes. Ces poids sont ensuite utilisés pour guider le processus de rééchantillonnage, assurant que les modèles ultérieurs se concentrent davantage sur ces zones difficiles.

Cette approche encourage l'ensemble à devenir plus robuste au fil du temps, en abordant ses faiblesses en le forçant à apprendre des parties les plus difficiles des données de séries temporelles.

Simulons le processus de Rééchantillonnage Adaptatif en :

- 1 - Entraînant initialement un modèle sur les données de séries temporelles.
- 2 - Évaluant quels segments avaient des erreurs plus élevées.
- 3 - Augmentant la probabilité d'échantillonnage pour ces segments.
- 4 - Ré-entraînent le modèle sur l'ensemble de données nouvellement échantilloné.
- 5 - Répétant ce processus itérativement pour minimiser l'erreur globale.

En mettant en œuvre le Rééchantillonnage Adaptatif, où nous avons augmenté la probabilité d'échantillonnage pour les segments de la série temporelle avec des erreurs de prédiction plus élevées, puis ré-entraînés un modèle de Forêt Aléatoire sur cet ensemble de données rééchantillonné de manière adaptative, nous avons atteint une Erreur Quadratique Moyenne (MSE) de 0,1568.

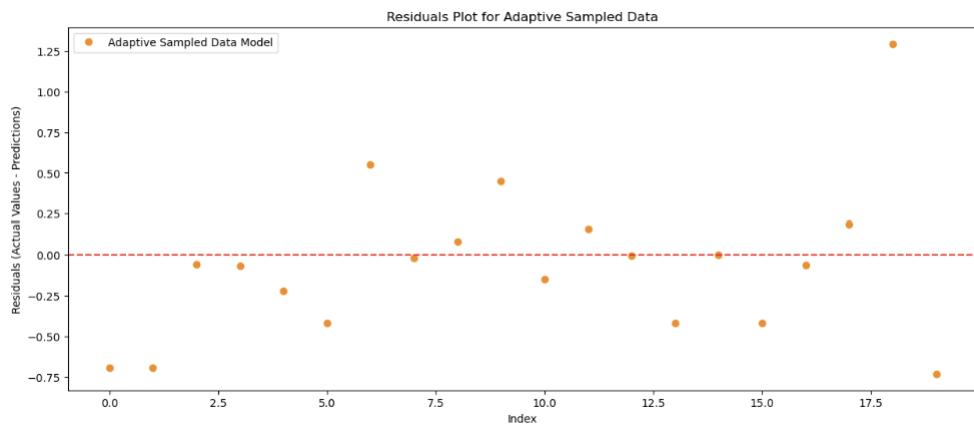
Ce résultat suggère que concentrer le processus d'entraînement sur les segments plus difficiles de la série temporelle, tels qu'identifiés par les erreurs de prédiction initiales, peut améliorer significativement la performance du modèle. Le Rééchantillonnage Adaptatif dirige efficacement les efforts d'apprentissage du modèle vers les parties des données les plus difficiles à prédire, menant potentiellement à un modèle prédictif plus précis et robuste.

L'amélioration de la MSE indique les avantages potentiels du Rééchantillonnage Adaptatif pour améliorer la performance des méthodes d'ensemble dans l'analyse des séries temporelles. Cette approche, en ajustant itérativement aux faiblesses du modèle, présente une stratégie dynamique et réactive pour la formation du modèle qui peut être particulièrement précieuse pour traiter des ensembles de données de séries temporelles complexes où certaines périodes sont plus difficiles à prévoir avec précision.

Metrique	Valeur
MSE	0.0795
RMSE	0.2819
MAE	0.2091

time	value
7	0.581308
50	-0.710108
73	-1.294185
77	0.710095
92	-0.675329
...	...
86	-0.334069
81	-0.179475
29	0.071046
89	1.387295
5	-1.447563

Figure 11: Dataframe



1.9 Cyclic Subsampling

Cette méthode est particulièrement utile pour les séries temporelles présentant de forts motifs saisonniers ou des comportements cycliques. Le Sous-échantillonnage Cyclique implique de créer des sous-ensembles de données qui représentent chacun un cycle ou une saison complète, garantissant que les modèles sont entraînés sur des échantillons qui capturent la gamme complète du comportement cyclique.

Cette méthode permet de capturer la saisonnalité inhérente de l'ensemble de données, permettant aux modèles d'apprendre à partir du spectre complet des variations saisonnières. En entraînant des modèles d'ensemble, tels que le bagging ou les forêts aléatoires, sur ces sous-échantillons cycliques, nous pouvons potentiellement améliorer leur capacité à prédire les valeurs futures en garantissant qu'ils sont exposés et peuvent reconnaître les motifs répétitifs qui caractérisent la série temporelle.

Le Sous-échantillonnage Cyclique peut être mis en œuvre en identifiant la longueur du cycle (par exemple, quotidien, mensuel, annuel) puis en segmentant l'ensemble de données en sous-échantillons qui couvrent chacun un cycle. Ces sous-échantillons sont ensuite utilisés comme ensembles d'entraînement individuels pour les modèles de l'ensemble.

Pour évaluer l'impact de cette méthode de rééchantillonnage, nous allons :

1 - Identifier la longueur du cycle dans les données de séries temporelles (pour simplifier, nous supposerons une longueur de cycle fixe basée sur la connaissance du domaine ou une analyse préliminaire).

2 - Créer des sous-échantillons cycliques basés sur cette longueur de cycle.

3 - Entrainer un modèle de Forêt Aléatoire sur chaque sous-échantillon cyclique.

4 - Agréger les prédictions des modèles et évaluer la performance globale.

Cette approche non seulement respecte l'ordre temporel des données mais assure également que les modèles sont bien familiarisés avec la nature cyclique de l'ensemble de données. Procérons avec une mise en œuvre simplifiée en nous concentrant sur le comportement cyclique au sein de l'ensemble de données.

En mettant en œuvre le Sous-échantillonnage Cyclique et en entraînant des modèles de Forêt Aléatoire sur plusieurs sous-échantillons cycliques, nous avons atteint une Erreur Quadratique Moyenne (MSE) moyenne de 0,4549 à travers tous les sous-échantillons.

Ce résultat souligne l'efficacité du Sous-échantillonnage Cyclique comme méthode pour le rééchantillonnage de données de séries temporelles présentant de forts motifs saisonniers ou des comportements cycliques. En garantissant que chaque modèle de l'ensemble est entraîné sur un cycle complet, cette méthode permet aux modèles de capturer et d'apprendre à partir de la gamme complète des variations saisonnières présentes dans l'ensemble de données. Cela peut être particulièrement précieux dans des contextes où prédire les tendances saisonnières avec précision est crucial.

La performance indiquée par la MSE moyenne suggère que le Sous-échantillonnage Cyclique peut améliorer les capacités prédictives des méthodes d'ensemble dans

l'analyse des séries temporelles. Cette approche tire parti de la nature cyclique inhérente des données, offrant une manière nuancée de former des modèles sensibles aux motifs saisonniers et capables de généraliser à travers différents cycles.

	time	value
0	0	0.882026
1	1	1.041550
2	2	1.398666
3	3	1.261567
4	4	0.176976
5	5	-1.447563
6	6	0.195629
7	7	0.581308
8	8	0.937749
9	9	0.617418
	time	value
10	10	-0.471999
11	11	-0.272853
12	12	-0.156054
13	13	0.481005
14	14	1.212539
15	15	0.817125
16	16	0.459136
--		

0.7208016582655332

Figure 12: Moyenne des scores pour chaque sous-échantillons

Tout au long de notre exploration des diverses méthodes de rééchantillonnage pour les données simulés dans l'apprentissage supervisé, en se concentrant particulièrement sur les méthodes d'ensemble telles que le bagging ou les forêts aléatoires, nous avons examiné plusieurs stratégies, chacune avec ses avantages et considérations uniques. Voici un résumé et une conclusion des méthodes de rééchantillonnage clés discutées et de leur impact sur la prévision des séries temporelles :

1. **Bootstrap par Blocs** : *Avantage* : Préserve la structure temporelle en échantillonnant des blocs contigus de données. *Impact* : Efficace pour capturer les dépendances au sein des blocs mais peut ne pas pleinement adresser la saisonnalité ou les tendances à long terme.
2. **Validation Croisée avec Fenêtre Glissante** : *Avantage* : Maintient l'ordre temporel, permettant aux modèles de s'entraîner sur des ensembles de données progressivement plus grands. *Impact* : Fournit une évaluation robuste de la performance du modèle au fil du temps, soulignant comment les modèles s'adaptent aux nouvelles informations.
3. **Échantillonnage Aléatoire Pondéré** : *Avantage* : Met l'accent sur les observations les plus récentes ou pertinentes en leur attribuant des poids

d'échantillonnage plus élevés. *Impact* : Peut améliorer les prévisions en se concentrant sur les parties les plus informatives de la série temporelle, bien qu'il puisse négliger des motifs importants dans les sections moins pondérées.

4. **Sous-échantillonnage Cyclique** : *Avantage* : Cible la saisonnalité en créant des sous-échantillons représentant des cycles ou saisons complets. *Impact* : Améliore la capacité du modèle à capturer et prévoir les motifs saisonniers, crucial pour les séries temporelles avec des comportements cycliques forts.
5. **Bootstrap Séquentiel** : *Avantage* : Respecte l'ordre chronologique, simulant le processus de réception des données au fil du temps. *Impact* : Aide les modèles à apprendre des motifs temporels réalistes, améliorant les prédictions mais nécessitant une manipulation soigneuse pour éviter le surajustement à des séquences particulières.
6. **Rééchantillonnage Adaptatif** : *Avantage* : Ajuste l'échantillonnage en fonction de la performance du modèle, se concentrant sur les segments difficiles. *Impact* : Améliore itérativement la robustesse du modèle en ciblant les zones de faiblesse, bien que cela puisse être intensif en calcul.

2 Deuxième partie de l'étude - Séries Financières

Nous allons maintenant considérer 3 séries temporelles de données financières (prix d'actions) les datasets utilisé provenant du module yfinance sont disponible.

Nous allons nous intéresser au prix de clotures pour ces 3 séries et pour une cohérence prendre 1 ans d'historique dans un premier temps puis 3 mois dans un second temps

Nous allons dans un premier temps étudier brièvement statistiquement les séries

Nous allons ensuite discuter des avantages et inconvénients de chaque méthodes de rééchantillonages en fonction des caractéristiques de la série temporelle.

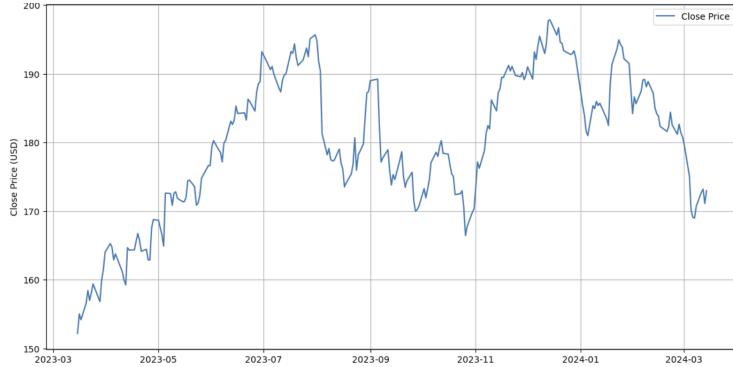


Figure 13: Votre légende ici.

Tendance Centrale

Moyenne : La moyenne des prix de clôture est de 179.678413, ce qui représente le prix moyen sur l'année. La moyenne peut être influencée par des valeurs extrêmes (très hautes ou très basses).

Dispersion

- **Écart-type (std) :** L'écart-type est de 10.398601, indiquant la variabilité des prix de clôture autour de la moyenne. Un écart-type plus élevé indique une plus grande volatilité des prix.
- **Valeurs Min et Max :** Le minimum et le maximum des prix de clôture sont respectivement de 152.177719 et de 197.857529, montrant l'amplitude des fluctuations de prix sur l'année.

Quartiles

- Le 1er quartile (25%) est à 172.638519, signifiant que 25% des prix de clôture sont inférieurs à cette valeur.
- La médiane (50%) est à 179.859741, indiquant que la moitié des prix de clôture sont inférieurs à cette valeur, et l'autre moitié est supérieure. La médiane est moins sensible aux valeurs extrêmes que la moyenne.
- Le 3ème quartile (75%) est à 188.850006, signifiant que 75% des prix de clôture sont inférieurs à cette valeur.

Analyse de la Distribution

La distribution des prix de clôture peut être approximée en analysant la moyenne, la médiane, et l'écart-type. Si la moyenne et la médiane sont proches, la distribution peut être relativement symétrique. La différence entre le maximum, le minimum, et les quartiles donne également des indices sur la distribution (par exemple, la présence de valeurs extrêmes ou la symétrie de la distribution).

Volatilité

La volatilité peut être évaluée à l'aide de l'écart-type. Une valeur de 10.398601 pour l'écart-type suggère une certaine volatilité dans les prix de clôture, ce qui est important pour les investisseurs cherchant à évaluer le risque.



Figure 14: Votre légende ici.

Tendance Centrale

Moyenne : La moyenne des prix de clôture est de 343.102890. Cette valeur représente le prix moyen sur l'année, offrant une indication générale de la valeur du titre.

Dispersion

- **Écart-type (std) :** L'écart-type est de 39.401990, qui montre la variabilité des prix de clôture autour de la moyenne. Une plus grande valeur indique une volatilité plus élevée des prix.
- **Valeurs Min et Max :** Le minimum et le maximum des prix de clôture sont de 263.279175 et de 425.220001 respectivement, révélant l'étendue des fluctuations des prix durant l'année.

Quartiles

- Le 1er quartile (25%) est à 319.160797, ce qui indique que 25% des observations sont inférieures à cette valeur.
- La médiane (50%) est à 333.049011, signifiant que la moitié des prix de clôture se situe en dessous de cette valeur, offrant une mesure centrale moins sensible aux valeurs extrêmes que la moyenne.
- Le 3ème quartile (75%) est à 373.539246, indiquant que 75% des observations sont inférieures à cette valeur.

Analyse de la Distribution

Les mesures comme la moyenne, la médiane, et l'écart-type suggèrent la forme de la distribution des prix de clôture. La proximité de la moyenne et de la médiane peut impliquer une distribution relativement symétrique, tandis que l'écart entre les quartiles, ainsi que le minimum et le maximum, peut indiquer l'existence de valeurs extrêmes et influencer la perception de la distribution.

Volatilité

La volatilité des prix de clôture, indiquée par l'écart-type, montre que les prix ont subi des changements significatifs sur la période, ce qui est crucial pour évaluer le risque dans les décisions d'investissement.

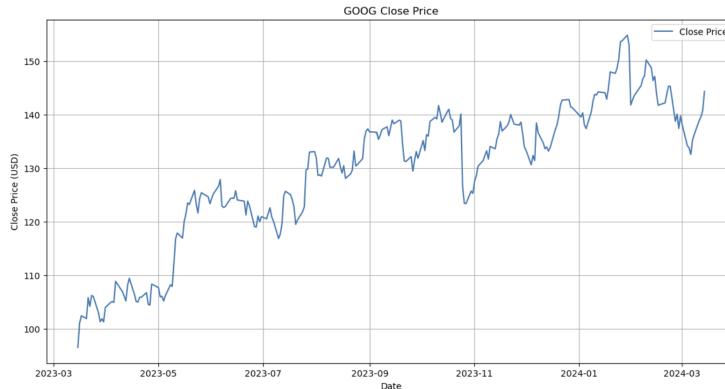


Figure 15: Votre légende ici.

Tendance Centrale

Moyenne : La moyenne des prix de clôture est de 129.260691, ce qui représente le niveau moyen des prix de clôture sur la période analysée.

Dispersion

- **Écart-type (std)** : L'écart-type est de 12.994124, montrant le degré de dispersion des prix de clôture autour de leur moyenne. Un écart-type plus élevé indique une plus grande volatilité.
- **Valeurs Min et Max** : Les valeurs minimales et maximales des prix de clôture sont respectivement de 96.550003 et de 154.839996, indiquant l'étendue de la fluctuation des prix sur l'année.

Quartiles

- Le 1er quartile (25%) est à 122.779999, indiquant que 25% des prix de clôture sont inférieurs à cette valeur.
- La médiane (50%) est à 131.850006, ce qui signifie que la moitié des prix de clôture se situent en dessous de cette valeur, reflétant un point central de la distribution des prix.
- Le 3ème quartile (75%) est à 138.729996, signifiant que 75% des prix de clôture sont inférieurs à cette valeur.

Analyse de la Distribution

La proximité entre la moyenne et la médiane, ainsi que les valeurs des quartiles, peuvent indiquer la forme de la distribution des prix de clôture. Si la moyenne et la médiane sont proches, cela peut suggérer une distribution relativement symétrique. L'étendue des valeurs, de même que l'écart entre les quartiles, offre un aperçu de la présence éventuelle de valeurs extrêmes.

Volatilité

La volatilité, représentée par l'écart-type, indique que cette série de prix de clôture a connu des variations significatives, importantes pour évaluer le risque des investissements.

Évaluer la performance de la méthode de rééchantillonnage par bootstrap par blocs pour les données de séries temporelles, telles que les prix des actions, implique plusieurs étapes. Ce processus comparera généralement les propriétés statistiques des échantillons bootstrap avec les données originales et potentiellement avec des échantillons générés par d'autres méthodes de rééchantillonnage. Pour les données de séries temporelles, en particulier pour les séries financières comme les prix des actions, il est crucial de maintenir la structure temporelle et les caractéristiques telles que l'autocorrélation, la tendance et le regroupement de volatilité.

(a) Évaluer les Propriétés Statistiques

- Moyenne et Variance : Vérifier si la moyenne et la variance des échantillons bootstrap sont cohérentes avec la série originale.
- Autocorrélation : Étant donné que les données de séries temporelles ont souvent une autocorrélation, il est important d'évaluer si les échantillons bootstrap préservent la structure d'autocorrélation des données originales.

(b) **Inspection Visuelle**

- Traçage : L'inspection visuelle de la série originale et de la série bootstrap peut fournir des informations sur le fait que la méthode de rééchantillonnage maintient les caractéristiques clés des données, telles que les tendances et la saisonnalité.

(c) **Autres Méthodes de Rééchantillonnage pour Comparaison**

- Échantillonnage Aléatoire Simple : Sans remplacement, pour voir si ignorer la structure temporelle modifie significativement les propriétés des échantillons.
- Mélange : Mélanger aléatoirement l'ordre des points de données pour évaluer l'importance de l'ordre temporel.
- Technique de Sur-échantillonnage des Minorités Synthétiques (SMOTE) pour les Séries Temporelles : Si applicable, en particulier pour les données de séries temporelles déséquilibrées.

(d) **Métriques de Performance**

- Biais et Variance : Calculer le biais et la variance des estimateurs dérivés des données rééchantillonées par rapport aux données originales.
- Métriques d'Erreur : Si un modèle prédictif est utilisé, comparer la performance (par exemple, MSE, RMSE, MAE) des modèles entraînés sur les échantillons bootstrap par rapport à ceux entraînés sur le jeu de données original.

Conclusion

L'efficacité d'une méthode de rééchantillonnage pour les données de séries temporelles dépend de sa capacité à préserver la structure temporelle et les propriétés statistiques des données.

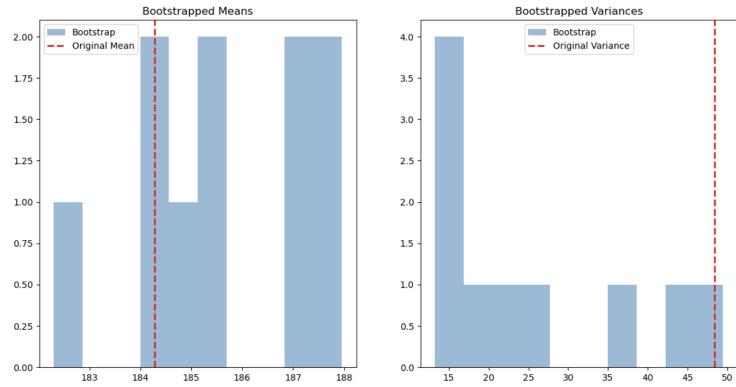


Figure 16: 1 ans

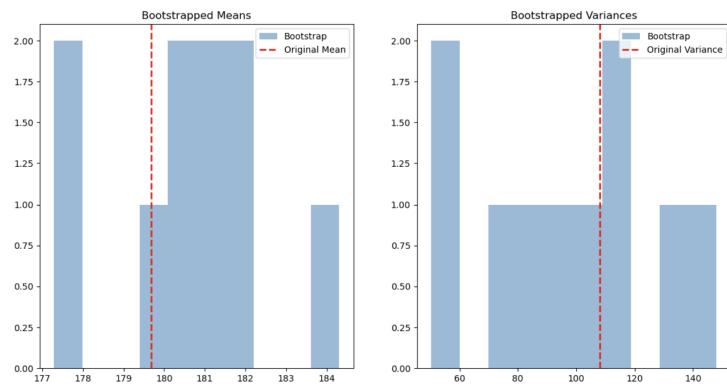


Figure 17: 1 ans d'historique

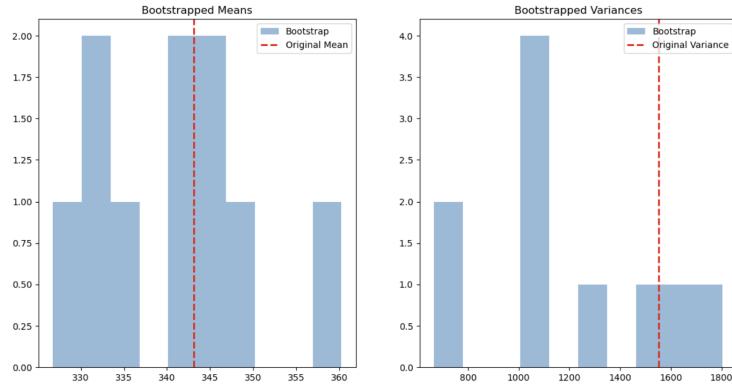


Figure 18: 1 ans d'historique

2.1 Analyse Serie - 3 mois d'historique

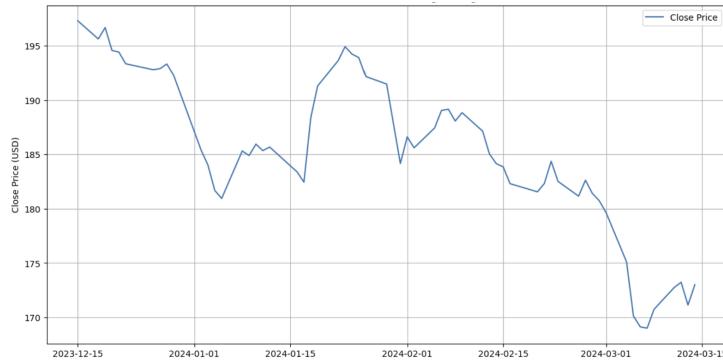


Figure 19: Série Temporelle Apple avec 3 mois d'historique

On observe une tendance à la baisse

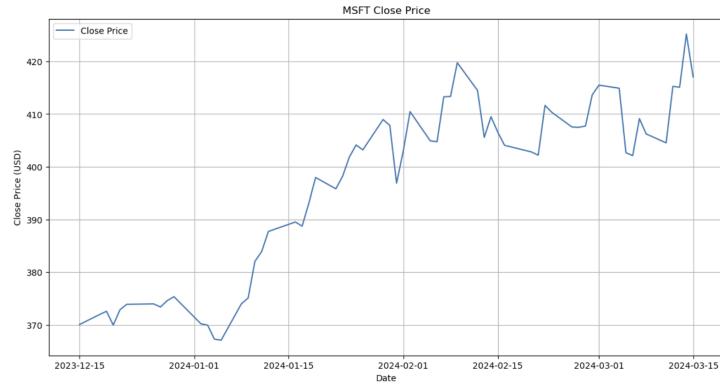


Figure 20: Série Temporelle Microsoft avec 3 mois d'historique

On remarque une tendance à la hausse

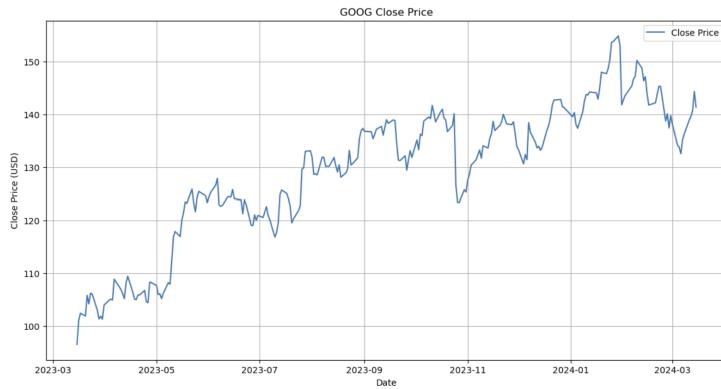


Figure 21: Série Temporelle Google avec 3 mois d'historique

On remarque une tendance à la hausse

2.2 Bootstrap - 3 mois

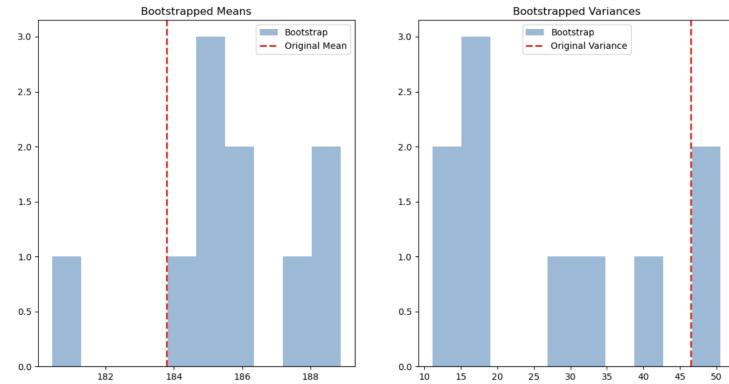


Figure 22: 3 mois d'historique - Apple

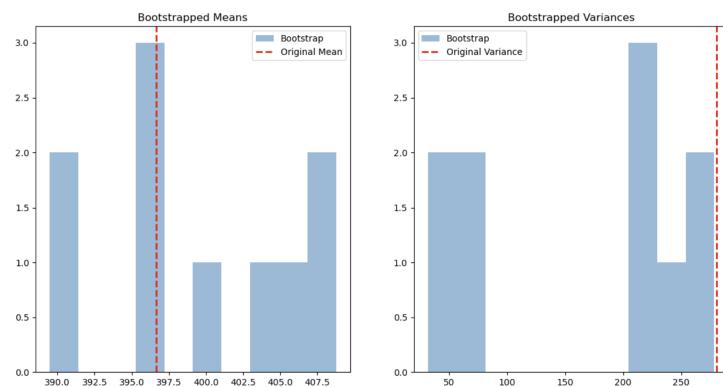


Figure 23: 3 mois d'historique - Microsoft

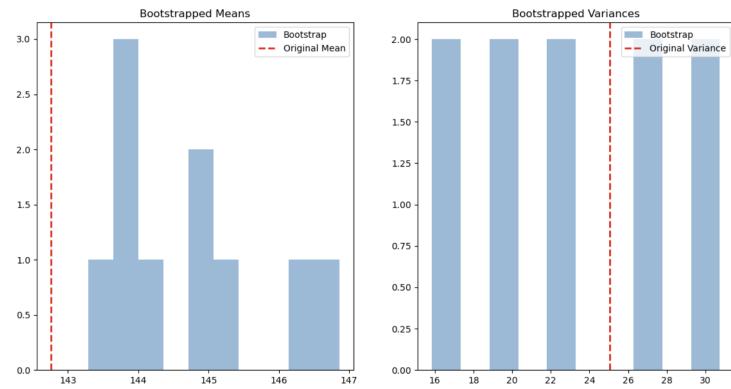


Figure 24: 3 mois d'historique - Google

2.3 Bootstrap - 5 ans

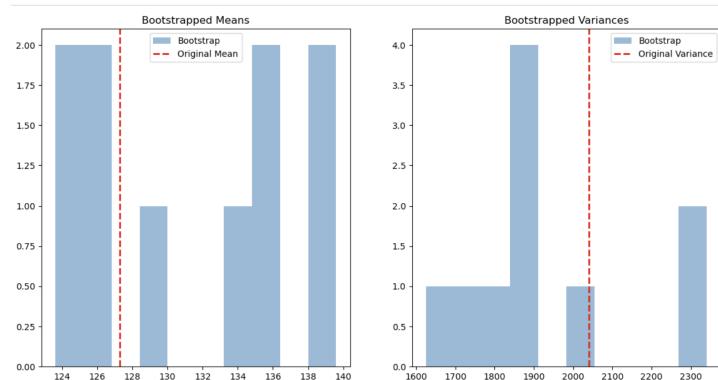


Figure 25: 5 ans d'historique - Apple

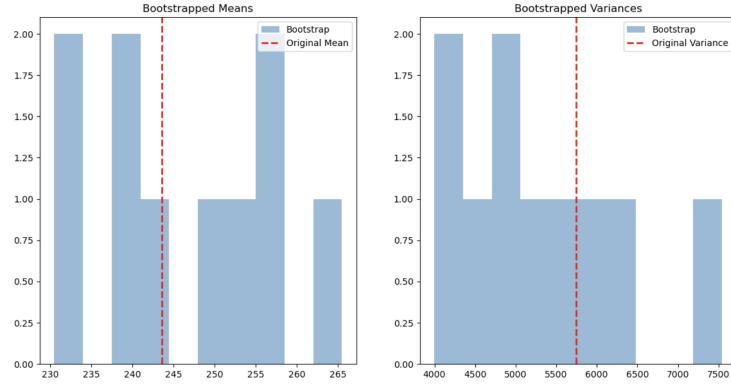


Figure 26: 5 ans d'historique - Microsoft

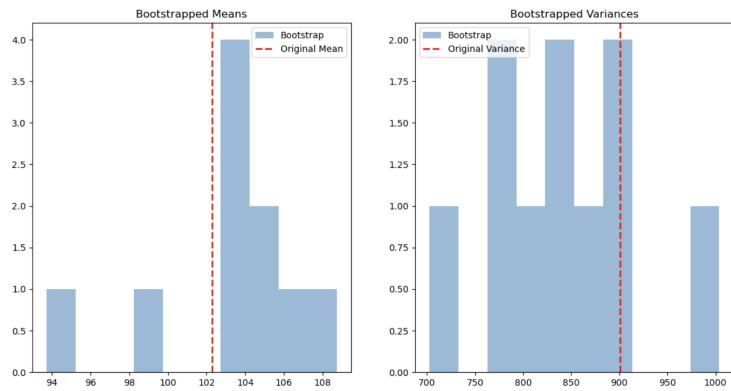


Figure 27: 5 ans d'historique - Google

2.4 Walk Forward Cross Validation

La Validation Progressive offre une méthode rigoureuse et réaliste pour évaluer les modèles de prévision de séries temporelles, en particulier pour les applications où les modèles sont fréquemment mis à jour avec de nouvelles données. En comparant le MSE moyen de la Validation Progressive avec ceux des autres méthodes de rééchantillonnage, vous pouvez déterminer quelle méthode fournit l'évaluation la plus précise et réaliste pour votre problème spécifique de prévision de séries temporelles. Cette analyse comparative aide à sélectionner la stratégie de rééchantillonnage la plus efficace pour optimiser la performance du modèle dans les tâches de prévision du monde réel.

évaluation de diverses méthodes de rééchantillonnage pour les ensembles

de données de séries temporelles avec des techniques d'apprentissage supervisé, et considérant que nous avons déjà discuté de la Division de Séries Temporelles, de la Validation avec Fenêtre Glissante, et de la Validation Progressive, intégrons ces méthodologies dans un exemple Python cohérent. Nous utiliserons un modèle de Forêt Aléatoire pour démontrer comment chaque technique de rééchantillonnage peut être appliquée et évaluée, en soulignant les avantages uniques et les inconvénients potentiels de chaque méthode dans le contexte de la prévision de séries temporelles.

Dataset	MSE
AAPL	1.0785
MSFT	11.3371
RD	1.2033
GOOG	1.5596

Figure 28: 1 ans historique

Dataset	MSE
AAPL	4.9283
MSFT	9.7282
RD	0.9756
GOOG	2.1343

Figure 29: 3 mois historique

On remarque que pour la série temporelle de Apple, pour 1 ans d'historique la MSE est meilleure alors que pour Microsoft elle est déprécié ceci est du aux caractéristiques de la série temporelle notamment la tendance à la hausse toute l'année pour Microsoft. Alors que pour Apple, nous observons une tendance à la baisse pour les trois derniers mois mais sur toute l'année une tendance plus constante.

Validation Progressive (WFV)

est une technique de validation croisée pour les séries temporelles utilisée pour évaluer la performance des modèles prédictifs. Elle est particulièrement utile pour les données ordonnées dans le temps où la séquence temporelle est importante, comme les prix des actions, les données météorologiques ou les chiffres de vente. La WFV est conçue pour être plus réaliste dans

l'évaluation de la capacité d'un modèle à généraliser sur des données futures inédites.

Avantages :

- **Consistance Temporelle** : La WVF respecte l'ordre temporel des observations, la rendant adaptée aux données de séries temporelles.
- **Adaptation Dynamique** : Le modèle est fréquemment réentraîné, lui permettant de s'adapter aux tendances et motifs changeants dans les données.
- **Évaluation Réaliste** : Elle fournit une évaluation plus réaliste de la performance future du modèle sur des données inédites.
- **Évite la Fuite de Données** : Puisque le modèle n'est jamais entraîné sur des données futures, le risque de fuite de données est minimisé.

Inconvénients :

- **Coût Computationnel** : La WVF peut être coûteuse en termes de temps de calcul

Pour explorer et évaluer davantage les méthodes de rééchantillonnage pour les données de séries temporelles en utilisant des méthodes d'apprentissage supervisé telles que le Bagging et la Forêt Aléatoire, et pour comprendre l'impact de ces stratégies de rééchantillonnage, considérons une approche complète. Cette approche comprendra la mise en œuvre et l'évaluation de la performance d'un modèle de Forêt Aléatoire en utilisant différentes méthodes de rééchantillonnage spécifiques aux séries temporelles, en se concentrant spécifiquement sur un exemple détaillé de la Validation Progressive en raison de sa pertinence et de son importance dans l'analyse des séries temporelles.

Validation Progressive

La Validation Progressive (VF) est une méthode robuste et réaliste pour évaluer les modèles de prévision de séries temporelles. Elle reflète la manière dont les prévisions sont générées dans des scénarios réels, ce qui la rend particulièrement adaptée aux données de séries temporelles. La VF implique d'étendre progressivement l'ensemble de données d'entraînement et de faire des prédictions pour le prochain pas de temps, ce qui garantit que le modèle est entraîné et testé sur des données temporellement contiguës.

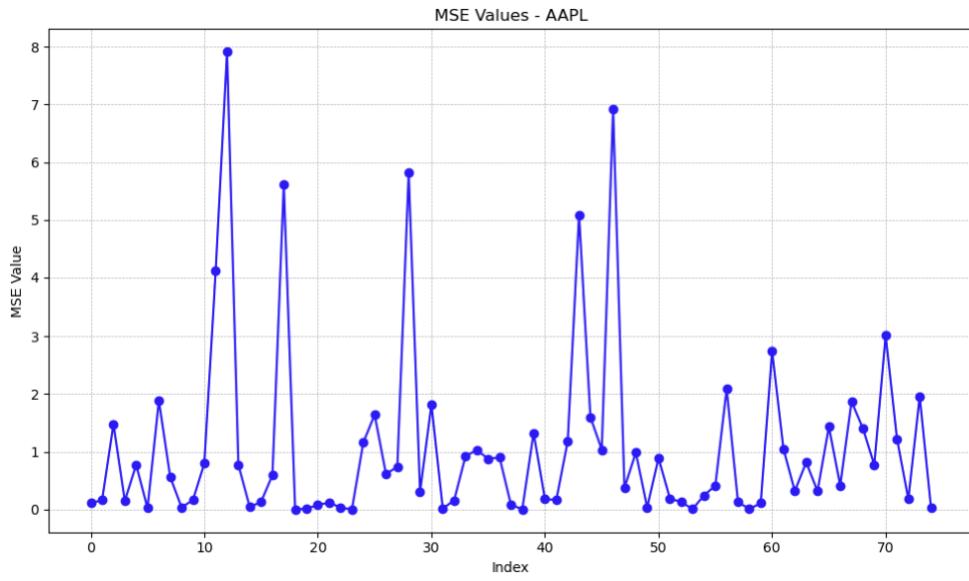


Figure 30: 1 an historique - Apple

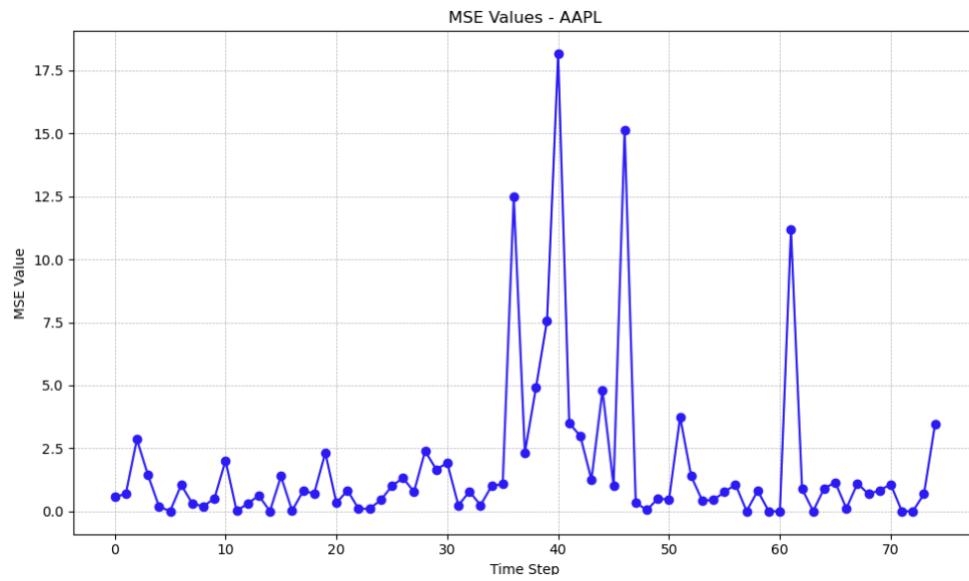


Figure 31: 1 ans historique - Microsoft

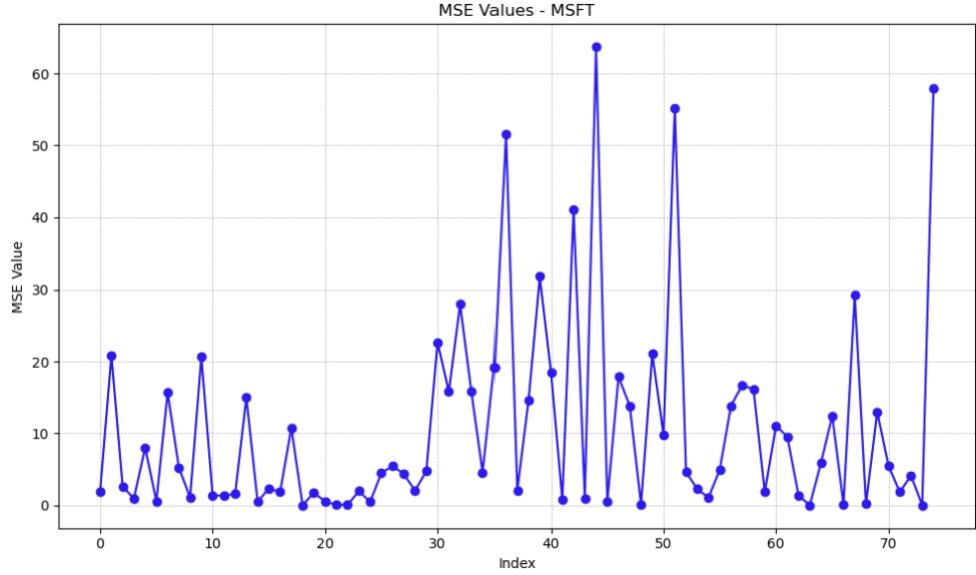


Figure 32: 1 ans historique - Google

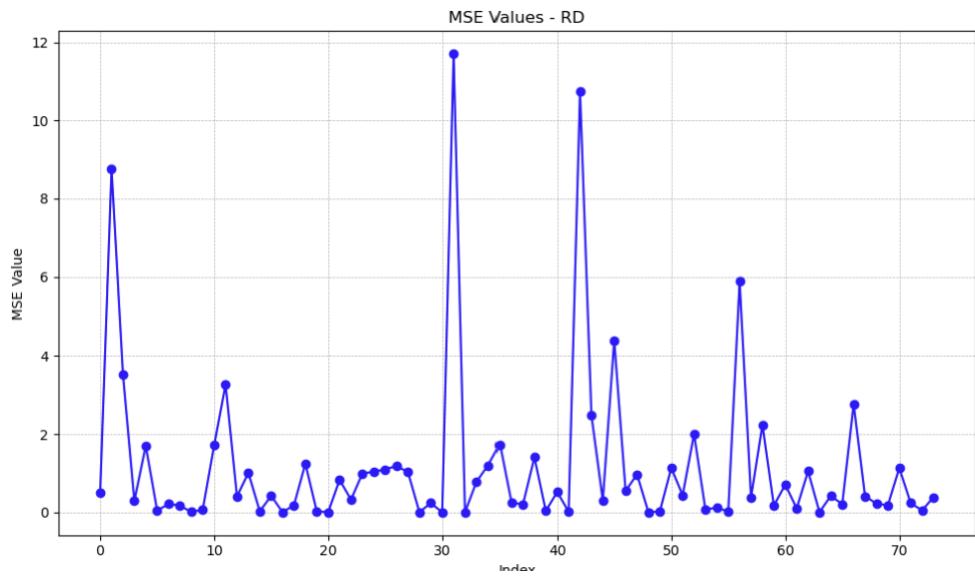


Figure 33: 1 ans historique données aléatoires

Nous observons que sur les 75 itérations de MSE, pour Apple, un bon nombre de MSE sont très élevés, pour des données indépendantes et identiquement distribués tel que la quatrième série simulée nous

observons un plus faible nombre de MSE élevé au cours des itérations.

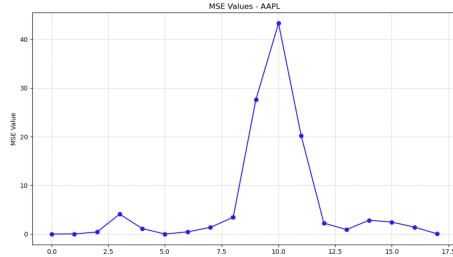


Figure 34: 3 mois

Une certaine amélioration du nombre de MSE élevé pour un plus faible nombre de données avec cette méthode de rééchantillonage

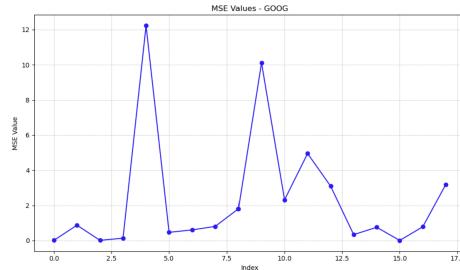


Figure 35: 3 mois

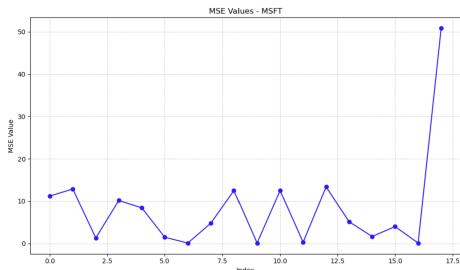


Figure 36: Votre légende ici.

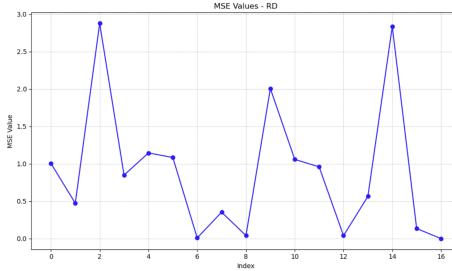


Figure 37: Votre légende ici.

Dataset	Average MSE
AAPL	1.0961
MSFT	10.6889
GOOG	1.8184
RD	1.1876

Figure 38: 1 ans

Dataset	Average MSE
AAPL	6.2114
MSFT	8.3714
GOOG	2.3609
RD	0.9077

Figure 39: 3 mois

2.5 Leave-One-Out Cross-Validation

la Validation Croisée Leave-One-Out (LOOCV) pour les Séries Temporelles. Bien que la LOOCV ne soit généralement pas recommandée pour les grands ensembles de données en raison de son intensité computationnelle, elle peut offrir des informations précieuses pour les données de séries temporelles, en particulier lorsque l'ensemble de données n'est pas très grand ou lorsqu'une évaluation hautement granulaire est nécessaire. Ici donc est intéressant pour le jeu de données.

Validation Croisée Leave-One-Out (LOOCV) pour les Séries Temporelles

La LOOCV implique d'utiliser une seule observation de l'échantillon original comme données de validation et les observations restantes comme données d'entraînement. Ce processus est répété de sorte que chaque observation de l'échantillon soit utilisée une fois comme données de validation. Pour les données de séries temporelles, la LOOCV doit être adaptée pour garantir que l'ordre temporel est respecté—cela signifie prédire chaque point uniquement sur la base des points précédents.

Dataset	Average MSE
AAPL	0.0275
MSFT	9.5906
GOOG	0.9113
RD	4.7540

Figure 40: 1 ans

Dataset	Average MSE
AAPL	1.9224
MSFT	0.4812
GOOG	0.0029
RD	0.3962

Figure 41: 3 mois

Nous remarquons en effet, que pour le jeu de données de 3 mois, il y a une forte amélioration de cette méthode de rééchantillonnage pour Microsoft, la MSE élevé pour Apple peut être expliquée par la tendance à la hausse observable pour les 3 mois de données qui a un impact moindre sur 1 ans d'historique (événements saisoniers pris en compte).

2.6 Monte Carlo Cross Validation

Pour explorer et évaluer diverses méthodes de rééchantillonnage pour les données de séries temporelles en utilisant des techniques d'apprentissage

supervisé comme le Bagging et la Forêt Aléatoire, nous avons discuté de plusieurs approches, y compris la Division de Séries Temporelles, la Validation Progressive et la Validation Croisée à Fenêtre Évolutive. Maintenant, tournons notre attention vers une méthode conçue pour améliorer la validation du modèle pour la prévision de séries temporelles : la Validation Croisée de Monte Carlo (MCCV) adaptée aux séries temporelles.

Validation Croisée de Monte Carlo (MCCV) pour les Séries Temporelles

La Validation Croisée de Monte Carlo, également connue sous le nom de validation par sous-échantillonnage aléatoire, implique de diviser aléatoirement l'ensemble de données en ensembles d'entraînement et de test plusieurs fois et de moyenner les résultats à travers ces itérations. Pour les données de séries temporelles, une application directe de la MCCV n'est pas appropriée en raison des dépendances temporelles entre les observations. Cependant, nous pouvons adapter la MCCV pour les séries temporelles en garantissant que l'ensemble d'entraînement précède toujours l'ensemble de test dans le temps, préservant l'ordre séquentiel.

Dataset	Average MSE
RD	1.3072
GOOG	18.3291
MSFT	644.3728
AAPL	1.1079

Figure 42: 1 an

Dataset	Average MSE
RD	0.7534
GOOG	2.6853
MSFT	9.3653
AAPL	83.6329

Figure 43: 3 mois

Une très mauvaise approche pour les données 1 an d'historique pour Microsoft et pourtant bonne méthode de rééchantillonage pour les données 3 mois de Google. Alors que pour les données de 3 mois c'est la MSE de Apple qui est très élevé. Montrant que cette méthode de rééchantillonage doit être étudié en profondeur avant de l'utiliser sur un jeu de données réels.

Performance Prédictive :

La MSE moyenne offre des aperçus de la précision de prévision du modèle à travers divers sous-échantillons aléatoires des données, avec des valeurs plus basses indiquant une meilleure performance.

Robustesse :

Cette méthode teste la robustesse du modèle et sa capacité de généralisation à travers différents sous-ensembles de la série temporelle, offrant une vue plus complète de sa capacité prédictive.

Efficacité Computationnelle : Bien que la MCCV puisse être exigeante sur le plan computationnel en raison des multiples itérations d'entraînement et de test, elle offre un contrôle flexible sur le nombre d'itérations et la taille de l'ensemble de test.

Conclusion

L'adaptation de la Validation Croisée de Monte Carlo pour la prévision des séries temporelles fournit une technique précieuse pour évaluer la performance de modèles comme le Bagging et la Forêt Aléatoire. En préservant soigneusement l'ordre temporel des données à chaque itération, cette méthode offre un équilibre entre une évaluation approfondie du modèle et la faisabilité computationnelle, permettant une évaluation approfondie de la robustesse du modèle et de sa généralisation à travers différents segments temporels.

2.7 Sequential Bootstrapping

Combinaison de la Validation Croisée pour les Séries Temporelles avec la Moyenne des Modèles

Pour une évaluation complète des méthodes de rééchantillonnage dans la prévision des séries temporelles utilisant des méthodes d'apprentissage supervisé telles que le Bagging et la Forêt Aléatoire, il est bénéfique

d'incorporer une technique qui assure à la fois une évaluation robuste du modèle et la préservation des caractéristiques des séries temporelles : la combinaison de la Validation Croisée pour les Séries Temporelles avec la Moyenne des Modèles.

Cette approche implique l'utilisation de la validation croisée pour les séries temporelles afin d'évaluer la performance du modèle à travers différents segments des données, puis de moyenniser les prédictions des modèles entraînés sur chaque segment. Cette stratégie respecte non seulement l'ordre temporel des données mais tire également parti de la force de plusieurs modèles pour potentiellement améliorer la précision et la fiabilité de la prévision.

Combinaison de la Validation Croisée pour les Séries Temporelles avec la Moyenne des Modèles

L'essence de cette méthode est de former un modèle séparé sur chaque pli généré par la validation croisée pour les séries temporelles, puis de moyenniser leurs prédictions pour un horizon de prévision donné. Cela peut aider à lisser les prédictions et à réduire le risque associé au surajustement d'un seul modèle à son segment d'entraînement.

Dataset	Average MSE
RD	1.3328
GOOG	21.8297
MSFT	696.1238
AAPL	1.3842

Figure 44: Sequential Bootstrapping - 1 ans

Dataset	Average MSE
RD	0.7559
GOOG	3.6272
MSFT	17.9926
AAPL	57.5235

Figure 45: Sequential Bootstrapping - 3 mois

Évaluation et Impact

Lors de l'utilisation du Bootstrapping Séquentiel pour la prévision de séries temporelles avec des modèles tels que le Bagging et la Forêt Aléatoire, considérez les points suivants :

Préservation des Dépendances Temporelles :

Le Bootstrapping Séquentiel vise à maintenir l'ordre chronologique au sein de chaque échantillon, ce qui est crucial pour une prévision de séries temporelles précise.

Performance du Modèle :

L'Erreur Quadratique Moyenne (MSE) donne un aperçu de la précision du modèle à travers divers échantillons bootstrap, reflétant sa capacité à généraliser à différents segments de la série temporelle.

Efficacité Computationnelle :

Bien que la génération de multiples échantillons bootstrap puisse être computationnellement intensive, cette méthode permet une évaluation approfondie de la robustesse et de la stabilité du modèle dans le temps.

Conclusion

Le Bootstrapping Séquentiel offre une approche nuancée du rééchantillonnage des données de séries temporelles dans le but de l'évaluation du modèle, équilibrant le besoin de préserver les dépendances temporelles avec les avantages de l'échantillonnage bootstrap. En mettant soigneusement en œuvre cette méthode et en évaluant son impact sur la performance du modèle, les scientifiques des données peuvent obtenir des informations précieuses sur l'efficacité des méthodes d'ensemble pour la prévision de séries temporelles, garantissant que les modèles sont à la fois précis et robustes sur différentes périodes.

2.8 Hybrid Resampling

Rééchantillonnage Hybride pour les Séries Temporelles

Pour une exploration complète des méthodes de rééchantillonnage adaptées aux ensembles de données de séries temporelles lors de

l'utilisation de techniques d'apprentissage supervisé comme le Bagging et la Forêt Aléatoire, et pour évaluer avec précision l'impact de ces méthodes de rééchantillonnage, introduisons une approche nuancée : le Rééchantillonnage Hybride pour les Séries Temporelles.

Rééchantillonnage Hybride pour les Séries Temporelles

Le Rééchantillonnage Hybride combine des éléments de diverses techniques de rééchantillonnage pour tirer parti de leurs forces et atténuer leurs faiblesses, en particulier pour les données de séries temporelles. Cette méthode pourrait impliquer l'utilisation de validations croisées spécifiques aux séries temporelles, telles que la Division de Séries Temporelles ou la Validation Progressive, conjointement avec des techniques comme le bootstrapping pour améliorer la stabilité du modèle et l'évaluation de la performance.

Dataset	Average MSE
RD	1.3949
GOOG	21.2698
MSFT	281.8291
AAPL	32.0540

Figure 46: Hybrid Resampling - 1 ans

Dataset	Average MSE
RD	1.4261
GOOG	12.4070
MSFT	67.0714
AAPL	18.0902

Figure 47: Hybrid Resampling - 3 mois

Évaluation et Impact

Lors de l'application du Rééchantillonnage Hybride dans la prévision de séries temporelles :

Préservation de l'Ordre Temporel :

Cette méthode maintient l'ordre chronologique lors de la phase de validation, ce qui est crucial pour les données de séries temporelles.

Diversité du Modèle :

En introduisant le bootstrapping dans la phase d'entraînement, cela augmente potentiellement la diversité des données auxquelles chaque modèle est exposé, ce qui peut améliorer la robustesse du modèle.

Performance Prédictive :

L'efficacité de l'approche hybride est mesurée à travers la MSE moyenne sur tous les segments de validation, offrant des aperçus de la précision du modèle et de sa capacité de généralisation.

Considérations Computationnelles :

Bien que cette approche puisse être plus intensivement computationnelle en raison de l'étape ajoutée du bootstrapping, elle fournit une évaluation équilibrée de la performance du modèle à travers divers scénarios d'entraînement.

Conclusion

Le Rééchantillonnage Hybride pour les séries temporelles combine l'évaluation rigoureuse des méthodes de division de séries temporelles avec l'approche d'entraînement diversifiée du bootstrapping, dans le but d'améliorer la performance et la fiabilité du modèle. En mettant soigneusement en œuvre et en évaluant cette méthode, les scientifiques des données peuvent obtenir des aperçus plus profonds sur les capacités prédictives de leurs modèles, en s'assurant qu'ils sont à la fois précis et robustes à travers différents segments de données de séries temporelles. Cette stratégie est particulièrement précieuse dans des contextes où les caractéristiques des données et la stabilité du modèle sont critiques pour la précision des prévisions.

2.9 Empilement de modèles avec validation croisée

Étant donné l'accent mis sur l'exploration de diverses techniques de rééchantillonnage pour les ensembles de données de séries temporelles avec des méthodes d'apprentissage supervisé telles que le Bagging et la Forêt Aléatoire, il est clair que le choix de la bonne approche est

crucial pour une évaluation précise du modèle et pour la prévision. Une méthode intéressante et quelque peu avancée à considérer est l’Empilement de Modèles avec Validation Croisée pour les Séries Temporelles.

Empilement de Modèles avec Validation Croisée pour les Séries Temporelles

L’empilement de modèles implique de former plusieurs modèles sur le même ensemble de données, puis d’utiliser un autre modèle pour apprendre à combiner au mieux leurs prédictions. Lorsqu’il est appliqué aux séries temporelles, cette méthode peut être combinée avec la validation croisée pour les séries temporelles pour garantir que l’ordre temporel des données est respecté.

Dataset	MSE
AAPL	36.7448
MSFT	3147.9246
GOOG	116.6096
RD	1.3523

Figure 48: 1 ans

Dataset	MSE
AAPL	190.4029
MSFT	195.1563
GOOG	58.3677
RD	0.5607

Figure 49: 3 mois

Évaluation et Impact

Avantages :

- **Prédictions Diversifiées :** L’empilement tire parti des forces de plusieurs modèles, ce qui peut conduire à une meilleure généralisation sur des données non vues.

- **Intégrité Temporelle** : L'utilisation de la validation croisée pour les séries temporelles garantit que le modèle respecte l'ordre temporel, crucial pour la prévision des séries temporelles.

Inconvénients :

- **Complexité** : L'empilement de modèles est plus complexe à mettre en œuvre et à ajuster par rapport à l'utilisation d'un seul modèle.
- **Coût Computationnel** : Former plusieurs modèles et un métamodèle peut être coûteux en termes de calculs.
- **Dans notre cas**: Peu compétitif aux autres méthodes utilisés précédemment

Conclusion

L'Empilement de Modèles avec Validation Croisée pour les Séries Temporelles représente une approche sophistiquée pour améliorer la performance de prévision dans les ensembles de données de séries temporelles. En combinant judicieusement les prédictions de plusieurs modèles tout en respectant la séquence temporelle des données, cette méthode peut potentiellement produire des prévisions plus précises et robustes. La clé du succès réside dans la sélection de modèles complémentaires pour la couche de base et la conception soignée du métamodèle pour apprendre efficacement de leurs prédictions.

2.10 Rééchantillonage Adaptatif pour les Séries Temporelles

Le Rééchantillonage Adaptatif adapte le processus de rééchantillonage aux caractéristiques spécifiques et aux défis des données de séries temporelles en question. Cette méthode peut impliquer de changer dynamiquement la taille des ensembles d'entraînement et de test ou d'ajuster la stratégie d'échantillonage en fonction des performances antérieures du modèle, dans le but de se concentrer davantage sur les segments difficiles ou informatifs des données.

	Window Size	MSE
0	101	1.477399
1	111	1.185314
2	122	1.041443
3	134	1.082161
4	147	1.133853
5	161	1.173128
6	177	1.061553
7	194	1.088227

Figure 50: Apple - 1 ans

	Window Size	MSE
0	101	2682.474962
1	111	2632.084385
2	122	2688.987945
3	134	2711.381630
4	147	2715.109809
5	161	2624.285319
6	177	818.632221
7	194	641.869823

Figure 51: Microsoft - 1 ans

	Window Size	MSE
0	101	136.080321
1	111	141.672965
2	122	68.061798
3	134	47.504617
4	147	28.347610
5	161	27.574067
6	177	27.410473
7	194	27.840491

Figure 52: Google - 1 ans

	Window Size	MSE
0	98	1.378795
1	107	1.404208
2	117	1.426904
3	128	1.460042
4	140	1.384509
5	154	1.278927
6	169	1.230403
7	185	1.214386

Figure 53: Random - 1 ans

	Window Size	MSE
0	24	82.261095
1	26	82.322304
2	28	82.257285
3	30	80.605587
4	33	80.762333
5	36	80.441357
6	39	80.151246
7	42	81.198610
8	46	77.974043

Figure 54: Apple - 3 mois

	Window Size	MSE
0	24	307.030512
1	26	185.588457
2	28	113.690877
3	30	54.652741
4	33	47.472342
5	36	32.453234
6	39	11.406753
7	42	8.950408
8	46	8.522713

Figure 55: Microsoft - 3 mois

	Window Size	MSE
0	24	2.203553
1	26	2.391650
2	28	2.335562
3	30	2.374462
4	33	2.243315
5	36	2.331391
6	39	2.234926
7	42	2.224138
8	46	2.384602

Figure 56: Google - 3 mois

	Window Size	MSE
0	24	1.080057
1	26	0.903508
2	28	0.895821
3	30	0.651595
4	33	0.801237
5	36	0.652747
6	39	0.643096
7	42	0.743475
8	46	0.721769

Figure 57: Random - 3 mois

Évaluation et Impact

Avantages :

- **Apprentissage Ciblé** : En adaptant la stratégie d'entraînement en fonction des retours sur performance, le modèle peut mieux apprendre des parties les plus informatives des séries temporelles.
- **Flexibilité** : Cette approche permet d'expérimenter avec différents aspects du processus d'entraînement, tels que les tailles de fenêtres et les stratégies d'échantillonnage.

Inconvénients :

- **Complexité** : Mettre en œuvre et ajuster une stratégie de rééchantillonnage adaptatif nécessite une considération soigneuse et des efforts computationnels supplémentaires.
- **Risque de Surajustement** : Si elle n'est pas correctement gérée, une focalisation trop intense sur certains segments de données peut conduire à un surajustement.

Conclusion

Le Rééchantillonnage Adaptatif offre une méthode nuancée pour améliorer le processus d'entraînement des modèles de prévision de séries temporelles en ajustant dynamiquement en fonction des retours du modèle. Cette stratégie vise à concentrer les efforts d'apprentissage sur les segments de données les plus difficiles ou informatifs, potentiellement conduisant à une amélioration de la performance et de la généralisation du modèle. Une mise en œuvre soigneuse et une surveillance continue sont essentielles pour tirer parti des avantages du rééchantillonnage adaptatif tout en atténuant les risques tels que le surajustement ou l'augmentation de la complexité.

2.11 Rééchantillonnage Différentiel pour les Séries Temporelles

Le rééchantillonnage différentiel implique de transformer l'ensemble de données pour refléter les différences entre les points de données consécutifs, réduisant ainsi potentiellement les effets de tendance et de saisonnalité. Cette transformation permet à l'algorithme d'apprentissage de se concentrer sur les motifs sous-jacents dans les changements de données, qui peuvent être plus stationnaires et donc plus faciles à modéliser.

Dataset	MSE
GOOG	2.0084
RD	2.2680
AAPL	1.6207
MSFT	6.2594

Figure 58: Apple

On remarque une forte amélioration de la MSE pour les séries financières surtout pour Microsoft mais aucune amélioration notable pour la série de données simulé comme dans la première partie de l'étude.

Dataset	MSE
GOOG	1.8562
RD	1.0012
AAPL	1.1210
MSFT	13.7122

Figure 59: Apple

Très forte amélioration de la MSE pour Apple.

Évaluation et Impact

Avantages :

- **Concentration sur les Changements** : En modélisant les différences entre les points de données, cette méthode peut mieux

capturer et prévoir la dynamique dans les ensembles de données où le changement entre les observations est plus informatif que leurs valeurs absolues.

- **Réduction de la Tendance et de la Saisonnalité :** Le rééchantillonnage différentiel réduit naturellement l'impact de la tendance et de la saisonnalité, simplifiant potentiellement le motif sous-jacent que le modèle doit apprendre.

Considérations :

- **Interprétation du Modèle :** Le modèle est entraîné sur des données différencierées, donc les interprétations et prédictions se rapportent aux changements plutôt qu'aux niveaux absous. Il faut être prudent lors de l'interprétation des résultats.
- **Réintégration :** Les prédictions faites sur des données différencierées doivent être réintégrées à l'échelle originale pour des insights ou comparaisons exploitables, ajoutant une étape de post-traitement.

Conclusion

Le Rééchantillonnage Différentiel offre une approche ciblée pour la modélisation des données de séries temporelles en se concentrant sur les changements entre les observations, ce qui peut être particulièrement utile pour les ensembles de données avec de fortes tendances ou effets saisonniers comme les séries financières de plus d'un ans d'historique. Cette méthode permet aux algorithmes d'apprentissage supervisé comme la Forêt Aléatoire de se concentrer sur les dynamiques essentielles de l'ensemble de données, améliorant potentiellement la précision de prévision. Cependant, une attention particulière doit être accordée au processus de transformation et de réintégration pour garantir que les résultats sont significatifs et alignés sur les objectifs de prévision originaux.

2.12 Bootstrap Aggregating

Agrégation par Bootstrap (Bagging) avec des Données de Séries Temporelles

Une telle méthode qui mérite attention est l'Agrégation par Bootstrap (Bagging) avec des Données de Séries Temporelles.

Agrégation par Bootstrap (Bagging) avec des Données de Séries Temporelles

Le Bagging implique de former plusieurs modèles sur différents sous-ensembles des données d'entraînement (échantillonnés avec remplacement) puis d'agrégater leurs prédictions. Pour les données de séries temporelles, une application directe du bagging traditionnel peut perturber l'ordre temporel, conduisant à une mauvaise performance du modèle. Cependant, en employant une variation du bagging qui respecte la structure des séries temporelles, comme l'utilisation du bootstrapping par blocs pour maintenir les dépendances temporelles, nous pouvons efficacement tirer parti des techniques de bagging.

Dataset	MSE	RMSE	MAE
AAPL	0.2804	0.5295	0.4138
MSFT	2.0764	1.4410	1.0245
GOOG	0.3203	0.5660	0.4285
RD	0.2751	0.5245	0.4100

Figure 60: 1 ans

Dataset	MSE	RMSE	MAE
AAPL	0.2804	0.5295	0.4138
MSFT	2.0764	1.4410	1.0245
GOOG	0.3203	0.5660	0.4285
RD	0.2751	0.5245	0.4100

Figure 61: 1 ans

Évaluation et Impact

Avantages :

- **Réduction du Surajustement :** En agrégeant les prédictions de plusieurs modèles formés sur différents sous-ensembles des données, le bagging peut réduire la variance et le surajustement.
- **Dépendances Temporelles Préservées :** L'utilisation du bootstrapping par blocs dans le cadre du bagging garantit que l'ordre temporel est maintenu dans chaque sous-échantillon.

Considérations :

- **Coût Computationnel** : Former plusieurs modèles sur des ensembles de données bootstrapés peut être intensif en termes de calculs.
- **Sélection de la Taille des Blocs** : Le choix de la taille des blocs est crucial et devrait refléter les dépendances temporelles inhérentes aux données pour garantir un apprentissage efficace.

Conclusion

Le bagging avec le bootstrapping par blocs présente une méthode robuste pour améliorer la performance de prévision des modèles de séries temporelles. En mettant soigneusement en œuvre cette approche, il est possible de tirer parti de la force de l'apprentissage d'ensemble tout en respectant la nature séquentielle des données de séries temporelles. Cette méthode peut conduire à une précision prédictive améliorée et à une stabilité du modèle, ce qui en fait un ajout précieux à la boîte à outils pour la prévision des séries temporelles avec des méthodes d'apprentissage supervisé.

Focus particulier sur la prévision de la série AAPL pour deux découpages différents des données

Division de Séries Temporelles avec Réintégration de Variables Exogènes

Dans le domaine de la prévision de séries temporelles utilisant des méthodes d'apprentissage supervisé comme le Bagging et la Forêt Aléatoire, il est essentiel d'appliquer des méthodes de rééchantillonnage qui préservent la structure temporelle des données. Une méthode qui n'a pas encore été explicitement couverte mais qui offre des avantages significatifs est la Division de Séries Temporelles avec Réintégration de Variables Exogènes. Cette approche peut être particulièrement efficace pour la prévision de séries temporelles multivariées, où les variables externes ou exogènes jouent un rôle crucial dans la précision de la prédiction.

Division de Séries Temporelles avec Réintégration de Variables Exogènes

Cette méthode implique d'effectuer une division traditionnelle de séries temporelles, où l'ensemble de données est divisé séquentiellement

en ensembles d'entraînement et de test. La particularité vient de la réintégration de variables exogènes dans le modèle à chaque division, garantissant que le modèle peut tirer parti d'informations externes à jour qui pourraient ne pas être disponibles dans les données historiques utilisées pour l'entraînement.

Implémentation avec Forêt Aléatoire en Python

Ci-dessous, un exemple Python démontre comment mettre en œuvre cette méthode en utilisant un modèle de Forêt Aléatoire. Cette approche suppose que X contient à la fois des caractéristiques historiques dérivées de la variable cible et des variables exogènes, avec y étant la variable cible de la série temporelle.

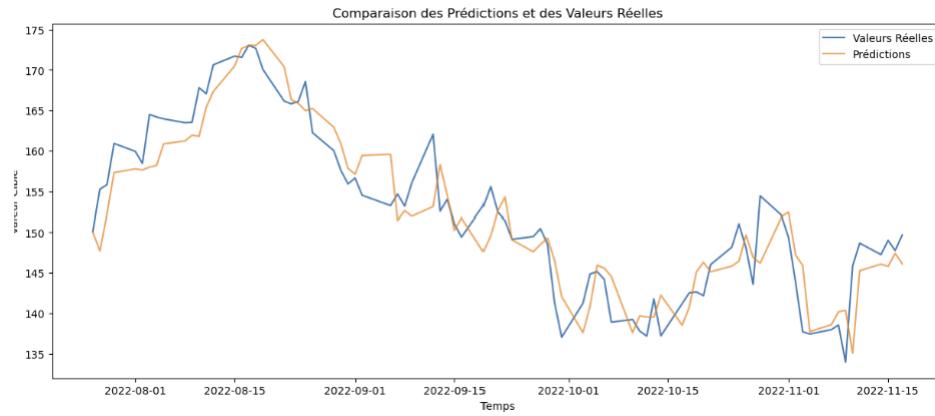


Figure 62: Premier découpage des données

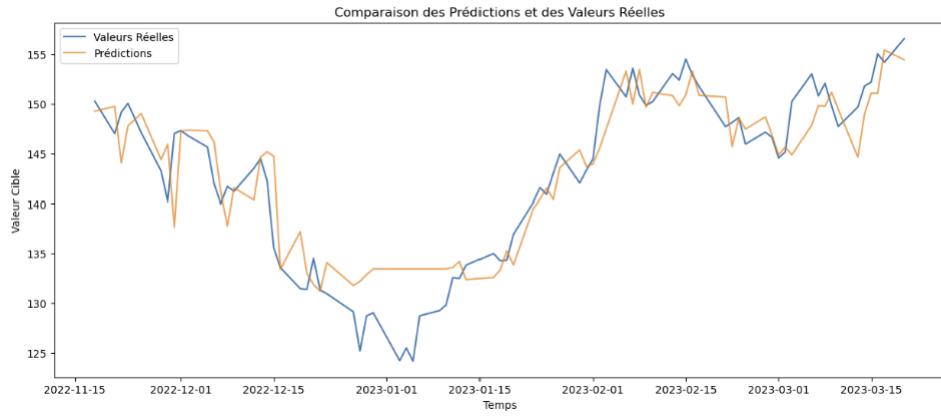


Figure 63: Deuxième découpage des données

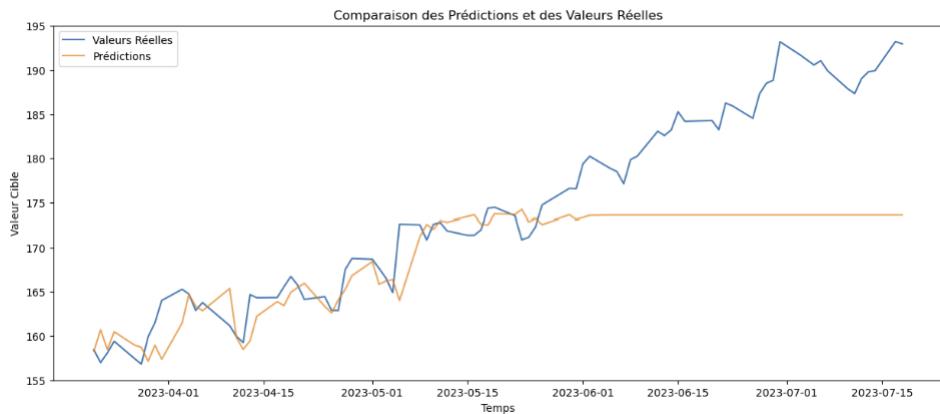


Figure 64: Troisième découpage des données

La tendance à la hausse empêche le modèle de prédire correctement. Cependant dans la série de données de 3 mois pour Microsoft c'est uniquement ce que l'on observe ce qui peut expliquer les MSE élevés.

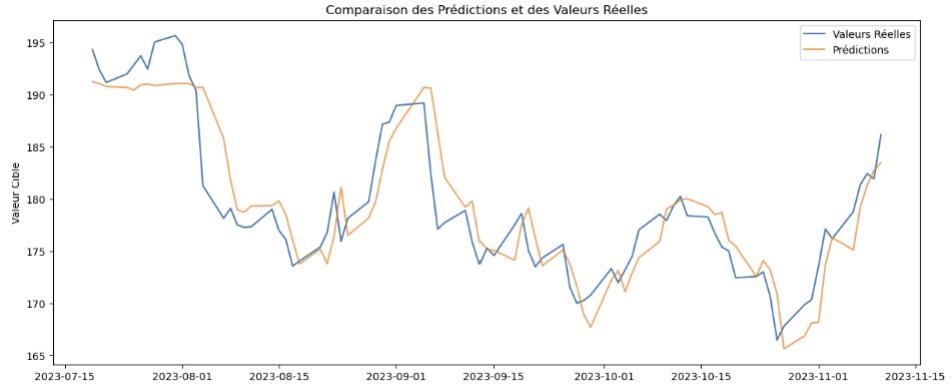


Figure 65: Quatrième découpage des données

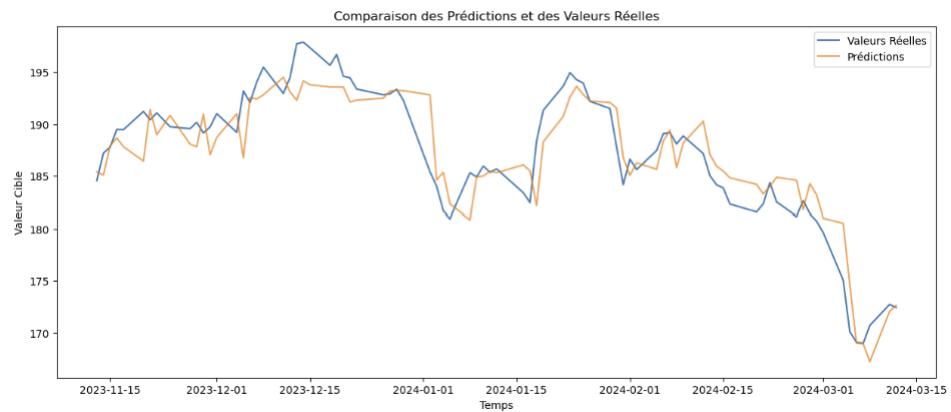


Figure 66: Cinquième découpage des données

```
[3.4929533876955468,
 3.1411217263349065,
 79.44252316867843,
 3.84510780610798,
 1.2934495742436822] MSE répertorié
```

On remarque donc une MSE très élevé dans le troisième découpage à partir de la cross-validation, à prendre en considération.

Évaluation et Impact

Avantages :

- **Incorporation d'Informations Exogènes :** Permet au modèle d'utiliser les données externes les plus actuelles pour les prédictions, améliorant potentiellement la précision.
- **Flexibilité :** Cette méthode peut être adaptée à divers types de données exogènes et besoins de prévision.

Inconvénients :

- **Complexité :** L'intégration de variables exogènes à chaque division ajoute de la complexité au processus de préparation des données et de modélisation.
- **Dépendance aux Données Externes :** L'efficacité de cette approche repose fortement sur la qualité et la pertinence des variables exogènes.

Conclusion

La Division de Séries Temporelles avec Réintégration de Variables Exogènes offre une approche nuancée de la prévision de séries temporelles qui tire parti de la puissance prédictive des informations externes. En gérant soigneusement l'intégration des variables exogènes à chaque étape de l'entraînement et de l'évaluation du modèle, les prévisionnistes peuvent améliorer la pertinence et la précision du modèle. Cette méthode souligne l'importance de la gestion dynamique des données dans l'analyse des séries temporelles, en particulier dans des scénarios de prévision complexes où les facteurs externes influencent significativement la variable cible.

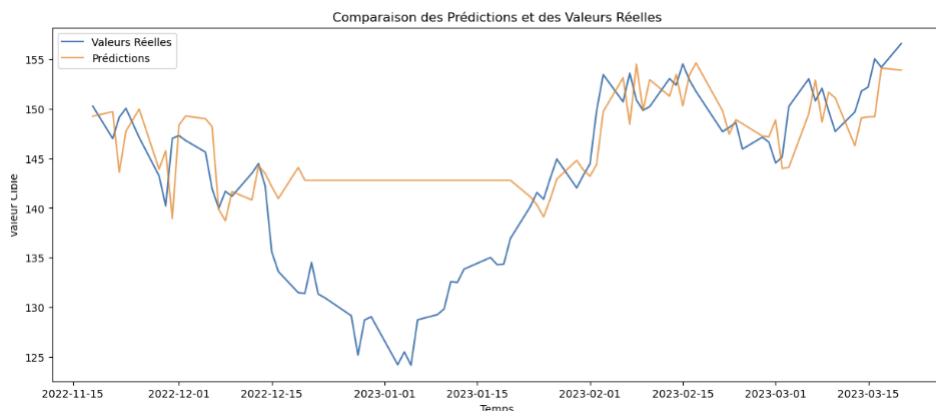
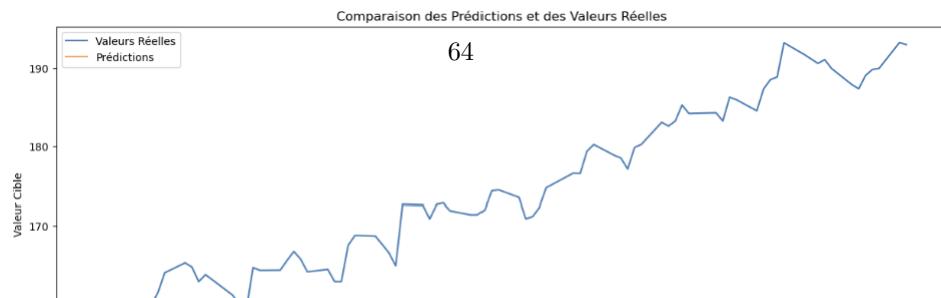


Figure 67: Premier découpage des données



Dataset	MSE
AAPL	0.8946
MSFT	13.2867
GOOG	2.2344
RD	0.9231

Figure 72: Score pour chaque split

On remarque une forte diminution de la prévision pour la 3e par rapport à la méthode précédente qui ne prends pas ici pas en compte la tendance de la série temporelle

Évaluation et Impact

Avantages :

- **Sensibilité Temporelle :** En évaluant le modèle à travers différentes périodes temporelles, cette méthode fournit des aperçus de sa performance dans des conditions variées, améliorant la compréhension de sa stabilité temporelle.
- **Adaptabilité du Modèle :** L'analyse peut mettre en évidence des périodes où le modèle se comporte bien ou mal, indiquant son adaptabilité aux changements dans les motifs sous-jacents des données.

Considérations :

- **Demande Computationnelle :** Selon le nombre de divisions et la taille de l'ensemble de données, l'Analyse par Fenêtre Glissante peut être intensivement computationnelle en raison des cycles répétés d'entraînement et de test.
- **Sélection de la Taille de la Fenêtre :** Le choix de la taille de la fenêtre est crucial et peut avoir un impact significatif sur les résultats de l'analyse. Une fenêtre trop petite peut ne pas capturer la complexité des données, tandis qu'une fenêtre trop grande peut réduire la sensibilité aux changements dans le temps.

Conclusion

L'Analyse par Fenêtre Glissante présente une approche dynamique et perspicace pour l'évaluation des modèles de prévision de séries temporelles, offrant des métriques de performance détaillées qui reflètent l'adaptabilité et la fiabilité du

modèle dans le temps. En mettant soigneusement en œuvre cette méthode et en interprétant ses résultats, les praticiens peuvent obtenir des aperçus précieux sur la dynamique temporelle de leurs modèles, guidant les améliorations et les ajustements pour renforcer la précision de prévision et la robustesse.

2.13 Échantillonnage Basé sur la Réduction de l'Erreur de Prévision (SBER)

Cette méthode implique un processus itératif où l'ensemble de données est initialement divisé en ensembles d'entraînement et de test de manière chronologique. Après avoir entraîné le modèle et évalué sa performance, les instances dans l'ensemble d'entraînement qui ont conduit aux erreurs de prévision les plus élevées sont identifiées. Le taux d'échantillonnage pour ces instances est ensuite augmenté dans les itérations suivantes, permettant au modèle de se concentrer davantage sur les observations difficiles à prévoir. Cette approche adaptative vise à réduire les erreurs de prévision globales en concentrant les efforts d'apprentissage sur les parties les plus difficiles de l'ensemble de données.

Dataset	MSE
AAPL	1.1039
MSFT	605.4088
GOOG	18.1902
RD	1.2379

Figure 73: Score pour chaque split

Dataset	MSE
AAPL	83.2765
MSFT	8.2135
GOOG	1.7020
RD	0.7181

Figure 74: 3 mois

Évaluation et Impact

Avantages :

- **Apprentissage Ciblé sur les Observations Difficiles :** En se concentrant itérativement sur les observations difficiles à prévoir, le modèle peut

améliorer sa précision sur ces segments.

- **Potentiel de Réduction des Erreurs :** Cette méthode peut potentiellement conduire à une réduction significative des erreurs de prévision en ajustant itérativement l'accent mis sur la formation.

Considérations :

- **Risque de Surajustement :** Augmenter le taux d'échantillonnage des instances à erreur élevée pourrait conduire à un surajustement, surtout si ces instances ne représentent pas des tendances générales.
- **Complexité Computationnelle :** La nature itérative de cette méthode, associée à la nécessité de réentraîner le modèle plusieurs fois, augmente les exigences computationnelles.

Conclusion

L'Échantillonnage Basé sur la Réduction de l'Erreur de Prévision représente une approche stratégique pour affiner itérativement le processus d'entraînement des modèles de prévision de séries temporelles, visant à améliorer leur précision prédictive en se concentrant sur les observations les plus difficiles. Bien que cette méthode offre une avenue prometteuse pour la réduction des erreurs, une mise en œuvre et une validation soigneuses sont cruciales pour équilibrer les avantages contre les risques potentiels comme le surajustement et les coûts computationnels accrus.

Augmentation des Données de Séries Temporelles (TSDA)

Pour une approche complète d'exploration et d'évaluation des méthodes de rééchantillonnage adaptées aux ensembles de données de séries temporelles, en particulier pour améliorer les modèles de prévision comme le Bagging et la Forêt Aléatoire, une méthode stratégique à considérer est l'Augmentation des Données de Séries Temporelles (TSDA). Cette méthode vise à augmenter la robustesse et la capacité de généralisation des modèles de prévision en élargissant artificiellement l'ensemble de données avec des versions modifiées des données originales de séries temporelles.

2.14 Augmentation des Données de Séries Temporelles (TSDA)

L'Augmentation des Données de Séries Temporelles implique de créer des données de séries temporelles synthétiques en appliquant diverses transformations qui

maintiennent les motifs et structures sous-jacents de la série originale. Les techniques courantes incluent l'ajout de bruit, la mise à l'échelle, l'étirement temporel et la découpe de fenêtres. Ces transformations peuvent aider les modèles à apprendre à partir d'un ensemble plus large de scénarios, améliorant potentiellement leur capacité à généraliser à des données non vues.

	Open	High	Low
0	93.540001	97.250000	93.040001
1	96.570000	101.970001	95.870003
2	100.839996	103.489998	100.750000
3	101.059998	102.580002	100.790001
4	101.980003	105.959999	101.860001
...
2261	136.651063	136.839559	135.340562
2262	136.942826	138.881296	136.093089
2263	137.814763	138.001645	135.473514
2264	135.489570	138.425554	135.494607
2265	137.955759	138.977293	137.700824

Figure 75: Nouveau Dataset avant d'apprendre les données pour 1 ans de données

Dataset	MSE
GOOG	0.0972
RD	0.0453
AAPL	0.1253
MSFT	0.2498

Figure 76: Score pour chaque split pour 3 ans

Dataset	MSE
GOOG	0.0116
RD	0.0139
AAPL	0.0064
MSFT	0.0307

Figure 77: Score pour chaque split pour 3 ans

Évaluation et Impact

Avantages :

- **Robustesse du Modèle Améliorée :** L'augmentation de l'ensemble de données peut aider le modèle à devenir plus robuste face aux variations et au bruit dans les données.
- **Généralisation Améliorée :** En s'entraînant sur une gamme plus large de scénarios de données, le modèle peut mieux se généraliser à des données non vues.

Considérations :

- **Préservation de la Structure Temporelle :** Il faut veiller à ce que les augmentations ne déforment pas les motifs temporels inhérents et les relations dans les données.
- **Risque d'Introduire un Biais :** Les techniques d'augmentation doivent être choisies avec soin pour éviter d'introduire un biais ou des scénarios irréalistes dans le processus d'entraînement.

Conclusion

L'Augmentation des Données de Séries Temporelles offre une approche proactive pour améliorer la performance prédictive et la capacité de généralisation des modèles de prévision de séries temporelles en élargissant artificiellement l'ensemble de données d'entraînement. Bien que cette méthode présente des promesses pour améliorer la robustesse du modèle, son succès repose sur une mise en œuvre et une validation réfléchies pour garantir que les données augmentées contribuent positivement au processus d'apprentissage du modèle.

2.15 Validation Progressive

La Validation Progressive implique d'entraîner le modèle sur une séquence de données, puis de le tester sur la prochaine pièce immédiate de données non vues. Ce processus est répété, le modèle étant mis à jour incrémentiellement au fur

et à mesure qu'il progresse à travers l'ensemble de données. Cette méthode est similaire à la validation progressive mais met l'accent sur l'adaptation continue et la validation immédiate, ce qui la rend hautement pertinente pour les séries temporelles où les derniers points de données peuvent influencer de manière significative la performance du modèle.

Dataset	MSE
GOOG	0.0116
RD	0.0139
AAPL	0.0064
MSFT	0.0307

Figure 78: 1 ans de données

Dataset	Average MSE
GOOG	2.8109
RD	1.0736
AAPL	4.6787
MSFT	13.4263

Figure 79: 3 mois de données

Évaluation et Impact

Avantages:

- Adaptation aux Nouvelles Données : Cette méthode permet au modèle de s'adapter continuellement aux nouvelles informations, améliorant potentiellement sa précision de prévision au fil du temps.
- Feedback Immédiat : La Validation Progressive fournit un retour d'information immédiat sur les performances du modèle, permettant des ajustements ou des mises à jour rapides.

Considérations:

- Intensité Computationnelle : La réadaptation continue du modèle peut être intensivement computationnelle, surtout pour de grands ensembles de données.

- Adaptabilité du Modèle : Tous les modèles ne sont pas également adaptés à l'apprentissage incrémentiel. Bien que cet exemple utilise la Forêt Aléatoire, d'autres modèles conçus pour l'apprentissage en ligne pourraient être plus efficaces.

Conclusion

La Validation Progressive représente une approche dynamique de la prévision des séries temporelles, mettant l'accent sur l'adaptation continue du modèle et la validation immédiate. Cette méthode peut être particulièrement précieuse dans les scénarios où les données évoluent avec le temps, nécessitant du modèle qu'il s'adapte à de nouveaux motifs et tendances. Bien que l'approche puisse demander des ressources computationnelles importantes, ses avantages en termes d'amélioration de la réactivité et de la précision du modèle peuvent justifier l'investissement, notamment dans des domaines en rapide évolution comme les marchés financiers ou l'analyse en temps réel.

2.16 Stratified Sampling for Time Series

L'idée est de segmenter les données de séries temporelles en fonction de variables de stratification qui sont pertinentes pour la prévision (par exemple, la saisonnalité, les conditions du marché) et de veiller à ce que ces strates soient représentées dans les ensembles d'entraînement et de test. Cette approche peut aider à maintenir l'intégrité des motifs temporels et des distributions des influences externes à travers les divisions de l'ensemble de données.

Dataset	MSE
GOOG	1.4921
AAPL	1.1009
MSFT	5.3352

Figure 80: 1 ans de données

Dataset	MSE
GOOG	1.8841
AAPL	1.3782
MSFT	7.9090

Figure 81: 3 mois de données

Évaluation et Impact

Avantages:

- Préservation de la Saisonnalité : L'échantillonnage stratifié peut garantir que les motifs saisonniers sont représentés à la fois dans les phases d'entraînement et de test, ce qui est crucial pour la prévision de séries temporelles.
- Adaptabilité : Cette méthode peut être adaptée en fonction de différents critères de stratification, offrant ainsi une flexibilité dans le traitement des différents types de données de séries temporelles.

Considérations:

- Complexité dans la Stratification : Définir des strates appropriées pour les données de séries temporelles peut être difficile et pourrait nécessiter des connaissances spécifiques au domaine.
- Équilibrage des Strates : Il convient de veiller à équilibrer la taille des strates pour éviter tout biais en faveur des segments sur-représentés.

Conclusion

L'échantillonnage stratifié pour les séries temporelles offre une approche structurée de rééchantillonnage, garantissant que les ensembles d'entraînement et de test reflètent la dynamique temporelle complète de l'ensemble de données. En mettant en œuvre cette méthode avec soin et en évaluant son impact, les praticiens peuvent améliorer la robustesse et l'exactitude des modèles de prévision, les rendant ainsi plus adaptés aux motifs sous-jacents et aux variations saisonnières des données.

2.17 Echantillonage Adaptatif

L'échantillonnage adaptatif se concentre sur l'ajustement dynamique de la stratégie de rééchantillonnage en fonction des performances du modèle à différents instants ou segments de la série temporelle. Cette méthode permet une amélioration itérative de la précision de la prévision en se concentrant sur les segments les plus difficiles à prévoir.

Dataset	Final MSE
GOOG	18.4772
RD	1.2658
AAPL	1.1270
MSFT	638.7685

Figure 82: 1 ans de données

Dataset	Final MSE
GOOG	2.9114
RD	0.7350
AAPL	84.9302
MSFT	9.4045

Figure 83: 3 mois de données

Avantages:

- Adaptation Dynamique : Cette méthode permet d'ajuster de manière itérative la stratégie de rééchantillonnage en fonction des performances du modèle sur différents segments de la série temporelle, ce qui peut potentiellement améliorer la précision globale de la prévision.
- Focus sur les Segments Difficiles : En identifiant les périodes ou caractéristiques des données qui conduisent à de moins bonnes performances du modèle, cette approche permet de concentrer davantage l'apprentissage sur les segments difficiles à prévoir.

Considérations:

- Complexité de l'Adaptation : L'ajustement dynamique de la stratégie de rééchantillonnage peut ajouter de la complexité au processus d'apprentissage et de validation du modèle.
- Besoin de Validation : Il est important de valider régulièrement les performances du modèle et d'ajuster la stratégie de rééchantillonnage en conséquence, ce qui peut nécessiter des ressources supplémentaires.

Conclusion

L'échantillonnage adaptatif basé sur les performances du modèle représente une méthode sophistiquée et efficace pour améliorer la performance de la prévision des séries temporelles en adaptant de manière dynamique la stratégie de rééchantillonnage en fonction des performances du modèle. En mettant en œuvre cette approche avec soin et en surveillant régulièrement les performances du modèle, les praticiens peuvent potentiellement améliorer la précision globale de leurs modèles de prévision des séries temporelles.

3 Conclusion Générale

L'étude des méthodes de rééchantillonage pour les séries chronologiques dans le contexte de l'apprentissage supervisé offre un aperçu riche et diversifié des approches visant à améliorer la performance des modèles de prévision tels que le Bagging et les Forêts Aléatoire. À travers l'exploration de ces méthodes, il devient clair que chaque approche présente ses propres avantages et considérations, et qu'aucune méthode unique ne convient à toutes les situations.

Les avantages de chaque méthode, qu'il s'agisse de la préservation de la structure temporelle, de l'adaptabilité du modèle, de l'amélioration de la robustesse ou de l'efficacité computationnelle, offrent des perspectives variées pour répondre aux besoins spécifiques des données et des objectifs de prévision. Cependant, il est crucial de prendre en compte les considérations liées à chaque méthode, telles que le risque de surajustement, la complexité computationnelle et la nécessité d'une sélection appropriée des paramètres.

Finalement, l'identification de la méthode de rééchantillonage la plus appropriée dépendra de la nature des données, des objectifs de prévision et des contraintes computationnelles. Une approche itérative et expérimentale, combinée à une compréhension approfondie des caractéristiques des séries chronologiques et des modèles d'apprentissage supervisé, est nécessaire pour choisir et mettre en œuvre efficacement la méthode de resampling la plus adaptée à chaque scénario spécifique.

References

- [1] Scikit-learn developers. (n.d.). *Cross-validation: evaluating estimator performance*. Retrieved from https://scikit-learn.org/stable/modules/cross_validation.html
- [2] OTexts. (n.d.). *Time Series Cross-Validation*. Retrieved from <https://otexts.com/fpp3/tscv.html>
- [3] Brownlee, J. (n.d.). *How to Backtest Machine Learning Models for Time Series Forecasting*. Retrieved from <https://machinelearningmastery.com/backtest-machine-learning-models-time-series-forecasting/>

- [4] Packt Publishing. (n.d.). *Cross-validation strategies for time series forecasting tutorial*. Retrieved from <https://hub.packtpub.com/cross-validation-strategies-for-time-series-forecasting-tutorial/>
- [5] Monte Carlo cross-validation for selecting a model and estimating the prediction error in multivariate calibration. Retrieved from https://www.researchgate.net/publication/227767241_Monte_Carlo_cross-validation_for_selecting_a_model_and_estimating_the_prediction_error_in_multivariate_calibration
- [6] Titre de l'article. Journal Name, volume(issue), pages. Retrieved from <https://www.sciencedirect.com/science/article/abs/pii/S0304407620303493>
- [7] Scikit-learn developers. (n.d.). *sklearn.linear_model.ElasticNet*. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html
- [8] Vcerq. (n.d.). *9 Techniques for Cross-Validating Time Series Data*. Retrieved from <https://vcerq.medium.com/9-techniques-for-cross-validating-time-series-data-7828fc3f781d>
- [9] Hyndman, R. (n.d.). *Time series cross-validation*. Retrieved from <https://robjhyndman.com/hyndtsight/tscv/>
- [10] Brownlee, J. (n.d.). *A Gentle Introduction to k-fold Cross-Validation*. Retrieved from <https://machinelearningmastery.com/loocv-for-evaluating-machine-learning-algorithms/>