Travaux Dirigés Notés – 3A_DDEFI :
Prédiction des Prix et Actifs du S&P500 grâce à du Machine Learning simple

Groupe 02 :
Nolan Bouigue, Rémi Chabo, Maxime Falanga, Joseph Rigaut, Sibylle Lehmann

I. Introduction et objectifs

Les marchés financiers sont des systèmes complexes, influencés par une multitude de facteurs économiques, techniques et comportementaux. Parmi ces marchés, le S&P 500, un indice regroupant les 500 plus grandes entreprises cotées aux États-Unis, joue un rôle clé en tant qu'indicateur de la performance du marché américain et, plus largement, de l'économie mondiale. Analyser et prévoir les tendances de cet indice est un défi important, qui suscite un intérêt constant de la part des différents acteurs.

Aujourd'hui, grâce au machine learning, il est possible d'exploiter de grandes quantités de données financières pour tenter de prédire les prix et les rendements des actifs. Ces outils permettent de détecter des relations complexes dans les données et d'obtenir des modèles capables d'anticiper certaines dynamiques de marché. Dans ce projet, nous allons utiliser des approches simples de machine learning pour explorer la prédiction des prix et actifs du S&P 500.

Nos objectifs sont les suivants :

- Collecter et préparer les données :
    - Rassembler les prix historiques et d'autres données économiques importantes, comme le VIX ou les taux d'intérêt.
    - Nettoyer et traiter ces données afin qu'elles soient exploitables pour les modèles de prédiction.
- Développer un modèle prédictif :
    - Implémenter différents modèles en utilisant des algorithmes de machine learning adaptés.
- Évaluer les modèles :
    - Vérifier l'efficacité des modèles avec des indicateurs comme MAE et RMSE, et voir lesquels donnent les meilleurs résultats.
- Analyser les résultats :
    - Comprendre les prédictions faites par les modèles et proposer des recommandations pour investir en fonction des résultats.

II. Méthodologie (préparation des données, choix des modèles)

A. Variables sélectionnées

Notre étude sera bornée dans le temps. Nous récupérons ainsi les données journalières sur les 5 dernières années du S&P500 via les variables suivantes :

start_date = '2019-12-19' / end_date = '2024-12-19' / time_interval = '1d'

Nous avons téléchargé les données suivantes via l'API Yahoo finance :

- Le **prix de clôture ajusté** et le **volume du S&P 500**, représentent respectivement la valeur finale ajustée de l'indice en fin de journée, tenant compte des événements corporatifs, et le nombre total de transactions effectuées sur ses actions durant la même période.
- L'indice **VIX**, indice de volatilité, mesure les attentes du marché en matière de volatilité future sur 30 jours pour le S&P 500, basé sur les options

Puis nous avons compléter ces données à l'aides des indicateurs économiques suivants afin de pouvoir par la suite modéliser les évolutions du S&P500 :

- **Taux de chômage (UNRATE)**, Le taux de chômage représente le nombre de chômeurs en pourcentage de la population active. Les données sur la population active sont limitées aux personnes âgées de 16 ans et plus, qui résident actuellement dans l'un des 50 États ou dans le district de Columbia, qui ne résident pas dans des institutions (par exemple, établissements pénitentiaires et psychiatriques, maisons pour personnes âgées) et qui ne sont pas en service actif dans les forces armées (publié mensuellement par la FED)
- **Taux des fonds fédéraux (FEDFUNDS)**, le taux des fonds fédéraux est le taux d'intérêt auquel les institutions de dépôt échangent des fonds fédéraux (soldes détenus auprès des banques de la Réserve fédérale) entre elles au cours d'une nuit (publié mensuellement par la FED)
- **Indice des prix à la consommation (CPIAUCSL)**, est un indice des prix d'un panier de biens et de services payés par les consommateurs urbains (publié mensuellement par la FED)
- **Ventes au détail (RSAFS)**, les ventes mensuelles anticipées pour le commerce de détail et les services de restauration fournissent des estimations préliminaires des ventes mensuelles pour les entreprises dans les secteurs du commerce de détail et des services de restauration (publié mensuellement par la FED)
- **Indice de confiance des consommateurs (UMCSENT)** , l'indice de confiance des consommateurs de l'Université du Michigan est un indice de confiance des consommateurs (publié mensuellement par l'Université du Michigan).
- **Indice de production industrielle (INDPRO)**, l'indice de la production industrielle mesure la production réelle de tous les établissements concernés situés aux États-Unis, indépendamment de leur propriété (publié mensuellement par la FED)
- **taux d'intérêt des bons du Trésor à 10 ans (DGS10)**, rendement à 10 ans estimé à partir des rendements moyens d'une variété de titres du Trésor à différentes échéances, dérivés de la courbe des rendements du Trésor.

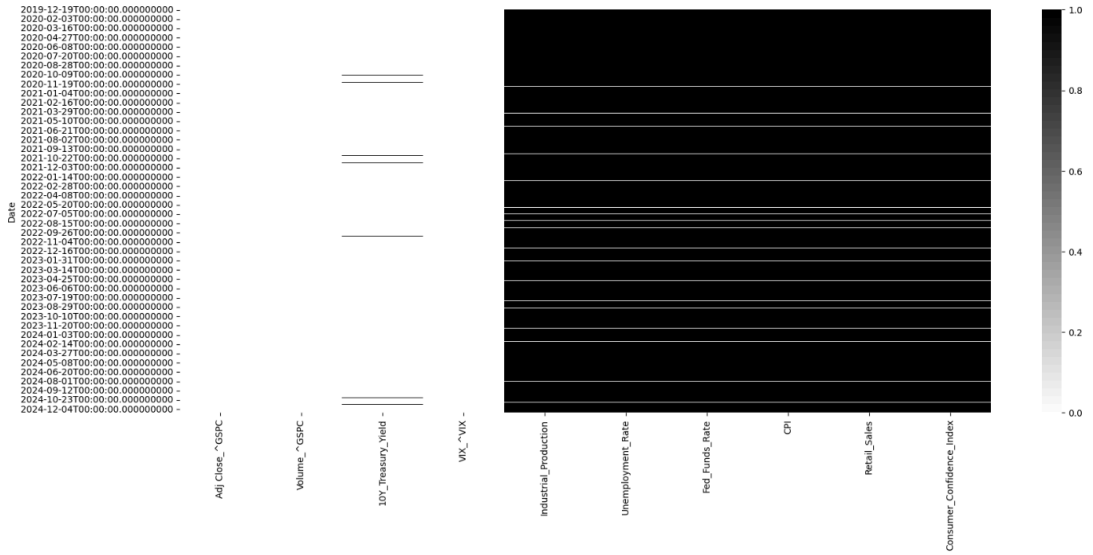
Les variables associées sont les suivantes :

Indice économique	Variable
prix de clôture ajusté	Adj Close_^GSPC
volume	Volume_^GSPC

VIX	VIX_^VIX
Taux de chômage (UNRATE)	Unemployment_Rate
Taux des fonds fédéraux (FEDFUNDS)	Fed_Funds_Rate
Indice des prix à la consommation (CPIAUCSL)	CPI
Ventes au détail (RSAFS)	Retail_Sales
Indice de confiance des consommateurs (UMCSENT)	Consumer_Confidence_Index
Indice de production industrielle (INDPRO)	Industrial_Production
Intérêts bons du Trésor à 10 ans (DGS10)	10Y_Treasury_Yield

Nous avons choisi de ne prendre que quelques variables macroéconomique américaine que l'on peut retreouver dans l'article de Gasparėnienė : « A Modelling of S&P 500 Index Price Based on U.S. Economic Indicators: Machine Learning Approach »[1]. Ici au lieu de prendre les 3-month Treasure bill nous avons choisi les bons du trésor à 10 ans afin d'améliorer la prédictibilité à moyen-terme du modèle.

Après avoir réuni ces données dans une même table nommée data, nous avons visualisé les données manquantes :



Heatmap des données manquantes - Python seaborn

Les données manquantes pour les bons du trésor ont été interpolées et les données mensuelles ont été transformées en données journalières grâce aux fonctions ffill et bfill.

Après ce premier traitement, les valeurs nous obtenons 1258 valeurs non nulles :

[1] Gasparėnienė, Ligita, et al. « A Modelling of S&P 500 Index Price Based on U.S. Economic Indicators: Machine Learning Approach ». Engineering Economics, vol. 32, no 4, octobre 2021, p. 362-75. inzeko.ktu.lt, https://doi.org/10.5755/j01.ee.32.4.27985.

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1258 entries, 2019-12-19 to 2024-12-18
Data columns (total 10 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Adj Close_^GSPC            1258 non-null   float64
 1   Volume_^GSPC               1258 non-null   int64
 2   10Y_Treasury_Yield         1258 non-null   float64
 3   VIX_^VIX                   1258 non-null   float64
 4   Industrial_Production      1258 non-null   float64
 5   Unemployment_Rate          1258 non-null   float64
 6   Fed_Funds_Rate             1258 non-null   float64
 7   CPI                        1258 non-null   float64
 8   Retail_Sales               1258 non-null   float64
 9   Consumer_Confidence_Index  1258 non-null   float64
dtypes: float64(9), int64(1)
memory usage: 108.1 KB
```

information sur les données - Python pandas

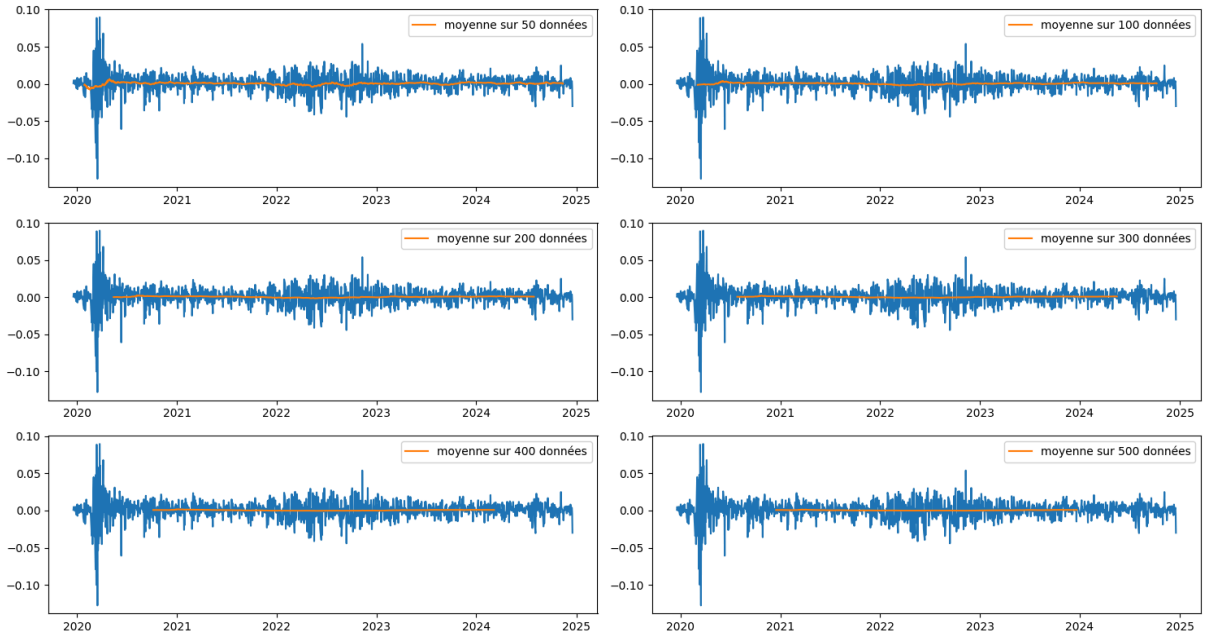
B. Variable cible

Par la suite, afin de prédire les rendements du S&P 500, nous avons créé une variable cible (que l'on souhaite prédire) :

```
# Calcul des rendements logarithmiques
data['Log_Returns'] = np.log(data['Adj Close_^GSPC'] / data['Adj Close_^GSPC'].shift(1))

# remplacer la première valeur NaN dans la colonne créée
data['Log_Returns'] = data['Log_Returns'].fillna(data['Log_Returns'].median())
```

En sachant qu'un modèle ARIMA va être utilisé par la suite, nous cherchons à savoir s'il existe une saisonnalité dans les données, afin d'éventuellement effectuer une différenciation :



Recherche de saisonnalité avec différentes moyennes mobiles - Python matplotlib

Nous n'observons pas de saisonnalité et nous utiliserons donc un modèle ARMA pour la modélisation par la suite.

Par ailleurs, grâce à un test de stationnarité (ADF - Augmented Dickey-Fuller test), nous rejetons l'hypothèse nulle de non-stationnarité de la série avec une p-value de :

p-value = 6.5168231600469545e-19.

La série est donc stationnaire et nous pourrons utiliser un modèle ARMA pour modéliser nos log-rendements.

C. Variables calculées

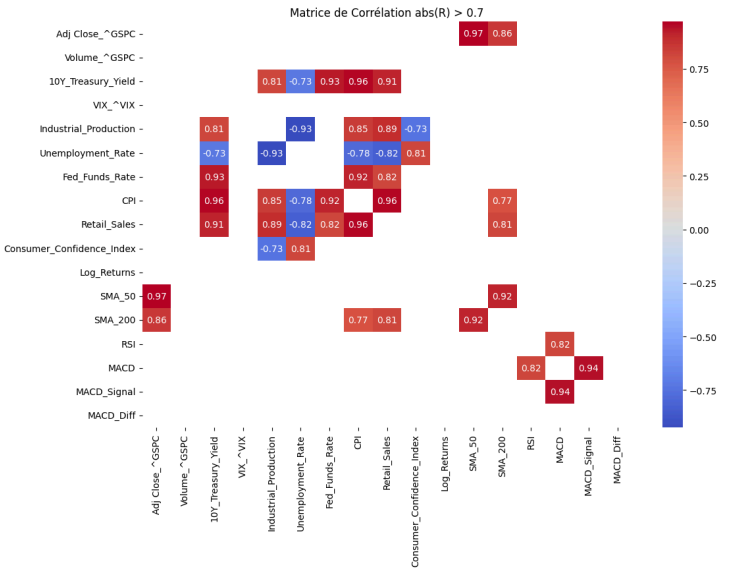
Nous avons enfin créé les variables :

- Moyenne mobile sur 50 et 200 données des log-rendements, **SMA_50** et **SMA_200**.
- **MACD** mesure l'élan de la tendance, calculée comme la différence entre une moyenne mobile exponentielle (EMA) rapide (sur 12 périodes) et une EMA lente (sur 26 périodes)
- **MACD_Signal** aide à repérer les signaux de retournement et représente la ligne de signal, qui est une EMA de la ligne MACD (généralement sur 9 périodes)
- **MACD_Diff** calculé comme la différence entre la ligne MACD et la ligne de signal facilitant l'interprétation graphique
- **RSI** (Relative Strength Index), mesure la vitesse et l'amplitude des mouvements de prix récents sur une échelle de 0 à 100 sur une période de 14 jours.

Les données manquantes dans ces colonnes (les n premières valeurs utilisées pour le calcul des moyennes mobiles) ont été remplacées par les valeurs moyennes.

D. Choix des variables

Une première analyse de corrélation a été faite pour voir l'interdépendance entre les différentes variables présentes dans notre jeu de données « data ».



Matrice de corrélation |R| > 0,7 - Python pandas

Puis, afin de sélectionner plus précisément les variables les plus importantes, nous utilisons la méthode RFE (Recursive Feature Elimination). Les variables sélectionnées sont les suivantes (nous avons choisi de ne prendre en compte que 5 variables afin d'obtenir des résultats plus facilement interprétables d'un point de vue économique) :

- 'Adj Close_^GSPC',
- '10Y_Treasury_Yield',
- 'Unemployment_Rate',
- 'Fed_Funds_Rate',
- 'Consumer_Confidence_Index'

Il est a noté que les variables sélectionnées ne sont pas celle qui obtiennent les meilleurs scores avec la méthode « SelectKBest » notamment à cause des corrélations lors de la modélisation de la régression linéaire avec l'algorithme RFE) :

Scores des features avec SelectKBest:

	Feature	Score
0	Adj Close_^GSPC	4.989837e+15
9	SMA_50	1.716521e+04
10	SMA_200	2.903410e+03
6	CPI	7.193717e+02
7	Retail_Sales	7.048655e+02
3	VIX_^VIX	5.454870e+02
5	Fed_Funds_Rate	3.604260e+02
2	10Y_Treasury_Yield	3.214506e+02
13	MACD_Signal	1.660940e+02
12	MACD	1.519688e+02
11	RSI	1.035453e+02
1	Volume_^GSPC	9.456049e+01
4	Unemployment_Rate	9.446997e+01
8	Consumer_Confidence_Index	3.851444e+00
14	MACD_Diff	1.469552e+00

Score des variables avec la méthode SelectKBest - Python sklearn

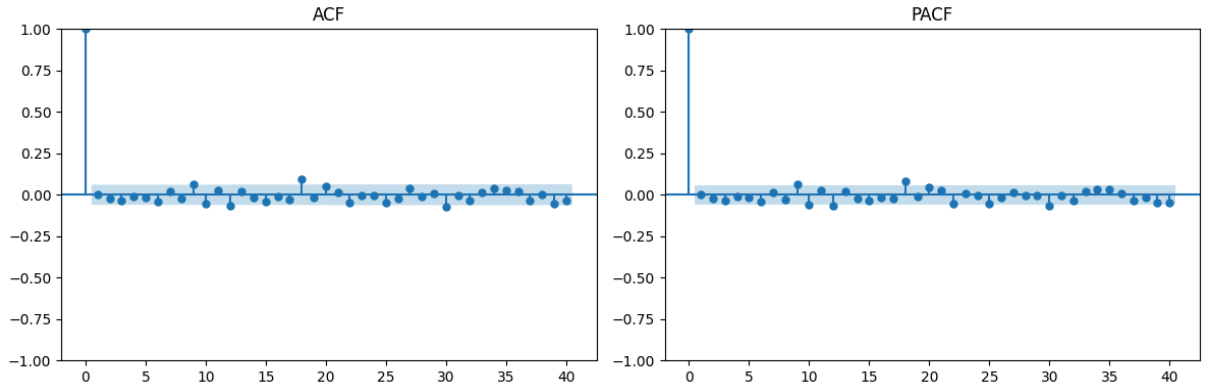
E. Choix des modèles

Dans un premier temps, nous avons utilisé un modèle de régression linéaire afin de modéliser et de prédire les rendements. La régression linéaire permet de comprendre directement l'impact des variables explicatives sur les rendements. Ce modèle est simple à implémenter, rapide à entraîner et fournit une base de comparaison robuste pour évaluer d'autres modèles peut être plus complexes. De plus, les données financières sont naturellement organisées dans le temps, ce qui exige des modèles capables de gérer les dépendances temporelles. La série des log-rendements montre une stationnarité d'où l'utilisation adaptée de la régression linéaire comme premier modèle.

Le second modèle utilisé est un modèle ARIMA (AutoRegressive Integrated Moving Average) spécifiquement conçu pour modéliser les séries temporelles, ce qui le rend idéal pour capturer les dépendances temporelles dans les données financières. Contrairement aux modèles de deep learning comme LSTM, ARIMA est plus simple à paramétrer et nécessite moins de données pour produire des résultats fiables. Il est également plus adapté dans un contexte où l'objectif est d'explorer et de comprendre les dépendances temporelles classiques. Par rapport à des modèles comme SARIMA ou Prophet, ARIMA est une solution

plus directe lorsqu'il s'agit de séries univariées sans composante saisonnière explicite.

Pour déterminer les paramètres p,q et d du modèle ARIMA, nous faisons une lecture de la fonction d'autocorrélation (ACF) et de la fonction d'autocorrélation partielle (PACF) des log-rendements.



Comme la série des log-rendements est stationnaire, la différenciation est inutile donc d=0. Une première lecture de l'ACF et de la PACF nous permet de proposer un modèle ARMA(2,3) sans beaucoup de précision. Nous utilisons donc la fonction *auto_arima* permettant de proposer le meilleur modèle ARMA possible. Celle-ci propose le modèle ARMA(3,2). On peut ensuite comparer les métriques et les p-values de ces deux modèles.

SARIMAX Results

Dep. Variable:	Log_Returns	No. Observations:	1059
Model:	ARIMA(2, 0, 3)	Log Likelihood	3333.377
Date:	Thu, 02 Jan 2025	AIC	−6652.754
Time:	16:02:08	BIC	−6617.998
Sample:	0	HQIC	−6639.581
	− 1059		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0005	0.000	1.686	0.092	−8.56e−05	0.001
ar.L1	−0.2378	0.548	−0.434	0.664	−1.311	0.836
ar.L2	0.5560	0.368	1.510	0.131	−0.166	1.278
ma.L1	0.2107	0.554	0.381	0.704	−0.875	1.296
ma.L2	−0.6042	0.351	−1.721	0.085	−1.292	0.084
ma.L3	−0.0031	0.046	−0.067	0.946	−0.093	0.087
sigma2	0.0001	3.51e−06	30.786	0.000	0.000	0.000

Ljung-Box (L1) (Q):	0.83	Jarque-Bera (JB):	178.28
Prob(Q):	0.36	Prob(JB):	0.00
Heteroskedasticity (H):	0.72	Skew:	−0.34
Prob(H) (two-sided):	0.00	Kurtosis:	4.89

ARMA(2,3)

SARIMAX Results

Dep. Variable:	y	No. Observations:	1058
Model:	SARIMAX(3, 0, 2)	Log Likelihood	3319.227
Date:	Thu, 02 Jan 2025	AIC	−6626.454
Time:	16:02:40	BIC	−6596.669
Sample:	0	HQIC	−6615.165
	− 1058		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	−0.7617	0.217	−3.509	0.000	−1.187	−0.336
ar.L2	−0.0363	0.033	−1.105	0.269	−0.101	0.028
ar.L3	−0.0624	0.027	−2.282	0.022	−0.116	−0.009
ma.L1	−0.2234	0.211	−1.061	0.289	−0.636	0.189
ma.L2	−0.7479	0.209	−3.583	0.000	−1.157	−0.339
sigma2	0.0001	3.66e−06	30.118	0.000	0.000	0.000

Ljung-Box (L1) (Q):	0.07	Jarque-Bera (JB):	157.70
Prob(Q):	0.79	Prob(JB):	0.00
Heteroskedasticity (H):	0.71	Skew:	−0.13
Prob(H) (two-sided):	0.00	Kurtosis:	4.87

ARMA(3,2)

Les deux modèles ont des AIC, BIC et Log Likelihood semblables cependant le modèle obtenu avec la fonction *auto_arima* possède davantage de coefficients significatifs. On poursuivra donc par la suite avec le modèle ARMA(3,2).

En ce qui concerne l'entraînement et la validation des modèles, nous avons utilisé une répartition classique de 80% pour l'entraînement et 20% pour le test. Donc 80% des données sont utilisées pour entraîner le modèle et lui permettre d'apprendre les relations entre les variables et 20% des données sont réservées pour évaluer la performance du modèle sur des données jamais vues pendant l'entraînement. 80/20 est un ratio standard qui assure un bon compromis entre la taille de l'ensemble d'entraînement, suffisamment grand pour garantir un apprentissage efficace, et la taille de l'ensemble de test, suffisamment représentatif pour évaluer les performances du modèle. Ce ratio est particulièrement intéressant dans notre cas car il est crucial que les données d'entraînement soient représentatives des conditions du marché.

De plus, nous avons utilisé une validation croisée temporelle. Cette technique permet de diviser les données en plusieurs fenêtres temporelles glissantes. Chaque fenêtre est utilisée pour entraîner un modèle, qui est ensuite testé sur les données immédiatement suivantes. Par exemple:Fenêtre 1: entraînement sur les données de t_1 à t_k, test sur t_{k+1} à t_{k+n}

Fenêtre 2: entraînement sur les données de t_2 à t_{k+1}, test sur t_{k+2} à t_{k+n+1}

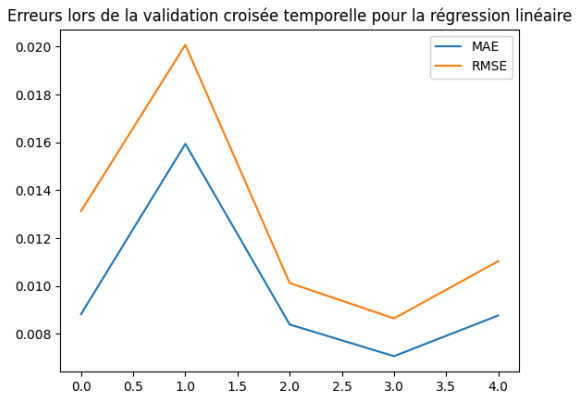
Et ainsi de suite.

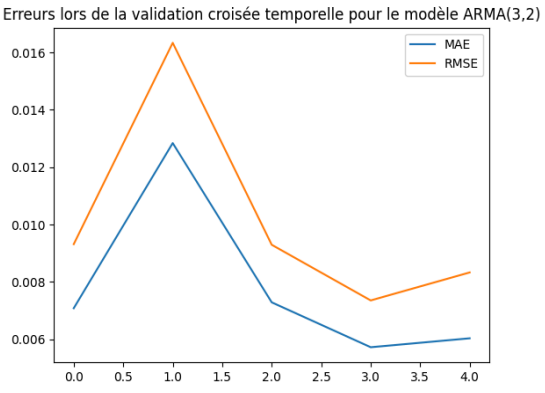
Chaque modèle est évalué indépendamment, et les performances moyennes sont calculées sur toutes les fenêtres. Cette méthode est particulièrement adaptée aux séries temporelles. Elle respecte l'ordre temporel des données et garantit qu'aucune information future n'est utilisée pour prédire des valeurs passées. Dans une modélisation de données financières comme celles-ci, les observations futures ne doivent pas influencer l'entraînement du modèle. On veut modéliser un scénario réel où les décisions sont prises en utilisant uniquement des informations disponibles à un instant donné. De plus, les marchés financiers évoluent constamment et une validation croisée temporelle aide à mesurer la robustesse du modèle à travers différentes périodes et en s'intéressant à la moyenne des performances.

Nous avons utilisé comme métrique la Mean Absolute Error (MAE) et la Root Mean Squared Error (RMSE) pour évaluer la performance des modèles. Ces deux métriques, courantes, permettent d'apprécier l'erreur entre les prédictions d'un modèle et les valeurs réelles. La MAE mesure l'erreur moyenne absolue entre les valeurs prédites et les valeurs réelles.

La RMSE mesure l'erreur quadratique moyenne entre les valeurs prédites et les valeurs réelles. La RMSE est particulièrement utile pour analyser l'impact des grandes erreurs dans les prédictions. En la combinant avec la MEA, on obtient une vue équilibrée des performances du modèle, en tenant compte à la fois des erreurs moyennes et des erreurs extrêmes.

Comme nous avons utilisé la méthode de validation croisée temporelle, nous avons donc fait la moyenne de la RMSE et la MEA de chaque modèle sur chaque fenêtre temporelle.





III. Résultats et analyse

Lors de l'utilisation de la validation croisée à 5 blocs, on a pu observer des erreurs quadratiques moyennes entre 0,8% et 1,6% pour la régression linéaire et entre 0,6% et 1,3% pour le modèle ARMA (3,2). On en déduit que pour la modélisation et la prévision des

résultats du S&P500 que notre modèle ARMA(3,2) sera plus adapté que le modèle de régression linéaire.

Concernant la régression linéaire, un des facteurs pouvant expliquer une moins bonne précision que le modèle ARMA est la forte corrélation des variables. En effet, dans un modèle de régression avec les moindres carrés ordinaires, la forte corrélation des variables explicatives entre elles conduit à une sous-performance du modèle. Il serait préférable dans cas d'utiliser une méthode de régression de type LASSO ou RIDGE afin de ne plus être affecté par ces problèmes de corrélations. Par ailleurs, dans le cas de modélisation de séries temporelles, la régression linéaire peut s'avérer limitée pour modéliser un phénomène souvent non linéaire.

Concernant le modèle ARMA, la méthode se reposant sur l'analyse des graphiques de corrélation et de corrélation partielle montre rapidement ses limites dans la précision. Ici, l'utilisation de la fonction auto-ARIMA nous permet d'obtenir un modèle dont les coefficients ont des p-value meilleures que le modèle ARMA(2,3) choisi au départ. Seuls deux coefficients ont une p-value supérieure au seuil de confiance de 5%. Par ailleurs, les résidus sont homoscédastiques et le test de Ljung-Box avec une p-value de 0,79 permet de ne pas rejeter l'hypothèse nulle de non-corrélation des données (les résidus sont indépendants).

Enfin, les données prises incluent la période du Covid-19 qui a impacté de manière significative l'économie américaine et mondiale. Le données du passées, notamment la période 2019 à 2021, ne sont donc pas forcément représentatives des tendances habituelles du marché conduisant à une baisse du pouvoir de prédictibilité du modèle.

IV. Conclusion et recommandations

1. **Stratégies d'investissement basées sur l'ARMA(3,2) :**

Utiliser les prédictions des rendements log pour identifier des phases de marché favorable (rendements positifs attendus) ou défavorables (rendements négatifs attendus). Une stratégie possible est de prendre des **positions longues** lorsque des rendements positifs sont prédits et des **positions courtes** lorsque des rendements négatifs sont attendus. Ces choix doivent être fait également en se basant sur les indicateurs RSI et MACD.

2. **Données et horizons temporels :**

Réévaluer le modèle en utilisant des données post-Covid-19 pour obtenir des prédictions basées sur des tendances économiques plus stables et étendre les horizons temporels pour inclure des périodes de marché plus variées, ce qui permettrait de généraliser les modèles.

3. **Intégration d'autres facteurs :**

Continuer à inclure des variables macroéconomiques pour affiner les prédictions, mais évaluer leur pertinence à travers des analyses de causalité (ex : test de Granger). Par ailleurs, des variables exogènes pourraient être ajoutées pour capturer des chocs inattendus (géopolitiques, climatiques, etc.).

Par ailleurs, certains articles comme celui de Chongda Liu[2]conseillent d'intégrer les indicateurs des autres places boursières comme le Nikkei daily return, USDJPY daily return, NASDAQ daily return ou les prix des matières premières (Gold price daily return et Crude oil daily return) afin d'améliorer les anticipations. Cela est dû à une forte interaction des places boursières due à la mondialisation.

[2] Liu, Chongda, et al. « Forecasting S&P 500 Stock Index Using Statistical Learning Models ». Open Journal of Statistics, vol. 6, no 6, novembre 2016, p. 1067‑75. www.scirp.org, https://doi.org/10.4236/ojs.2016.66086.