Prédiction du prix de l'indice CAC40 : Méthodes ARIMA & LSTM

Préparé par :

CLEMENT Landy
MAROUF Chaimaa
OUTADA Mouad

PLAN

1. **Choix et description de notre base de données**
2. **Méthodes et modélisation**
3. **Conclusion**
4. **Annexes**

1. Choix et description de notre base de données

L'objectif de ce projet est de mettre en œuvre différents modèles de prévision pour prédire le prix de l'indice CAC 40 en se basant sur différents modèles de séries temporelles (ARIMA, LSTM...).
L'indice CAC 40 est le premier indice boursier de la Bourse de Paris, créée en 1987 et composé de 40 sociétés françaises sélectionnées sur la base de la combinaison des deux classements suivants :

- Le classement du montant des capitaux échangés sur le marché réglementé observé sur douze mois ;
- Le classement de la capitalisation boursière flottante à la Date de révision.

2. Méthodes et modélisation

- Premièrement, nous avons commencé par l'importation des données (voir annexe 1), ainsi que la série des prix mensuels de l'indice CAC40 (voir annexe 2) sujet de notre prédiction. Ensuite, nous allons procéder à la visualisation des données (voir annexe 3).

- A noter : Une série chronologique stationnaire est une série dont les propriétés ne dépendent pas du moment où la série est observée. Ainsi, les séries temporelles avec des tendances, ou avec une saisonnalité, ne sont pas stationnaires. D'autre part, une série de bruit blanc est stationnaire - peu importe le moment où vous l'observez, elle devrait avoir à peu près la même apparence à tout moment.
  La stationnarité est importante car de nombreux outils analytiques, tests statistiques et modèles utiles en dépendent.
  De ce fait, nous avons vérifié la stationnarité des données de prix du CAC40 et nous avons commencé par décomposer le signal (voir annexe 4).

- Nous avons vérifié l'autocorrélation et l'autocorrélation partielle : A partir du graphique ACF, nous avons pu constater qu'il n'y a pas de saisonnalité dans les données et nous avons remarqué que les 25 premiers lags ont une corrélation significative avec la série originale. Nous remarquons que le premier retard contribue de manière importante à la série originale. Les autres retards contribuent de manière très faible (voir annexe 5).

- Ensuite, il a été nécessaire de vérifier la stationnarité avec le test de Dickey-Fuller (seuil de 5%) et nous avons obtenu les résultats suivants :

```
Results of Dickey-Fuller Test:
Test Statistic                -1.216250
p-value                        0.666567
#Lags Used                     0.000000
Number of Observations Used    385.000000
dtype: float64
As the p-value is not less than our threshold of 5% we fail to reject the null-hypothesis, thus this series is non-stationary
```

Comme la valeur p n'est pas inférieure à notre seuil de 5%, nous ne rejetons pas l'hypothèse nulle, donc cette série est non stationnaire.

- Rendre la série stationnaire en différenciant son logarithme (voir annexe 5) et nous avons eu les résultats ci-dessous :

```
Test Statistic                -1.879905e+01
p-value                        2.022968e-30
#Lags Used                     0.000000e+00
Number of Observations Used    3.840000e+02
dtype: float64
```

Comme la valeur p est inférieure à 1%, nous rejetons l'hypothèse nulle, donc la différence logarithmique est stationnaire avec 99% de confiance.

- Deuxièmement, nous allons faire une prédiction des séries temporelles avec ARIMA :

ARIMA est l'acronyme de AutoRegressive Integrated Moving Average (dans ce contexte, l'intégration" est l'inverse de la différenciation). Le modèle complet peut être écrit comme suit :

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t,$$

Où Yt' est la série différenciée (elle peut avoir été différenciée plus d'une fois). Les "prédicteurs" du côté droit comprennent à la fois les valeurs décalées de Yt' et les erreurs décalées. Nous appelons cela un modèle ARIMA(p,d,q), où :

p = ordre de la partie autorégressive ;
d = degré de première différenciation impliqué ;
q = ordre de la partie moyenne mobile.

Il y a de nombreux modèles abordés dans la prévision des séries temporelles ne sont que des cas particuliers du modèle ARIMA (voir annexe 7)

- Ensuite, nous avons préparé nos données pour la prévision et nous ajusterons nos paramètres p, d et q à l'aide de la fonction auto_arima afin d'ajuster le meilleur modèle et de faire des prévisions.
- Nous avons procédé à la division des données, en choisissant un ratio de division.
- Ajustement du modèle.

- Faire les prédictions.
- Evaluation du modèle ARIMA.
- Traçage des prédictions vs le test (voir annexe 8) : Nous avons remarqué que les prédictions sont presque linéaires et capturent la tendance mais pas le bruit parce que nous essayons de prédire plusieurs étapes à l'avance. Dans le prochain modèle, nous essaierons de prédire un pas en avant sur la base d'une fenêtre d'entraînement.
- Troisièmement, nous avons procédé à la prévision des séries chronologiques avec les LSTM. Long Short Term Memory networks (LSTMs) LSTM) sont un type particulier de RNN, capable d'apprendre des dépendances à long terme.

Le problème des réseaux neuronaux récurrents est qu'ils disposent d'une mémoire à court terme pour retenir les informations précédentes dans le neurone actuel. Cependant, cette capacité diminue très rapidement pour les séquences plus longues. Pour y remédier, les modèles LSTM ont été introduits afin de pouvoir retenir les informations antérieures encore plus longtemps.

Le problème des réseaux neuronaux récurrents est qu'ils stockent simplement les données précédentes dans leur "mémoire à court terme". Une fois la mémoire épuisée, ils suppriment simplement les informations retenues depuis le plus longtemps et les remplacent par de nouvelles données.

Le modèle LSTM tente d'échapper à ce problème en ne conservant que des informations sélectionnées dans la mémoire à court terme. Cette mémoire à court terme est stockée dans ce qu'on appelle l'état cellulaire. En outre, il existe également l'état caché, que nous connaissons déjà grâce aux réseaux neuronaux normaux.
Les LSTM sont généralement utilisés pour reconnaître des motifs dans des séquences de données, tels que ceux qui apparaissent dans les données des capteurs, les cours de la bourse ou le langage naturel.

- Et de la même manière, nous avons commencé par la préparation des données : choisir les étapes pour le retour en arrière.
- Ensuite, Construction du modèle LSTM.
- Ajuster le modèle LSTM.
- Tracer la perte (voir annexe 10).
- Procéder à la prédiction par LSTM (voir annexe 11).

3. Conclusion

Comme nous pouvons le voir, le MAE est très faible, mais nous ne pouvons pas comparer avec le MAE du modèle précèdent car ce n'est pas le même type de prévision, car dans la dernière méthode nous prévoyons une étape à la fois par contre au début nous avons procédé ç la prévoyance de nombreuse étapes à l'avance.

4. Annexes

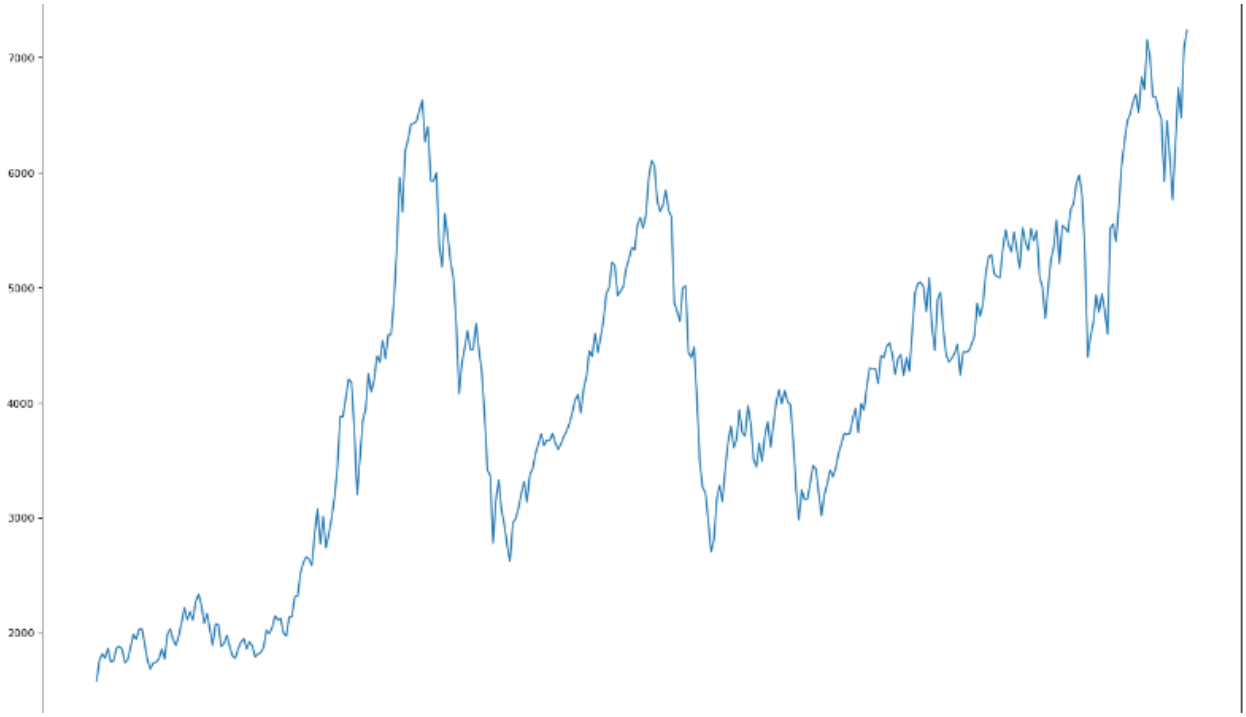
```
import numpy as np
import pandas as pd
import time
import datetime
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import statsmodels.api as sm
from statsmodels.graphics.tsaplots import plot_acf
from statsmodels.graphics.tsaplots import plot_pacf
from pylab import rcParams
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.arima.model import ARIMA
import pmdarima
from pmdarima import auto_arima
from  sklearn.metrics import mean_absolute_error
import datetime as dt
import tensorflow as tf
from tensorflow import keras
from sklearn.preprocessing import MinMaxScaler
import datetime as dt
from sklearn.metrics import mean_absolute_error
```

Annexe 1 : Importation des données nécessaires

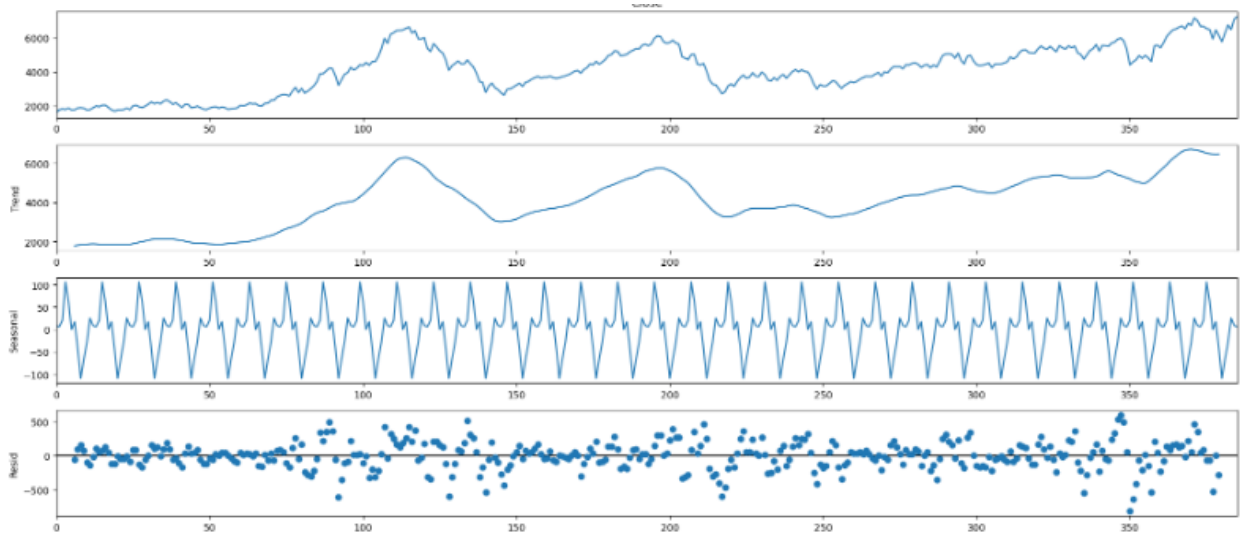
	Date	Open	High	Low	Close	Adj Close	Volume
0	1991-01-01	1504.000000	1600.000000	1425.000000	1580.000000	1580.000000	0
1	1991-02-01	1577.000000	1762.000000	1565.000000	1760.000000	1760.000000	0
2	1991-03-01	1746.000000	1855.000000	1726.000000	1816.000000	1816.000000	0
3	1991-04-01	1815.000000	1859.000000	1762.000000	1779.000000	1779.000000	0
4	1991-05-01	1793.000000	1864.400024	1785.500000	1861.800049	1861.800049	0
...	...	...	...	...	...	...	...
381	2022-10-01	5697.470215	6293.149902	5654.439941	6266.770020	6266.770020	1530909300
382	2022-11-01	6329.759766	6743.600098	6191.729980	6738.549805	6738.549805	1663727000
383	2022-12-01	6784.600098	6823.100098	6388.229980	6473.759766	6473.759766	1448945200
384	2023-01-01	6521.069824	7117.529785	6518.209961	7082.419922	7082.419922	1248137000
385	2023-02-01	7087.200195	7233.939941	7059.609863	7233.939941	7233.939941	249589900

386 rows × 7 columns

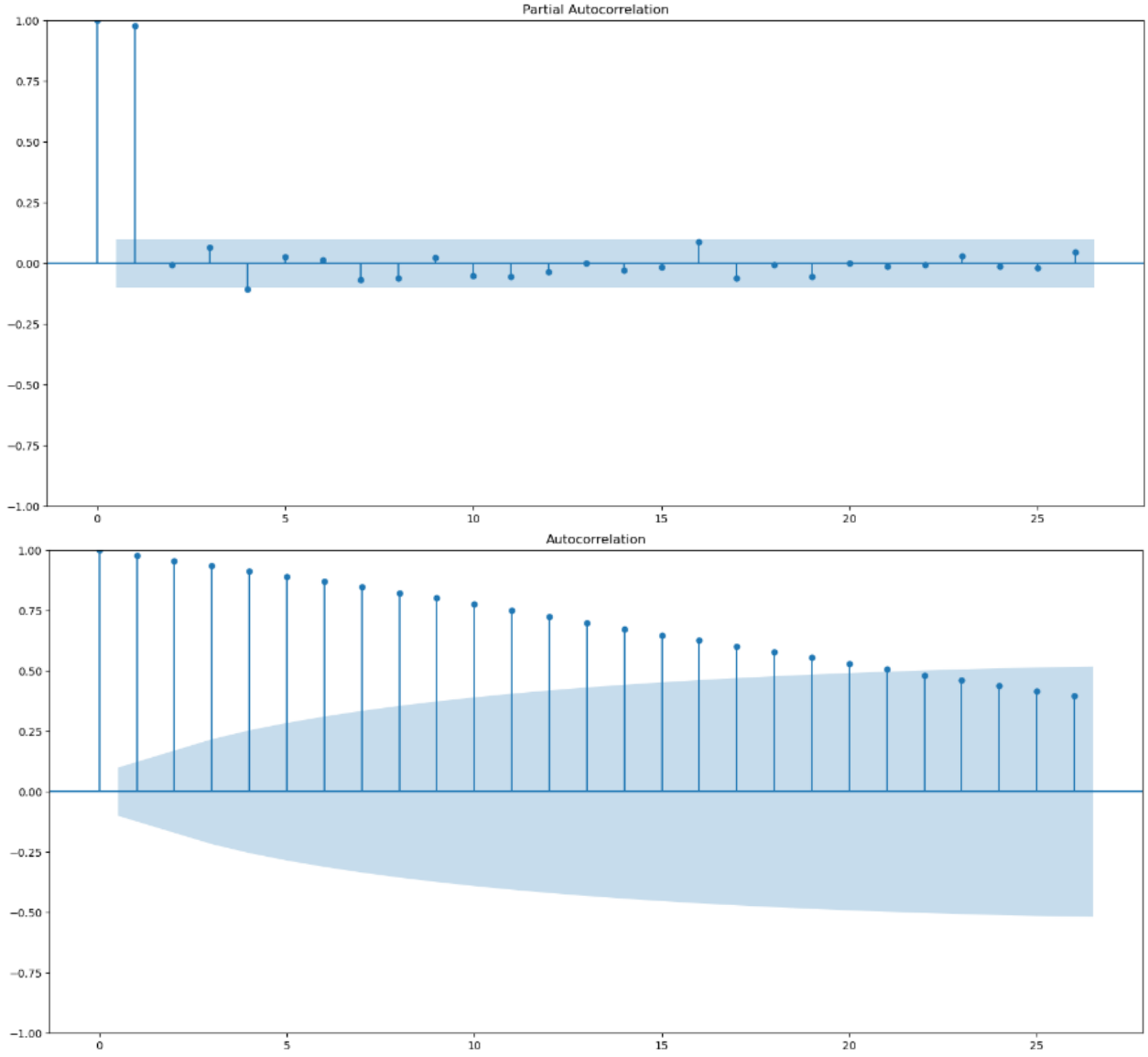
Annexe 2 : Série temporelle de l'indice CAC40



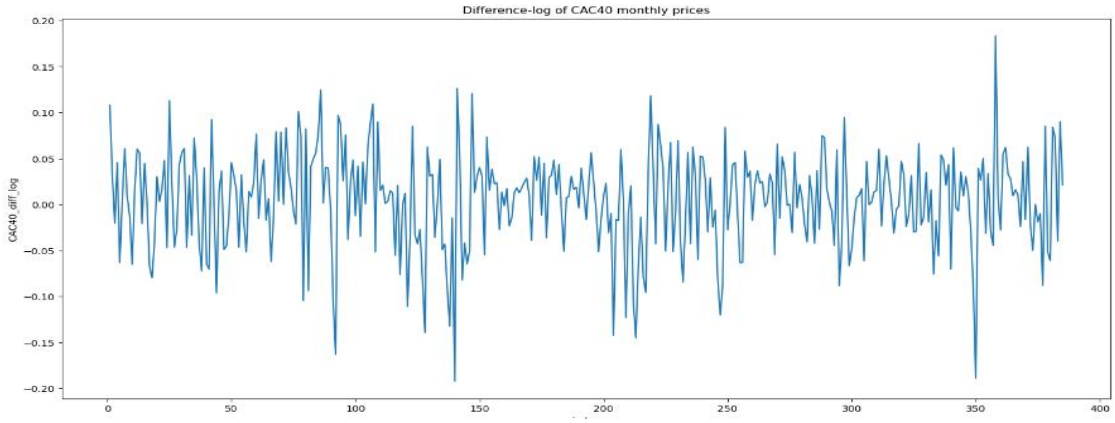
Annexe 3 : visualisation des données



Annexe 4 : Décomposition du signal



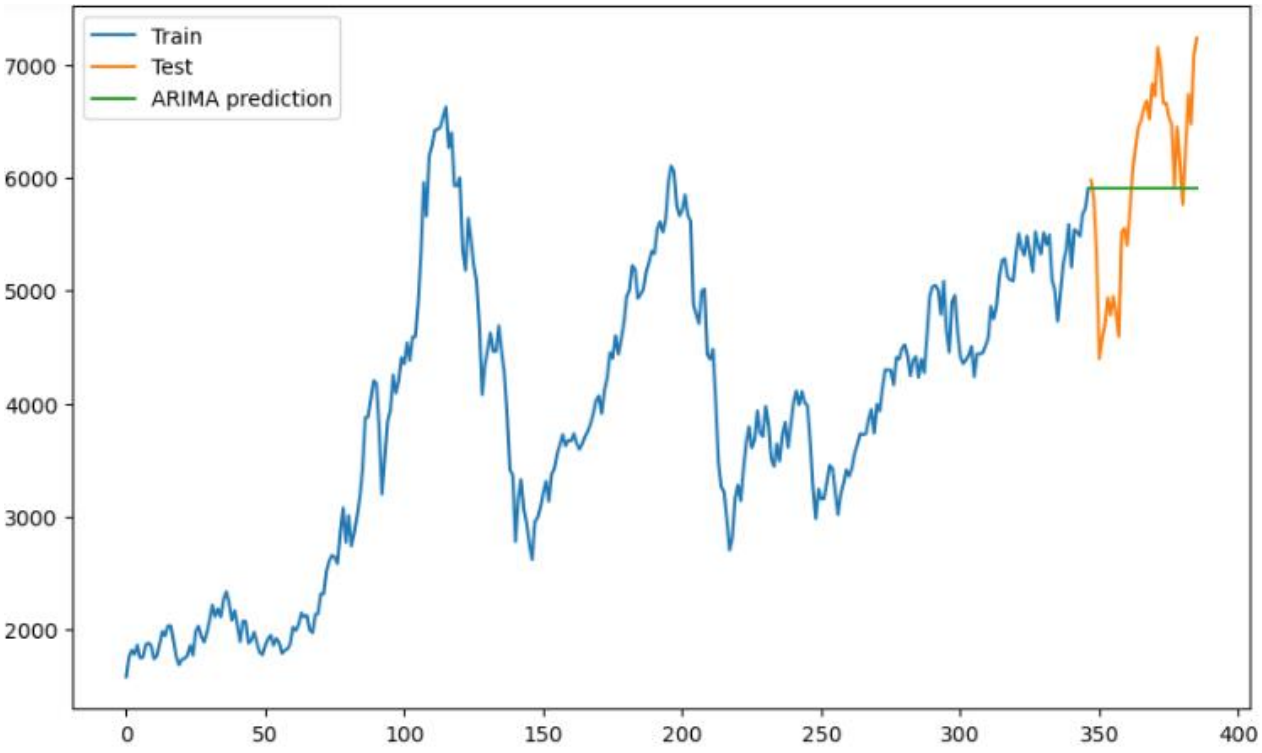
Annexe 5 : Vérification de l'autocorrélation et l'autocorrélation partielle



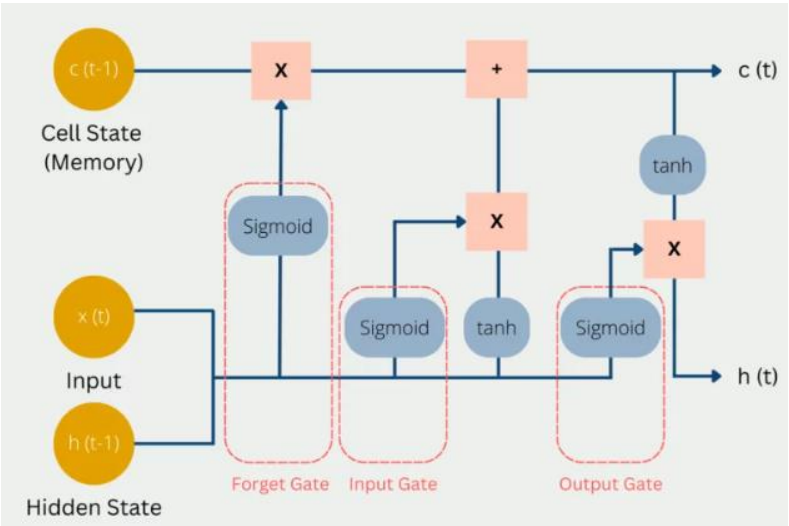
Annexe 6 : Différenciation du logarithme de la série temporelle

White noise	ARIMA(0,0,0)
Random walk	ARIMA(0,1,0) with no constant
Random walk with drift	ARIMA(0,1,0) with a constant
Autoregression	ARIMA(p,0,0)
Moving average	ARIMA(0,0,q)

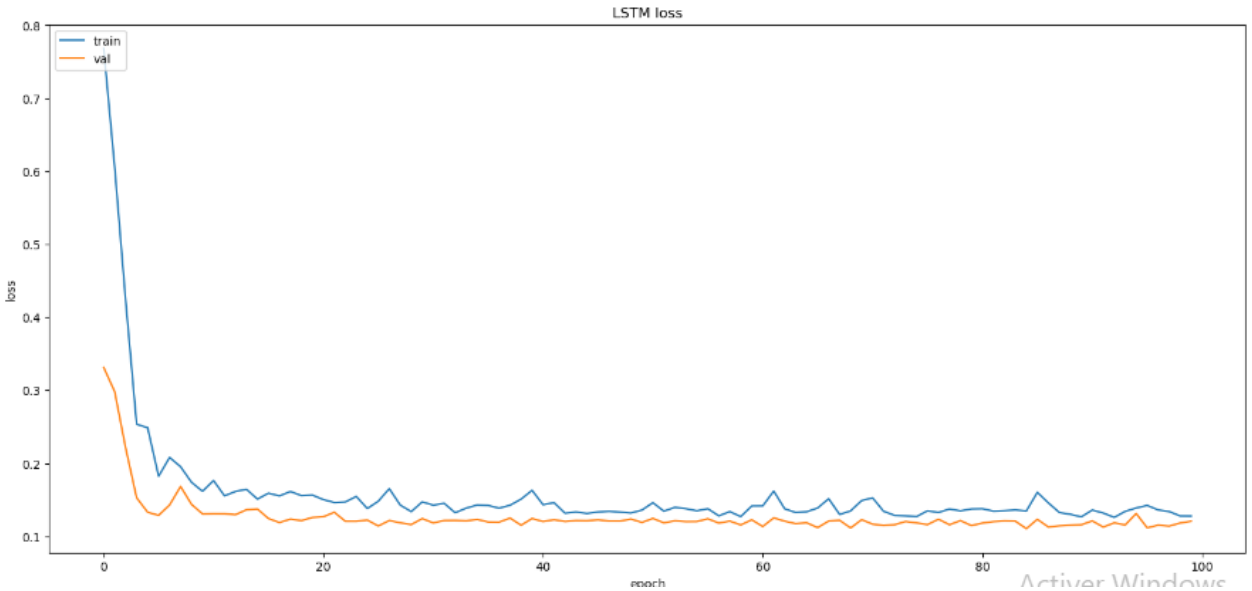
Annexe 7 : Cas particuliers du modèle ARIMA



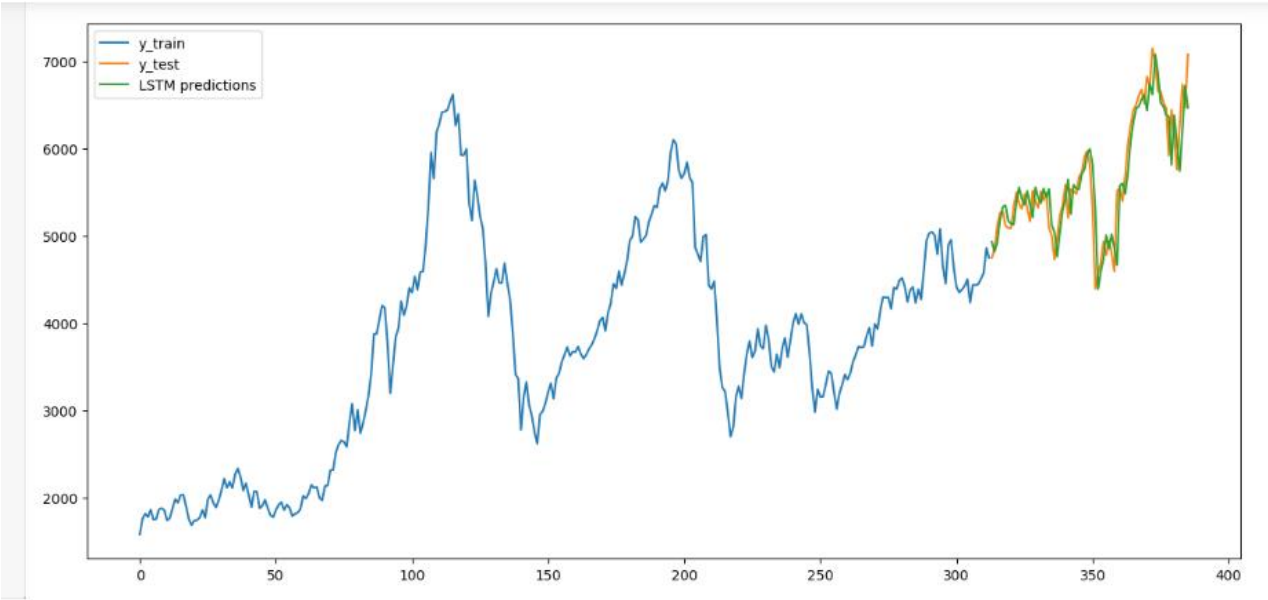
Annexe 8 : l'indice CAC 40 prédit en model ARIMA



Annexe 9 : Modèle LSTM



Annexe 10 : Traçage de la perte LSTM



Annexe 11 : l'indice CAC 40 prédit en model LSTM et comparaison avec ARIMA