



HIDDEN MARKOV MODELS FOR STOCK MARKET PREDICTION

Mars 2024

Yakoub Biram , Alain Kuissu



TABLE DES MATIÈRES

1	Introduction	3
2	Fondements Théoriques des HMM	4
2.1	Motivation	4
2.2	Principes de Base des HMM	5
2.3	Algorithmes Fondamentaux	6
2.3.1	Algorithme Forward pour le problème d'évaluation	7
2.3.2	Algorithme de Viterbi pour le problème de décodage	7
2.3.3	Algorithme de Baum-Welch pour le problème d'estimation de paramètres	9
3	HMM dans la Finance	11
3.1	Application des HMM dans la Finance	11
3.1.1	Modélisation des séries chronologiques	11
3.2	Étude de Cas détaillée	12
4	Comparaison avec d'autres modèles de prédiction	13
4.1	Comparaison avec ARIMA	13
4.1.1	Contexte et Hypothèse	14
4.1.2	Approche et Méthodologie	14
4.2	Résultats et Comparaison	14
4.3	Comparaison avec LSTM	14
4.3.1	Contexte et Hypothèse	14
4.3.2	Approche et Méthodologie	15
4.4	Résultats et Comparaison	16
5	Analyse détaillée des états cachés du HMM pour les actions Apple, Google, et l'indice SPY	17
5.1	Analyse pour Google	17
5.2	Analyse pour Apple	18
5.3	Analyse pour SPY	18
5.4	Conclusion	18
6	Conclusion	20

1

INTRODUCTION

La prévision du marché boursier a toujours été un domaine d'intérêt majeur pour les investisseurs, les traders et les analystes financiers, en raison de son potentiel de génération de profits significatifs. Historiquement, diverses méthodes, telles que l'analyse technique et fondamentale, ont été utilisées pour prédire les mouvements du marché. Récemment, avec l'avancement de la technologie, des méthodes plus sophistiquées comme les réseaux de neurones artificiels et le machine learning sont devenues populaires. Parmi celles-ci, les modèles de Markov cachés (HMM) se révèlent être des outils puissants. Ces modèles, en raison de leur capacité à capturer les états cachés et les transitions probabilistes entre ces états, offrent une compréhension plus profonde des dynamiques de marché. Contrairement aux méthodes traditionnelles, les HMM peuvent efficacement modéliser les séquences temporelles et les comportements non linéaires, ce qui est essentiel dans la compréhension des marchés financiers, souvent caractérisés par une volatilité et des tendances imprévisibles.

L'utilisation des HMM dans la prédiction boursière représente un progrès significatif, permettant une analyse plus nuancée et une prédiction potentiellement plus précise des mouvements de marché. Cela peut conduire à des stratégies d'investissement plus informées, réduisant les risques et augmentant les opportunités de gain. En outre, l'intégration des avancées technologiques telles que le machine learning et l'analyse de données volumineuses, combinées aux HMM, ouvre de nouvelles voies pour explorer des modèles complexes de données de marché, offrant ainsi une perspective nouvelle et enrichissante sur la prédiction boursière.

Nous nous penchons sur la mise en œuvre des HMM pour la prédiction des prix des actions, abordant les problèmes d'évaluation, de décodage et d'estimation des paramètres. Cette étude s'inspire des recherches précédentes qui ont utilisé les HMM dans des contextes similaires, mais avec l'ajout de nos propres innovations et approches méthodologiques.

Notre objectif est de démontrer comment les HMM, adaptés aux spécificités du marché boursier, peuvent conduire à des prédictions plus précises et informatives, contribuant ainsi à des stratégies d'investissement plus efficaces et à une meilleure compréhension des dynamiques du marché.

Article de référence : "*Hidden Markov Models for Stock Market Prediction*"[2]

2

FONDEMENTS THÉORIQUES DES HMM

2.1 MOTIVATION

Une chaîne de Markov est un modèle stochastique qui décrit une séquence d'événements possibles dans laquelle la probabilité de chaque événement dépend uniquement de l'état atteint lors de l'événement précédent. Pour ce travail, nous nous concentrons spécifiquement sur les chaînes de Markov du premier ordre, qui respectent la **propriété de Markov** : Soit $S = \{S_1, S_2, \dots, S_n\}$ l'ensemble de tous les états possibles, et $X = \{X_k | X_k \in S, k = 1, \dots, T\}$ la série temporelle des états. La propriété de Markov stipule que pour tout $k \geq 0$ et les états X_0, \dots, X_k :

$$P(X_{k+1} = S_j | X_k = S_i, X_{k-1}, \dots, X_0) = P(X_{k+1} = S_j | X_k = S_i) \quad (1)$$

En d'autres termes, la probabilité de passer à un certain état S_j au temps $k + 1$ ne dépend que de l'état actuel S_i au temps k et non des états précédents. Cette propriété nous permet de calculer la distribution de probabilité de la chaîne de Markov à n'importe quel moment futur en se basant exclusivement sur son état actuel.

Formellement, une chaîne de Markov du premier ordre est définie par l'ensemble des états S et une matrice de probabilité de transition $A = [a_{ij}]$, où a_{ij} représente la probabilité de passer de l'état S_i à l'état S_j en une étape. C'est une matrice $n \times n$ avec n le nombre d'états de la chaîne. Elle est également dite stochastique car la somme des éléments de ses lignes donne 1 à chaque fois.

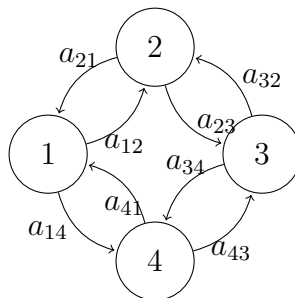


FIGURE 1 – Illustration d'une chaîne de Markov à 4 états

Les éléments de la matrice de transition doivent satisfaire les contraintes suivantes :

- $0 \leq a_{ij} \leq 1$ pour tout i, j .
- $\sum_{j=1}^n a_{ij} = 1$ pour tout i .

Soit $x_k \in \mathbb{R}^n$ le vecteur contenant les probabilités d'être dans chaque état au temps k , le système évolue selon :

$$x_{k+1}^T = x_k^T A \quad (2)$$

Nous désignons la distribution initiale des probabilités comme $\pi = (\pi_1, \pi_2, \dots, \pi_n)$. La connaissant, nous pouvons construire le vecteur x_k pour tout temps k grâce à la formule précédente :

$$x_k^T = \pi A^k \quad (3)$$

Une chaîne de Markov peut donc être représentée par le couple (A, π) avec $S = \{S_1, S_2, \dots, S_n\}$ l'ensemble des états possibles.

Dans le cas de ces chaînes de Markov, nous assumons que les états de la chaîne représentent une réalité observable, ce qui n'est pas toujours le cas.

C'est dans ce cadre que nous introduisons les chaînes de Markov cachées qui pourront nous permettre de modéliser des cas où les états de la chaîne sont cachés (non observables mais représentent bien une certaine réalité même si elle est inconnue) mais ces états cachés peuvent émettre des observations (sorte d'effet observable à chaque fois que nous sommes dans un état caché) avec une certaine probabilité.

Nous nous intéresserons au cadre théorique de ce modèle dans la suite.

2.2 PRINCIPES DE BASE DES HMM

Les chaînes de Markov se sont avérées être des outils précieux pour modéliser des données séquentielles dans divers domaines. Cependant, de nombreux scénarios du monde réel impliquent des états sous-jacents qui influencent les observations mais ne sont pas directement observables. Cette limitation a conduit au développement des Modèles de Markov Cachés (HMM), qui étendent le modèle de chaîne de Markov de base en introduisant des états cachés ou non observables qui affectent les données observées.

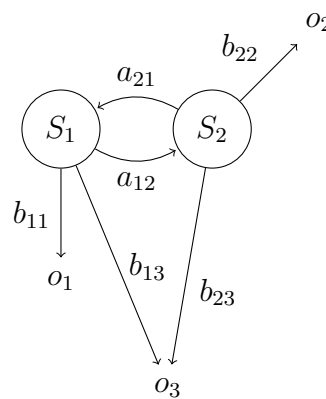


FIGURE 2 – Illustration d'une chaîne de Markov cachée à 2 états cachés et 3 symboles émis

Le processus d'état caché d'un HMM est une chaîne de Markov, où chaque état génère une observation ayant une certaine distribution de probabilité qui ne dépend que de l'état lui-même. Soit $\{O_k | O_k \in O, k = 1, \dots, T\}$ une séquence observée, où O est l'ensemble des

observations possibles supposées de cardinal m . On suppose que les états cachés évoluent selon la propriété de Markov. La probabilité que le symbole $o \in O$ soit émis par l'état S_i est décrite par la fonction de probabilité d'émission $b_i : o \mapsto b_i(o) \in [0, 1], \forall i$.

Dans la section actuelle et les sections suivantes, nous faisons référence à la matrice de probabilité d'émission $B = \{b_{ij} = b_i(o_j) = P(o_j \text{ émis au temps } t | X_t = S_i)\}$.

Une chaîne de Markov cachée peut donc être définie par le quintuplet de paramètres (n, m, A, B, π) avec : n le nombre total d'états cachés possibles, m le nombre total d'observations possibles, A la matrice de transition entre les états cachés, B la matrice d'émission et π la distribution initiale des états cachés, sachant que O est l'ensemble des observations possibles et S est l'ensemble des états cachés possibles.

À partir de là, nous pouvons construire un algorithme pour générer des observations $\{O_k | O_k \in O, k = 1, \dots, T\}$.

Algorithm 1 Génération d'une séquence d'observations à partir d'un HMM

- 1: **Données** : Distribution initiale π , Matrice de transition A , la matrice d'émission B
 - 2: **Résultat** : Séquence d'observations générée $\{O_k | O_k \in O, k = 1, \dots, T\}$
 - 3: Initialiser l'état caché $X_1 = S_i$ selon la distribution initiale π et plus précisément la probabilité π_i d'être en S_i initialement.
 - 4: $t \leftarrow 1$
 - 5: **while** la longueur de la séquence d'observations désirée n'est pas atteinte (T dans ce cas)
do
 - 6: Générer une observation $O_t = o_k$ à partir de l'état caché S_i selon la matrice d'émission B et plus précisément la probabilité b_{ik} d'émettre o_k sachant qu'on est en S_i
 - 7: Mettre à jour l'état caché $X_{t+1} = S_j$ en fonction de la transition de l'état caché actuel $X_t = S_i$ selon la matrice de transition A et plus précisément la probabilité a_{ij} de transiter de S_i à S_j
 - 8: $t \leftarrow t + 1$
 - 9: **end while**
 - 10: Retourner la séquence d'observations générée
-

2.3 ALGORITHMES FONDAMENTAUX

Ayant les briques de base de ce modèle, nous aimerions répondre à 3 questions permettant ainsi d'avoir 3 applications utiles des HMM :

- Connaissant les paramètres probabilistes du modèle que nous noterons $\lambda = (A, B, \pi)$, comment trouver la probabilité que le modèle génère une séquence $G = \{O_k | O_k \in O, k = 1, \dots, T\}$? Autrement dit, quelle est $P(G|\lambda)$? C'est le problème d'évaluation pour les HMM.
- Quelle séquence d'états cachés explique le mieux l'observation $G = \{O_k | O_k \in O, k = 1, \dots, T\}$? Autrement dit, sachant les paramètres λ du HMM, quelle est la séquence $X =$

$\{X_k | X_k \in S, k = 1, \dots, T\}$ d'états cachés qui maximise $P(X|G, \lambda)$? C'est le problème de décodage pour les HMM.

- Ayant une séquence d'observation $G = \{O_k | O_k \in O, k = 1, \dots, T\}$, comment retrouver les paramètres λ du HMM ayant permis de les générer ? Autrement dit, quel est le paramètre λ qui maximise $P(G|\lambda)$? C'est le problème d'estimation des paramètres d'un HMM.

La réponse à chacune de ces questions est donnée par des algorithmes que nous allons analyser.

2.3.1 • ALGORITHME FORWARD POUR LE PROBLÈME D'ÉVALUATION

Le problème d'évaluation dans les HMM implique le calcul de la probabilité d'une séquence d'observations connue, étant donné le modèle. Plus précisément, étant donné un HMM avec les paramètres $\lambda = (A, B, \pi)$, et une séquence observée $G = \{O_k | O_k \in O, k = 1, \dots, T\}$, nous voulons calculer $P(O|\lambda)$. La procédure utilisée pour résoudre ce problème consiste à calculer une variable dite **Forward**, notée $\alpha_k(i) = P(O_1, O_2, \dots, O_k, X_k = S_i | \lambda)$, qui représente la probabilité d'être dans l'état S_i au temps k et d'observer la séquence jusqu'au temps k . Elle est calculée de manière récursive pour chaque pas de temps k et chaque état S_i .

À l'étape d'initialisation, la variable forward pour l'étape temporelle initiale $k = 1$ est calculée comme suit :

$$\alpha_1(i) = P(O_1, X_1 = S_i | \lambda) = P(X_1 = S_i) \cdot P(O_1 | X_1 = S_i, \lambda) = \pi_i \cdot b_i(O_1), \quad \forall i,$$

où π_i est la probabilité initiale d'être dans l'état S_i et $b_i(O_1)$ est la probabilité d'émission pour la première observation O_1 .

Pour chaque pas de temps suivant $k > 1$, la variable forward $\alpha_k(i)$ est calculée comme la somme des probabilités de tous les chemins possibles qui auraient pu générer les observations jusqu'au temps k et atteindre l'état S_i . Cela est exprimé par la formule suivante :

$$\alpha_k(i) = \left(\sum_{j=1}^n \alpha_{k-1}(j) \cdot a_{ji} \right) \cdot b_i(O_k), \quad \forall i. \quad (4)$$

En calculant de manière récursive les variables forward pour chaque pas de temps, l'algorithme évalue la probabilité d'observer l'ensemble de la séquence, étant donné les paramètres du HMM.

Enfin, pour obtenir la probabilité globale de la séquence observée, la procédure forward calcule la somme des variables forward au dernier pas de temps T :

$$P(G|\lambda) = \sum_{i=1}^n \alpha_T(i). \quad (5)$$

Cet algorithme a une complexité en $O(n^2T)$.

2.3.2 • ALGORITHME DE VITERBI POUR LE PROBLÈME DE DÉCODAGE

Le problème de décodage consiste à déterminer la séquence la plus probable d'états cachés étant donné une séquence d'observations et le modèle. Étant donné un HMM avec les paramètres

Algorithm 2 Algorithme Forward pour les HMM

- 1: **Données** : Distribution initiale π , Matrice de transition A , la matrice d'émission B , Séquence observée $G = \{O_k | O_k \in O, k = 1, \dots, T\}$
 - 2: **Résultat** : Probabilité de la séquence observée $P(G|\lambda)$ avec $\lambda = (A, B, \pi)$
 - 3: Initialiser la variable Forward pour l'étape temporelle initiale $k = 1$:
 - 4: $\alpha_1(i) = \pi_i \cdot b_i(O_1), \forall i$
 - 5: **for** $k = 2$ to T **do**
 - 6: **for** $i = 1$ to n **do**
 - 7: $\alpha_k(i) = \left(\sum_{j=1}^n \alpha_{k-1}(j) \cdot a_{ji} \right) \cdot b_i(O_k)$
 - 8: **end for**
 - 9: **end for**
 - 10: Calculer la probabilité globale de la séquence observée :
 - 11: $P(G|\lambda) = \sum_{i=1}^n \alpha_T(i)$
 - 12: Retourner $P(G|\lambda)$
-

Algorithm 3 Algorithme de Viterbi pour les HMM

- 1: **Données** : Distribution initiale π , Matrice de transition A , la matrice d'émission B , Séquence observée $G = \{O_k | O_k \in O, k = 1, \dots, T\}$
 - 2: **Résultat** : Séquence d'états cachés la plus probable $X^* = \{X_k^* | X_k^* \in S, k = 1, \dots, T\}$
 - 3: Initialiser la variable Viterbi pour l'étape temporelle initiale $k = 1$:
 - 4: $v_1(i) = \pi_i \cdot b_i(O_1), \forall i$
 - 5: $ptr_1(i) = 0, \forall i$
 - 6: **for** $k = 2$ to T **do**
 - 7: **for** $j = 1$ to n **do**
 - 8: $v_k(j) = \max_{i=1}^n (v_{k-1}(i) \cdot a_{ij}) \cdot b_j(O_k)$
 - 9: $ptr_k(j) = \arg \max_{i=1}^n (v_{k-1}(i) \cdot a_{ij})$
 - 10: **end for**
 - 11: **end for**
 - 12: $P^* = \max_{i=1}^n v_T(i)$
 - 13: $X_T^* = \arg \max_{i=1}^n v_T(i)$
 - 14: **for** $k = T - 1$ to 1 **do**
 - 15: $X_k^* = ptr_{k+1}(X_{k+1}^*)$
 - 16: **end for**
 - 17: Retourner la séquence d'états cachés la plus probable X^*
-

$\lambda = (A, B, \pi)$, et une séquence observée G , nous voulons trouver la séquence d'états cachés X qui maximise $P(X|G, \lambda)$.

La solution au problème de décodage est couramment abordée en utilisant l'algorithme de Viterbi. L'algorithme fonctionne en calculant de manière itérative le chemin le plus probable vers chaque état à chaque pas de temps, en considérant à la fois l'observation actuelle et les probabilités des états précédents. À chaque étape temporelle, il calcule la probabilité d'être dans chaque état et suit le chemin le plus probable menant à cet état.

2.3.3 • ALGORITHME DE BAUM-WELCH POUR LE PROBLÈME D'ESTIMATION DE PARAMÈTRES

Le problème d'estimation des paramètres dans les HMM implique d'ajuster les paramètres du modèle pour maximiser la probabilité d'une séquence observée. Étant donné une séquence d'observations G , nous voulons estimer les valeurs optimales pour les paramètres π , A et B qui maximisent $P(G|\pi, A, B)$.

La solution à ce problème est généralement abordée en utilisant l'algorithme de Baum-Welch, également connu sous le nom d'algorithme forward-backward.

L'algorithme de Baum-Welch est une implémentation spécifique de l'algorithme Expectation-Maximization (EM), adapté pour les HMM. Il effectue de manière itérative trois étapes principales : forward, backward et mise à jour. Dans l'étape forward, l'algorithme calcule la probabilité d'être dans un état spécifique à chaque pas de temps, étant donné la séquence observée jusqu'à ce point. C'est le problème d'évaluation expliqué précédemment.

Dans l'étape backward, l'algorithme calcule la probabilité d'observer la partie restante de la séquence à partir d'un état donné à chaque pas de temps. Il complète la procédure forward en calculant la variable backward, notée $\beta_k(i)$, qui représente la probabilité d'être dans l'état S_i au temps k et d'observer la séquence restante du temps $k + 1$ jusqu'à la fin. La variable backward est calculée de manière récursive pour chaque pas de temps k et chaque état S_i . À l'étape d'initialisation, la variable backward pour le dernier pas de temps est définie comme : $\beta_T(i) = 1, \forall i$.

Pour chaque pas de temps $k < T$, la variable backward $\beta_k(i)$ est calculée comme la somme des probabilités de tous les chemins possibles qui peuvent être générés à partir de l'état S_i au temps k jusqu'à la fin de la séquence. Cela est exprimé par la formule suivante :

$$\beta_k(i) = \sum_{j=1}^n \beta_{k+1}(j) \cdot a_{ij} \cdot b_j(O_{k+1}) \quad (6)$$

Après avoir calculé les variables forward et backward, elles sont utilisées pour affiner les paramètres du HMM en fonction de la séquence observée. En fait, soit $\xi_k(i, j)$ la probabilité d'être dans l'état S_i au temps k et dans l'état S_j au temps $k + 1$, étant donné le modèle et la séquence d'observations. Elle peut être calculée comme suit :

$$\xi_k(i, j) = \frac{\alpha_k(i) \cdot a_{ij} \cdot b_j(O_{k+1}) \cdot \beta_{k+1}(j)}{\sum_{r=1}^n \sum_{h=1}^n \alpha_k(r) \cdot a_{rh} \cdot b_h(O_{k+1}) \cdot \beta_{k+1}(h)} \quad (7)$$

De plus, $\gamma_k(i)$ est la probabilité d'être dans l'état S_i au temps k , étant donné la séquence d'observations et le modèle :

$$\gamma_k(i) = \frac{\alpha_k(i) \cdot \beta_k(i)}{\sum_{j=1}^n \alpha_k(j) \cdot \beta_k(j)} \quad (8)$$

Nous pouvons relier ces quantités :

$$\gamma_k(i) = \sum_{j=1}^n \xi_k(i, j) \quad (9)$$

De plus, $\sum_{k=1}^{T-1} \gamma_k(i)$ représente le nombre attendu de transitions à partir de l'état S_i , et $\sum_{k=1}^{T-1} \xi_k(i, j)$ représente le nombre attendu de transitions de l'état S_i à l'état S_j .

L'étape de mise à jour implique d'estimer les nouvelles valeurs pour les probabilités d'état initiales, les probabilités de transition et les probabilités d'émission. Pour les probabilités d'état initiales, les valeurs mises à jour sont calculées comme les probabilités forward-backward normalisées au premier pas de temps :

$$\pi_i = \gamma_1(i) \quad (10)$$

Les probabilités de transition sont mises à jour en fonction du nombre attendu de transitions entre les états, normalisé par le nombre attendu de transitions à partir de chaque état :

$$a_{ij} = \frac{\sum_{k=1}^{T-1} \xi_k(i, j)}{\sum_{k=1}^{T-1} \gamma_k(i)} \quad (11)$$

L'estimation pour la probabilité d'émission $b_j(o)$ - la probabilité d'observer le symbole o à partir de l'état S_j - est calculée en évaluant le nombre attendu de fois où o est émis depuis l'état j . Cette valeur est ensuite normalisée par le nombre attendu de transitions à partir de l'état S_j :

$$b_j(o) = \frac{\sum_{k=1}^T (\gamma_k(j) \cdot \delta(O_k, o))}{\sum_{k=1}^T \gamma_k(j)} \quad (12)$$

où $\delta(O_k, o)$ est la fonction delta de Kronecker qui évalue à 1 lorsque O_k est égal au symbole observé o , et à 0 sinon.

Algorithm 4 Algorithme de Baum-Welch pour les HMM

Require: Une séquence d'observations $G = O_1, O_2, \dots, O_T$

Ensure: Les paramètres estimés π , A et B du HMM

- 1: Initialiser π , A et B avec des valeurs arbitraires
 - 2: **repeat**
 - 3: Calculer les variables forward $\alpha_k(i)$ pour tout k et i selon l'équation (4)
 - 4: Calculer les variables backward $\beta_k(i)$ pour tout k et i selon l'équation (6)
 - 5: Calculer les probabilités $\xi_k(i, j)$ et $\gamma_k(i)$ pour tout k , i et j selon les équations (7) et (8)
 - 6: Mettre à jour les paramètres π , A et B selon les équations (10), (11) et (12)
 - 7: **until** La convergence des paramètres est atteinte
-

Ce processus affine itérativement les paramètres du modèle jusqu'à convergence, où ils se stabilisent, ce qui signifie que la vraisemblance pour la séquence est maximisée. Nous rajouterons aussi un paramètre `max_iter` qui va nous permettre de stopper la boucle et contrôler si on tombe sur un maximum local satisfaisant. La preuve faisant intervenir le calcul de la vraisemblance et prouvant son optimisation peut être vue dans le document "Derivation of Baum-Welch Algorithm for Hidden Markov Models". Le choix des conditions initiales est crucial car il peut avoir un impact significatif sur l'estimation des paramètres du HMM. En sélectionnant des conditions initiales appropriées, l'algorithme de Baum-Welch a plus de chances de converger vers certains paramètres fournissant une modélisation précise de la dynamique sous-jacente.

3

HMM DANS LA FINANCE

3.1 APPLICATION DES HMM DANS LA FINANCE

3.1.1 • MODÉLISATION DES SÉRIES CHRONOLOGIQUES

Les modèles de Markov cachés (HMM) et leur variante, le modèle de mélange gaussien Hidden Markov Model (GMMHMM), sont largement utilisés dans le domaine financier pour modéliser les séries chronologiques de prix d'actifs tels que les actions, les obligations et les devises. Ces modèles permettent de capturer les régimes sous-jacents du marché et d'effectuer des prévisions sur les tendances futures.

Dans notre étude, nous avons appliqué un GMMHMM à la modélisation des séries chronologiques des prix des actions de Google. Les états cachés du modèle étant des variables inconnues, nous considérons le nombre d'états cachés comme un hyperparamètre du modèle que nous ferons varier afin de ne garder que le modèle ayant donné les meilleures performances. En utilisant les données historiques des prix des actions, nous avons entraîné le modèle à estimer les paramètres de transition entre les états cachés ainsi que les paramètres des distributions de probabilité gaussienne pour chaque état.

Une fois le modèle entraîné et validé, nous avons utilisé les informations apprises pour prédire les valeurs futures du prix à la fermeture de marché de Google. Cette prédiction des valeurs futures a été réalisée en utilisant les estimations des états cachés futurs et en exploitant les distributions de probabilité des observations pour chaque état.

Dans la figure ci-dessous, nous présentons un exemple de prédiction des prix futurs des actions de Google à l'aide de notre modèle GMMHMM entraîné. Cette visualisation illustre comment notre approche permet de capturer les tendances et les fluctuations des prix, offrant ainsi des perspectives précieuses pour la prise de décision en matière d'investissement.

Cette approche illustre l'efficacité des HMM et de leurs variantes comme le GMMHMM dans l'analyse financière, offrant une modélisation précise des séries temporelles. Le document "*Stock Market Trend Analysis Using Hidden Markov Model and Long Short Term Memory*"

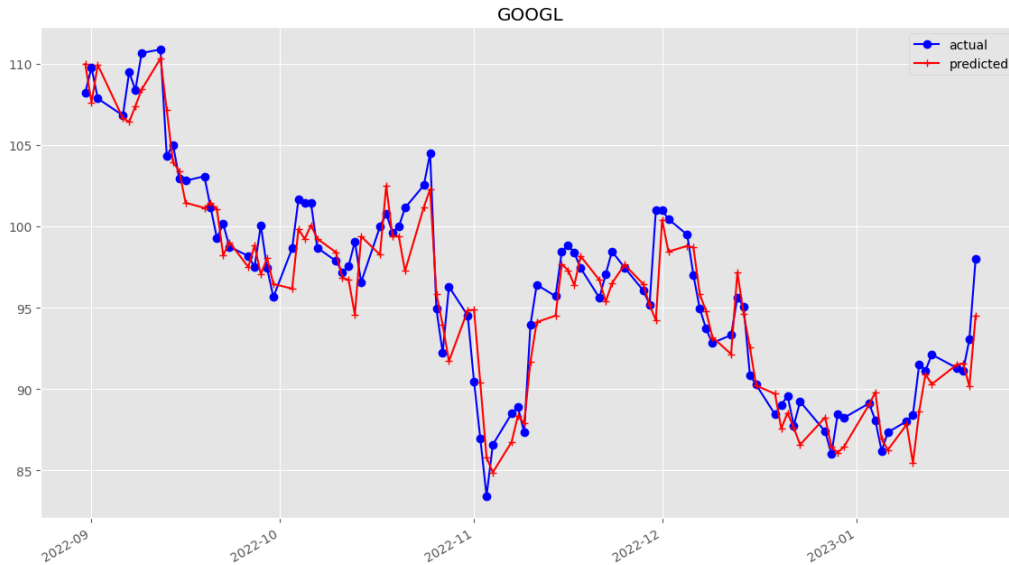


FIGURE 3 – Prédiction des prix des actions de Google

[1] approfondit cette idée en combinant les HMM avec des LSTM, enrichissant l'analyse des tendances du marché. Il explore également des modèles avancés comme le XGB-HMM, qui appliquent des techniques de pointe pour une prédiction affinée des mouvements du marché financier.

3.2 ÉTUDE DE CAS DÉTAILLÉE

On dispose de variables très limitées pour chaque jour, à savoir le prix d'ouverture de l'action pour ce jour, le prix de clôture, le prix le plus élevé de l'action et le prix le plus bas de l'action. On va les utiliser pour calculer les prix des actions avec notre modèle. On peut calculer le prix de clôture de l'action pour un jour, étant donné le prix d'ouverture de l'action pour ce jour, et les données des d jours précédents. Notre prédicteur aurait une latence de d jours.

Au lieu d'utiliser directement les prix d'ouverture, de clôture, les plus bas et les plus hauts d'une action, on extrait les variations fractionnaires de chacun d'entre eux qui seront utilisées pour entraîner notre HMM. On définit ces paramètres comme suit :

$$O_k = \left(\frac{\text{Clôture} - \text{Ouverture}}{\text{Ouverture}}, \frac{\text{Haut} - \text{Ouverture}}{\text{Ouverture}}, \frac{\text{Ouverture} - \text{Bas}}{\text{Ouverture}} \right)$$

$$O_k = (\text{frac}_{\text{change}}, \text{frac}_{\text{high}}, \text{frac}_{\text{low}})$$

Pour le prédicteur de prix d'actions HMM, on peut représenter une seule observation comme un vecteur pour ces paramètres, à savoir $O_t = \langle \text{frac}_{\text{change}}, \text{frac}_{\text{high}}, \text{frac}_{\text{low}} \rangle$. On peut diviser des tableaux ou des matrices en sous-ensembles d'entraînement et de test aléatoires. Comme on entraîne notre HMM avec des données séquentielles, on ne veut pas diviser les données de manière aléatoire.

Une fois que notre modèle est entraîné, on doit prédire le prix de clôture de l'action. Comme mentionné précédemment, on veut prédire le prix de clôture de l'action pour un jour donné sachant qu'on connaît le prix d'ouverture. Cela signifie que si on est capable de prédire frac_{change} pour un jour donné, on peut calculer le prix de clôture comme suit :

$$Cloture = Ouverture * (\text{frac}_{change} + 1)$$

Ainsi, notre problème se résume à calculer le vecteur d'observation $O_{t+1} = \langle \text{frac}_{change}, \text{frac}_{high}, \text{frac}_{low} \rangle$ pour un jour donné, étant donné les données d'observation pour t jours, o_1, \dots, o_t , et les paramètres du HMM, c'est-à-dire trouver la valeur de O_{t+1} qui maximise la probabilité a posteriori de $P(O_{t+1}|O_1, \dots, O_t, \lambda)$:

$$\begin{aligned} O_{t+1} &= \arg \max_{O_{t+1}} P(O_{t+1}|O_1, \dots, O_t, \lambda) \\ &= \arg \max_{O_{t+1}} \frac{P(O_1, \dots, O_t, O_{t+1}|\lambda)P(\lambda)}{P(O_1, \dots, O_t|\lambda)} \\ &= \arg \max_{O_{t+1}} P(O_1, \dots, O_t, O_{t+1}|\lambda) \end{aligned}$$

Une fois qu'on a éliminé tous les paramètres indépendants de O_{t+1} de l'équation de maximisation, on se retrouve avec le problème de trouver la valeur de O_{t+1} qui optimise la probabilité de $P(O_1, \dots, O_{t+1}|\lambda)$. Si on suppose que frac_{change} est une variable continue, l'optimisation du problème serait difficile sur le plan computationnel.

Ainsi, on va diviser ces changements fractionnaires en quelques valeurs discrètes comprises entre deux variables finies et trouver un ensemble de changements fractionnaires, $O_{t+1} = \langle \text{frac}_{change}, \text{frac}_{high}, \text{frac}_{low} \rangle$, qui maximiseraient la probabilité, $P(O_1, \dots, O_{t+1}|\lambda)$.

On obtient ainsi la figure 3 pour la prédiction du cours de l'action Google. Nous avons fixé le nombre d'états cachés à 4 car c'est le nombre d'états donnant les meilleures métriques. Cependant, par souci d'interprétabilité des états, il ne faut pas hésiter à le fixer selon le domaine (la finance de marché dans notre cas). Par exemple, une approche consisterait à définir les états "Haussier" ou "Baissier" comme états cachés car ceux-ci représentent une certaine réalité du domaine. N'oublions pas que plus il y a d'états cachés, plus il y a de paramètres dans le HMM et cela peut devenir un souci du point de vue computationnel.

4

COMPARAISON AVEC D'AUTRES MODÈLES DE PRÉDICTION

4.1 COMPARAISON AVEC ARIMA

4.1.1 • CONTEXTE ET HYPOTHÈSE

Considérons une série chronologique représentant les prix de clôture quotidiens de l'action Google sur une période donnée. Les données financières, caractérisées par leur volatilité et leurs régimes de marché variables, présentent un défi significatif pour la modélisation prédictive. L'hypothèse est que les HMM, grâce à leur capacité à modéliser les transitions entre différents états cachés (par exemple, des régimes de marché), peuvent capturer plus efficacement la dynamique sous-jacente des prix de clôture par rapport aux modèles ARIMA, qui supposent une dépendance linéaire constante dans le temps.

4.1.2 • APPROCHE ET MÉTHODOLOGIE

Pour le HMM, nous définissons plusieurs états cachés correspondant à différents régimes de marché. Le modèle est entraîné sur les données historiques des prix de clôture pour estimer les probabilités de transition entre ces états, ainsi que les paramètres de distribution de probabilité pour les observations (prix de clôture) dans chaque état.

En parallèle, un modèle ARIMA est ajusté sur les mêmes données, en identifiant les paramètres optimaux (p , d , q) qui minimisent une mesure d'erreur, comme l'AIC (Akaike Information Criterion) ou le BIC (Bayesian Information Criterion), pour capturer la tendance et la saisonnalité dans les données de prix de clôture.

4.2 RÉSULTATS ET COMPARAISON

Les résultats montrent que le HMM parvient à identifier de manière significative les transitions entre différents régimes de marché, permettant des prédictions de prix de clôture qui s'alignent étroitement sur les mouvements réels observés, même dans des périodes de haute volatilité ou lors de changements de tendance abrupts. En revanche, l'ARIMA, en se basant sur une relation linéaire et en supposant une homogénéité temporelle, échoue à anticiper ces changements de régime, résultant en des prédictions qui peuvent s'écarter significativement de la réalité lors de fluctuations importantes du marché.

L'exemple met en évidence la supériorité des HMM dans la modélisation des séries chronologiques financières où les dynamiques de marché sont non linéaires et régies par des régimes changeants.

4.3 COMPARAISON AVEC LSTM

4.3.1 • CONTEXTE ET HYPOTHÈSE

Les réseaux de neurones LSTM (Long Short-Term Memory) sont renommés pour leur capacité à capturer des dépendances à long terme dans les données séquentielles, ce qui les rend appropriés pour la modélisation des séries chronologiques comme les prix des actions. Toutefois,

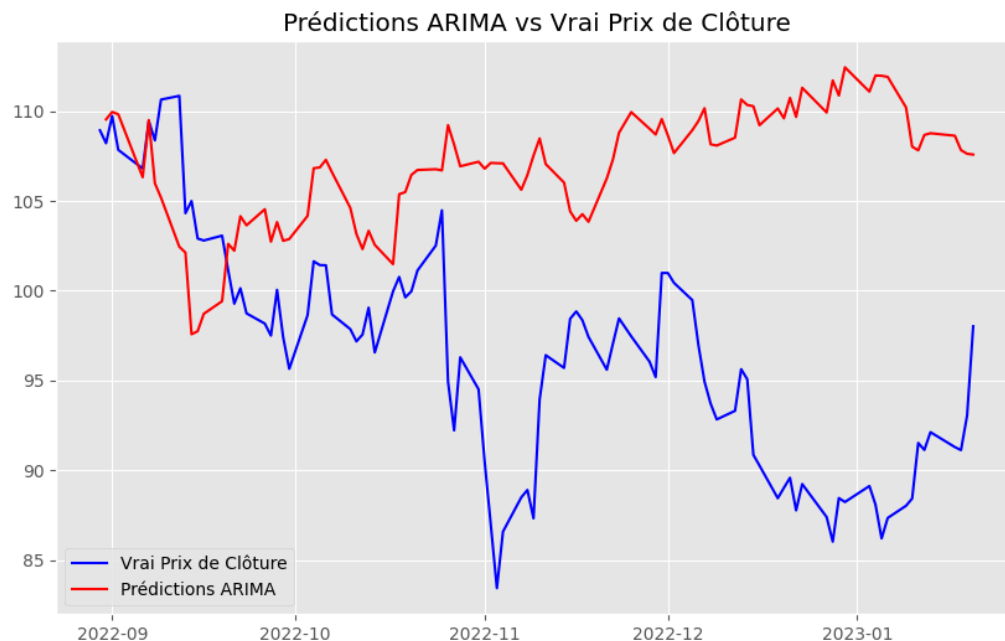


FIGURE 4 – Prédiction des prix des actions de Google avec ARIMA

la complexité de leur architecture et le besoin de grandes quantités de données pour l'entraînement posent des défis, particulièrement dans des contextes où les données sont limitées. L'hypothèse est que, bien que les LSTM puissent théoriquement surpasser les HMM en termes de performance grâce à leur capacité à modéliser des dépendances complexes, les contraintes pratiques rendent les HMM plus efficaces dans certains cas, notamment avec des ensembles de données plus restreints.

4.3.2 • APPROCHE ET MÉTHODOLOGIE

Pour l'approche LSTM, un réseau est configuré avec des couches LSTM suivies par des couches denses pour prédire les prix de clôture des actions. Le nombre de couches, de neurones par couche, et d'autres hyperparamètres doivent être soigneusement choisis, souvent via un processus d'essai et d'erreur ou des techniques de recherche d'hyperparamètres, pour optimiser la performance du modèle.

Le modèle LSTM est entraîné sur les données historiques des prix de clôture, en utilisant une partie des données pour l'entraînement et une autre pour la validation. Les performances sont évaluées en fonction de la précision des prédictions de prix et comparées à celles obtenues avec le modèle HMM.

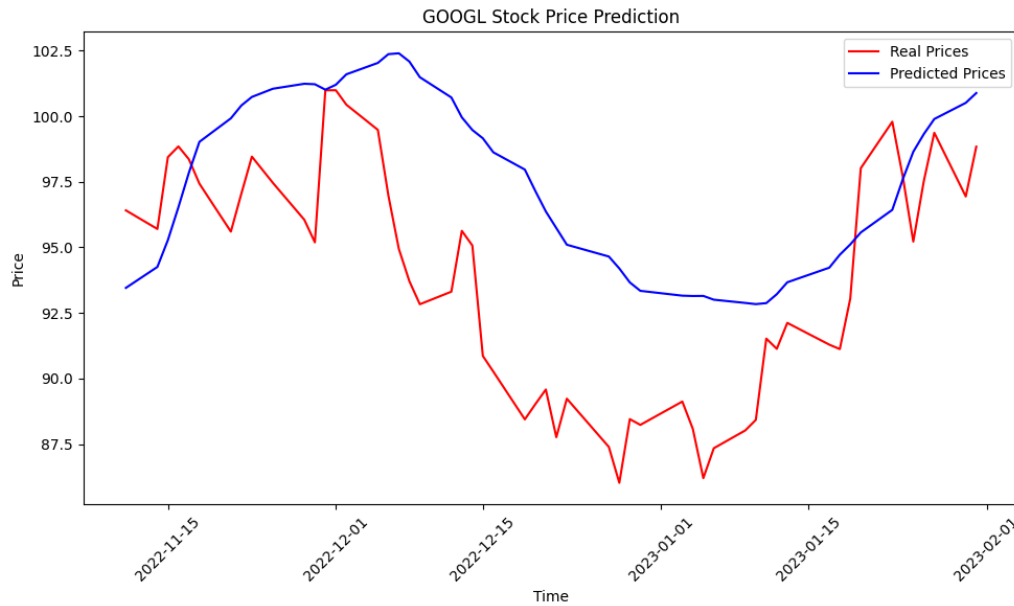


FIGURE 5 – Prédiction des prix des actions de Google avec LSTM

4.4 RÉSULTATS ET COMPARAISON

Les LSTM peuvent potentiellement offrir une modélisation plus précise en capturant les dépendances à long terme et les motifs complexes dans les séries de prix. Cependant, cette étude constate que, compte tenu de la taille limitée de l'ensemble de données et des défis associés au réglage de l'architecture LSTM, les résultats ne sont pas substantiellement meilleurs que ceux obtenus avec le modèle HMM, qui est nettement moins complexe à configurer et à entraîner.

De plus, les LSTM peuvent être sujets au surajustement, particulièrement avec des ensembles de données plus petits, ce qui peut réduire leur capacité à généraliser et à effectuer des prédictions précises sur des données non vues. En revanche, les HMM, avec leur structure simplifiée et leur efficacité dans la modélisation des transitions entre différents régimes de marché, ont démontré une robustesse remarquable, s'alignant de près avec les tendances réelles des prix de clôture, même avec un volume de données limité.

En complément de notre étude, nous avons également exploré des implémentations de réseaux LSTM-CNN (Convolutional Neural Networks) disponibles sur GitHub, qui sont spécifiquement conçues pour la prédiction des prix des actions. Ces implémentations fournissent un cadre pour évaluer comment une architecture bien conçue et optimisée peut potentiellement améliorer les performances de prédiction. Des exemples de ces codes peuvent être trouvés sur la page GitHub suivante : https://github.com/Soroush98/LSTM-CNN_Stock. Malgré les résultats prometteurs obtenus avec les modèles HMM dans notre étude, il est crucial de reconnaître que des architectures LSTM sophistiquées, lorsqu'elles sont correctement ajustées et entraînées, pourraient offrir des performances exceptionnelles. Cela souligne que la sélection du modèle optimal dépend fortement du contexte spécifique, y compris la disponibilité des données, l'objectif

de la modélisation, et les contraintes de temps et de ressources. Par conséquent, il est difficile de conclure de manière définitive quel modèle est supérieur sans prendre en compte ces facteurs nuancés.

5

ANALYSE DÉTAILLÉE DES ÉTATS CACHÉS DU HMM POUR LES ACTIONS APPLE, GOOGLE, ET L'INDICE SPY

Cette section fournit une analyse approfondie des résultats obtenus en appliquant un modèle Hidden Markov Model (HMM) à deux états sur les séries temporelles des prix de clôture des actions Apple, Google, et de l'indice SPY (S&P 500 ETF). Le but est de décrypter les caractéristiques inhérentes à chaque état caché et d'interpréter ces états en termes de dynamiques du marché.

5.1 ANALYSE POUR GOOGLE

Le modèle HMM appliqué aux données de Google identifie deux états distincts avec les caractéristiques suivantes sur 98 jours :

État 0 (48 observations) :

- Moyenne des changements fractionnels : -0.0047
- Écart type : 0.017
- Valeur minimale : -0.043
- Valeur maximale : 0.0283

Cet état semble représenter un régime de marché moins actif pour Google, avec des changements de prix généralement négatifs mais relativement faibles en amplitude.

État 1 (50 observations) :

- Moyenne des changements fractionnels : 0.041
- Écart type : 0.02
- Valeur minimale : -0.052
- Valeur maximale : 0.065

À l'opposé, l'État 1 suggère un régime de marché plus volatile avec des changements de prix plus positifs en moyenne, indiquant des périodes potentiellement plus propices à des gains.

5.2 ANALYSE POUR APPLE

Pour Apple, le modèle HMM révèle une répartition très déséquilibrée entre les deux états sur la même période :

État 0 (1 observation) :

- Moyenne des changements fractionnels : -0.019
- Écart type : 0 (une seule observation)
- Minimum et maximum des données : -0.019

Ce régime pourrait correspondre à un jour atypique où Apple a connu un mouvement de prix distinctif par rapport au reste de la période étudiée.

État 1 (97 observations) :

- Moyenne des changements fractionnels : -0.0003
- Écart type : 0.02
- Valeur minimale : -0.041
- Valeur maximale : 0.0592

L'État 1 domine la période observée, représentant un régime de volatilité modérée avec des changements fractionnels moyens proches de zéro.

5.3 ANALYSE POUR SPY

L'analyse de l'indice SPY montre également une prédominance d'un état sur l'autre :

État 0 (95 observations) :

- Moyenne des changements fractionnels : -0.00064
- Écart type : 0.012
- Valeur minimale : -0.048
- Valeur maximale : 0.023

Cet état semble indiquer un régime de marché stable pour l'indice SPY, avec une faible volatilité et des changements de prix minimes.

État 1 (3 observations) :

- Moyenne des changements fractionnels : -0.022
- Écart type : 0.007
- Valeur minimale : -0.03
- Valeur maximale : -0.012

En contraste, l'État 1 représente des jours avec des baisses plus prononcées, suggérant des moments de stress ou de corrections sur le marché.

5.4 CONCLUSION

L'application du HMM à deux états sur les données de prix de clôture pour Google, Apple, et SPY fournit des insights précieux sur les différents régimes de marché. Pour Google, le modèle

distingue clairement entre des périodes de volatilité faible et élevée. Pour Apple, un état domine largement, indiquant un régime de volatilité modérée, tandis que l'indice SPY montre un état de stabilité prédominant avec quelques jours de baisses notables. Ces analyses mettent en lumière l'utilité du HMM dans la détection de patterns et de transitions significatives dans les dynamiques des prix sur les marchés financiers, offrant ainsi des perspectives qui pourraient être cruciales pour la prise de décision en matière d'investissement et de gestion des risques.

En identifiant ces deux états distincts pour chaque série temporelle, le HMM fournit une méthode concise mais puissante pour modéliser et anticiper les comportements de marché. Pour Google et SPY, la distinction entre les régimes de volatilité faible et élevée pourrait être exploitée pour ajuster les stratégies de trading, par exemple en augmentant la prudence ou en cherchant des opportunités de gain durant les périodes de volatilité élevée. Pour Apple, l'existence d'un seul jour dans l'État 0 soulève des questions intéressantes sur ce qui a causé ce comportement anormal et comment de telles anomalies peuvent être interprétées et potentiellement exploitées.

En résumé, cette analyse démontre la capacité du HMM à deux états non seulement à simplifier la compréhension des séries temporelles financières mais aussi à révéler des informations sous-jacentes qui ne sont pas immédiatement apparentes. Les résultats soulignent l'importance de l'analyse des états cachés pour déchiffrer la nature des régimes de marché, ce qui peut être d'une aide précieuse pour les analystes financiers, les gestionnaires de portefeuille, et les investisseurs cherchant à optimiser leurs décisions basées sur l'évolution des prix des actions.

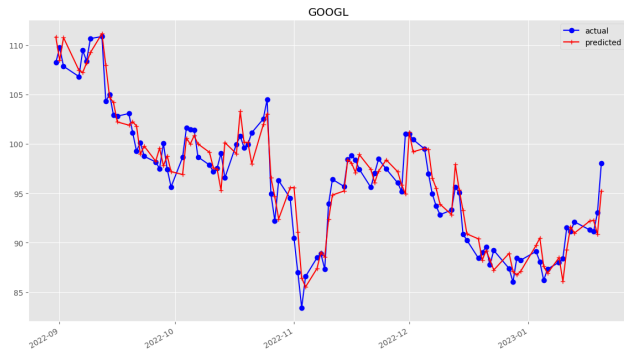


FIGURE 6 – Prédictions du HMM pour les prix de clôture de Google

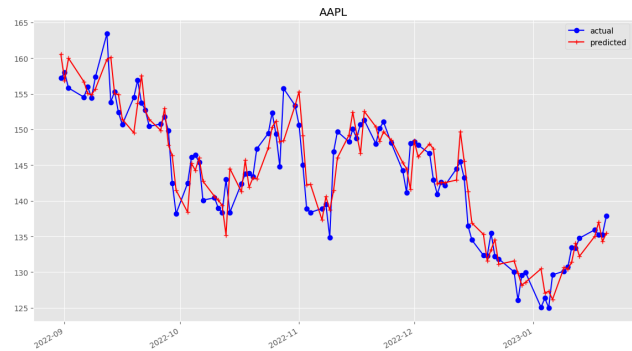


FIGURE 7 – Prédictions du HMM pour les prix de clôture d'Apple

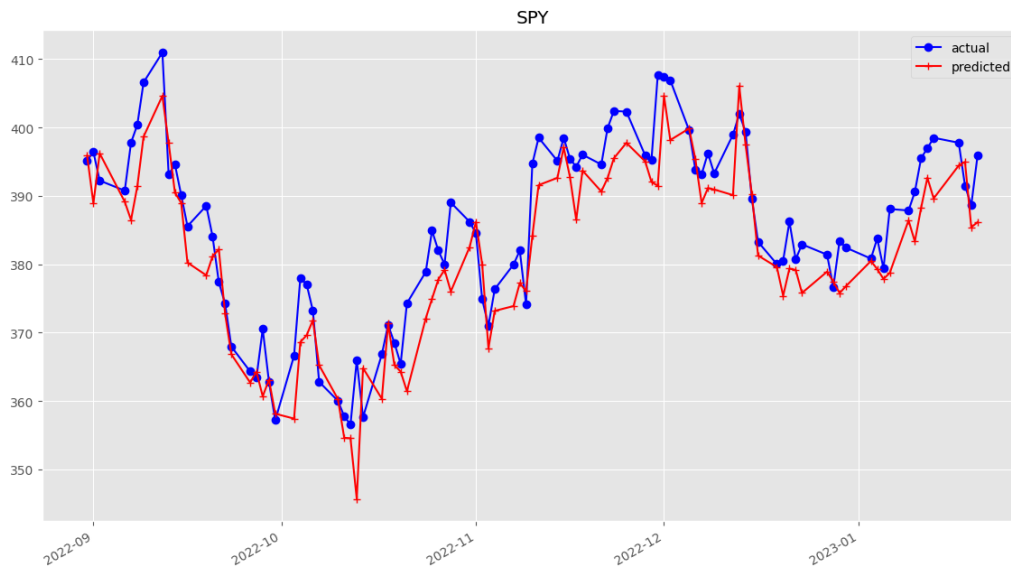


FIGURE 8 – Prédictions du HMM pour l'indice SPY

FIGURE 9 – Prédictions du HMM pour Google, Apple, et l'indice SPY

6 CONCLUSION

Dans ce travail, nous avons exploré l'application des modèles de Markov cachés (HMM) pour la modélisation des séries chronologiques financières, en mettant un accent particulier sur les prix de clôture des actions de Google, Apple et de l'indice SPY. Notre analyse a démontré que les HMM peuvent capturer efficacement les transitions entre différents états du marché et offrir des prédictions significatives sur les comportements futurs des prix des actions.

Nous avons également comparé les performances des HMM avec celles d'autres modèles prédictifs populaires tels que les ARIMA et les LSTM. Bien que les LSTM, en particulier,

possèdent une capacité théorique à modéliser des dépendances à long terme plus complexes, notre étude a révélé que les HMM, en raison de leur simplicité et efficacité, peuvent être préférables dans des contextes où les données sont limitées ou quand la rapidité et la simplicité de développement sont cruciales.

En outre, l'exploration de combinaisons innovantes de modèles, comme l'intégration des HMM avec des modèles génératifs basés sur les réseaux de neurones, suggère une voie prometteuse pour de futures recherches. Cette direction est inspirée par des travaux récents tels que l'article [3], qui illustrent le potentiel de ces approches hybrides.

Pour conclure, notre recherche souligne l'importance de choisir le modèle approprié en fonction du contexte spécifique et des données disponibles. Elle ouvre également la voie à de futures études sur l'intégration des HMM avec des techniques d'intelligence artificielle plus avancées pour améliorer la précision et l'efficacité de la modélisation des séries temporelles financières.

RÉFÉRENCES

- [1] Qian Liu, Chaojie Wang, and Chengjun Wang. Stock market trend analysis using hidden markov model and long short term memory, 2021.
- [2] Lucia Schiavone Luigi Catello, Ludovica Ruggiero and Mario Valentino. Hidden markov models for stock market prediction, 2023.
- [3] Zhihang Niu, Bo Wu, Hao Wang, Runze Wu, and Yunchuan Chen. Powering hidden markov model by neural network based generative model, 2019.