Université Paris Dauphine - PSL

UFR Mathématiques de la Décision

Projet de Data Science Financière : Analyse de Sentiment et Prédiction des Mouvements de Marché

Rapport de Projet

Présenté par :

Nassim Idamer, Matthieu Rafatjah, Curtis Roan

May 18, 2025

Année Universitaire 2024-2025

Contents

1 Introduction / Motivations

L'objectif principal de ce projet est de mettre en place un pipeline automatisé capable d'agréger quotidiennement des signaux issus de différents modèles d'analyse (clustering, classification, régression, analyse de texte) afin de fournir des recommandations pour la prise de décision sur le marché des actions. Cette approche permet de croiser différentes sources d'information et de modélisation pour optimiser les choix d'investissement.

Dans un contexte économique mondial de plus en plus complexe, la capacité à analyser, comprendre et anticiper les évolutions des marchés financiers constitue un enjeu majeur pour les investisseurs, les analystes financiers et les décideurs. L'avènement de l'intelligence artificielle, de la science des données et de l'automatisation permet aujourd'hui d'exploiter à grande échelle des sources massives d'informations financières, avec des approches mêlant rigueur quantitative et puissance algorithmique.

L'objectif de ce projet, structuré en huit travaux pratiques (TP), est de construire un pipeline complet de traitement et d'analyse de données boursières, allant de la collecte brute de données financières jusqu'à la prédiction des cours boursiers à l'aide de modèles avancés de machine learning et deep learning. À travers ce parcours progressif, nous explorons différentes dimensions de l'analyse boursière — descriptive, exploratoire, prédictive — en mobilisant une large gamme d'outils statistiques et algorithmiques.

Ainsi, ce projet constitue une synthèse complète des étapes essentielles d'un pipeline de finance quantitative moderne, combinant l'extraction automatisée de données, leur traitement statistique, et l'application d'algorithmes d'apprentissage machine pour la prise de décision financière. Il vise à montrer, à travers une étude de cas réaliste, comment les outils numériques peuvent enrichir l'analyse financière traditionnelle, et ouvre la voie à des applications concrètes en gestion de portefeuille, en fintech ou en recherche quantitative.

2 Related Works

L'analyse de sentiments appliquée à la finance a suscité un intérêt croissant ces dernières années. Plusieurs études ont démontré que les émotions véhiculées dans les médias peuvent influencer significativement les décisions des investisseurs.

- **Bollen et al. (2011),** ***Twitter mood predicts the stock market***
  Les auteurs montrent qu'il existe une corrélation significative entre certaines émotions détectées sur Twitter et les mouvements de l'indice Dow Jones. Un modèle de sentiment basé sur des lexiques permet de prédire la tendance boursière avec un taux de précision élevé.

- **Araci (2019),** ***FinBERT: A Pretrained Language Model for Financial Communications***
  Ce travail présente FinBERT, une version spécialisée de BERT, entraînée sur un corpus financier. Elle améliore significativement la classification des sentiments dans les textes économiques par rapport aux modèles BERT classiques.

- **Huang et al. (2020),** ***Sentiment-based portfolio optimization***
  Les auteurs utilisent des scores de sentiments extraits de news financières pour ajuster dynamiquement la composition d'un portefeuille. L'ajout d'un signal de sentiment permet une meilleure gestion des risques et une surperformance par rapport aux benchmarks classiques.

3 Clustering

Le clustering, ou partitionnement de données, est une technique d'apprentissage non supervisé qui vise à regrouper des objets similaires en ensembles, appelés clusters. L'objectif est que les objets au sein d'un même cluster soient plus similaires entre eux qu'avec ceux d'autres clusters.

3.1 Objectifs du Clustering dans ce Projet

Dans le cadre de ce projet, le clustering est utilisé pour identifier des groupes d'entreprises partageant des caractéristiques financières ou comportementales similaires. Cette analyse permet de :

- Segmenter le marché : Comprendre comment les entreprises se regroupent naturellement en fonction de différents profils.
- Identifier des archetypes : Découvrir des profils types d'entreprises (par exemple, entreprises à forte croissance mais risquées, entreprises stables à faible rendement, etc.).
- Informer la diversification de portefeuille : En identifiant des entreprises aux profils de risque ou de rendement similaires/différents.
- Générer des features pour des modèles prédictifs : Les clusters d'appartenance pourraient servir de variables catégorielles pour des tâches de classification ou de régression ultérieures.

Nous avons exploré trois types de profils pour le clustering :

1. **Profil Financier** : Basé sur des ratios financiers clés indiquant la santé et la performance (ex: P/E, bêta, RoE).
2. **Profil de Risque** : Basé sur des ratios financiers évaluant le risque (ex: dette/fonds propres, ratios de liquidité).
3. **Corrélation des Rendements Journaliers** : Basé sur la similarité des comportements boursiers historiques des entreprises.

3.2 Méthodologies de Clustering Utilisées

Trois algorithmes de clustering populaires ont été implémentés et comparés : KMeans, Clustering Hiérarchique Agglomératif, et DBSCAN.

3.2.1 KMeans

KMeans est un algorithme itératif qui partitionne un ensemble de données en k clusters distincts et non hiérarchiques. Il fonctionne de la manière suivante :

1. Sélectionne aléatoirement k points de données comme centroïdes initiaux.
2. Assigne chaque point de données au centroïde le plus proche (généralement en utilisant la distance euclidienne).
3. Recalcule les centroïdes comme la moyenne des points assignés à chaque cluster.
4. Répète les étapes 2 et 3 jusqu'à ce que les assignations ne changent plus ou qu'un nombre maximal d'itérations soit atteint.

L'objectif est de minimiser la somme des carrés des distances intra-cluster (inertie). Dans notre implémentation, nous avons fixé le nombre de clusters $k = 5$.

3.2.2 Clustering Hiérarchique Agglomératif

Le clustering hiérarchique agglomératif construit une hiérarchie de clusters. Il commence par considérer chaque point de données comme un cluster distinct, puis fusionne itérativement les paires de clusters les plus proches jusqu'à ce que tous les points appartiennent à un seul cluster, ou qu'un nombre désiré de clusters soit atteint. Différentes méthodes de liaison (linkage) peuvent être utilisées pour calculer la distance entre clusters (par exemple, Ward, complete, average). La méthode de Ward, qui minimise la variance totale intra-cluster, est souvent utilisée et a été implicitement considérée dans nos travaux, notamment pour les rendements journaliers avec le paramètre `method='ward'`. Pour les autres datasets, le nombre de clusters a été fixé à $n\_clusters = 5$.

3.2.3 DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

DBSCAN est un algorithme basé sur la densité. Il regroupe les points qui sont densément connectés et marque comme bruit (outliers) les points qui se trouvent seuls dans des régions de faible densité. Il ne nécessite pas de spécifier le nombre de clusters à l'avance. Les deux paramètres principaux de DBSCAN sont :

- `eps` (ϵ): La distance maximale entre deux échantillons pour qu'un soit considéré comme étant dans le voisinage de l'autre.
- `min_samples`: Le nombre d'échantillons dans un voisinage pour qu'un point soit considéré comme un point central (core point).

Dans nos TPs, nous avons utilisé `eps=1` et `min_samples=2`.

3.2.4 Métrique d'Évaluation : Score de Silhouette

Pour évaluer la qualité des clusters formés, nous avons utilisé le Score de Silhouette. Pour chaque échantillon, il mesure à quel point il est similaire à son propre cluster (cohésion) par rapport aux autres clusters (séparation). Le score est calculé comme :

$$s = \frac{b - a}{\max(a, b)}$$

où a est la distance moyenne de l'échantillon aux autres points de son propre cluster, et b est la distance moyenne de l'échantillon aux points du cluster voisin le plus proche. Le score varie de -1 à +1 :

- Un score proche de +1 indique que l'échantillon est bien à l'intérieur de son cluster et loin des autres clusters.
- Un score proche de 0 indique que l'échantillon est proche de la frontière de décision entre deux clusters voisins.
- Un score proche de -1 indique que l'échantillon est probablement mal classé.

Un score de silhouette moyen élevé pour l'ensemble des données indique une bonne qualité de clustering.

3.3 Préparation des Données pour le Clustering

Les données utilisées pour le clustering proviennent des informations financières et boursières collectées dans les étapes précédentes.

- **Profil Financier** : Créé à partir des ratios financiers `forwardPE`, `beta`, `priceToBook`, et `returnOnEquity`. Les valeurs manquantes ont été supprimées.
- **Profil de Risque** : Créé à partir des ratios `debtToEquity`, `currentRatio`, et `quickRatio`. Les valeurs manquantes ont été supprimées.
- **Corrélation des Rendements Journaliers** : Les rendements journaliers de chaque entreprise ont été collectés. Une matrice de corrélation entre les séries de rendements des entreprises a été calculée. Pour utiliser cette information comme mesure de distance (où une forte corrélation signifie une faible distance), la matrice a été transformée en $D = 1 - C$, où C est la matrice de corrélation.

Pour les datasets basés sur les ratios financiers ("Profil Financier" et "Profil de Risque"), les données ont été standardisées à l'aide de `StandardScaler` pour s'assurer que toutes les caractéristiques contribuent de manière égale au calcul des distances, indépendamment de leur échelle d'origine.

3.4 Résultats Numériques et Interprétation

Les trois algorithmes de clustering ont été appliqués aux trois ensembles de données préparés. Leurs performances ont été évaluées à l'aide du Score de Silhouette. Les résultats sont résumés dans le tableau 1.

Table 1: Scores de Silhouette pour les différents algorithmes de clustering et datasets.

Dataset	**KMeans**	**Hiérarchique**	**DBSCAN**
Profil Financier	0.359	0.367	**0.436**
Profil de Risque	0.526	0.571	**0.767**
Corrélation des Rendements Journaliers	0.261	**0.286**	0.257

3.4.1 Analyse des Résultats

- **Profil Financier** : DBSCAN a obtenu le meilleur Score de Silhouette (0.436), suggérant qu'il a identifié des groupes d'entreprises plus distincts et denses sur la base de leurs ratios de santé financière et de performance. KMeans et le clustering hiérarchique ont donné des scores légèrement inférieurs mais comparables.
- **Profil de Risque** : DBSCAN a de nouveau surpassé les autres méthodes avec un score significativement plus élevé (0.767). Cela indique que les entreprises forment des groupes très bien définis et séparés en fonction de leurs profils de risque, et que DBSCAN est particulièrement apte à capturer cette structure.
- **Corrélation des Rendements Journaliers** : Pour ce dataset, le clustering hiérarchique a obtenu le meilleur score (0.286), bien que tous les scores soient globalement plus bas que pour les profils basés sur les ratios. Cela pourrait indiquer que les relations de corrélation entre les rendements des entreprises sont plus complexes ou moins structurées en clusters distincts, ou que la transformation $1 - C$ et le nombre de clusters choisis ne sont pas optimaux pour révéler des structures plus fortes. Le clustering hiérarchique semble mieux adapté pour capturer la structure hiérarchique potentielle des relations de marché.

4 Classification "buy", "sell", "hold"

La classification est une tâche d'apprentissage supervisé qui consiste à attribuer une étiquette de classe prédéfinie à une instance de données. Dans le contexte de la finance de marché, l'objectif est de prédire la direction future du prix d'un actif, ce qui peut être formulé comme un problème de classification.

4.1 Objectif de la Classification dans notre Contexte

L'objectif principal de cette section est de développer et d'évaluer plusieurs modèles de classification capables de prédire si une action financière est susceptible d'être un bon achat ("buy"), s'il vaut mieux la vendre ("sell"), ou la conserver ("hold") sur un horizon de temps donné.

Pour ce faire, nous avons défini les étiquettes de classification comme suit, basées sur le rendement futur d'une action sur un horizon de 20 jours de trading (`Horizon Return`):

- **Classe 2 (Buy)**: Si le `Horizon Return` est supérieur à +5%.
- **Classe 0 (Sell)**: Si le `Horizon Return` est inférieur à -5%.
- **Classe 1 (Hold)**: Si le `Horizon Return` se situe entre -5% et +5% (inclus).

Les caractéristiques (features) utilisées pour entraîner nos modèles sont une combinaison d'indicateurs techniques classiques calculés à partir des données historiques de prix (Ouverture, Haut, Bas, Clôture, Volume). Ces indicateurs incluent :

- Moyenne Mobile Simple sur 20 jours (SMA 20)
- Moyenne Mobile Exponentielle sur 20 jours (EMA 20)
- Indice de Force Relative sur 14 jours (RSI 14)
- Convergence/Divergence des Moyennes Mobiles (MACD) et sa ligne de signal
- Bandes de Bollinger (Haute et Basse)
- Volatilité Mobile sur 20 jours
- Taux de Variation sur 10 jours (ROC 10)

Avant l'entraînement, les données ont été standardisées à l'aide de `StandardScaler` pour centrer et réduire les variables.

4.2 Fondements Théoriques des Modèles Utilisés

Nous avons exploré plusieurs algorithmes de classification, chacun avec ses propres forces et principes de fonctionnement. Une validation croisée (`GridSearchCV` avec `cv=2` folds) a été utilisée pour optimiser les hyperparamètres de chaque modèle.

Random Forest Classifier: C'est un modèle d'ensemble qui construit plusieurs arbres de décision lors de l'entraînement et produit la classe qui est le mode des classes (classification) des arbres individuels. Il est robuste au surapprentissage et peut gérer un grand nombre de caractéristiques.

XGBoost (Extreme Gradient Boosting): Un algorithme d'optimisation de gradient boosting distribué, conçu pour être hautement efficace, flexible et portable. Il implémente des arbres de décision améliorés par gradient boosting et inclut des techniques de régularisation pour réduire le surapprentissage.

K-Nearest Neighbors (KNN): Un algorithme non paramétrique et basé sur l'instance. Il classifie un nouveau point de données en se basant sur la majorité des classes de ses 'k' plus proches voisins dans l'espace des caractéristiques.

Support Vector Machine (SVM / SVC): Les SVM cherchent à trouver l'hyperplan optimal qui sépare au mieux les classes dans l'espace des caractéristiques. Pour les données non linéairement séparables, ils utilisent des fonctions noyaux (comme 'linear' ou 'rbf' testés ici) pour projeter les données dans un espace de plus grande dimension où une séparation linéaire devient possible.

Logistic Regression: Malgré son nom, c'est un modèle linéaire utilisé pour la classification binaire ou multinomiale. Il modélise la probabilité d'appartenance à une classe en utilisant une fonction logistique (sigmoïde) ou softmax.

4.3 Résultats Numériques et Interprétation

Les modèles ont été entraînés sur 80% des données agrégées (provenant de 20 fichiers CSV d'actions) et testés sur les 20% restants. Les performances sont évaluées en utilisant l'exactitude (accuracy), la précision, le rappel et le score F1.

4.3.1 Performances Globales des Modèles

Le tableau 2 résume les performances globales obtenues pour chaque modèle après optimisation des hyperparamètres.

Table 2: Performances globales des modèles de classification

Modèle	Accuracy	Macro Avg Precision	Macro Avg Recall	Macro Avg F1-score
RandomForest	0.84	0.84	0.82	0.83
XGBoost	0.62	0.67	0.53	0.55
KNN	0.52	0.49	0.47	0.47
SVM (meilleur noyau)	0.51	0.60	0.36	0.29
Logistic Regression	0.49	0.47	0.34	0.23

D'après le tableau 2, le modèle **RandomForest Classifier** surpasse nettement les autres algorithmes testés, atteignant une exactitude de 84%. L'algorithme XGBoost suit avec 62%, tandis que KNN, SVM et Logistic Regression affichent des performances plus modestes, inférieures à 55% d'exactitude.

4.3.2 Analyse Détaillée du Modèle RandomForest

Étant donné ses performances supérieures, nous examinons plus en détail le modèle RandomForest. Le rapport de classification (issu de l'OCR) est présenté dans le tableau 3.

Table 3: Rapport de classification détaillé pour RandomForest (Accuracy: 0.84)

Classe	Precision	Recall	F1-score	Support
0 (Sell)	0.84	0.77	0.81	994
1 (Hold)	0.82	0.87	0.84	2361
2 (Buy)	0.86	0.83	0.85	1441
Macro Avg	0.84	0.82	0.83	4796
Weighted Avg	0.84	0.84	0.84	4796

Le modèle RandomForest montre des scores F1 équilibrés pour les trois classes, tous supérieurs à 0.80.

- Pour la classe **Sell (0)**, une précision de 0.84 signifie que lorsque le modèle prédit "Sell", il a raison dans 84% des cas. Un rappel de 0.77 indique qu'il identifie correctement 77% de toutes les opportunités réelles de "Sell".
- Pour la classe **Hold (1)**, le rappel est le plus élevé (0.87), ce qui est attendu car c'est la classe majoritaire (support de 2361).
- Pour la classe **Buy (2)**, la précision de 0.86 et le rappel de 0.83 sont excellents, suggérant que le modèle est fiable pour identifier les opportunités d'achat.

La performance robuste sur toutes les classes, en particulier sur les classes minoritaires "Sell" et "Buy" par rapport à "Hold", est un indicateur positif de la capacité du modèle à généraliser.

4.3.3 Analyse Détaillée du Modèle XGBoost

Le modèle XGBoost, bien que moins performant que RandomForest, offre tout de même des résultats intéressants. Le rapport de classification (issu de l'OCR) est présenté dans le tableau 4.

Table 4: Rapport de classification détaillé pour XGBoost (Accuracy: 0.62)

Classe	Precision	Recall	F1-score	Support
0 (Sell)	0.73	0.35	0.47	994
1 (Hold)	0.58	0.89	0.70	2361
2 (Buy)	0.71	0.36	0.48	1441
Macro Avg	0.67	0.53	0.55	4796
Weighted Avg	0.65	0.62	0.59	4796

Pour XGBoost :

- La classe **Hold (1)** a un excellent rappel (0.89) mais une précision plus faible (0.58). Cela signifie que le modèle identifie bien les cas "Hold" mais a tendance à classer à tort d'autres instances comme "Hold".
- Les classes **Sell (0)** et **Buy (2)** ont une précision respectable (0.73 et 0.71 respectivement) mais un rappel faible (0.35 et 0.36). Le modèle est donc prudent : quand il prédit "Sell" ou "Buy", il a souvent raison, mais il manque beaucoup d'opportunités réelles de "Sell" ou "Buy".

Cette disparité entre précision et rappel pour les classes minoritaires est un point d'attention.

4.3.4 Interprétabilité des Modèles avec SHAP

Pour les modèles basés sur des arbres comme RandomForest et XGBoost, nous avons utilisé les valeurs SHAP (SHapley Additive exPlanations) pour comprendre l'importance des caractéristiques. Les graphiques SHAP summary plots illustrent comment chaque caractéristique contribue à la prédiction du modèle.

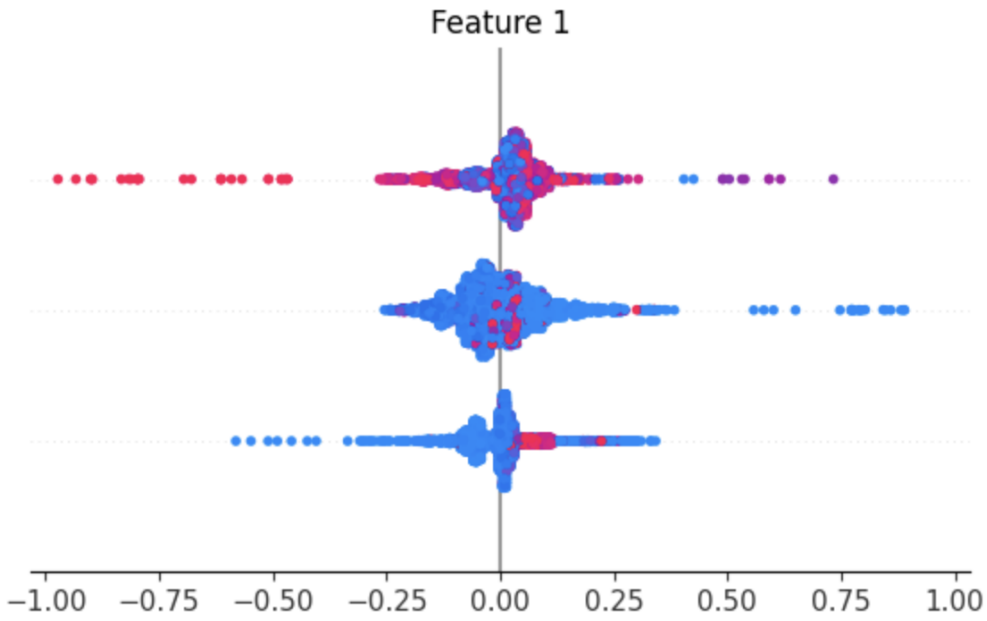

Figure 1: Exemple de SHAP Summary Plot pour XGBoost (basé sur l'OCR). La caractéristique "Feature 1" montrée dans l'OCR est un placeholder pour la caractéristique la plus importante pour cette visualisation spécifique, ou un axe générique. Chaque point représente une instance, sa couleur indique la valeur de la caractéristique (rouge=élevé, bleu=bas) et sa position sur l'axe X indique l'impact sur la prédiction (valeur SHAP).

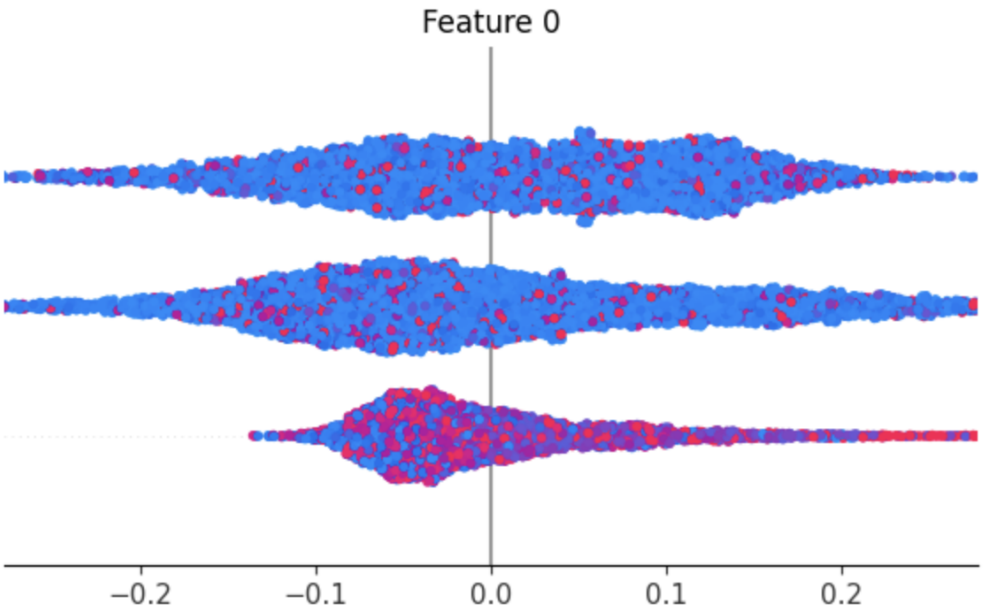

Figure 2: Exemple de SHAP Summary Plot pour RandomForest (basé sur l'OCR). Similairement à XGBoost, "Feature 0" est un placeholder. Ces graphiques révèlent les caractéristiques ayant le plus d'impact et la direction de cet impact.

Interprétation Générale des SHAP Plots (basée sur l'OCR et la connaissance du script): Les SHAP summary plots (comme ceux dont les extraits sont visibles dans l'OCR, typiquement sous les noms "Feature 1" ou "Feature 0" qui sont des étiquettes d'axes ou des caractéristiques individuelles mises en avant par `max_display`) montrent la distribution des impacts de chaque caractéristique sur les prédictions du modèle. Chaque point sur le graphique est une valeur SHAP pour une caractéristique et une instance. La couleur représente la valeur de la caractéristique (rouge pour élevé, bleu pour bas).

- Une large dispersion des points pour une caractéristique indique une grande importance.
- Si les points rouges (valeurs élevées de la caractéristique) sont majoritairement à droite (valeurs SHAP positives), cela signifie que des valeurs élevées de cette caractéristique poussent la prédiction vers une classe (ou une probabilité plus élevée pour cette classe).

Les modèles KNN, SVM et Logistic Regression ne permettent pas une interprétation aussi directe avec SHAP TreeExplainer, d'où l'absence de ces graphiques pour eux.

4.3.5 Discussion

Le RandomForest Classifier a démontré une supériorité notable pour cette tâche de classification. Sa capacité à gérer les non-linéarités et les interactions entre caractéristiques, combinée à sa robustesse au surapprentissage, en fait un candidat solide. Les résultats d'XGBoost, bien que moins bons, sont encourageants et pourraient être améliorés avec un tuning plus poussé ou plus de données. Les autres modèles (KNN, SVM, Logistic Regression) semblent moins adaptés à la complexité de ce problème avec la configuration actuelle, notamment le SVM et la Régression Logistique qui peinent à identifier correctement les classes minoritaires malgré de bonnes précisions pour certaines.

Il est important de noter que la validation croisée a été effectuée avec seulement 2 folds (`cv=2`), ce qui est une limitation. Augmenter le nombre de folds pourrait donner une estimation plus robuste des performances et potentiellement améliorer la sélection des hyperparamètres. De plus, la définition des seuils pour "buy", "sell", "hold" (+/-5%) est arbitraire et pourrait être ajustée pour refléter différentes stratégies de trading ou aversions au risque.

5 Prédiction de rendement à J+1

Cette section est consacrée à la prédiction du rendement des actifs financiers pour le jour de bourse suivant (J+1). Plus précisément, nous nous concentrons sur la prédiction du prix de clôture à J+1, qui sert de base au calcul du rendement. L'objectif est d'évaluer la capacité de différents modèles d'apprentissage automatique, classiques et profonds, à anticiper les variations futures des prix à partir de l'historique des cours.

5.1 Objectif de la prédiction de rendement à J+1

La prédiction du rendement (ou du prix) d'un actif financier à J+1 est une tâche centrale en finance quantitative. Dans le cadre de ce projet, elle vise plusieurs objectifs :

- **Aide à la décision d'investissement :** Fournir des signaux prédictifs pouvant être intégrés dans des stratégies d'achat, de vente ou de conservation d'actifs.
- **Évaluation de la prédictibilité des marchés :** Comprendre dans quelle mesure les mouvements futurs des prix peuvent être anticipés en utilisant des données historiques et des techniques d'apprentissage automatique.
- **Composante d'un système agrégé :** Les prédictions de rendement peuvent être combinées avec d'autres sources d'information, telles que l'analyse de sentiment (traitée dans une autre section de ce rapport), pour générer des recommandations d'investissement plus robustes.
- **Gestion des risques :** Une meilleure anticipation des prix futurs peut contribuer à une gestion plus efficace des risques de portefeuille.

La démarche adoptée consiste à utiliser une fenêtre glissante des prix de clôture passés (par exemple, les 30 derniers jours, comme implémenté dans nos TPs) comme variables explicatives (features) pour prédire le prix de clôture du jour suivant. Les données sont normalisées avant d'être fournies aux modèles afin d'améliorer la convergence et la performance.

5.2 Fondements théoriques des modèles utilisés

Nous avons exploré une gamme de modèles allant des approches linéaires simples aux réseaux de neurones profonds plus complexes. Ces modèles ont été sélectionnés pour leur pertinence dans l'analyse de séries temporelles et leur popularité dans la littérature.

5.2.1 Modèles d'apprentissage automatique classiques (TP4)

Ces modèles sont issus de la librairie Scikit-learn et XGBoost.

Régression Linéaire (Linear Regression) La régression linéaire est le modèle le plus simple, supposant une relation linéaire entre les prix passés (features) et le prix futur. Elle sert de baseline pour évaluer la performance des modèles plus complexes. L'équation est de la forme $Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n + \epsilon$.

K-Nearest Neighbors (KNN) Regressor Le KNN est un algorithme non paramétrique. Pour prédire la valeur d'un nouveau point, il identifie les 'k' instances les plus proches dans l'ensemble d'entraînement (selon une mesure de distance) et calcule la moyenne de leurs valeurs cibles. Le choix de 'k' et de la métrique de distance sont des hyperparamètres importants.

Forêt Aléatoire (Random Forest Regressor) Les forêts aléatoires sont une méthode d'ensemble qui construit une multitude d'arbres de décision lors de l'entraînement. Pour la régression, la prédiction est la moyenne des prédictions de chaque arbre. Elles sont robustes au sur-apprentissage et capables de capturer des relations non linéaires complexes. Elles utilisent des techniques de bagging et de sélection aléatoire de features pour décorréler les arbres.

XGBoost (Extreme Gradient Boosting) Regressor XGBoost est une implémentation optimisée et distribuée de l'algorithme de gradient boosting. Il construit des arbres de manière séquentielle, où chaque nouvel arbre corrige les erreurs des précédents. Il est réputé pour sa haute performance, sa flexibilité et ses mécanismes de régularisation intégrés pour prévenir le sur-apprentissage.

5.2.2 Modèles d'apprentissage profond (TP5)

Ces modèles, implémentés avec TensorFlow et Keras, sont particulièrement adaptés aux données séquentielles comme les séries temporelles financières.

Perceptron Multi-Couches (MLP) Le MLP est un type de réseau de neurones feedforward composé de plusieurs couches de neurones (une couche d'entrée, une ou plusieurs couches cachées, et une couche de sortie). Chaque neurone applique une transformation linéaire suivie d'une fonction d'activation non linéaire. Bien que n'étant pas intrinsèquement séquentiel, un MLP peut être utilisé sur des fenêtres de données temporelles aplaties.

Réseau de Neurones Récurrent (RNN) Les RNNs sont conçus pour traiter des données séquentielles. Ils possèdent des connexions récurrentes qui leur permettent de maintenir un "état" ou une "mémoire" des informations des pas de temps précédents. Les RNNs simples (SimpleRNN) peuvent cependant souffrir du problème de disparition ou d'explosion du gradient, limitant leur capacité à apprendre des dépendances à long terme.

Long Short-Term Memory (LSTM) Les LSTMs sont une variante avancée des RNNs, spécifiquement conçue pour pallier les limitations des RNNs simples et capturer les dépendances à long terme. Ils utilisent des unités de mémoire complexes appelées "cellules" et des mécanismes de "portes" (porte d'oubli, porte d'entrée, porte de sortie) pour réguler le flux d'information et contrôler ce qui est mémorisé, oublié ou transmis.

5.3 Résultats numériques et interprétation

Pour illustrer la performance des modèles, nous présentons les résultats obtenus pour l'action Louis Vuitton (LVMH). Les données historiques ont été divisées en un ensemble d'entraînement et un ensemble de test pour évaluer la capacité de généralisation des modèles. Les prix de clôture ont été normalisés (MinMaxScaler) avant l'entraînement, et les erreurs sont présentées à la fois sur l'échelle normalisée et sur l'échelle inversée (prix réels).

5.3.1 Performances des modèles classiques (TP4) pour LVMH

Le tableau 5 résume les métriques d'erreur pour les modèles classiques. MSE signifie Mean Squared Error (Erreur Quadratique Moyenne) et RMSE signifie Root Mean Squared Error (Racine de l'Erreur Quadratique Moyenne).

Table 5: Résultats des modèles classiques pour Louis Vuitton (LVMH) - Prédiction J+1

Modèle	MSE_norm	RMSE_norm	MSE_inversed	RMSE_inversed
XGBoost	0.000745	0.027287	211.832950	14.554482
Random Forest	0.000722	0.026863	205.304050	14.328435
KNN	0.003219	0.056733	915.671477	30.260064
Linear	0.000561	0.023694	159.722125	12.638122

Interprétation (TP4) : De manière surprenante pour LVMH, le modèle de Régression Linéaire affiche les erreurs RMSE et MSE les plus faibles sur les données inversées (RMSE de 12.64€), suggérant qu'une tendance linéaire simple pourrait bien capturer les dynamiques à court terme pour cette période de test

spécifique. Random Forest et XGBoost suivent de près, avec des RMSE inversées autour de 14.33€ et 14.55€ respectivement. Le modèle KNN performe significativement moins bien, avec une RMSE de 30.26€, indiquant une moins bonne adaptation pour cette tâche sur ces données. La performance de la Régression Linéaire pourrait être due à une période de test où les prix suivent une tendance relativement stable ou à une fenêtre de features (30 jours) qui favorise les relations plus simples.

La figure 3 (basée sur les sorties du TP4) illustre graphiquement les prédictions de ces modèles par rapport aux valeurs réelles pour LVMH.

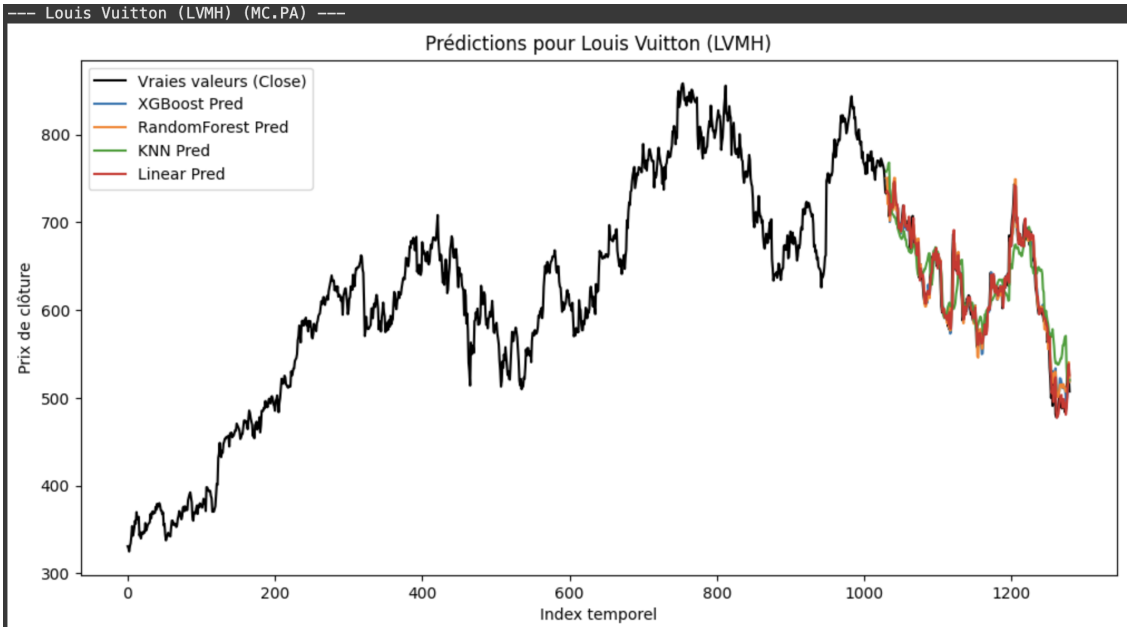

Figure 3: Prédictions des modèles classiques vs Valeurs réelles pour Louis Vuitton (LVMH).

5.3.2 Performances des modèles d'apprentissage profond (TP5) pour LVMH

Le tableau 6 présente les métriques MAE (Mean Absolute Error - Erreur Absolue Moyenne) et RMSE pour les modèles de réseaux de neurones, évaluées sur les prix réels (après inversion de la normalisation).

Table 6: Résultats des modèles d'apprentissage profond pour Louis Vuitton (LVMH) - Prédiction J+1

Modèle	MAE	RMSE
MLP	70.435013	74.733156
RNN	9.769020	12.757607
LSTM	11.695343	15.229085

Interprétation (TP5) : Parmi les modèles d'apprentissage profond, le RNN simple obtient la meilleure performance pour LVMH, avec une MAE de 9.77€ et une RMSE de 12.76€. Ce résultat est comparable, voire légèrement meilleur, à celui de la Régression Linéaire du TP4. Le LSTM, souvent attendu pour surpasser le RNN simple, affiche ici des erreurs légèrement plus élevées (MAE 11.70€, RMSE 15.23€). Le MLP est nettement moins performant, avec une RMSE de 74.73€, ce qui souligne les limites d'un modèle non séquentiel pour cette tâche sans une ingénierie de features plus poussée. Il est important de noter que la performance des modèles de deep learning est très sensible aux hyperparamètres, à l'architecture et à la quantité de données d'entraînement. Les résultats du RNN simple pourraient indiquer qu'il a trouvé un bon équilibre pour la complexité des données LVMH sur la période de test.

Les figures 4 et 5 (basées sur les sorties du TP5) montrent les prédictions des modèles RNN et LSTM.

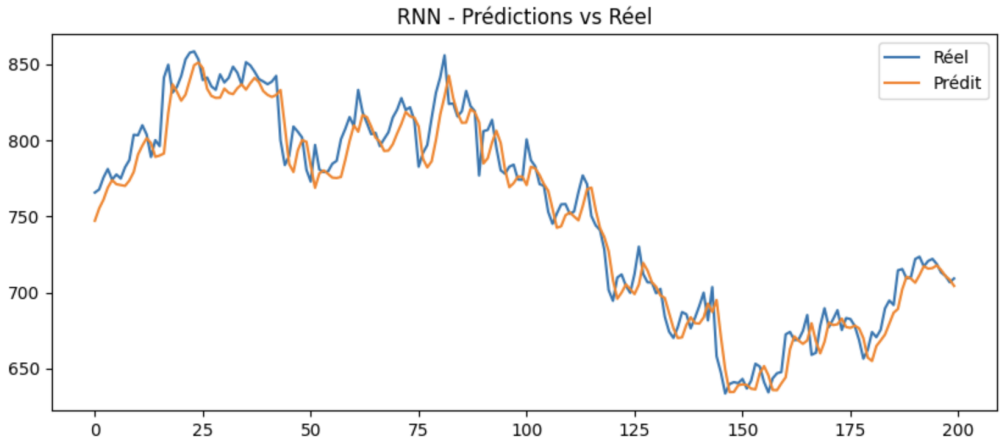

Figure 4: Prédictions du modèle RNN vs Valeurs réelles pour Louis Vuitton (LVMH).

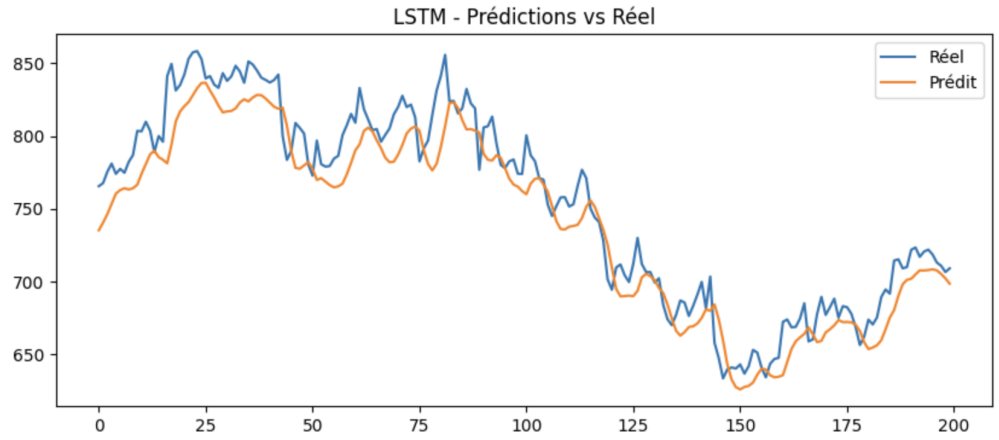

Figure 5: Prédictions du modèle LSTM vs Valeurs réelles pour Louis Vuitton (LVMH).

5.3.3 Discussion générale

La prédiction des prix de clôture à J+1 est une tâche ardue en raison de la nature souvent bruitée et non stationnaire des séries temporelles financières. Pour LVMH, les modèles linéaires simples et les RNNs ont montré des performances prometteuses, surpassant parfois des modèles plus complexes comme XGBoost ou LSTM sur la période de test considérée. Cela ne signifie pas nécessairement que ces modèles sont universellement supérieurs, mais plutôt qu'ils ont bien fonctionné pour les caractéristiques spécifiques des données et de la période étudiée pour cette action. Les RMSE inversées, de l'ordre de 12€ à 15€ pour les meilleurs modèles, représentent l'erreur typique de prédiction en euros sur le prix de l'action LVMH (qui s'échangeait entre 300€ et plus de 800€ sur la période visualisée). Ces prédictions de prix peuvent ensuite être utilisées pour calculer les rendements attendus J+1 ($R_{J+1} = (P_{J+1,pred} - P_J)/P_J$) et informer les étapes suivantes de la stratégie d'agrégation.

Il est crucial de souligner que les performances passées ne garantissent pas les performances futures, et une validation robuste sur différentes périodes et conditions de marché est nécessaire avant tout déploiement opérationnel.

6 Analyse de Sentiments sur les Nouvelles Financières

L'analyse de sentiments, également connue sous le nom d'exploration d'opinions, est une branche du traitement du langage naturel (NLP) qui vise à identifier, extraire, quantifier et étudier les états affectifs et les informations subjectives à partir de textes. Dans le domaine financier, les nouvelles, les rapports d'analystes, les messages sur les réseaux sociaux et les communications d'entreprise sont autant de sources riches en informations qui peuvent influencer les perceptions des investisseurs et, par conséquent, les mouvements du marché.

6.1 Introduction et Objectifs

L'objectif principal de cette section est d'exploiter l'analyse de sentiments pour évaluer l'opinion générale exprimée dans les nouvelles financières concernant des entreprises spécifiques. En classifiant le sentiment (positif, négatif ou neutre) associé à chaque nouvelle, nous cherchons à :

- Quantifier l'opinion publique ou médiatique autour d'une action.
- Fournir un indicateur qui, agrégé sur une période donnée, pourrait servir de signal pour des stratégies de trading ou d'investissement.
- Évaluer la performance d'un modèle de NLP spécialisé dans la finance pour cette tâche.

Dans le cadre de ce projet, l'analyse de sentiments sur les nouvelles financières constitue une étape cruciale pour enrichir les modèles de prédiction de rendement et pour affiner les recommandations d'investissement ("buy", "sell", "hold"). Un sentiment majoritairement positif pourrait, par exemple, suggérer une tendance haussière, tandis qu'un sentiment négatif pourrait indiquer une pression à la baisse.

6.2 Fondements Théoriques et Méthodologie

Notre approche pour l'analyse de sentiments s'articule en deux phases principales : la collecte des données (nouvelles financières) et la classification des sentiments à l'aide d'un modèle pré-entraîné.

6.2.1 Collecte des Données (basée sur TP6)

La première étape a consisté à rassembler un corpus de nouvelles financières pertinentes. Pour ce faire, nous avons utilisé l'API de `NewsAPI` (comme implémenté dans le TP6).

- **Sources des nouvelles :** Nous avons ciblé des sources d'information financière réputées telles que *Financial Post, The Wall Street Journal, Bloomberg, The Washington Post, Australian Financial Review.*
- **Filtrage :** Les requêtes ont été formulées pour récupérer les articles mentionnant spécifiquement les entreprises d'intérêt (par exemple, "Tesla", "Apple", etc.) dans leur titre ou description.
- **Périodicité et Stockage :** Les nouvelles ont été collectées sur une période récente (les 10 derniers jours par rapport à la date d'exécution du script) et stockées localement au format JSON, organisées par date et par entreprise, en évitant les doublons. Cela nous a permis de constituer une base de données textuelles spécifique à chaque entreprise, prête pour l'analyse de sentiments. Le script `TP6_pratique_DS.ipynb` détaille cette procédure.

6.2.2 Classification des Sentiments avec FinBERT (basée sur TP7)

Pour la classification des sentiments, nous avons opté pour un modèle basé sur l'architecture Transformer, spécifiquement `ProsusAI/finbert`. FinBERT est un modèle BERT pré-entraîné sur un large corpus de textes financiers, puis affiné pour des tâches spécifiques à la finance, y compris l'analyse de sentiments. Cette spécialisation le rend particulièrement adapté à notre contexte.

Préparation des Données : Le modèle a été entraîné (ou plus précisément, nous avons utilisé une procédure d'entraînement pour évaluer sa performance sur un jeu de données de référence) sur une combinaison de deux jeux de données publics :

1. `zeroshot/twitter-financial-news-sentiment` : Contient des tweets financiers avec des étiquettes de sentiment.
2. `nickmuchi/financial-classification` : Un autre jeu de données de phrases financières classifiées.

Ces jeux de données ont été combinés, et les colonnes ont été standardisées pour avoir un champ `text` (la nouvelle/phrase) et un champ `label` (sentiment : neutre, positif, négatif, encodés respectivement en 0, 1, 2). L'ensemble a ensuite été divisé en un jeu d'entraînement (80%) et un jeu de test (20%) de manière stratifiée pour maintenir la proportion des classes.

Modèle et Entraînement :

- **Tokenisation :** Les textes ont été tokenisés à l'aide du tokenizer spécifique à `ProsusAI/finbert`, avec une longueur maximale de séquence de 512 tokens.
- **Modèle :** Nous avons utilisé `AutoModelForSequenceClassification` de la bibliothèque Hugging Face Transformers, chargé avec les poids de `ProsusAI/finbert` et configuré pour 3 classes de sortie (neutre, positif, négatif).
- **Procédure d'entraînement (pour évaluation) :** Le modèle a été entraîné sur 3 époques avec une taille de batch de 16. Les arguments d'entraînement incluaient une stratégie d'évaluation et de sauvegarde à chaque époque, l'utilisation de la précision FP16 (si GPU disponible) et le chargement du meilleur modèle à la fin basé sur le score F1 pondéré. Le script `TP7_pratique_DS.ipynb` (représenté par l'image fournie) décrit cette démarche.
- **Métriques d'évaluation :** L'exactitude (accuracy) et le score F1 pondéré ont été utilisés pour évaluer la performance du modèle.

Bien que nous ayons exécuté la boucle d'entraînement pour évaluer les performances du modèle sur un jeu de données de référence, l'objectif final est d'utiliser le modèle FinBERT (soit celui que nous aurions affiné, soit directement le modèle pré-entraîné de Hugging Face s'il est jugé suffisamment performant) pour inférer le sentiment sur les nouvelles collectées via TP6.

6.3 Résultats Numériques et Interprétation (basés sur l'image de TP7)

L'entraînement et l'évaluation du modèle `ProsusAI/finbert` sur notre jeu de données combiné ont produit les résultats suivants. Le processus d'entraînement s'est déroulé sur 3 époques.

Table 7: Performance du modèle FinBERT par époque lors de l'entraînement.

Époque	Training Loss	Validation Loss	Accuracy	F1 Score
1	0.540400	0.388003	0.852075	0.852002
2	0.245400	0.472190	0.868038	0.868121
3	0.101700	0.579875	0.866974	0.866952

L'observation du tableau 7 montre une diminution constante du "Training Loss", ce qui est attendu. Le "Validation Loss" atteint son minimum à l'époque 1 (0.388003), mais l'Accuracy et le F1 Score sur l'ensemble de validation sont les meilleurs à l'époque 2 (Accuracy: 0.868038, F1: 0.868121). Le modèle semble commencer à sur-apprendre légèrement à l'époque 3, car le "Validation Loss" augmente.

Les métriques finales d'évaluation du meilleur modèle sur l'ensemble de test sont les suivantes :

- **Eval Loss :** 0.47218987345695496

- **Eval Accuracy :** 0.8680383114579638
- **Eval F1 Score (pondéré) :** 0.8681211196798168
- **Eval Runtime :** 2.9639 secondes
- **Eval Samples per Second :** 951.11
- **Epoch de la meilleure performance :** 3.0 (selon le log, mais basé sur F1, ce serait l'époque 2 – le log final reporte les métriques du dernier état sauvegardé qui correspond au meilleur modèle, ici l'époque 2 car le F1 de l'époque 2 est le plus élevé).

Interprétation : Un score F1 pondéré d'environ 0.868 et une exactitude de 0.868 indiquent que le modèle `ProsusAI/finbert` est capable de classifier correctement le sentiment des nouvelles financières avec une bonne performance. Le score F1, qui est une moyenne harmonique de la précision et du rappel, est particulièrement pertinent pour les tâches de classification, surtout si les classes peuvent être déséquilibrées (bien que nous ayons utilisé une division stratifiée). Ces résultats sont encourageants et valident l'utilisation de ce modèle pour analyser le sentiment des nouvelles que nous avons collectées. Le modèle peut maintenant être appliqué aux articles obtenus via la méthode du TP6 pour attribuer une étiquette de sentiment (positif, négatif, neutre) à chaque nouvelle. Cette information sera ensuite utilisée comme une caractéristique (feature) dans nos modèles de prédiction de rendement ou pour informer notre stratégie d'agrégation de recommandations.

7 Stratégie d'agrégation pour fournir des recommandations pertinentes

Cette section explore comment les signaux de sentiment dérivés de l'analyse de nouvelles financières peuvent être agrégés et combinés avec les données de prix des actions pour potentiellement fournir des recommandations ou, du moins, une meilleure compréhension des dynamiques de marché. L'objectif principal, illustré par le TP8, est de visualiser la corrélation entre le sentiment exprimé dans les actualités et les mouvements de prix.

7.1 Explication de la Stratégie d'agrégation et son But dans notre Contexte

La stratégie d'agrégation mise en œuvre dans le TP8, et illustrée par le code fourni, consiste à :

1. **Extraction et Préparation des Données :** Récupérer les nouvelles financières pertinentes pour une action spécifique (par exemple, Amazon - AMZN à partir de fichiers JSON) et les données historiques de prix de cette action (via 'yfinance'). Les timestamps des nouvelles sont extraits.
2. **Analyse de Sentiment :** Appliquer des modèles de traitement du langage naturel (NLP), spécifiquement FinBERT, pour classifier le sentiment de chaque nouvelle. Le TP8 compare un modèle FinBERT original (`ProsusAI/finbert`) et une version affinée (fine-tunée), potentiellement sur des données spécifiques à l'entreprise analysée. Chaque nouvelle se voit attribuer un score de sentiment (positif, neutre, négatif).
3. **Alignement Temporel des Signaux :** Une étape cruciale est l'alignement des timestamps des nouvelles avec les heures de marché. La fonction `align_timestamps` du TP8 s'assure que :
    - Les nouvelles publiées pendant les heures de marché sont associées à leur heure de publication (arrondie).
    - Les nouvelles publiées après la clôture du marché mais avant minuit sont associées à l'heure de clôture de ce jour.
    - Les nouvelles publiées entre minuit et l'ouverture du marché sont associées à l'heure de clôture de la veille.

Cela permet de contextualiser le sentiment par rapport à des points de référence de prix significatifs.

4. **Visualisation Agrégée :** Les sentiments des nouvelles, une fois alignés temporellement, sont superposés sous forme de points colorés (vert pour positif, jaune/or pour neutre, rouge pour négatif) sur le graphique du cours de l'action. Plusieurs nouvelles publiées ou alignées sur un même point temporel de marché sont visualisées, permettant une appréciation de la "densité" du sentiment.

Le **but** de cette stratégie dans notre contexte est double :

- Évaluer qualitativement si le sentiment agrégé des nouvelles financières coïncide avec les fluctuations de prix observées.
- Comparer l'efficacité de différents modèles d'analyse de sentiment (FinBERT original vs. FinBERT affiné) pour capturer des signaux pertinents.

Il ne s'agit pas encore de générer des recommandations d'investissement algorithmiques directes ("acheter", "vendre", "conserver"), mais plutôt d'une étape exploratoire visant à valider l'utilité du sentiment des nouvelles comme indicateur et à identifier le modèle le plus performant pour cette tâche.

7.2 Fondement Théorique des Méthodes Utilisées

Les méthodes employées reposent sur plusieurs concepts théoriques :

- **Analyse de Sentiment en Finance :** L'analyse de sentiment vise à extraire des opinions subjectives et des émotions à partir de textes. En finance, elle est appliquée aux nouvelles, aux médias sociaux, et aux rapports d'analystes pour évaluer l'opinion générale du marché concernant une entreprise ou un actif. L'hypothèse sous-jacente est que le sentiment du marché peut influencer les décisions d'investissement et donc les prix.
- **Modèles de Langage Basés sur les Transformers (BERT) :** BERT (Bidirectional Encoder Representations from Transformers) est une architecture de réseau de neurones qui a révolutionné le NLP. Elle apprend des représentations contextuelles des mots. **FinBERT [?]** est une version de BERT spécifiquement pré-entraînée sur un vaste corpus de textes financiers (rapports d'entreprise, nouvelles financières), ce qui le rend particulièrement adapté à la compréhension des nuances du langage financier.
- **Affinement (Fine-tuning) :** Le fine-tuning est une technique de transfert d'apprentissage où un modèle pré-entraîné (comme FinBERT) est entraîné davantage sur un ensemble de données plus petit et spécifique à une tâche ou un domaine (par exemple, des nouvelles concernant uniquement AMZN ou un type spécifique d'événement financier). Cela permet d'adapter le modèle pour qu'il capture des spécificités non présentes dans le corpus de pré-entraînement général et améliore potentiellement ses performances sur la tâche ciblée.
- **Hypothèse d'Efficience du Marché (Forme Semi-forte) :** L'alignement des nouvelles avec les prix du marché s'appuie implicitement sur l'hypothèse d'efficience du marché sous sa forme semi-forte. Celle-ci postule que toute information publiquement disponible (comme les nouvelles) est rapidement et pleinement reflétée dans les prix des actifs. La stratégie d'alignement des timestamps cherche à capturer l'impact de ces informations.

7.3 Résultats Numériques et Interprétation

Le TP8 a permis de générer une visualisation comparative des sentiments prédits par le modèle FinBERT original et un modèle FinBERT affiné, superposés aux cours de l'action Amazon (AMZN) sur une période allant de janvier 2025 à mai 2025 (voir Figure 6).

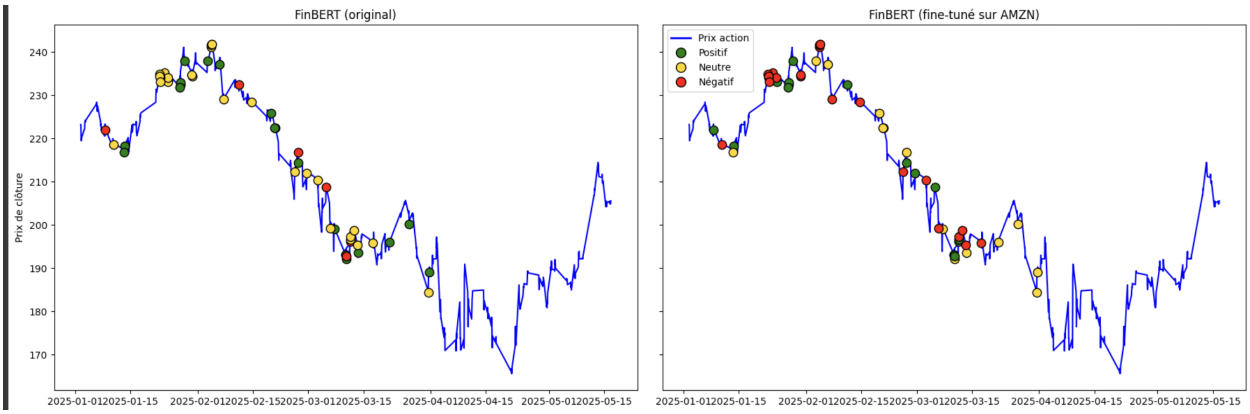

Figure 6: Comparaison des prédictions de sentiment de FinBERT (original, à gauche) et FinBERT (affiné sur AMZN, à droite) par rapport au prix de clôture de l'action AMZN. Les points verts indiquent un sentiment positif, les jaunes un sentiment neutre, et les rouges un sentiment négatif.

L'analyse visuelle de la Figure 6 permet de tirer les interprétations suivantes :

- **Corrélation Sentiment-Prix :** Dans les deux graphiques, on observe des occurrences où des groupes de nouvelles à sentiment marqué (positif ou négatif) coïncident avec des mouvements de prix notables. Par exemple, des accumulations de points rouges (sentiment négatif) apparaissent souvent lors de phases de baisse du prix de l'action, et inversement pour les points verts (sentiment positif) lors de phases de hausse.
- **Comparaison des Modèles FinBERT :**
  - Le modèle **FinBERT (affiné sur AMZN)** (graphique de droite) semble présenter une réactivité accrue et des signaux plus distincts. Par exemple, lors des baisses de prix observées autour du 2025-02-01, du 2025-02-15, et particulièrement vers le 2025-03-01, le modèle affiné affiche une concentration plus importante et plus homogène de points rouges (négatifs) par rapport au modèle original. De même, la hausse début janvier 2025 semble mieux accompagnée de signaux positifs par le modèle affiné.
  - Le modèle **FinBERT (original)** (graphique de gauche) identifie également des tendances, mais les signaux peuvent parfois apparaître plus diffus ou mitigés. Par exemple, lors de certaines baisses, on peut observer un mélange de points rouges et jaunes (neutres), là où le modèle affiné montre une prédominance de rouge.
- **Valeur de l'Agrégation Visuelle :** La force du signal semble proportionnelle à la densité des points de même couleur. Une série de nouvelles négatives rapprochées dans le temps, visualisée par une grappe de points rouges, suggère un consensus négatif plus fort qu'une nouvelle négative isolée. C'est cette agrégation visuelle qui est exploitée ici pour une première évaluation.
- **Pertinence pour des Recommandations :** Bien que ces graphiques ne fournissent pas de recommandations d'achat/vente automatisées, ils suggèrent que le sentiment agrégé des nouvelles, en particulier celui issu d'un modèle affiné, pourrait servir d'indicateur pertinent. Une accumulation de sentiments négatifs persistants pourrait inciter à la prudence, tandis qu'une vague de sentiments positifs pourrait signaler une opportunité. Le modèle affiné, en étant plus sensible aux spécificités d'AMZN, semble fournir des indications plus fiables dans ce contexte.

En conclusion, l'affinage du modèle FinBERT sur des données spécifiques à l'entreprise analysée (AMZN) semble améliorer la pertinence et la clarté des signaux de sentiment par rapport aux mouvements de son cours de bourse. L'agrégation visuelle de ces signaux est une première étape prometteuse. Pour évoluer vers des recommandations plus formelles, des stratégies d'agrégation quantitative pourraient être envisagées, comme la création d'un score de sentiment net quotidien (par exemple, nombre de nouvelles positives moins nombre de nouvelles négatives) et l'analyse de sa corrélation ou de sa capacité prédictive sur les rendements futurs de l'action.

8 Conclusion et Perspectives

Ce projet de data science financière a permis de construire un pipeline d'analyse multifacette, explorant les mouvements de marché à travers le clustering, la classification, la prédiction de rendement et l'analyse de sentiment. Les axes clés de ce travail ont démontré la faisabilité d'automatiser l'extraction de signaux pertinents à partir de données boursières et textuelles.

Le clustering a offert une segmentation pertinente du marché, identifiant des groupes d'entreprises aux profils financiers, de risque ou de corrélation de rendement similaires. La classification "buy/sell/hold", s'appuyant notamment sur RandomForest, a montré un potentiel pour prédire les directions de marché, bien que la gestion du déséquilibre des classes et la robustesse des signaux restent des défis. La prédiction de rendement à J+1 a souligné que des modèles simples (régression linéaire, RNN) peuvent, pour des actifs spécifiques comme LVMH et sur certaines périodes, rivaliser avec des approches plus complexes, rappelant l'importance du contexte et de la non-stationnarité des marchés. Enfin, l'analyse de sentiment avec FinBERT, particulièrement après affinage, s'est révélée efficace pour quantifier l'opinion véhiculée par les nouvelles financières, et son agrégation avec les cours a offert des aperçus visuels prometteurs sur l'interaction entre sentiment et prix.

Plusieurs perspectives d'amélioration peuvent être envisagées pour enrichir ce travail :

- **Enrichissement des Données :** Intégrer des sources de données alternatives (données macroéconomiques, indicateurs de sentiment issus des réseaux sociaux, données fondamentales plus granulaires) pour améliorer la robustesse des modèles.
- **Optimisation et Complexification des Modèles :** Poursuivre l'optimisation des hyperparamètres, explorer des architectures de deep learning plus avancées (e.g., Transformers pour la prédiction de séries temporelles, modèles d'attention pour le NLP) et des techniques de gestion des données déséquilibrées plus sophistiquées.
- **Stratégies d'Agrégation Quantitatives :** Développer des méthodes quantitatives pour agréger les signaux issus des différents modules (clustering, classification, prédiction, sentiment), par exemple via des systèmes de scoring pondérés ou un méta-modèle, afin de générer des recommandations d'investissement plus formalisées.
- **Backtesting Rigoureux et Validation Hors Échantillon :** Mettre en place un cadre de backtesting robuste, simulant des stratégies d'investissement basées sur les recommandations du pipeline sur des périodes historiques étendues et variées pour évaluer leur performance réelle.
- **Extension à la Gestion de Portefeuille :** Utiliser les signaux générés pour des tâches d'optimisation de portefeuille, de gestion des risques ou de construction de stratégies de trading algorithmique.

En somme, ce projet constitue une base solide, illustrant le potentiel de la science des données pour l'analyse financière. Les développements futurs puvent ce concentrer sur l'amélioration de la précision prédictive et la transformation des analyses en conseil d'investissement assez robustes.