Université Paris Dauphine - PSL

Pipeline d'agrégation de signaux boursiers

Groupe Boonekamp, Escudie, Barbier

18 mai 2025

https://github.com/Vbarbier1809/Impact$_s$*tocks*

Résumé

L'objectif de ce projet est de mettre en place un pipeline automatisé permettant d'agréger quotidiennement l'ensemble des signaux prédits par des modèles de clustering, classification, régression et d'analyse textuelle, afin de fournir des recommandations pour la prise de décision sur le marché des actions. Ce rapport détaille les motivations, les méthodes employées, les résultats obtenus et leur interprétation.

Table des matières

1 Introduction et motivations

Le marché boursier moderne est caractérisé par une grande volatilité et une quantité massive de données hétérogènes : cours historiques, actualités financières, indicateurs macroéconomiques et perceptions sentimentales capturées dans les médias. Pour réussir dans un tel environnement, il ne suffit pas d'analyser un seul type de signal : il faut combiner efficacement techniques de clustering pour identifier des groupes d'entreprises similaires, modèles de classification pour émettre des recommandations "buy/sell/hold", approches de régression pour prévoir les rendements à court terme et méthodes d'analyse de texte pour évaluer le sentiment des investisseurs. Ce projet vise à intégrer ces domaines de la data science, apprentissage non supervisé, supervisé, prédictif et traitement du langage naturel, au sein d'un pipeline automatisé, capable d'agréger quotidiennement l'ensemble des signaux afin de générer des conseils d'investissement robustes et adaptatifs. L'intérêt majeur de cette démarche réside dans la synergie entre ces méthodes et la mise à jour dynamique des recommandations, offrant aux acteurs financiers un outil pragmatique et complet pour optimiser leurs décisions de trading.

2 Travaux connexes

Plusieurs travaux ont exploré l'application des techniques de data science au domaine financier. En clustering, Fama et French ont introduit un modèle factoriel pour regrouper les actions selon leurs caractéristiques fondamentales, montrant l'intérêt de segmenter le marché pour expliquer les rendements [Fama and French, 1993]. Plus récemment, Ntakaris et al. ont comparé différentes méthodes non supervisées (K-means, spectral clustering) pour identifier dynamiquement des portefeuilles homogènes en fonction des tendances de marché [Ntakaris et al., 2018].

Dans la classification des signaux « buy/sell/hold », plusieurs études ont exploité des modèles de machine learning supervisé. Par exemple, Sirignano et Cont (2019) ont entraîné des réseaux de neurones profonds sur des séries temporelles financières pour générer des recommandations de trading automatiques, obtenant une performance supérieure aux approches classiques de Random Forest [**?**]. En régression, Fischer et Krauss (2018) ont démontré la supériorité des LSTM sur les modèles ARIMA pour la prédiction de prix à court terme [**?**].

Enfin, pour l'analyse de sentiment, des travaux tels que Huang et al. (2020) ont adapté des modèles pré-entraînés de type BERT au langage financier (FinBERT), permettant d'extraire finement les émotions véhiculées par les titres d'articles et leur impact sur les cours [**?**]. L'agrégation de ces signaux (clustering, classification, régression, sentiment) reste toutefois peu traitée dans la littérature, ce qui justifie l'approche intégrée de ce projet.

3 Clustering

Le clustering vise à segmenter l'univers d'entreprises étudiées selon différents critères pertinents (financiers, profil de risque, similitude de comportement de cours), afin de dégager des groupes homogènes qui faciliteront la construction de portefeuilles diversifiés et la génération de recommandations adaptées.

3.1 Objectifs

1. **Profil financier** Identifier des groupes d'entreprises aux caractéristiques de valorisation et de performance financière similaires.
2. **Profil de risque** Regrouper les entreprises selon leur exposition au risque financier et opérationnel.
3. **Corrélation des rendements quotidiens** Classer les sociétés en fonction de la similarité de l'évolution de leurs cours de clôture jour par jour.

3.2 Méthodologie

1. Clustering financier (K-Means) L'algorithme K-Means est adapté pour partitionner efficacement des données continues en clusters bien séparés, minimisant la somme des distances intra-clusters. Il offre une exécution rapide et un contrôle direct sur le nombre de groupes, essentiel pour comparer différents choix de segmentation.

- *Features retenues* : `columns_financial = [ 'forwardPE', 'beta', 'priceToBook', 'operatingMargins', 'returnOnEquity', 'profitMargins' ]`
- *Pourquoi ces indicateurs ?* :
  - **forwardPE** : mesure la valorisation anticipée par le marché, pertinente pour distinguer les entreprises de croissance.
  - **beta** : quantifie la sensibilité au marché global, reflétant la volatilité systématique.
  - **priceToBook** : compare le prix du marché à la valeur comptable, utile pour repérer les titres décotés ou surévalués.
  - **operatingMargins** et **profitMargins** : indiquent la capacité de l'entreprise à générer des bénéfices à différents niveaux d'activité.
  - **returnOnEquity** : évalue l'efficacité dans l'utilisation des capitaux propres pour créer de la valeur.
- *Pré-traitement* : suppression des valeurs manquantes et standardisation avec `StandardScaler` pour homogénéiser l'échelle des features.
- *Choix du nombre de clusters* : déterminé par la méthode du coude (inertie vs nombre de clusters) pour un compromis optimal entre compacité et simplicité.
- *Avantages* : rapidité, interprétabilité des centroïdes comme profils financiers types, robustesse sur features normalisées.

2. Clustering des profils de risque (Hiérarchique) Le clustering hiérarchique agglomératif (linkage de Ward) permet de découvrir une hiérarchie naturelle des entreprises basée sur leur risque, sans pré-définir un nombre fixe de groupes.

- *Features retenues* : `columns_risk = ['debtToEquity', 'beta', 'operatingMargins', 'profitMargins', 'returnOnAssets', 'trailingEps' ]`
- *Pourquoi ces indicateurs ?* :
  - **debtToEquity** : reflète l'endettement et le levier financier.
  - **returnOnAssets** : mesure la rentabilité par rapport aux actifs totaux.
  - **trailingEps** : montre la performance passée en termes de bénéfice par action.
  - **beta**, **profitMargins**, **operatingMargins** : apportent des informations complémentaires sur la volatilité et la santé opérationnelle.
- *Pré-traitement* : élimination des `NA`, standardisation des données.
- *Dendrogramme* : construction avec `scipy.cluster.hierarchy.linkage(..., method='ward')` pour visualiser la structure multi-niveaux.
- *Avantages* : pas besoin de spécifier K, description de la structure emboîtée, identification de sous-groupes et de liens de similarité.

3. Clustering par corrélation des rendements quotidiens (DBSCAN) DBSCAN est privilégié pour détecter des groupes de comportements similaires dans l'espace des corrélations, avec détection automatique des outliers.

- *Features calculées* : vecteurs issus de la matrice de corrélation des rendements quotidiens ($\text{corr}(R)$) pour chaque entreprise.
- *Pré-traitement* : imputation des valeurs manquantes par la moyenne, standardisation.
- *Paramètres clés* : choix de ε et `min_samples` pour contrôler la sensibilité et assurer une séparation solide entre clusters et outliers.
- *Avantages* : flexible pour la forme des clusters, robustesse au bruit et détection automatique des outliers.

3.3 Résultats et interprétation

3.3.1 3.3.1 Profil financier

Le choix de $k = 5$ clusters a été validé par la méthode du coude (cf. Figure 1). On observe un point d'inflexion – « coude » – à $k = 5$, ce qui justifie ce nombre de groupes.

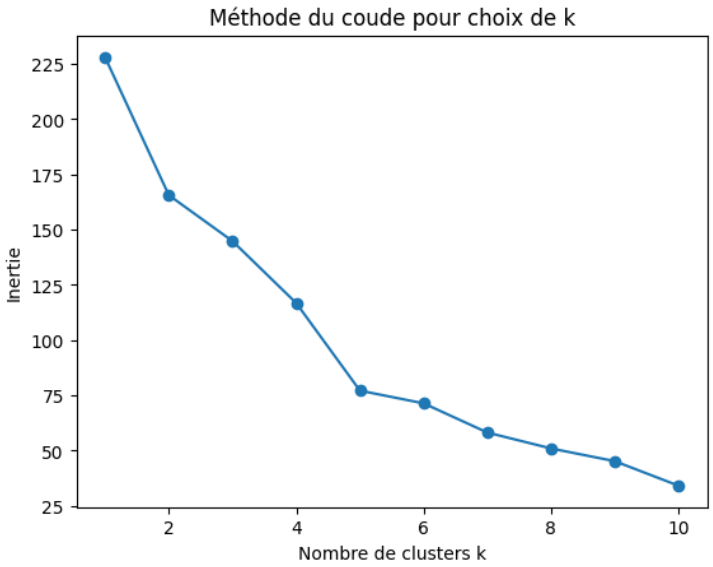

FIGURE 1 – Méthode du coude pour déterminer le nombre optimal de clusters financiers.

La projection t-SNE (Figure 2) confirme la séparation nette des 5 clusters. On y observe notamment que Microsoft, Alphabet (Google) et Meta (Facebook) partagent le même cluster, reflétant leur profil financier à forte valorisation et marges élevées.

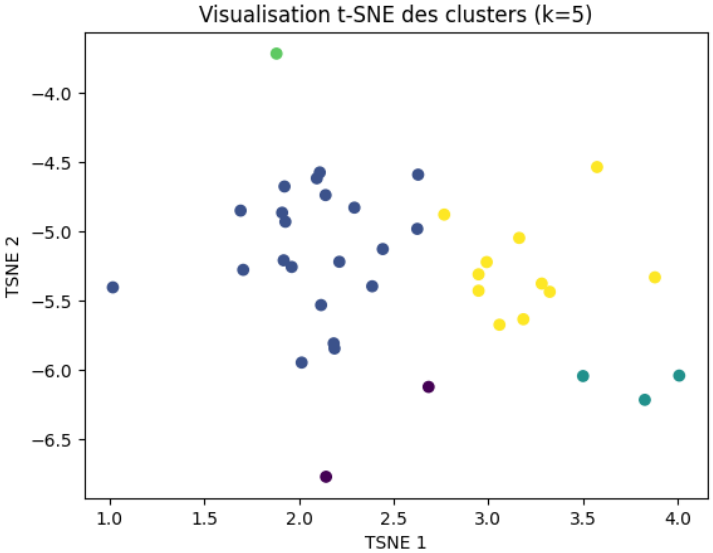

FIGURE 2 – Visualisation t-SNE des clusters financiers ($k = 5$).

Le Tableau 1 présente les centroïdes moyens de chaque cluster. Le Cluster 0, avec un *forwardPE* moyen de 58.3 et des marges supérieures à 14

Cluster	forwardPE	beta	priceToBook	operatingMargins	returnOnEquity	profitMargins
0	58.32	2.06	13.53	0.142	0.244	0.148
1	16.97	0.74	2.74	0.133	0.147	0.099
2	22.55	1.48	32.80	0.423	1.198	0.339
3	7.55	0.29	79.00	0.089	0.090	0.056
4	16.99	1.05	6.69	0.429	0.319	0.351

TABLE 1 – Caractéristiques moyennes des clusters financiers (*columns_financial*).

3.3.2 3.3.2 Profil de risque

Le dendrogramme présenté en Figure 3, construit avec la méthode de linkage de Ward, met en évidence trois groupes distincts lorsque l'on coupe l'arbre à une distance adaptée.

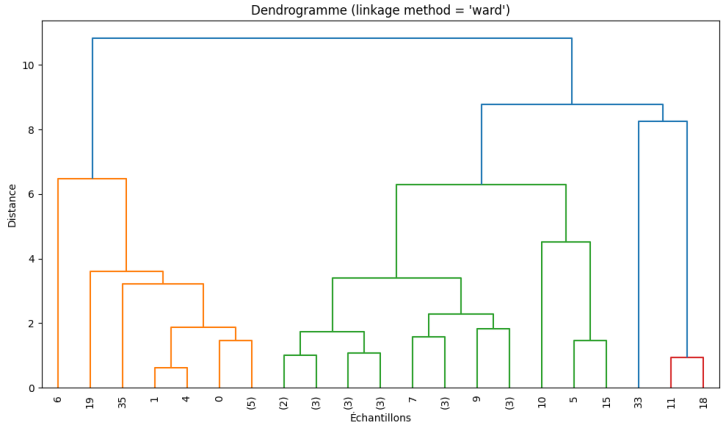

FIGURE 3 – Dendrogramme hiérarchique des profils de risque (linkage = Ward).

Le Tableau 2 récapitule les centres moyens (centroïdes) de chaque cluster sur les indicateurs de risque. On note par exemple un endettement très élevé (debtToEquity moyen de 460) dans le Cluster 0, contre moins de 45 dans le Cluster 1.

Cluster	debtToEquity	beta	operatingMargins	profitMargins	returnOnAssets	trailingEps
0	460.34	1.05	0.259	0.205	0.0306	208.33
1	44.97	1.23	0.400	0.329	0.2046	23.78
2	55.87	0.84	0.12	0.089	0.04	6.5

TABLE 2 – Caractéristiques moyennes des clusters de risque (*columns_risk*).

Les entreprises se répartissent ainsi :

- **Cluster 0 (risque élevé)** : Oracle, Goldman Sachs, SoftBank.
- **Cluster 1 (risque modéré)** : Apple, Microsoft, Alphabet, Meta, NVIDIA, Adobe, Netflix, Qualcomm, Visa, ASML, Tata Consultancy Services.
- **Cluster 2 (risque plus faible)** : Amazon, Tesla, Tencent, Alibaba, IBM, Intel, Sony, AMD, Cisco, Johnson Johnson, Pfizer, ExxonMobil, SAP, Siemens, LVMH, TotalEnergies, Shell, Baidu, JD.com, BYD, Toyota, Hyundai.

3.3.3 3.3.3 Corrélation des rendements quotidiens

Pour choisir le paramètre ε de DBSCAN, nous avons tracé la distance au 4^{e} plus proche voisin (Figure 4). Le « coude » se situe autour de $\varepsilon \approx 1.0$.

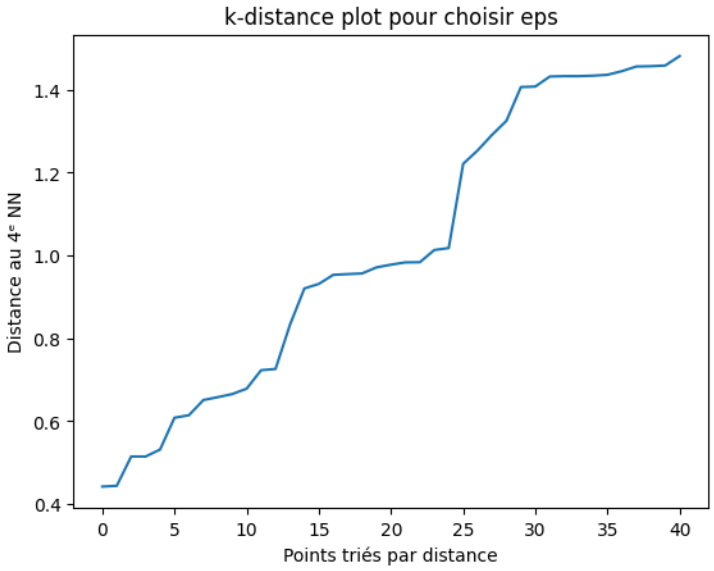

FIGURE 4 – k-distance plot ($k = 4$) pour déterminer ε de DBSCAN.

Appliqué aux vecteurs de corrélation des rendements quotidiens, DBSCAN génère majoritairement un unique cluster global et de nombreux outliers (Figure 5). Malgré de nombreux essais en ajustant ε et `min_samples`, aucune structure de sous-groupes dense et cohérente n'est apparue : le profil de corrélation est trop homogène parmi les entreprises considérées.

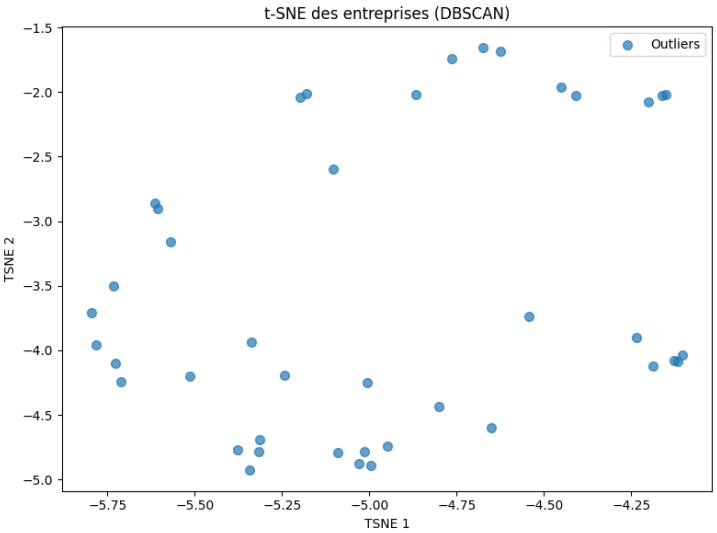

FIGURE 5 – Projection t-SNE des résultats DBSCAN sur corrélations quotidiennes.

Un dendrogramme construit sur la même matrice de distances corrélées (Figure 6) révèle également un enchevêtrement progressif des séries, sans coupe naturelle claire pour former plusieurs clusters.

Interprétation La faible hétérogénéité des corrélations journalières, due aux mouvements globaux synchronisés du marché, empêche l'identification de groupes de comportement vraiment distincts. Par conséquent, nous n'avons pas retenu de partitionnement sur ce critère dans le pipeline final.

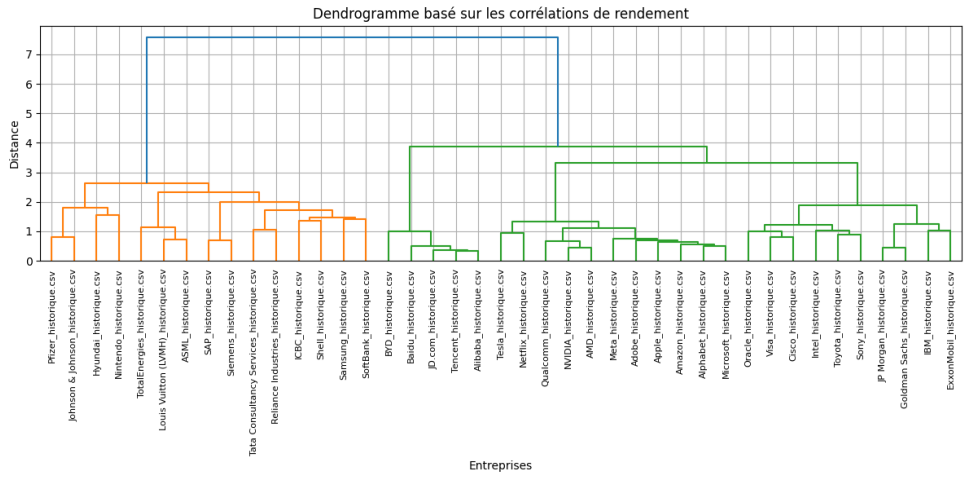

FIGURE 6 – Dendrogramme des séries de rendements basé sur les corrélations (linkage Ward).

3.3.4 3.3.4 Silhouette Score

Le *Silhouette Score* est une métrique d'évaluation de la qualité d'un clustering qui combine deux notions :

— **Cohésion** (distance moyenne aux autres points du même cluster) ;
— **Séparation** (distance moyenne au cluster le plus proche).

Pour chaque point, le score vaut

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

où a est la cohésion et b la séparation. La moyenne de s sur tous les points se situe entre -1 et $+1$: un score proche de $+1$ signifie des clusters bien séparés et compacts, un score négatif indique probablement un mauvais cluster.

Scores obtenus :

— **KMeans (k=5)** : 0.272
— **Hiérarchique (k=3)** : 0.197
— **DBSCAN** ($\varepsilon = 1.0$, `min_samples`=**4**) : `nan`

On constate que KMeans offre le meilleur compromis cohésion/séparation, tandis que DBSCAN, faute de former des noyaux denses suffisants, ne produit pas de score défini.

4 Classification (“buy”, “sell”, “hold”)

4.1 Objectif

Utiliser des algorithmes de classification supervisée pour prédire, à partir des indicateurs techniques calculés sur les prix de clôture des 41 entreprises, une recommandation d'investissement ("*buy*" / "*hold*" / "*sell*"). Les features retenues sont :

— SMA sur 20 jours (`'SMA 20'`)
— EMA sur 20 jours (`'EMA 20'`)
— RSI 14 jours (`'RSI 14'`)
— MACD (`'MACD'`) et sa ligne de signal (`'MACD Signal'`)
— Bandes de Bollinger haute et basse (`'Bollinger High'`, `'Bollinger Low'`)
— Volatilité glissante sur 20 jours (`'Rolling Volatility 20'`)
— Taux de variation à 10 jours (`'ROC 10'`)

L'objectif est de comparer cinq modèles (XGBoost, Random Forest, K-Nearest Neighbors, Régression Logistique, Support Vector Machine) afin d'identifier celui offrant le meilleur compromis entre précision prédictive, robustesse et temps de calcul.

4.2 Méthodes

Après prétraitement (normalisation et étiquetage "buy" / "hold" / "sell"), chaque modèle a fait l'objet d'un `GridSearchCV` pour ajuster ses hyperparamètres.

4.2.1 4.2.1 XGBoost

XGBoost est un algorithme de gradient boosting optimisé, rapide et capable de gérer les fortes corrélations entre variables.

```
param_grid = {
    'n_estimators': [50, 100],
    'max_depth': [3, 5, 7],
    'learning_rate': [0.01, 0.1],
    'subsample': [0.8, 1.0],
}
```

La métrique `eval_metric="logloss"` a été retenue car elle pénalise fortement les prédictions surestimées et améliore la calibration des probabilités, essentielle pour un système de conseil financier.

4.2.2 4.2.2 Random Forest

Random Forest agrège plusieurs arbres de décision pour réduire le surapprentissage et la variance.

```
param_grid = {
    'n_estimators': [50, 100],
    'max_depth': [None, 5, 10],
    'min_samples_split': [2, 5],
    'min_samples_leaf': [1, 2]
}
```

Le scoring utilisé est l'`accuracy`, mesurant la proportion de prédictions exactes.

4.2.3 4.2.3 K-Nearest Neighbors (KNN)

KNN classe chaque point selon la majorité de ses k voisins les plus proches.

```
param_grid = {
    'n_neighbors': [3, 5, 7],       % nombre de voisins
    'weights': ['uniform', 'distance'], % pondération des votes
    'metric': ['euclidean', 'manhattan'] % distance utilisée
}
```

Ces paramètres influencent la forme des frontières et la sensibilité au bruit.

4.2.4 4.2.4 Régression Logistique

La régression logistique est un modèle linéaire probabiliste qui estime, via la fonction sigmoïde, la probabilité d'appartenance à chaque classe. Elle sert de baseline rapide et interprétable, sans hyperparamètres majeurs à optimiser.

4.2.5 4.2.5 Support Vector Machine (SVM)

Les SVM trouvent un hyperplan maximisant la marge entre classes.

```
param_grid = {
    'C': [0.1, 1.0, 10],
    'kernel': ['linear', 'rbf'],
    'gamma': ['scale', 'auto']
}
```

Le paramètre C gère la tolérance aux erreurs, le kernel et γ déterminent la complexité de la frontière de décision.

4.3 4.3 Résultats

Modèle	CV Score
XGBoost	0.999991
Random Forest	0.999934
K-Nearest Neighbors	0.97
Régression Logistique	0.998
SVM	0.998

TABLE 3 – Performance en validation croisée des différents classificateurs.

Le modèle retenu est **XGBoost**, dont les meilleurs hyperparamètres trouvés sont :

$$\{\texttt{learning\_rate} = 0.1,\ \texttt{max\_depth} = 5,\ \texttt{n\_estimators} = 100,\ \texttt{subsample} = 0.8\}.$$

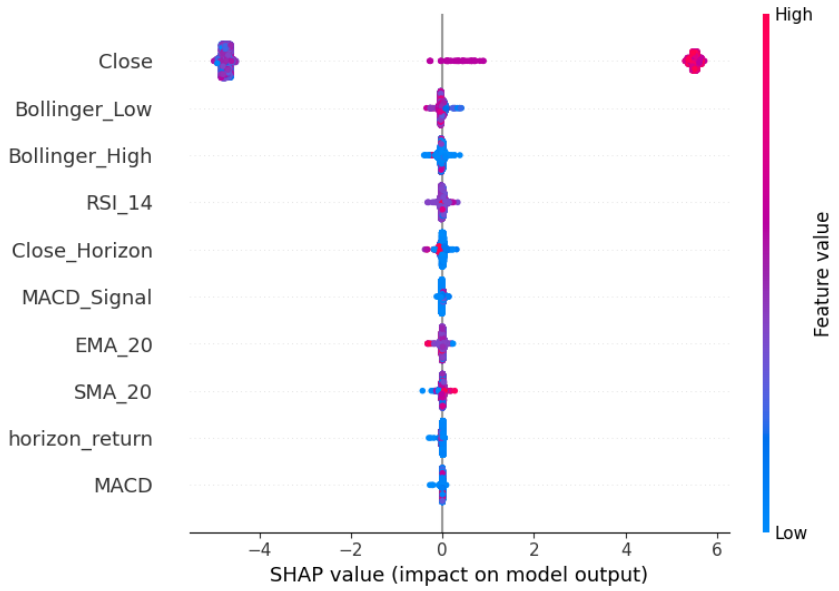

FIGURE 7 – Diagramme summary SHAP pour XGBoost.

Analyse des valeurs SHAP La Figure 7 met en évidence la contribution de chaque feature à la prédiction :

— **Close** : les valeurs extrêmes influencent le plus fortement le passage en « sell » (SHAP positif) ou « buy » (SHAP négatif).

— **Bollinger High / Bollinger Low** : un large écart de bande (volatilité accrue) tend à générer un signal de vente.
— **RSI_14** : les niveaux de surachat (>70) et de survente (<30) sont corrélés respectivement aux conseils « sell » et « buy ».
— **SMA_20 / EMA_20** : reflètent la tendance moyenne, favorisant plutôt les décisions « hold » lorsque la courbe est stable.
— **MACD / MACD Signal** : les croisements de MACD au-dessus ou au-dessous de sa ligne de signal annoncent respectivement des retournements haussiers (« buy ») ou baissiers (« sell »).
— **ROC_10** et **Rolling Volatility 20** : moins influents, mais utiles pour affiner les signaux autour de la zone « hold ».

5 Prédiction de rendement à J+1

5.1 Objectif

Utiliser des algorithmes de régression pour prédire la valeur de clôture des actions à l'horizon. Ce problème de séries temporelles est abordé à la fois avec des modèles classiques de machine learning (Régression linéaire, Random Forest Regressor, KNN et XGBoost) et des approches de deep learning (LSTM, MLP et RNN), afin de comparer performance prédictive, capacité à capturer la dynamique séquentielle et robustesse aux variations de marché.

5.2 Méthodes

Nous avons testé plusieurs modèles de régression supervisée, dont la régression linéaire et XGBoost Regressor. Les meilleures performances ont été obtenues avec XGBoost (voir section 5.3). La régression linéaire, malgré sa capacité à parfaitement calquer la tendance des cours – du fait de son hypothèse de relation linéaire directe et de son absence de régularisation sur la complexité non linéaire – introduit un biais important : elle est trop lissée et ne capture pas les fluctuations à court terme, ce qui peut donner des prédictions systématiquement décalées. Pour tirer parti de la structure temporelle des données, nous avons implémenté un réseau de neurones récurrent de type LSTM (Long Short-Term Memory). Ce modèle intègre trois portes clés :
— *Porte Forget* : décide quelles informations passées doivent être oubliées.
— *Porte Input* : régule quelles nouvelles informations sont ajoutées à la mémoire.
— *Porte Output* : contrôle la sortie générée par l'état de la mémoire.
Cette architecture permet de conserver l'information pertinente sur de longues séquences, d'atténuer le problème de disparition du gradient et de modéliser efficacement les dépendances temporelles. Le LSTM a ainsi surpassé les modèles classiques en capturant à la fois la tendance générale et les variations soudaines des cours.

5.3 5.3 Résultats de prédiction

Nous illustrons ci-dessous les performances obtenues pour l'action Apple.

5.3.1 5.3.1 Modèles Machine Learning

Analyse La régression linéaire, malgré son faible MSE, est trop lissée et ne suit pas fidèlement les fluctuations rapides du cours, ce qui introduit un biais. Les modèles basés sur les arbres (Random Forest, XGBoost) capturent mieux la non-linéarité et offrent des performances similaires, XGBoost étant légèrement supérieur.

Modèle	MSE	RMSE	Meilleurs paramètres
Régression linéaire	11.6872	3.4187	{}
Random Forest	1071.7249	32.7372	{'max_depth' :5, 'n_estimators' :100}
K-Nearest Neighbors	1427.1011	37.7770	{'n_neighbors' :10}
XGBoost	1069.7872	32.7076	{'learning_rate' :0.1, 'max_depth' :3, 'n_estimators' :100}

TABLE 4 – Performances des modèles ML sur l'action Apple.

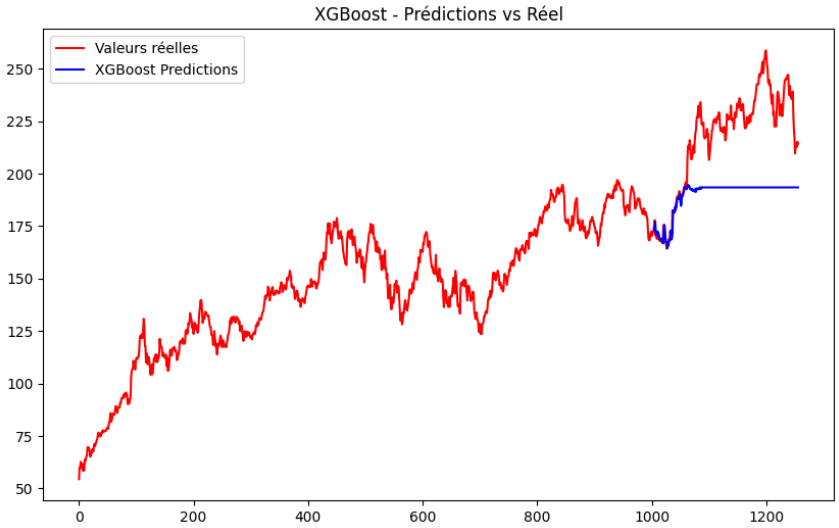

FIGURE 8 – Valeurs réelles vs. prédictions XGBoost (Apple).

5.3.2 5.3.2 Modèles Deep Learning

Les métriques utilisées sont :

— **MAE (Mean Absolute Error)** : erreur absolue moyenne, interprétable en unités de prix.

— **RMSE (Root Mean Squared Error)** : racine de l'erreur quadratique moyenne, plus sensible aux grandes erreurs.

Modèle	MAE	RMSE
LSTM	3.3632	4.4114
MLP	3.4489	4.3775
RNN	3.3209	4.2965

TABLE 5 – Performances des réseaux neuronaux sur Apple.

Analyse Le LSTM, grâce à sa mémoire à long terme, capte à la fois la tendance et les variations soudaines, ce qui se reflète dans un MAE/RMSE légèrement meilleur que pour les MLP ou RNN classiques.

6 Analyse de sentiments sur les news financières

6.1 Objectif

Le but premier de cette étape est de collecter automatiquement les titres et descriptions d'articles financiers relatifs à chacune des entreprises étudiées sur les dix derniers jours. En

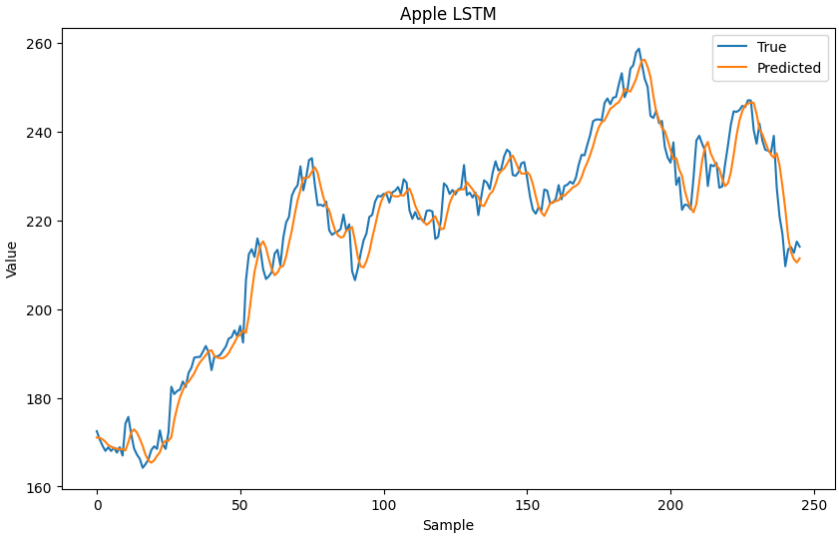

FIGURE 9 – Prédictions LSTM vs réel (Apple).

interrogeant l'API NewsAPI (sources : Financial Post, The Wall Street Journal, Bloomberg, The Washington Post) pour chaque nom de société, nous extrayons tous les articles pertinents en anglais, puis les regroupons par date de publication. Cette collecte régulière et structurée permet d'alimenter notre pipeline de sentiment analysis et de mesurer l'impact immédiat des événements médiatiques sur la dynamique des cours.

6.2 Méthodes

Le processus de traitement se décompose en deux volets complémentaires :

1. **Extraction et parsing** :
   — Utilisation de la bibliothèque `requests` pour appeler l'endpoint `/v2/everything` de NewsAPI dans une fenêtre glissante de 10 jours.
   — Filtrage des réponses JSON pour ne conserver que les articles mentionnant explicitement le nom de la société dans le titre ou la description.
   — Agrégation dans un dictionnaire `news_by_date[YYYY-MM-DD]` et export en JSON (fichier `news_data2.json`).
2. **Fine-tuning de BERT** :
   — Chargement d'un modèle BERT pré-entraîné (textttbert-base-uncased ou texttFinBERT).
   — Tokenisation des titres et descriptions, création de jeux de données PyTorch.
   — Configuration de `TrainingArguments` (incluant `evaluation_strategy="epoch"`, taux d'apprentissage, taille de batch).
   — Entraînement supervisé sur un corpus financier annoté (Twitter Financial News Sentiment, Financial Phrase Bank) pour classer chaque article en *LABEL_0* (négatif), *LABEL_1* (neutre) ou *LABEL_2* (positif).

6.3 Résultats et interprétation

L'exécution du script principal génère un DataFrame `df_results` où chaque ligne contient :
— Un `timestamp` d'article,
— Le `text` (titre + description),
— Le `sentiment` prédit (*LABEL_0*/*LABEL_1*/*LABEL_2*),

— L'`aligned_timestamp`, calé sur l'heure de marché la plus proche.

Par exemple, plusieurs articles sur Tesla du 30/04 ont été classés *LABEL_2* (positif) ou *LABEL_0* (négatif) selon leur contenu. Chaque entreprise dispose en moyenne de 20 à 80 articles alignés par date.

Ces résultats, sauvegardés dans `news_data2.json`, constituent la base de données pour corréler les chocs médiatiques aux variations horodatées des cours dans la phase d'agrégation du pipeline.

Timestamp	**text**	**Sentiment**
2025-04-26 14 :00 :00-04 :00	Tesla Hikes Canadian Prices and Pushes Its...	2 (POSITIVE)
2025-05-01 08 :00 :00-04 :00	Tesla Chair Denies Reported CEO Search and Backs...	2 (POSITIVE)
2025-05-01 04 :00 :00-04 :00	Tesla sales plunge 59 in April...	0 (NEGATIVE)
2025-04-25 15 :00 :00-04 :00	Tech giants push U.S. stocks higher with Tesla...	1 (NEUTRAL)

Table 6 – Analyse de sentiments des texts scrappés

7 Conclusion

Références

[Fama and French, 1993] Fama, E. F. and French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33 :3–56.

[Ntakaris et al., 2018] Ntakaris, A., Louppe, G., Georges, M., and Gramfort, A. (2018). Stock market clustering with spectral methods. In *International Conference on Pattern Recognition*.