DOCUMENT DE SPECIFICATION

Équipe n°236 : Détection d'outliers dans des données de marché

GRANGIER Tom

MOUTON Cyprien

STAHL Benjamin

LIN Johnny

HAUG-THOMAS Victor-Emmanuel

LI Chenjie

Sommaire

1. Contexte :

Pour ce projet réalisé dans le cadre du PI², nous nous attelons à exploiter les possibilités offertes par des algorithmes d'intelligence artificielle pour repérer des données aberrantes au sein des historiques de données financières, en particulier les prix et les volatilités. Notre objectif premier est de choisir judicieusement des algorithmes, utilisant le langage Python pour leur mise en œuvre. Nous prévoyons d'acquérir nos données financières à partir de Yahoo Finance, une source fiable et accessible pour les marchés financiers.

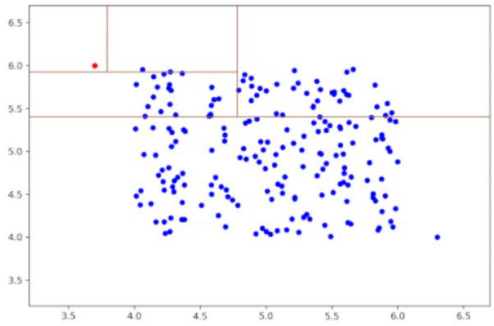
Pour rendre notre démarche pratique, nous effectuerons des tests manuels en insérant délibérément des données erronées. Cela nous permettra d'évaluer la capacité de nos modèles à gérer et à identifier des anomalies, même lorsqu'elles sont introduites artificiellement. Les résultats de nos détections d'anomalies seront ensuite présentés grâce à une interface graphique, facilitant ainsi la compréhension et l'utilisation des informations obtenues.

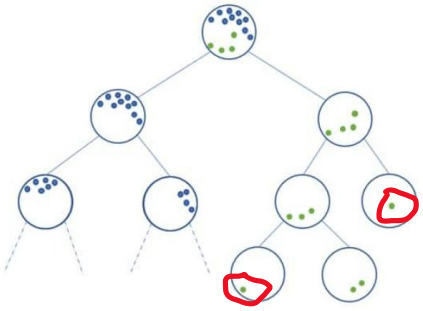
Nous allons donc implémenter plusieurs algorithmes et méthodes différentes afin de trouver lequel peut être le plus pertinent.

2. Approches algorithmiques (IA)

2.1 Détection de valeurs aberrantes avec l'isolation Forest :

L'isolation forest est un algorithme de classification non supervisée. L'algorithme marche de la façon suivante, on procède à un partitionnement aléatoire du jeu de données, ce qui génère deux sous-ensembles de données distincts. Ces étapes sont réitérées jusqu'à l'isolation d'une donnée spécifique. Ce processus est reproduit de façon récursive ce qui crée un arbre avec une valeur aberrante dans les dernières branches. Ce processus peut être reproduit un grand nombre de fois pour avoir plusieurs arbres ainsi les valeurs sont aberrantes sont ciblés plus efficacement.





Une fois la forêt construite, le score d'anomalie pour une observation particulière est déterminé en mesurant la profondeur moyenne de cette observation à travers tous les arbres de la forêt. Les anomalies auront tendance à être isolées plus rapidement et auront donc une profondeur moyenne plus faible.

Paramètres importants :

- Nombre d'arbres : Il est souvent recommandé d'utiliser plusieurs arbres pour construire une forêt robuste, et la performance de l'algorithme peut augmenter avec le nombre d'arbres.

- Profondeur maximale des arbres : Il est important de faire attention à la profondeur maximale des arbres pour éviter le surajustement. Des arbres trop profonds peuvent conduire à une sensibilité excessive aux anomalies.

Avantages :

- Résistance aux valeurs aberrantes : Contrairement à d'autres méthodes, l'Isolation Forest est moins sensible aux valeurs aberrantes dans le jeu de données. Les valeurs aberrantes ont tendance à être isolées plus rapidement lors de la construction des arbres.

- Algorithme d'apprentissage non supervisé : L'apprentissage non supervisé permet de nous dispenser de l'étiquetage manuel des données.

- Efficacité avec des grandes dimensions : L'Isolation Forest fonctionne bien même avec un grand nombre de dimensions, ce qui est souvent un défi pour d'autres méthodes de détection d'anomalies.

Inconvénients :

- Sensibilité à la taille des données : L'efficacité de l'Isolation Forest peut diminuer avec des ensembles de données très petits. Elle peut également ne pas fonctionner aussi bien lorsque la proportion d'anomalies est extrêmement faible.

- Non adapté aux données hautement corrélées : Si les données sont fortement corrélées, l'Isolation Forest peut ne pas être aussi efficace, car elle se base sur l'idée que les anomalies sont plus faciles à isoler.

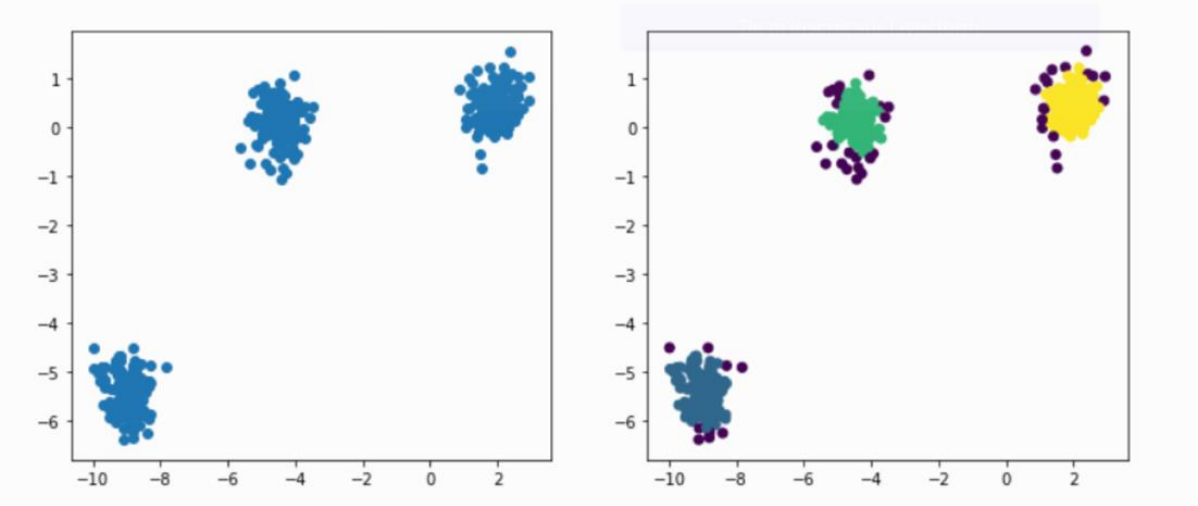
- Interprétabilité limitée : Les résultats de l'Isolation Forest peuvent parfois être difficiles à interpréter. Comprendre pourquoi une observation particulière est considérée comme une anomalie peut être complexe, en particulier avec un grand nombre de dimensions.

Complexité :

la complexité de l'isolation forest est en O(n*h), où "n" est le nombre d'échantillons dans l'ensemble de données et "h" est la profondeur moyenne de l'arbre.

2.2 Détection de valeurs aberrantes par l'algorithme DBSCAN :

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) est un algorithme de clustering qui fonctionne en identifiant des regroupements de points dans l'espace de données basés sur leur densité. Les points suffisamment proches les uns des autres sont regroupés dans un même cluster, tandis que ceux qui ne sont pas dans des zones denses sont considérés comme des anomalies ou des données aberrantes (bruit). Dans le contexte de la détection des données aberrantes et de la filtration des données, DBSCAN est particulièrement utile car il permet d'isoler ces anomalies de manière efficace, sans nécessiter la spécification préalable du nombre de clusters, ce qui facilite l'analyse et la purification des ensembles de données complexes ou bruités.



Son fonctionnement ?

- Pour chaque observation on regarde le nombre de points à au plus une distance ε de celle-ci. On appelle cette zone le ε-voisinage de l'observation.
- Si une observation compte au moins un certain nombre de voisins (min_samples) y compris elle-même, elle est considérée comme une observation cœur. On a alors décelé une observation à haute densité.
- Toutes les observations au voisinage d'une observation cœur appartiennent au même cluster. Il peut y avoir des observations cœur proche les unes des autres. Par conséquent de proche en proche on obtient une longue séquence d'observations cœur qui constitue un unique cluster.
- Toute observation qui n'est pas une observation cœur et qui ne comporte pas d'observation cœur dans son voisinage est considérée comme une anomalie.

La distance choisie dépendra du contexte de l'utilisation de l'algorithme, les plus utilisées étant :

- Distance euclidienne ;
- Distance de Minkowski ;
- Distance de Manhattan ;
- Distance de Tchebychev ;
- Distance de Canberra.

Avantages :

- **Pas de Pré-spécification du Nombre de Clusters :** DBSCAN ne nécessite pas que l'utilisateur définisse le nombre de clusters à l'avance, contrairement à d'autres méthodes comme K-means.
- **Détection Efficace des Anomalies :** Il identifie naturellement les données aberrantes comme des points qui ne font pas partie des clusters principaux.
- **Capacité à Gérer Différentes Formes de Clusters :** DBSCAN peut découvrir des clusters de formes arbitraires, pas seulement sphériques.
- **Robustesse au Bruit :** L'algorithme est conçu pour gérer efficacement le bruit et les points aberrants dans les données.
- **Adaptabilité :** Peut être utilisé avec différents types de données et dans divers contextes d'analyse.

Inconvénients :

- **Sensibilité aux Paramètres :** Le choix des paramètres **ε (distance de voisinage) et min_samples** (nombre minimum de points pour former un cluster) est crucial et peut être difficile à déterminer.
- **Performance Diminuée dans les Espaces de Grande Dimension :** DBSCAN peut ne pas bien fonctionner avec des données de haute dimensionnalité, un phénomène connu sous le nom de malédiction de la dimensionnalité.
- **Difficultés avec les Densités de Clusters Variées :** L'algorithme peut avoir du mal à identifier correctement les clusters si les densités de points varient considérablement au sein de l'ensemble de données.
- **Complexité Computationnelle :** Peut-être gourmand en termes de ressources de calcul, en particulier pour de très grands ensembles de données.

Complexité:

- **Avec Indexation Spatiale :** Si l'algorithme utilise une structure de données d'indexation spatiale (comme un R-tree ou un k-d tree) pour optimiser la recherche des voisins, la complexité est typiquement de l'ordre de O(n*log(n)). Cette structure permet une recherche rapide des points voisins, ce qui est crucial pour la performance de l'algorithme.
- **Sans Indexation Spatiale :** Si l'algorithme n'utilise pas de structure d'indexation spatiale, la complexité peut s'élever à O(n^2). Dans ce cas, pour chaque point de données, l'algorithme doit parcourir l'ensemble des points pour identifier ses voisins, ce qui est beaucoup plus coûteux en termes de temps de calcul.

Mémoire :

- **Avec Indexation Spatiale :** L'utilisation de structures d'indexation spatiale augmente la consommation de mémoire, car ces structures doivent être stockées en plus des données elles-mêmes.
- **Sans Indexation Spatiale :** Même sans indexation spatiale, DBSCAN doit souvent maintenir des structures de données supplémentaires, comme des matrices de distance ou des listes de voisins, ce qui peut également être coûteux en mémoire, surtout pour de grands ensembles de données.

2.3 Détection de valeurs aberrantes avec l'algorithme des KNN :

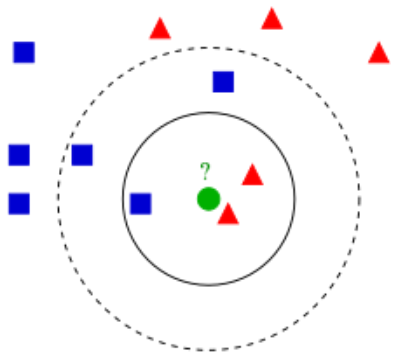
L'algorithme des k plus proches voisins est une méthode de détection d'anomalies basée sur la densité locale. Il est largement utilisé dans le domaine de l'apprentissage automatique et de la détection d'anomalies.

Le fonctionnement de l'algorithme KNN repose sur l'idée que les anomalies sont souvent entourées de points de données normaux. L'algorithme se divise en trois étapes principales :

1) Calcul de la distance : Tout d'abord, pour chaque point de données, la distance entre ce point et tous les autres points du jeu de données est calculée. La distance la plus couramment utilisée est la distance euclidienne, mais d'autres mesures de distance peuvent également être employées.

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$$

2) Sélection des k plus proches voisins : Ensuite, pour chaque point de données, les k points les plus proches (les plus similaires) sont sélectionnés en fonction de la distance calculée précédemment. La valeur de k est un paramètre choisi par l'utilisateur et détermine la sensibilité de l'algorithme à la détection d'anomalies.



Comment classer le point vert ? Si k = 3 (cercle en ligne pleine) il est classé comme un triangle car il y a deux triangles et seulement un carré dans le cercle considéré. Si k = 5 (cercle en ligne pointillée) il est classé comme un carré (3 carrés face à deux triangles dans le cercle externe).

3) Évaluation et classification : Enfin, pour chaque point de données, on évalue les k plus proches voisins et on détermine si le point en question est une anomalie ou non.

Pour la détection de valeurs aberrantes en finance :

Une fois les distances sont calculées pour chaque point, on estime la densité locale autour de chaque point en utilisant la distance moyenne des k-NN les plus proches.

Pour chaque point, on peut lui attribuer un score d'anomalie en comparant sa densité locale à celle des autres points. Les points ayant une densité locale significativement plus faible que la moyenne pourront ainsi être considérés comme des valeurs aberrantes avec un score d'anomalie plus élevé.

On peut également définir un seuil pour déterminer quels points sont considérés comme des valeurs aberrantes en fonction des scores d'anomalie. Les points ayant un score supérieur au seuil sont identifiés comme des valeurs aberrantes.

La valeur k dans l'algorithme k-NN définit le nombre de voisins qui seront vérifiés lors du calcul de la densité locale d'un point de donnée (le nombre de voisins qui seront utilisés pour estimer la densité autour du point donné).

Le choix de la valeur de k est important car il influence la sensibilité de l'algorithme à la détection des valeurs aberrantes.

1. Trop petit k : Si k est trop petit, par exemple égal à 1, cela signifie que seul le point le plus proche est pris en compte pour estimer la densité locale. Dans ce cas, l'algorithme peut être sensible au bruit et considérer des points isolés comme des valeurs aberrantes, même s'ils ne le sont pas réellement.
2. Trop grand k : Si k est trop grand, par exemple égal à la taille totale de l'échantillon de données, cela signifie que la densité locale est estimée sur une région plus large de l'espace des caractéristiques. Dans ce cas, il peut être plus difficile de détecter des valeurs aberrantes qui se trouvent dans des régions moins densément peuplées.
3. Choix basé sur le contexte : La valeur de k doit être choisie en fonction du contexte spécifique de nos données, du nombre d'échantillons disponibles et de la densité attendue des points. Par exemple, si on s'attend à ce que les valeurs aberrantes soient rares et isolées, une valeur de k plus petite peut être plus appropriée. Si on s'attend à ce que les valeurs aberrantes soient plus fréquentes ou qu'elles se produisent dans des régions densément peuplées, une valeur de k plus grande peut être plus adaptée.

Avantages :

-Facile à implémenter et simple à comprendre

-Il s'adapte facilement : à mesure que de nouveaux échantillons d'apprentissage sont ajoutés, l'algorithme s'ajuste pour tenir compte de toute nouvelle donnée, toutes les données d'apprentissage étant stockées en mémoire.

-Peu d'hyperparamètres : KNN ne nécessite qu'une valeur k et une mesure de distance, ce qui est peu par rapport aux autres algorithmes d'apprentissage automatique.

Inconvénients :

Mise à l'échelle difficile : puisque KNN est un algorithme paresseux, il utilise plus de mémoire et de stockage de données par rapport aux autres discriminants. Cela peut être coûteux en termes de temps et d'argent. Davantage de mémoire et de stockage augmenteront les dépenses de l'entreprise et plus de données peuvent prendre plus de temps à calculer.

Sensibilité au choix des paramètres : La performance de l'algorithme KNN peut être influencée par le choix des paramètres, en particulier la valeur de k, qui correspond au nombre de voisins à considérer. Un mauvais choix de k peut entraîner une détection inexacte des anomalies.

Difficulté à gérer les déséquilibres de classe : Lorsque les anomalies sont rares par rapport aux données normales, l'algorithme KNN peut avoir du mal à détecter efficacement ces anomalies, car les voisins proches seront principalement des points de données normaux.

2.4 Détection de valeurs aberrantes en utilisant des Réseaux de Neurones Récurrents (RNN)

Les Réseaux de Neurones Récurrents (RNN) sont une classe de réseaux de neurones artificiels spécialement conçus pour traiter des séquences de données. Contrairement aux réseaux de neurones traditionnels qui traitent chaque entrée indépendamment, les RNN possèdent des boucles internes leur permettant de conserver une forme de "mémoire" des entrées précédentes. Cette caractéristique les rend idéalement adaptés à l'analyse de séries temporelles, comme celles que l'on rencontre dans les données de marché financier.

Structure et Fonctionnement des RNN

Les RNN sont composés de couches de neurones où les connexions entre les neurones forment une boucle, permettant l'information de circuler d'un pas de temps à l'autre.

- Calcul de l'État Caché

L'état caché dans un Réseau de Neurones Récurrent est un élément fondamental qui permet à ces réseaux de capturer et de traiter des informations séquentielles ou temporelles

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$

- h_t : État caché à l'instant t
- x_t : Entrée à l'instant t
- W_{xh}, W_{hh} : Matrices des poids
- b_h : Vecteur de biais
- f : Fonction d'activation (tanh ou ReLU)

Cette formule calcule l'état caché actuel comme une fonction de l'entrée actuelle et de l'état caché précédent. Elle permet au RNN de "se souvenir" des informations passées et de les utiliser pour influencer les calculs actuels, ce qui est crucial pour la modélisation de séries temporelles comme les données financières.

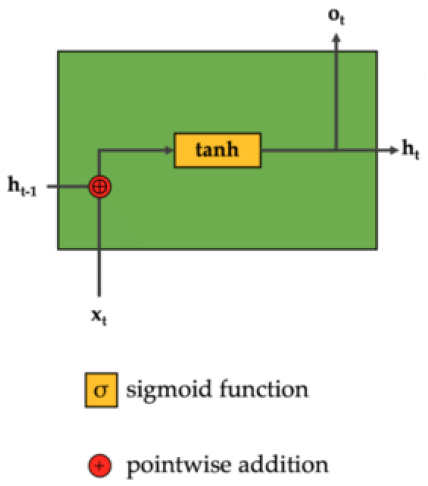
- Formule de sortie du RNN

$$y_t = W_{hy}h_t + b_y$$

- y_t : Sortie à l'instant t
- W_{hy} : Matrice de poids de l'état caché à la sortie
- b_y : vecteur de biais de la sortie

Cette formule transforme l'état caché en une sortie qui peut être utilisée pour des prédictions ou des classifications. Dans le contexte des données financières, cette sortie pourrait représenter une prédiction du prix futur d'une action, basée sur les informations précédentes capturées par l'état caché.

- Schéma classique d’un RNN



Les entrées x_t et h_{t-1} sont pondérées par leurs matrices de poids respectives et sommées ensemble. La somme pondérée est ensuite passée à travers une fonction d'activation. Ici, c'est la fonction tanh qui est utilisée, qui est courante dans les couches cachées des RNN pour ajouter de la non-linéarité. La tanh est une fonction d'activation non-linéaire qui produit une sortie dans l'intervalle [-1, 1]. Elle aide à gérer les problèmes de gradient et à permettre au réseau de traiter des séquences de données avec des dépendances temporelles.

On a ensuite un nouvel état caché généré (x_t) à l'instant t après l'application de la fonction d'activation. Cet état caché peut être utilisé pour la prochaine étape de temps ou comme partie de la sortie du réseau.

Et puis o_t la sortie à l'instant t qui est calculée à partir de l'état caché

Introduction aux LSTM (variante de RNN)

Les LSTM sont capables de capturer des dépendances à long terme dans les séries temporelles. Cela les rend particulièrement adaptés pour des tâches comme la détection d'anomalies dans les données financières, où il est crucial de comprendre les séquences de données sur de longues périodes.

Structure et Fonctionnement des LSTM

Les LSTM ajoutent une structure complexe de "portes" à l'architecture RNN de base pour contrôler le flux d'informations. Chaque cellule LSTM a trois de ces portes :

- La porte d'oubli (forget gate) :

Détermine les parties de l'état de la cellule précédent C_{t-1} qui doivent être oubliées.

$$f_t = \sigma_g\left(W_f x_t + U_f h_{t-1} + b_f\right)$$

x_t : Vecteur d'entrée au temps t
h_{t-1}: État caché du temps t-1
f_t : Activations de la porte d'oubli au temps t
W, U : Matrices de poids (respectivement pour l'entrée et l'état caché précédent).
b : Vecteurs de biais.
σ : Fonction d'activation sigmoïde

- La porte d'entrée (input gate) :

Décide quelles nouvelles informations seront stockées dans l'état de la cellule.

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$

i_t: Activations de la porte d'entrée au temps t

- La porte de sortie (output gate) :

Détermine quelles informations de l'état de la cellule seront utilisées dans l'état caché h_t

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$

o_t : Activations de la porte de sortie au temps t.

- Bloc de candidat pour l'état de la cellule (cell candidate) :

Propose une mise à jour de l'état de la cellule.

$$\widetilde{C_t} = \sigma_c(W_C x_t + U_C h_{t-1} + b_C)$$

$\widetilde{C_t}$: Nouveau candidat pour l'état de la cellule au temps t.

- Mise à jour de l'État de la cellule (cell state update) :

Combinaison de l'état de la cellule précédent et du bloc de candidat pour former le nouvel état de la cellule.

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C_t}$$

C_{t-1}: État de la cellule au temps t-1
C_t : Nouvel état de la cellule au temps t.
$\odot$: Produit de Hadamard (multiplication élément par élément).

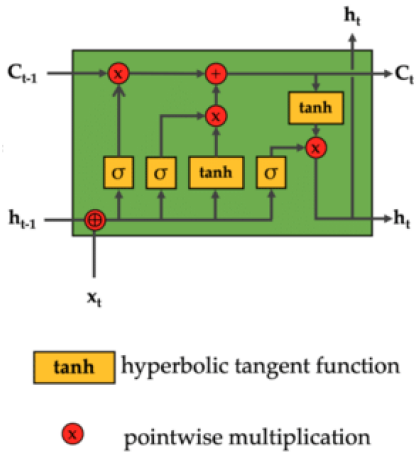
- Mise à jour de l'état caché (Hidden state update) :

L'état caché est mis à jour en utilisant l'état de la cellule filtré par la porte de sortie.

$$h_t = o_t \odot \sigma_h(C_t)$$

h_t : Nouvel état caché au temps t

- Schéma d’un LSTM



$x_t et h_{t-1}$: Entrée actuelle et état caché précédent, qui sont combinés et traités à travers différentes portes pour mettre à jour l'état de la cellule.

Les portes d'oubli, d'entrée et de sortie régulent le flux d'informations dans et hors de la cellule, décident des informations à garder ou à éliminer et préparent la sortie de l'état caché.

Les opérations point par point (représentées par des "x" rouges) indiquent des multiplications élément par élément, par exemple entre l'état de la cellule précédent C_{t-1} et la sortie de la porte d’oublie

L'état de la cellule C_t est la mémoire interne de la cellule LSTM qui transporte l'information à travers les pas de temps, tandis que l'état caché h_t est utilisé pour les calculs de la sortie actuelle et comme entrée pour la cellule au pas de temps suivant.

Complexité des LSTM :

-En Temps : La complexité en temps des LSTM pour une seule opération de mise à jour est généralement O(n) (avec n la taille de l’entrée). Cependant, en raison de la nature séquentielle des LSTM, où chaque étape temporelle dépend de la sortie de l'étape précédente, les opérations ne peuvent pas être facilement parallélisées. Ce qui veut dire que pour une séquence de longueur T, la complexité en temps serait O(T * n)

-En Espace : En ce qui concerne la complexité en espace, les LSTM ont une complexité O(m^2 + m * n) où m est le nombre d'unités dans le réseau LSTM et n est la dimensionnalité de l'entrée

Avantages des LSTM :

-Gestion des dépendances à long terme : Les LSTM sont spécialement conçus pour traiter des séquences de données avec des dépendances temporelles longues et complexes, ce qui est essentiel pour analyser les séries temporelles financières.

-Capacité de modélisation avancée : Ils peuvent capturer des patterns complexes dans les données de marché, ce qui les rend particulièrement aptes à identifier des tendances subtiles et des signaux faibles qui pourraient indiquer des anomalies.

-Réduction du problème de disparition du Gradient : Grâce à leur architecture unique de portes, les LSTM surmontent le problème du vanishing du gradient souvent rencontré avec les RNN traditionnels.

-Flexibilité et Adaptabilité : Ils peuvent être ajustés pour divers types de données et de tâches, ce qui les rend adaptés à diverses configurations de données financières.

Inconvénients de LSTM :

-Complexité et coût computationnel : Les LSTM sont plus complexes en termes de structure et nécessitent des ressources de calcul plus importantes, ce qui peut entraîner des temps d'entraînement plus longs et une consommation accrue de ressources.

-Risque de surajustement (overfitting) : En raison de leur capacité à modéliser des relations complexes, ils peuvent surajuster sur des données de marché bruitées ou limitées, apprenant le "bruit" comme s'il s'agissait de patterns significatifs.

-Complexité de la mise au point : Le réglage des hyperparamètres (comme le nombre de couches, la taille de l'état caché, les taux d'apprentissage, etc.) peut être plus difficile et nécessiter une expertise en modélisation plus approfondie.

-Interprétabilité : Les LSTM, comme de nombreux modèles de deep learning, peuvent manquer de transparence dans leur prise de décision, rendant difficile la compréhension et l'explication des raisons pour lesquelles certaines données sont identifiées comme des anomalies.

3. Approche statistique

3.1 Détection de valeurs aberrantes avec fonctions densité de noyau :

La détection des données aberrantes avec des fonctions de densité de noyau est une approche statistique qui vise à identifier des observations aberrantes dans un ensemble de données. Elle repose sur l'utilisation de fonctions de densité de noyau, et permet de comparer la densité de chaque point en la comparant avec les densités de ses proches voisins. La méthode vise à détecter les points de données qui s'écartent significativement du modèle attendu. Cette approche s'avère particulièrement utile dans des domaines tels que l'analyse de séries temporelles financières, où la détection précise des anomalies peut fournir des informations importantes pour la prise de décision.

$$LDE(\mathbf{x}_j) \propto \frac{1}{m} \sum_{\mathbf{x}_i \in mNN(\mathbf{x}_j)} \frac{1}{(2\pi)^{\frac{dim}{2}} h(\mathbf{x}_i)^{dim}} \exp(-\frac{\mathbf{rd}_k(\mathbf{x}_j, \mathbf{x}_i)^2}{2h(\mathbf{x}_i)^2})$$
$$= \frac{1}{m} \sum_{\mathbf{x}_i \in mNN(\mathbf{x}_j)} \frac{1}{(2\pi)^{\frac{dim}{2}} (h \cdot d_k(\mathbf{x}_i))^{dim}} \exp(-\frac{\mathbf{rd}_k(\mathbf{x}_j, \mathbf{x}_i)^2}{2(h \cdot d_k(\mathbf{x}_i))^2}).$$

Avec, $\mathbf{rd}_k(\mathbf{y}, \mathbf{x}) = \max(d(\mathbf{y}, \mathbf{x}), d_k(\mathbf{x}))$.

m le nombre de proches voisins

d_k (x) la distance du kième voisin par rapport à x

Le LDE (local density estimate) s'appuie sur une estimation non paramétrique de la densité d'un point de la base de données, en utilisant la distance de réacheminement. Le LDE comporte une analyse locale de la distribution des données, et calcule la densité en considérant les voisins les plus proches. Cela permet de prendre en compte le fais que tous les points n'ont pas la même densité et donc de détecter des anomalies avec plus de précision.

Nous pourrons ensuite utiliser la LDF (local density factor) qui est évalué en un point de la base de données et qui est défini comme le rapport de la LDE moyenne de ses m voisins les plus proches avec la LDE au point (nous utiliserons c=0.01) :

$$LDF(\mathbf{x}_j) \propto \frac{\sum_{\mathbf{x}_i \in mNN(\mathbf{x}_j)} \frac{LDE(\mathbf{x}_i)}{m}}{LDE(\mathbf{x}_j) + c \cdot \sum_{\mathbf{x}_i \in mNN(\mathbf{x}_j)} \frac{LDE(\mathbf{x}_i)}{m}}.$$

Avantages :

-Permet une analyse locale qui s'adapte aux fluctuations du marché.

-La LDF a montré des résultats similaires à d'autres méthodes de détection (LOF) montrant des résultats valables.

-La LDF est paramétrable ce qui permettrait de s'adapter en fonction de la base de données sur laquelle on l'utilise.

Inconvénients :

-Cette méthode ne permet pas de prendre en comptes les évènements exogènes.

-La LDF n'as pas été testé sur des très grandes base de données donc on ne connait pas sa précision dans ce cas-là.

- Les valeurs LDF tendent à être plus élevées pour les échantillons aux limites des clusters, ce qui peut entraîner une détection accrue d''outliers dans ces zones, même s'ils ne sont pas nécessairement

3.2 Méthode du z-score

Le score Z, ou score standard, également connue sous le nom de méthode de détection des données aberrantes basée sur les valeurs aberrantes standardisées, est un moyen de décrire un point de données selon sa relation avec la moyenne et l'écart-type d'un groupe de points. L'établissement du score Z revient simplement à représenter les données dans une distribution dont la moyenne est définie sur 0 et dont l'écart-type est défini sur 1.

L'objectif du score Z est d'éliminer les effets de localisation et d'échelle des données, ce qui permet de comparer directement des ensembles de données différents. L'idée derrière la méthode de score Z pour détecter les valeurs aberrantes est qu'une fois que vous avez centré et remis à l'échelle les données, toute valeur trop éloignée de zéro est considérée comme aberrante.

Voici comment la méthode du Z-score fonctionne :

1) Calculez la moyenne (μ) et l'écart type (σ) de l'ensemble de données.
2) Calculez le Z-score pour chaque observation en soustrayant la moyenne de l'observation et en divisant le résultat par l'écart type : Z = (observation - μ) / σ.
3) Définissez un seuil pour le Z-score au-delà duquel une observation est considérée comme aberrante. Un seuil couramment utilisé est de ± 3, ce qui signifie que toute observation ayant un Z-score inférieur à -3 ou supérieur à 3 est considérée comme aberrante.
4) Identifiez les observations qui dépassent le seuil défini. Ce sont les valeurs aberrantes potentielles dans votre ensemble de données.

Le Z-score :

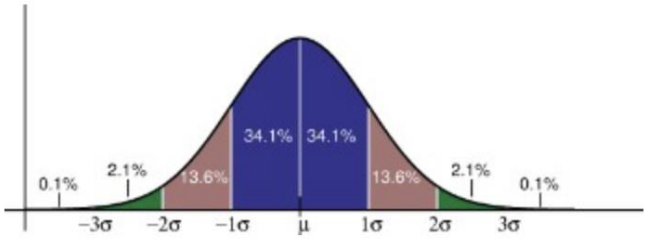
Le Z-score d'une observation est défini comme suit :

$$Z_i = \frac{X_i - \bar{X}}{s_d}$$

Avec $\bar{X}$ est la moyenne de l'échantillon et s_d est la déviation standard (c'est l'écart-type des observations).

Exemple :

En supposant une distribution normale des données (courbe en cloche), la formule "moyenne + 3*écart-type" capture 99,7 % des observations. Statistiquement, toute valeur qui se trouve en dehors de cette plage est considérée comme une anomalie.



Avantages :

- **Facilité d'utilisation** : La méthode du score Z est relativement simple à comprendre et à mettre en œuvre. Elle est basée sur des concepts statistiques de base tels que la moyenne et l'écart-type, ce qui la rend accessible même aux personnes qui ne sont pas des experts en statistiques.

- **Interprétation intuitive** : Les valeurs des scores Z peuvent être facilement interprétées. Une valeur de score Z élevée indique une observation très éloignée de la moyenne, ce qui suggère une forte probabilité d'être une donnée aberrante. En revanche, une valeur de score Z proche de zéro indique que l'observation est proche de la moyenne, ce qui suggère qu'elle est probablement normale.

- **Flexibilité** : La méthode du score Z peut être utilisée avec différents types de données et dans différentes distributions. Elle peut être appliquée aux données univariées ainsi qu'aux données multivariées en utilisant des scores Z multivariés.

Inconvénients :

- **Sensibilité aux valeurs extrêmes** : La méthode du score Z est sensible aux valeurs extrêmes. Si un ensemble de données contient des valeurs aberrantes extrêmes, cela peut fausser les calculs de la moyenne et de l'écart-type, ce qui peut entraîner une mauvaise détection des données aberrantes.

- **Distribution normale requise** : La méthode du score Z repose sur l'hypothèse que les données suivent une distribution normale. Si les données ne sont pas normalement distribuées, les scores Z peuvent être moins fiables pour détecter les données aberrantes.

- **Dépendance à la taille de l'échantillon** : La méthode du score Z peut être influencée par la taille de l'échantillon. Pour de petits échantillons, les estimations de la moyenne et de l'écart-type peuvent être moins précises, ce qui peut affecter la précision de la détection des données aberrantes.

3.3 Méthode des 3 écarts-types

En statistique, la règle 68-95-99,7 (ou règle des trois sigmas ou règle empirique) indique que pour une loi normale, presque toutes les valeurs se situent dans un intervalle centré autour de la moyenne et dont les bornes se situent à trois écarts-types de part et d'autre de celle-ci.

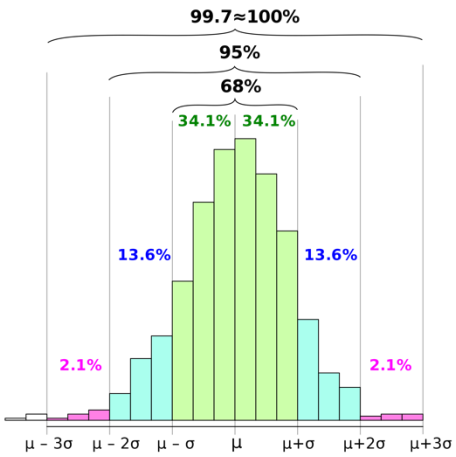
Environ 68,27 % des valeurs se situent à moins d'un écart-type de la moyenne.
De même, environ 95,45 % des valeurs se situent à moins de deux écarts-types de la moyenne.
La quasi-totalité (99,73 %) des valeurs se situent à moins de trois écarts-types de la moyenne.

$$\mathbb{P}(\mu - \ \sigma \leq x \leq \mu + \ \sigma) \approx 0,6827$$
$$\mathbb{P}(\mu - 2\sigma \leq x \leq \mu + 2\sigma) \approx 0,9545$$
$$\mathbb{P}(\mu - 3\sigma \leq x \leq \mu + 3\sigma) \approx 0,9973$$

La règle des trois sigmas exprime une heuristique fréquemment utilisée : la plupart des valeurs se situent à moins de trois fois l'écart-type de la moyenne. Pour de nombreuses applications pratiques, ce pourcentage de 99,7 % peut être considéré comme une quasi-certitude. L'usage de cette heuristique dépend cependant du domaine : ainsi en sciences sociales, un résultat est considéré comme significatif si son intervalle de confiance est au moins de 95 %, soit de l'ordre de deux sigmas, alors qu'en physique des particules, le seuil de significativité se situe autour de cinq sigmas (soit un intervalle de confiance à 99,999 94 %).

La règle des trois sigmas est également applicable à d'autres distributions que la loi normale. En effet, l'inégalité de Bienaymé-Tchebychev permet d'affirmer que pour toute variable aléatoire, au moins 88,8 % des réalisations se situent dans un intervalle de trois sigmas.

Ces valeurs numériques (68 %, 95 % et 99,7 %) proviennent de la fonction de répartition de la loi normale.



Avantages :

- **Simplicité :** La méthode des 3 écarts-types est facile à comprendre et à appliquer. Elle ne nécessite pas de connaissances statistiques avancées et peut être utilisée par des personnes ayant une compréhension de base des concepts de moyenne et d'écart-type.

- **Robustesse aux valeurs extrêmes** : Contrairement à la méthode du score Z, la méthode des 3 écarts-types est moins sensible aux valeurs extrêmes. Étant donné que la détection des valeurs aberrantes est basée sur une plage fixe de 3 écarts-types, les valeurs extrêmes ont moins d'impact sur la détection des données aberrantes.

- **Indépendance de la distribution** : La méthode des 3 écarts-types ne repose pas sur l'hypothèse de normalité des données. Elle peut être utilisée avec différents types de distributions, ce qui la rend plus flexible dans l'application à différents ensembles de données.

Inconvénients :

- **Limitation à la distribution normale** : Bien que la méthode des 3 écarts-types soit indépendante de la distribution, elle peut ne pas être aussi efficace avec des distributions non normales. Dans certaines distributions asymétriques ou à queues longues, les observations aberrantes peuvent ne pas être détectées avec précision en utilisant cette méthode.

- **Sensibilité à la taille de l'échantillon** : Tout comme la méthode du score Z, la méthode des 3 écarts-types peut être influencée par la taille de l'échantillon. Avec de petits échantillons, les estimations de la moyenne et de l'écart-type peuvent être moins précises, ce qui peut affecter la précision de la détection des données aberrantes.

- **Manque de seuil adaptatif** : La méthode des 3 écarts-types utilise un seuil fixe de 3 écarts-types pour détecter les données aberrantes. Cela peut ne pas être optimal dans certains cas où la variance des données est élevée ou faible, car le seuil fixe peut être trop restrictif ou pas assez sensible pour détecter les valeurs aberrantes.

4. Choix des 3 méthodes à implémenter

A la suite de nos recherches présentées précédemment, nous avons choisi les trois méthodes nous semblant les plus pertinentes que nous allons implémenter et comparer pour la suite du projet.

- Nous avons fait le choix de **l'algorithme KNN** en raison de sa capacité à détecter des anomalies dans les données financières en se basant sur la proximité des points dans un espace multidimensionnel. En utilisant les caractéristiques des prix, KNN peut identifier des observations aberrantes en comparant leurs similitudes avec celles des voisins les plus proches. Sa simplicité d'implémentation en Python et son adaptabilité aux données financières en font un choix pertinent dans le cadre de notre projet.

- Le choix d'utiliser la méthode de détection d'anomalies basée sur les **fonctions de densité de noyau** dans ce projet est justifié par sa capacité à modéliser la distribution des données financières de manière non paramétrique. En observant les variations de densité, on pourra donc identifier les valeurs aberrantes en comparant la densité de chaque point avec celles de ses voisins ce qui offre une bonne flexibilité d'usage. L'implémentation en Python semble être accessible, offrant ainsi une solution puissante et adaptable pour repérer des anomalies dans les historiques de données.

- Le choix des **Réseaux de Neurones Récurrents** (RNN), et plus spécifiquement des Long Short-Term Memory (LSTM), pour notre modèle de détection d'anomalies dans les données financières est motivé par leur excellence dans la capture des séquences temporelles et des dépendances à long terme inhérentes à ce type de données. Les LSTM sont particulièrement adaptés pour apprendre des comportements complexes et non linéaires des marchés financiers. Ils offrent une manière sophistiquée et adaptée de repérer les anomalies, en assurant une détection précise et contextuellement informée des irrégularités dans les séries temporelles financières