Compte rendu

Table des matières

I. Introduction 1

II. Architecture du code 1

III. Pre-processing 1

IV. Sélection des données financières 2

V. Extraction des variables 2

VI. Réduction du nombre de variables 3

VII. Numérisation des valeurs de la dataframe 4

VIII. IA causale 4

IX. Conclusions et Ouvertures 6

I. Introduction

Nous avons à notre disposition un corpus de 32320 articles abordant des sujets divers. L'objectif est de pouvoir en extraire des informations financières et établir des liens de causalités entre des événements et les mouvements boursiers. Pour cela, nous allons dans un premier temps traiter la donnée afin de la rendre utilisable par des modèles de NLP, puis filtrer les articles liés à la finance grâce à ChatGPT. Par la suite, nous envoyons une requête à ChatGPT afin d'extraire les informations essentielles dans ces textes afin de créer les variables. Il est nécessaire de retraiter la sortie de ChatGPT afin de réduire le nombre de variables. Nous encodons les informations et obtenons une table qui est fournie en entrée d'une IA causale afin de déterminer les relations entre chaque variable.

II. Architecture du code

Nous avons principalement travaillé sur "causal_find", dans lequel les étapes suivantes sont appliquées : ChatGPT essaie de trouver par lui-même les variables et nous utilisons un algorithme et une seconde fois ChatGPT afin de réduire le nombre de variables trouvés. Nous proposons un deuxième dossier "causal_direct", où nous imposons dans une requête ChatGPT les variables et nous demandons à ChatGPT de relever les informations essentielles et de les classer selon ces variables. Un fichier "Synthèse bibliographique" contient le résumé de certains articles sur le sujet et qui pourrait amener à des pistes d'amélioration.

III. Pre-processing

Cette partie du code met en forme la donnée afin de pouvoir l'utiliser et la donner à chatGPT. Le corpus se compose de 9 colonnes : « modifyAt », « createAt », « titre »,

« source », « idArticle », « texte », « date », « auteur », « vecteur ». Nous n'utilisons que les données « titre » et « texte ».

Le traitement consiste en la suppression des doublons, la suppression des valeurs manquantes, la création d'une nouvelle colonne « content » qui contient le titre et le corps de l'article.

Après traitement, il reste 4515 articles. Dans notre cas, nous n'avons pas le temps de traiter l'ensemble du corpus et nous nous restreignons dans notre projet à étudier les 1000 premiers articles du corpus traité.

Remarque :

- Nous pourrions inclure dans l'étude la source, la date et l'auteur fournis par chaque article.
- Nous avons fait le choix de ne pas supprimer la ponctuation ou de mettre en minuscule car ChatGPT traite correctement les phrases ponctuées.

IV. Sélection des données financières

Cette partie du code sélectionne les articles financiers ou proches. Nous lançons une requête à ChatGPT par article afin de classer ces derniers (1 si lien avec la finance, 0 sinon). Le point positif ici est l'utilisation de ChatGPT au lieu du traitement classique par des modèles pré-entraînés (il était difficile de télécharger ces modèles sur nos machines).

Remarque :

- ChatGPT sélectionne correctement les articles comme nous le souhaitons, c'est-à-dire les articles qui ont un lien avec la finance ou qui peuvent impacter le marché.
- Cette étape est chronophage (vingtaine de minutes pour 500 articles), il faudrait étudier des pistes d'exploration afin d'optimiser cette partie. Il serait peut-être plus rapide dans ce cas d'utiliser des modèles pré-entraînés.
- On pourrait supprimer cette étape dès le départ, en fournissant un corpus de textes liés à la Finance. Il faudrait extraire (algorithme de scrapping) les articles de sources et presses financières.

V. Extraction des variables

Dans cette partie, nous cherchons à extraire des variables de traitement, de résultat et de confusion liées à la hausse ou la baisse des marchés financiers. A priori ces variables sont inconnues et doivent donc être déterminées par ChatGPT.

Pour se faire, nous émettons une requête par article afin d'extraire ces variables. ChatGPT nous renvoie les informations essentielles classées par variable sous format JSON. Ces informations sont des mots ou des bouts de phrases contenus dans l'article.

Nous demandons à ChatGPT de créer une autre variable « Indicateur » qui peut prendre les valeurs « POSITIF », « NEUTRE » et « NEGATIF ». Dans cette variable, ChatGPT donne son sentiment sur l'article et son impact en bourse et nous nous servirons de celle-ci en tant que sortie du modèle causal.

Les informations extraites sont renvoyées dans des dictionnaires ayant le format suivant :

{
« Variable 1 » : [« valeur1 », « valeur 2 », …],
« Variable 2 » : [« valeur 1 », « valeur 3 », …],
…
}

Remarque :

- La requête peut être trop longue, nous devrions la simplifier.
- La requête est chronophage.
- Parfois la réponse à la requête n'est pas dans le format attendu. Nous relançons la requête dans ce cas.
- L'extraction de variables génèrent parfois des variables qui peuvent être proches, mais sous des désignations différentes (sans emplois et chômage par exemple). Il est nécessaire de les regrouper afin de diminuer le nombre de variables créées. Nous constatons par la suite des problèmes en nombre de tokens ou de groupes réalisés.
- Certaines variables créées sont difficiles à remettre en contexte dans la réduction de variables (exemple de variable : « performance », mais performance de quoi ? Entreprise, Sport…)

VI. Réduction du nombre de variables

Comme nous n'imposons pas les noms de variables à ChatGPT, nous obtenons un grand nombre de variables. Les variables ne sont pas forcément communes entre les articles, il est donc pour le moment impossible d'utiliser de l'IA causale. Nous souhaitons réduire ce nombre de variables afin d'en avoir moins de 15 en créant de nouvelles variables plus générales, regroupant les anciennes variables entre elles. Cela permet d'avoir plus d'informations par catégorie.

Pour cela nous regroupons toutes les variables identifiées dans une liste et nous envoyons une requête à ChatGPT. Il essaie de construire des catégories dans lesquelles les variables d'entrée sont stockées. Une variable peut appartenir à plusieurs catégories.

La sortie se présente sous la forme suivante :

{
« Catégorie 1 » : [Variable 1, Variable 2, …],
« Catégorie 2 » : [Variable 2, Variable 3, …],
…
}

Nous avons développé un algorithme qui permet de réaliser le « mapping » des informations essentielles de chaque article vers les catégories. Il s'agit d'inverser le précédent dictionnaire puis de créer une nouvelle dataframe dans laquelle les lignes sont les articles, les colonnes

sont les catégories et les valeurs sont les informations essentielles de chaque article classé dans une des catégories.

Remarque :

- Parfois certaines variables ne sont pas attribuées à une catégorie et nous perdons l'information extraite
- Si la liste donnée en entrée à ChatGPT est trop longue, le nombre de tokens peut dépasser la limite autorisée.

Piste à explorer :

Une des contraintes du cahier des charges était de laisser ChatGPT trouver lui-même les variables d'intérêt à partir de chaque article du corpus, contrairement aux autres algorithmes de NER (« Name Entity recognition »). Nous avons réalisé cette étude jusqu'au bout. Il serait possible de :

A. proposer directement à ChatGPT les variables d'intérêt. L'utilisateur peut avoir connaissance du sujet et s'attendre à un certain résultat.
B. de demander à ChatGPT de proposer des variables d'intérêt sans l'accès au corpus.

Cela permettrait d'éviter de réduire le nombre de variables (partie 5) qui entraîne des erreurs, oublis et perte d'information. De plus, ces variables d'intérêt sont globalement faciles à identifier.

VII. Numérisation des valeurs de la dataframe

Nous obtenons finalement la dataframe recherchée.

Afin d'utiliser l'IA causale, nous devons transformer chaque valeur (étant un string) en réel. En pratique, cela se fait en deux étapes. Tout d'abord, les strings (mots ou groupe de mots) sont transformés en vecteur, puis une autre méthode est utilisée afin de transformer ces vecteurs en réels.

L'état de l'art suggère d'utiliser des algorithmes tels Word2Vec puis TF.IDF afin de numériser. D'autres algorithmes comme BERTTopic peuvent utiliser d'autres méthodes tel une ACP.

Afin d'utiliser au maximum ChatGPT, nous décidons d'utiliser des fonctions d'OpenAI. Premièrement, nous importons puis appliquons le modèle "text-embedding-ada-002" afin de créer des vecteurs numpy à partir des mots. Dans un second temps, nous utilisons une ACP issue de la librairie d'OpenAI afin de projeter chaque valeur sur la colonne correspondante. Nous obtenons ainsi un dataframe pouvant être utilisé dans un algorithme d'IA causale.

VIII. IA causale

Plusieurs librairies d'IA causale existent.

La plus connue, DoWhy, exige en entrée une dataframe de données numériques et un graphe de structure causal des variables. Il est donc nécessaire au préalable d'établir des

relations causales pour que DoWhy réalise des inférences dessus. Nous l'avons expérimenté sur une des machines de notre groupe (la seule qui arrivait à installer la librairie) et nous a donné des résultats.

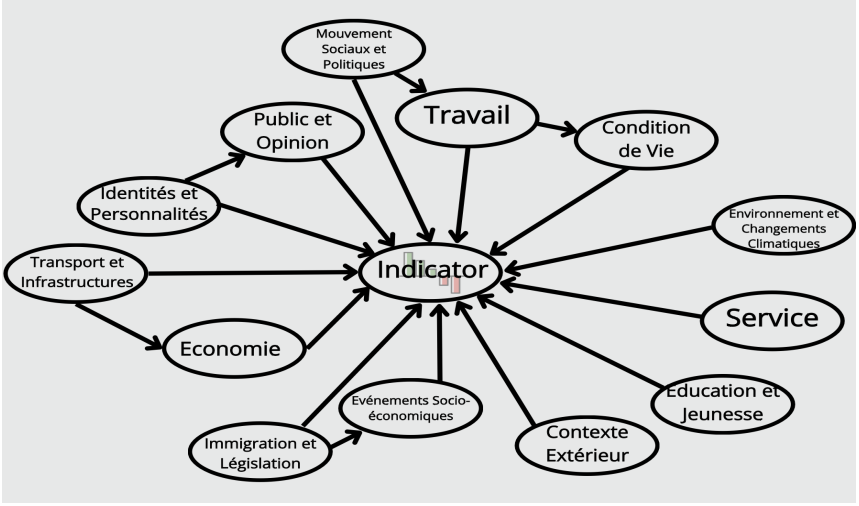
Afin de construire en amont le graphe à donner à DoWhy nous avons utiliser la bibliothèque python CDT (causal discovery toolbox) qui permet d'établir des relations causales en utilisant des algorithmes de recherche comme PC, GES (Greedy Equivalence Search) ou GIES. Nous avons principalement essayé GIES, qui donnait les meilleurs résultats sur des dataframes de démonstration, mais nous n'avons pas eu le temps de réaliser un benchmark des différents algorithmes sur notre dataframe.

La librairie DoWhy sort en résultat les coefficients de causalité et un graphique, cependant nous n'arrivons pas à afficher le graphique de la librairie. Nous avons donc construit à la main le graphique en se basant sur les coefficients donnés en sortie de l'algorithme.

Coefficients en sortie de DoWhy:


```
{
 "Conditions de vie->Indicateur": 16.17162366412843,
 "Contexte Extérieur->Indicateur": 9.827960318024417,
 "Environnement et Changements Climatiques->Indicateur": 6.795064259420889,
 "Identités et Personnalités->Indicateur": 24.075101406402,
 "Immigration et Législation->Indicateur": 1.4420653909180796,
 "Mouvements Sociaux et Politiques->Indicateur": 6.2751465279986585,
 "Public et Opinion->Indicateur": 4.09189660290318,
 "Services->Indicateur": 4.362397155117872,
 "Technologie et Innovation->Indicateur": 9.065833177362181,
 "Transport et Infrastructures->Indicateur": 1.4845571771991863,
 "Travail->Indicateur": 5.54351041262706,
 "Économie->Indicateur": 3.822559593869478,
 "Éducation et Jeunesse->Indicateur": 3.933364804952988,
 "Événements Socio-économiques->Indicateur": 3.1089195090755877
}
```


Graphique de la structure causale



IX. Conclusions et Ouvertures

Lors du processing dans le notebook ipython, nous pouvons repérer à quelques reprises des erreurs dans les outputs de ChatGPT. Il nous a été nécessaire de reprendre à la main et de corriger ces erreurs (par exemple, des variables non attribuées à des catégories, des outputs qui ne sont pas sous le bon format…). Cela peut être dû à des requêtes mal formulées ou à la complexité de la requête. Il peut être nécessaire de simplifier certaines d'entre elles.

Nous avons étudié une partie du corpus, mais non son intégralité. Nous avons dû faire face à plusieurs reprises à des limites de tokens (nombre par minute ou prompt trop long). Un travail d'optimisation sur les requêtes est nécessaire et peut conclure à de nouvelles étapes ou à la suppression de certaines étapes.

Il peut être intéressant d'utiliser l'intégralité du corpus et des données « source », « date », « auteur ».

D'un autre côté, nous avions pensé à inclure le temps dans notre modèle. Par exemple, l'apprentissage progressif de l'IA en incluant des articles de journal ayant une date de publication de 0 à T. Cela pourrait donner une dynamique au modèle causal. Par exemple, si une variable "crise sanitaire" est créée, elle aura plus d'impact entre 2020 et 2022 que les autres années. Les articles sont datés et contiennent des informations temporelles.

Nous n'avons pas eu le temps d'étudier et de benchmarker les méthodes et modèles cités dans certains articles et références. Par exemple, les modèles principalement utilisés sont Word2Vec et TF.IDF mais nous avons utilisé des modèles de OpenAI afin de numériser les données textuelles.

Une autre piste de réduction de dimension serait d'utiliser des modèles tels Word2Vec et TF.IDF. Certains articles de recherche s'appuient sur ces modèles.

Certaines variables d'intérêt sont sous-représentées et n'interviennent donc pas dans le modèle. Certaines méthodes peuvent permettre de remédier :

- Premièrement, nous avons dupliqué ces lignes afin de mettre plus de poids à ces variables. Cela était une solution "facile" pour l'extrait réduit.
- Deuxièmement, il est possible de regrouper deux variables entre elles et d'avoir une variable plus globale. Nous avons pu essayer également cette méthode. Le point positif est le fait que nous ne modifions pas l'échantillon de textes. Cependant, cela induit des tâches manuelles dans notre notebook et une intervention de l'utilisateur et non de ChatGPT.
- La dernière possibilité était d'échantillonner à nouveau en cherchant des articles sur le thème correspondant. Nous n'avons pas eu le temps pour essayer cette méthode.

Comme abordé précédemment, nous nous interrogeons sur la détection des variables d'intérêt par ChatGPT. Certaines erreurs peuvent apparaître lors du processus et le nombre de tokens utilisés peut dépasser la limite autorisée. La seconde piste à étudier serait de demander à ChatGPT d'extraire les informations essentielles et de classer dans des variables

définies au préalable. Un notebook a été créé dans le dossier causal_direct sur le google drive. L'étude de cette piste n'a pas pu être terminée.

- ÉVALUATION DU MODELE dans la classification des variables