

Projet de « Statistiques Avancées pour Big Data » Master 2 MBFA

Université Paris Nanterre

GROUPE DE :

TAHIR Badr-eddine (33%)

AZLY Amine (33%)

KAMELA Alesterd (33%)

Tables de matières

PHASE 1 : CART	1
I. Description du modèle CART	1
A. Définition et Histoire	1
B. Types d'Arbres dans CART	1
II. Fonctionnement de l'Algorithme CART	2
A. Structure de l'Arbre	2
B. Critères de Division	2
C. Élagage de l'Arbre	3
III. Détails Techniques de CART	3
A. Impureté de Gini	3
B. Processus de Division	4
IV. Avantages et Limitations	5
A. Avantages	5
B. Limitations	5
V. Conclusion	5
VI. Cost Complexity Pruning	6
A. Introduction	6
B. Fondements Théoriques et CCP	7
C. Sélection de l'Arbre Optimal et Avantages du CCP	8
D. Applications Pratiques et Limitations du CCP	9
VII. Démonstration	10
A. Démonstration de l'Équivalence dans le Cost Complexity Pruning (CCP)	10
B. Exemple :	11
PHASE 2 : CHAID MODEL	14
I. CHAID	14
C. Decision tree components in CHAID analysis	14
D. Chi_square	15
E. CHAID Algorithm	15
F. Example: CHAID Algorithm	17
II. Comparaison de CHAID et CART	19
PHASE 3 : Mise en pratique	20
I. Modèle Random Forest	21
II. Modèle CART	24
III. Modèle CHAID (sur SPSS)	26
Conclusion générale	28
APPENDIX:	1
Sources:	8

PHASE 1 : CART

I. Description du modèle CART

A. Définition et Histoire

L'algorithme CART (Classification and Regression Trees) est une avancée significative dans le domaine des modèles d'arbres de décision. Conçu pour résoudre à la fois des problèmes de classification et de régression, il offre une polyvalence rare en pouvant traiter aussi bien des variables cibles discrètes que continues. À l'origine présenté en 1984 par un groupe éminent de statisticiens, à savoir Leo Breiman, Jerome Friedman, Richard Olshen et Charles Stone, l'algorithme CART est devenu un pilier dans le domaine de l'apprentissage automatique.

L'une des caractéristiques distinctives de CART réside dans sa capacité à produire des modèles prédictifs robustes tout en restant compréhensibles. Cette dualité est cruciale, car elle permet aux praticiens et aux chercheurs de bénéficier d'une interprétabilité claire tout en maintenant des performances élevées. CART exploite les relations intrinsèques entre les caractéristiques des données d'entrée et la variable cible, établissant ainsi un modèle sous forme d'arbre. Dans cette représentation, chaque embranchement symbolise une décision basée sur une caractéristique particulière, tandis que chaque feuille représente une prédiction finale. Cette structure arborescente permet une visualisation intuitive des processus décisionnels sous-jacents, facilitant ainsi la compréhension du modèle par les utilisateurs.

B. Types d'Arbres dans CART

Arbres de Classification

Les arbres de classification sont utilisés quand la variable cible est catégorique, c'est-à-dire qu'elle peut être divisée en groupes distincts. Par exemple, un arbre de classification peut être utilisé pour déterminer si un email est un spam ou non.

Argument 1 : Le principal avantage de l'arbre de classification est sa capacité à modéliser des frontières de décision complexes, permettant de capturer des motifs non linéaires et des interactions entre les caractéristiques.

Argument 2 : Ils sont également intuitifs à comprendre et à interpréter, car les règles de classification peuvent être exprimées sous forme de chemin simple dans l'arbre, ce qui est utile pour des décisions basées sur des preuves explicites.

Arbres de Régression

Les arbres de régression, d'autre part, s'attaquent aux problèmes où la variable cible est une quantité continue. Ils sont souvent utilisés dans des situations où il est nécessaire de faire des prédictions quantitatives, comme la prévision des ventes ou l'estimation de la qualité d'un vin en fonction de ses propriétés chimiques.

Argument 1 : Un avantage des arbres de régression est qu'ils ne supposent pas une relation linéaire entre les caractéristiques et la variable cible, ce qui les rend adaptés pour modéliser des relations qui pourraient autrement nécessiter des transformations complexes des données.

Argument 2 : De plus, les arbres de régression peuvent naturellement gérer les caractéristiques avec des échelles différentes et ne nécessitent pas de normalisation préalable, ce qui simplifie la préparation des données.

II. Fonctionnement de l'Algorithme CART

A. Structure de l'Arbre

Nœuds et Points de Décision

Dans l'arbre CART, chaque nœud représente une question ou une condition basée sur une seule caractéristique des données. Cela peut être comparé à un jeu de "oui ou non" où chaque réponse conduit à une nouvelle question jusqu'à ce qu'une conclusion soit atteinte.

Argument 1 : Cette méthode hiérarchique permet une segmentation détaillée de l'espace de caractéristiques, ce qui rend le modèle capable d'isoler des comportements spécifiques pour différentes combinaisons de caractéristiques.

Argument 2 : Les nœuds internes sont donc des points de décision qui augmentent la granularité de la prédiction et permettent au modèle d'appréhender des relations non linéaires entre les variables.

Nœuds Feuilles et Prédictions

Les nœuds feuilles, ou nœuds terminaux, sont l'aboutissement des chemins de décision au sein de l'arbre. Ils ne contiennent pas de sous-division supplémentaire et attribuent une valeur prédictive basée sur les données qui ont suivi le chemin jusqu'à ce point.

Argument 1 : L'existence de nœuds feuilles distincts pour des chemins de caractéristiques différents permet au modèle de fournir des prédictions personnalisées pour chaque ensemble de conditions.

Argument 2 : La précision des prédictions dans les nœuds feuilles dépend de la pureté des données qui y parviennent, c'est-à-dire à quel point les données dans chaque feuille sont homogènes par rapport à la variable cible.

B. Critères de Division

Approche Gloutonne et Meilleure Division

CART utilise une stratégie gloutonne, ce qui signifie qu'à chaque étape, le modèle cherche la division qui offre le plus grand bénéfice immédiat en termes de pureté, sans nécessairement considérer les bénéfices à long terme ou les divisions ultérieures.

Argument 1 : Bien que l'approche gloutonne ne garantisse pas la solution optimale globale, elle est efficace en termes de temps de calcul et donne souvent des résultats pratiquement utiles.

Argument 2 : Cette méthode permet de simplifier le problème de recherche en réduisant considérablement l'espace des solutions possibles, ce qui rend l'entraînement de l'arbre gérable même avec de grands ensembles de données.

Impureté de Gini et Réduction Résiduelle

Pour la classification, l'impureté de Gini est employée pour évaluer la qualité d'une division en mesurant la fréquence à laquelle un élément aléatoire serait mal classé si on le classait au hasard selon la distribution des classes dans le sous-ensemble.

Pour la régression, la réduction résiduelle - souvent mesurée par l'erreur quadratique moyenne - est utilisée pour déterminer comment une division réduite l'hétérogénéité des valeurs cibles au sein d'un sous-ensemble.

C. Élagage de l'Arbre

Prévention du Surajustement

L'élagage est un processus où l'on retire des branches de l'arbre pour simplifier le modèle et améliorer sa capacité de généralisation, c'est-à-dire sa performance sur des données non vues lors de l'entraînement.

Argument 1 : Sans élagage, un arbre de décision peut "apprendre par cœur" le jeu de données d'entraînement, incluant le bruit et les anomalies, ce qui dégrade sa capacité à faire des prédictions précises sur de nouvelles données.

Argument 2 : Un arbre élagué a tendance à avoir une structure plus simple et est moins susceptible d'être affecté par des variations spécifiques aux données d'entraînement, conduisant à une meilleure robustesse.

Méthodes d'Élagage

Élagage de Complexité de Coût : Cette méthode évalue l'augmentation de l'erreur de prédiction par rapport à la complexité de l'arbre et enlève les branches qui ne justifient pas leur existence en termes de précision.

Élagage de Gain d'Information : Il s'agit de mesurer le gain d'information - une réduction de l'incertitude - apporté par chaque branche et de supprimer celles qui apportent peu d'informations supplémentaires par rapport au risque de surajustement qu'elles introduisent

III. Détails Techniques de CART

A. Impureté de Gini

Concept d'Impureté de Gini

L'impureté de Gini est une mesure statistique qui quantifie la fréquence à laquelle un élément aléatoire serait mal classé si on le classait selon la distribution des classes dans un sous-ensemble. Elle est fondamentale dans le processus de décision de CART pour la classification.

Argument 1 : Une impureté de Gini de 0 signifie que le nœud est parfaitement pur, c'est-à-dire que toutes les instances dans ce nœud appartiennent à une seule classe, ce qui indique une séparation claire sans ambiguïté.

Argument 2 : À l'inverse, une impureté de Gini de 1 indique une répartition égale des instances parmi toutes les classes possibles, suggérant aucune domination d'une classe particulière et donc une grande incertitude quant à la classification.

Utilisation de l'Impureté de Gini dans CART

Dans la pratique, CART recherche la division qui minimise l'impureté de Gini après la division. Cette minimisation conduit à des groupes de données plus homogènes qui améliorent la précision de la classification.

Argument 1 : L'utilisation de l'impureté de Gini permet de prendre en compte toutes les classes possibles lors de la division, ce qui offre une vue d'ensemble et évite de favoriser injustement une classe particulière.

Argument 2 : Elle est préférée dans les arbres de décision car contrairement à d'autres mesures comme l'entropie, elle nécessite moins de calculs tout en fournissant des informations similaires sur la pureté des divisions.

B. Processus de Division

Sélection du Meilleur Point de Division

Le processus de division est au cœur de l'algorithme CART. Il consiste à choisir la caractéristique et le seuil de cette caractéristique qui séparent le mieux les données en sous-ensembles les plus homogènes possible.

Argument 1 : Identifier le meilleur point de division est crucial car il influence directement la qualité du modèle final. Une bonne division au début peut significativement augmenter la pureté des nœuds feuilles et donc la précision globale du modèle.

Argument 2 : Ce processus itératif est appliqué récursivement à chaque sous-ensemble généré par une division, ce qui permet de construire un arbre de décision complexe et multi-niveaux adapté aux spécificités des données.

Critères d'Arrêt

Le processus de division continue jusqu'à ce que des critères d'arrêt soient atteints. Ces critères peuvent inclure la pureté maximale d'un nœud (par exemple, impureté de Gini de 0), un nombre minimal d'instances dans un nœud ou une profondeur maximale de l'arbre.

Argument 1 : Les critères d'arrêt empêchent la surconstruction de l'arbre et le surajustement aux données d'entraînement, ce qui est crucial pour la capacité de généralisation du modèle.

Argument 2 : Ils sont également un moyen de contrôler la complexité du modèle et d'éviter des arbres excessivement grands qui seraient difficiles à comprendre et à interpréter.

C. Pseudo-code et Modélisation

Pseudo-code

Le pseudo-code pour l'algorithme CART décrit une structure de contrôle algorithmique qui guide le processus de construction de l'arbre.

Argument 1 : Le pseudo-code sert à démontrer de manière abstraite la logique derrière l'algorithme, offrant une représentation simplifiée qui peut être traduite en code dans n'importe quel langage de programmation.

Argument 2 : Il met en évidence la nature récursive de l'algorithme, illustrant comment chaque division crée de nouvelles opportunités pour des divisions ultérieures, dans un processus itératif.

Modélisation de l'Arbre

La modélisation de l'arbre CART est un processus qui émerge de l'évaluation répétée des variables d'entrée et des points de division possibles, menant à la création d'un modèle qui reflète la structure sous-jacente des données.

Argument 1 : Ce processus de modélisation est dynamique et s'adapte continuellement aux données, ce qui permet de créer des arbres personnalisés pour différents ensembles de données avec leurs spécificités uniques.

Argument 2 : La modélisation résultante offre un cadre visuel et logique pour la prise de décision, où chaque chemin à travers l'arbre peut être interprété comme une série de règles conditionnelles conduisant à une prédiction.

IV. Avantages et Limitations

A. Avantages

Résultats Simples et Interprétables

Les arbres de décision CART sont appréciés pour leur clarté et leur facilité d'interprétation, essentielles dans les domaines exigeant de justifier les décisions prises. Leur représentation visuelle simplifie la détection d'erreurs potentielles, renforçant ainsi leur utilité pratique.

Algorithme Non Paramétrique et Non Linéaire

CART se distingue par sa capacité à modéliser des relations non linéaires sans suppositions préalables sur la distribution des données, ce qui le rend particulièrement résistant aux variations et adapté à une vaste gamme de contextes.

Sélection Implicite des Caractéristiques

L'algorithme optimise la sélection des caractéristiques, évitant le surchargement du modèle avec des données superflues et réduisant la nécessité de prétraitement des données, ce qui contribue à une meilleure performance globale et à une économie de temps dans le développement de modèles.

Résilience aux Valeurs Aberrantes

La résistance naturelle de CART aux valeurs aberrantes, due à sa méthode de division basée sur les relations et non sur les valeurs extrêmes, le rend précieux dans des situations où le nettoyage des données est complexe.

Modèles Compréhensibles avec Peu de Supervision

Les modèles CART nécessitent peu de supervision dans leur développement, permettant aux analystes de se concentrer sur l'interprétation des résultats plutôt que sur le réglage technique, favorisant ainsi leur adoption par les décideurs.

B. Limitations

Risque de Surajustement

Le surajustement reste un défi pour les arbres de décision CART, qui peuvent devenir trop complexes et capter le bruit au lieu des tendances sous-jacentes, nécessitant une attention particulière lors de l'élague pour assurer une bonne généralisation.

Variance Élevée

La structure de l'arbre CART peut varier considérablement avec de petites modifications des données d'entraînement, conduisant à une performance inégale et à une instabilité du modèle qui peut souvent être atténuée par des méthodes d'ensemble comme les forêts aléatoires.

Structure Potentiellement Instable

L'instabilité structurelle peut survenir lorsque plusieurs caractéristiques influencent fortement la sélection des points de division, posant des défis pour la réplication des résultats et nécessitant une analyse approfondie pour assurer la stabilité et la fiabilité du modèle.

V. Conclusion

Nous pouvons dire que L'algorithme CART est un outil versatile et robuste dans le domaine de l'apprentissage automatique, offrant des applications variées et des modèles compréhensibles.

Malgré certaines limitations comme le surajustement, sa capacité à gérer divers types de données le rend inestimable pour de nombreuses disciplines scientifiques et commerciales.

VI. Cost Complexity Pruning

A. Introduction

- Brève introduction aux arbres de décision

Les arbres de décision sont des outils d'analyse prédictive utilisés dans l'apprentissage supervisé, un domaine de l'intelligence artificielle et du machine Learning. Ils sont utilisés pour modéliser les probabilités de décisions successives et leurs résultats possibles.

- La Structure d'un Arbre de Décision se présente comme suite :

Nœuds : Chaque nœud représente une caractéristique ou un attribut.

Branches : Les branches sont les décisions ou les résultats de ces caractéristiques.

Feuilles : Les feuilles de l'arbre représentent les résultats finaux ou les décisions.

Nous pouvons également mentionner d'autres notions tel que la **Profondeur**, **Le Terminal**, **la Branche**, **le Sous Arbre**

- Principe de Fonctionnement :

Le principe de fonctionnement des arbres de décision repose sur des décisions prises à chaque nœud, basées sur des caractéristiques spécifiques des données. Ces décisions dirigent la formation de branches qui se subdivisent de plus en plus, aboutissant finalement à un nœud feuille où une prédiction ou un résultat final est établi. Cette méthode systématique permet une analyse détaillée et une classification précise. Les applications des arbres de décision sont vastes et variées, s'étendant à de nombreux domaines tels que la finance, la médecine et l'ingénierie. Dans ces secteurs, ils sont utilisés pour classer les données ou prédire des résultats, offrant ainsi des outils puissants pour la prise de décision et l'analyse prédictive.

- Définition et importance du Cost Complexity Pruning (CCP) dans la prévention du surajustement

Définition et Fonctionnement du Cost Complexity Pruning (CCP)

Le Cost Complexity Pruning (CCP) est une technique essentielle dans l'optimisation des arbres de décision. Son objectif principal est de simplifier ces arbres en supprimant les branches qui apportent peu à l'efficacité globale du modèle. Cette méthode repose sur l'équilibre entre la réduction de la complexité de l'arbre, afin de prévenir le surajustement, et le maintien d'une précision acceptable du modèle. Le paramètre clé dans ce processus est le paramètre de complexité, appelé alpha, qui aide à réguler l'élagage en équilibrant la perte de précision due à l'élagage avec la simplicité obtenue. Le processus implique une évaluation minutieuse pour identifier et retirer les branches qui, comparativement à leur complexité, contribuent le moins à l'amélioration de la précision de l'arbre, allégeant ainsi le modèle sans sacrifier significativement son efficacité.

Rôle Crucial du CCP dans la Prévention du Surajustement

Dans le contexte des arbres de décision, le surajustement survient lorsqu'un arbre devient trop complexe et trop ajusté aux données d'entraînement, ce qui diminue sa capacité à généraliser efficacement sur de nouvelles données. Le CCP joue un rôle crucial dans la réduction de cette complexité excessive. En élaguant judicieusement les branches superflues, le CCP contribue à la

construction de modèles d'arbres de décision plus robustes et généralisables, ce qui améliore notablement leur performance sur des données non rencontrées auparavant. Les arbres élagués grâce au CCP se distinguent par leur facilité d'interprétation et de compréhension, rendant les décisions prises par le modèle plus transparentes. De plus, ces arbres simplifiés tendent à être plus performants dans des applications réelles, évitant les erreurs et les complications associées à des modèles surdimensionnés et trop complexes.

B. Fondements Théoriques et CCP

- Explication de base des arbres de décision

Les arbres de décision sont structurés de manière arborescente, où chaque nœud représente une question ou un test basé sur les caractéristiques des données. De ces nœuds, se déploient des branches, symbolisant les chemins de décision possibles, et menant finalement à des feuilles qui représentent les résultats ou décisions finales. Le fonctionnement de ces arbres repose sur une séquence logique de questions, guidant progressivement vers une décision ou une catégorie spécifique.

- Problématique du surajustement et introduction au concept de complexité dans les modèles d'arbres

Le surajustement est un problème courant dans les modèles d'arbres de décision, où le modèle s'adapte de manière excessive aux données d'entraînement, réduisant ainsi sa capacité à généraliser correctement sur de nouvelles données. La complexité d'un arbre, mesurée par le nombre de ses nœuds et branches, est souvent un indicateur de ce phénomène. Un arbre trop complexe peut exceller sur les données d'entraînement, mais échouer à prédire correctement sur de nouvelles données, soulignant l'importance d'une gestion adéquate de la complexité pour maintenir l'efficacité du modèle.

- Définition détaillée du CCP

C'est une technique destinée à réduire la complexité des arbres de décision, visant à prévenir le surajustement. Son objectif est de trouver un équilibre entre le détail et la précision du modèle, et sa simplicité et généralisation. La méthode consiste à éliminer les branches qui contribuent faiblement à l'exactitude globale du modèle, simplifiant ainsi l'arbre tout en conservant son efficacité prédictive.

- Rôle du paramètre alpha et processus général du CCP

Le paramètre alpha est crucial dans le processus de CCP, agissant comme un régulateur de l'importance des branches de l'arbre. Des valeurs élevées d'alpha mènent à des arbres plus simples, tandis que des valeurs faibles permettent une plus grande complexité. La sélection optimale de ce paramètre est donc essentielle pour atteindre un modèle qui est à la fois précis et généralisable, équilibrant efficacement la simplicité et la complexité.

- Processus Général du CCP

Le processus de CCP débute avec la construction d'un arbre de décision complet. Chaque branche est ensuite évaluée pour déterminer son impact sur la performance du modèle. Les branches dont l'élimination n'augmente que minimalement l'erreur de prédiction, ajustée par le paramètre alpha, sont retirées. Le résultat est un modèle équilibré qui offre à la fois précision

dans ses prédictions et une capacité de généralisation, évitant le surajustement tout en restant efficace.

III. Construction et Optimisation d'un Arbre Maximal

- Explication de la construction initiale de l'arbre et visualisation

La construction d'un arbre de décision maximal débute par la création d'un modèle complet qui intègre toutes les divisions possibles basées sur les données d'entraînement. Cet arbre est structuré de manière arborescente, avec des nœuds qui posent des questions basées sur les caractéristiques des données, des branches représentant les réponses possibles, et des feuilles indiquant les décisions finales. Visuellement, cet arbre est souvent dense et complexe, reflétant la richesse et la diversité des informations contenues dans les données.

- Limites d'un arbre non élagué

Un arbre de décision qui n'est pas élagué peut se heurter à des problèmes significatifs. Principalement, il a tendance à s'adapter excessivement aux particularités des données d'entraînement, ce qui compromet sa capacité à généraliser sur de nouvelles données - un phénomène connu sous le nom de surajustement. De plus, la complexité de ces arbres peut rendre leur compréhension et leur interprétation difficiles, entravant ainsi leur utilité pratique.

- Calcul du coût de complexité pour chaque nœud et importance du paramètre alpha

Dans le processus d'optimisation, le coût de complexité de chaque nœud est évalué pour déterminer sa contribution à la performance globale du modèle. Le paramètre alpha joue un rôle crucial dans ce contexte, agissant comme un seuil pour l'élagage des nœuds. Ce paramètre aide à équilibrer la complexité de l'arbre avec la précision nécessaire, permettant ainsi de trouver un compromis optimal entre un modèle détaillé et un modèle simplifié.

- Mécanisme de sélection des branches et impact du paramètre alpha sur l'élagage

Le CCP implique un élagage basé sur le paramètre alpha, où les branches qui contribuent peu à l'exactitude du modèle par rapport à leur complexité sont supprimées. Ce processus d'optimisation nécessite un choix judicieux de la valeur d'alpha pour atteindre un équilibre entre l'élagage nécessaire, la précision du modèle et sa simplicité. Le résultat est un arbre équilibré qui offre non seulement une précision dans ses prédictions mais aussi une capacité améliorée à généraliser efficacement sur de nouvelles données.

C. Sélection de l'Arbre Optimal et Avantages du CCP

- Sélection de l'Arbre

Dans le processus du Cost Complexity Pruning, la sélection de l'arbre optimal est un équilibre entre la complexité et la précision. Ce choix est crucial pour assurer l'efficacité du modèle. Différents niveaux du paramètre alpha sont testés pour déterminer quelle configuration offre le meilleur équilibre. La validation croisée joue un rôle essentiel ici, car elle permet d'évaluer la performance de l'arbre sur des ensembles de données non utilisés pendant l'entraînement, garantissant ainsi une sélection plus fiable et objective.

- Avantages du CCP

Le CCP présente plusieurs avantages significatifs. Il réduit le surajustement, un aspect essentiel pour maintenir la fiabilité des modèles prédictifs. De plus, il simplifie le modèle, rendant sa structure plus facile à comprendre et à interpréter, et réduit les erreurs liées à la complexité.

Enfin, il améliore la généralisation du modèle, augmentant ainsi sa pertinence et son efficacité dans des situations réelles.

D. Applications Pratiques et Limitations du CCP

- Applications du CCP

Le CCP trouve des applications dans une variété de domaines. En médecine, par exemple, il peut être utilisé pour affiner les diagnostics en éliminant les facteurs moins pertinents, améliorant ainsi la précision diagnostique. Dans le secteur financier, il aide à construire des modèles de risque plus robustes et précis. Ces applications montrent la flexibilité et l'utilité du CCP dans des contextes variés.

- Limitations et Stratégies

Bien que le CCP soit puissant, il comporte des défis, notamment le choix optimal du paramètre alpha. Une valeur inadéquate peut soit sursimplifier le modèle, soit ne pas suffire à contrôler le surajustement. Pour surmonter ces défis, des stratégies telles que la validation croisée et les analyses de sensibilité sont recommandées. Elles permettent de naviguer plus efficacement dans la sélection d'alpha, assurant ainsi un équilibre optimal entre complexité et précision.

VI. Conclusion

Nous pouvons dire que Le Cost Complexity Pruning est une méthode cruciale pour équilibrer la complexité et la précision dans les arbres de décision. Il joue un rôle clé dans la réduction du surajustement et la simplification des modèles, améliorant ainsi leur capacité de généralisation et leur applicabilité pratique.

- Importance et Perspectives Futures

Le CCP est un outil essentiel dans l'arsenal de l'apprentissage automatique, particulièrement dans l'optimisation des arbres de décision. Avec l'augmentation de la taille et de la complexité des ensembles de données, l'importance du CCP est destinée à croître, offrant des perspectives prometteuses pour son application dans des modèles de plus en plus sophistiqués et diversifiés.

Tableau Récapitulatif entre l’Algorithme CART et le “Cost Complexity Pruning”

Critère	CART	CCP
Objectif	Construire l'arbre de décision complet.	Élaguer l'arbre de décision pour éviter le surajustement.
Méthode de division	Utilise des critères comme l'impureté de Gini ou l'erreur quadratique moyenne pour choisir le meilleur point de division à chaque nœud.	Utilise le coût de complexité pour évaluer l'impact de l'élagage des branches sur la performance du modèle.
Critères de sélection	La meilleure division est celle qui maximise la pureté ou minimise l'erreur au nœud suivant.	La branche est élaguée si son retrait améliore ou maintient le coût de complexité ajusté par un paramètre alpha.

Utilisation d'Alpha	Non applicable (pas de paramètre alpha dans la construction initiale de l'arbre).	Alpha est un paramètre clé qui équilibre la précision et la complexité de l'arbre.
Résultat	Un arbre potentiellement complexe et profond avec de nombreuses branches.	Un arbre simplifié avec moins de branches, optimisé pour la généralisation.
Prévention du surajustement	Peut nécessiter des méthodes supplémentaires pour prévenir le surajustement, comme la limitation de la profondeur de l'arbre ou le pré-élagage.	Intègre la prévention du surajustement dans le processus grâce au paramètre alpha qui contrôle l'élagage.
Validation	Nécessite une évaluation sur un ensemble de test ou via la validation croisée pour confirmer la performance.	La performance est évaluée après élagage pour chaque valeur d'alpha, souvent en utilisant une courbe de validation pour sélectionner le meilleur alpha.

VII. Démonstration

A. Démonstration de l'Équivalence dans le Cost Complexity Pruning (CCP)

Notre Objectif consiste à Démontrer l'équivalence du Cost complexity Pruning qui s'écrit :

$$C_{\alpha}(T - T_t) - C_{\alpha}(T) \leq 0 \Leftrightarrow g(t) - \alpha \leq 0$$

Où

- $C_{\alpha}(T)$ représente le coût de complexité de l'arbre T ,
- $C_{\alpha}(T - T_t)$ est le coût après élagage du sous-arbre T_t ,
- $g(t)$ est le gain de l'élagage du nœud t , et α est le paramètre de complexité.

Notre objectif est de démontrer l'équivalence cruciale du Cost Complexity Pruning qui s'écrit : $C_{\alpha}(T - T_t) - C_{\alpha}(T) \leq 0 \Leftrightarrow g(t) - \alpha \leq 0$. Cette équivalence joue un rôle clé dans la décision d'élaguer ou non un sous-arbre dans un arbre de décision.

Étape 1 : Compréhension du Coût de Complexité

Formule : $C_{\alpha}(T) = \text{Erreur}(T) + \alpha \times \text{Nombre de feuilles}(T)$

Explication : Le coût de complexité $C_{\alpha}(T)$ d'un arbre de décision peut être défini comme la somme de l'erreur de prédiction de l'arbre T et d'un terme pénalisant proportionnel au nombre de feuilles de l'arbre, multiplié par le paramètre de complexité α . Ce coût vise à équilibrer la précision de l'arbre avec sa simplicité.

Étape 2 : Effet de l'Élagage sur le Coût

Observation : L'élagage du sous-arbre T_t de T réduit le nombre de feuilles, affectant ainsi le coût de complexité du nouvel arbre.

Coût Après Élagage : $C_{\alpha}(T - T_t)$ est le coût de complexité de l'arbre T après élagage du sous-arbre T_t .

Explication : Lorsqu'un sous-arbre T_t est retiré de l'arbre T , le coût de complexité de l'arbre résultant $C_{\alpha}(T - T_t)$ change également. Ce changement est dû à la réduction du nombre de feuilles, et cela diminue le terme de complexité tout en pouvant potentiellement augmenter l'erreur de prédiction.

Étape 3 : Calcul du Gain d'Élagage

Formule de Gain : $g(t) = C_{\alpha}(T) - C_{\alpha}(T - T_t)$

Interprétation : Le gain d'élagage $g(t)$ est calculé comme la **différence entre le coût de complexité de l'arbre avant et après l'élagage**. Ce gain mesure l'efficacité de l'élagage en termes de réduction du coût global de l'arbre.

Étape 4 : Établissement de l'Équivalence

- **Transformation de l'Équivalence:**

Réarrangeons l'équivalence initiale $C_\alpha(T - T_t) - C_\alpha(T) \leq 0$ en $C_\alpha(T) - C_\alpha(T - T_t) \geq 0$.

- **Substitution du Gain d'Élagage:**

En substituant $g(t)$ dans l'équivalence, nous obtenons $g(t) \geq 0$.

- **Équivalence Finale:**

En ajoutant α des deux côtés, $g(t) - \alpha \geq -\alpha$. Étant donné que $-\alpha \leq 0$, cela implique $g(t) - \alpha \geq 0$, ce qui correspond à la deuxième partie de notre équivalence initiale.

Interprétation :

La transformation de l'équivalence indique que l'élagage est bénéfique si et seulement si le gain réalisé par l'élagage, compensé par le coût supplémentaire introduit par le paramètre α , est inférieur ou égal à zéro. Cette équivalence est fondamentale pour guider le processus d'élagage dans le CCP.

Étape 5 : Implications Pratiques

- **Critère d'Élagage:**

Un sous-arbre T_t devrait être élagué si $g(t) - \alpha \leq 0$, ce qui indique que l'élagage du sous-arbre améliore ou maintient l'équilibre entre la précision et la simplicité de l'arbre.

- **Guidage du Processus de CCP :**

Cette condition guide le processus d'élagage, indiquant quand un nœud doit être conservé ou éliminé pour optimiser la structure de l'arbre.

Conclusion

Cette démonstration unifiée souligne le rôle crucial du paramètre α et du gain d'élagage dans le processus de CCP. Elle offre une base pour prendre des décisions objectives d'élagage, visant à obtenir un arbre de décision optimal et généralisable.

B. Exemple :

"Prédiction de la Durabilité des Bâtiments en Utilisant CART et l'Amélioration par le Cost Complexity Pruning"

Contexte du Sujet

La durabilité des bâtiments est un enjeu crucial dans le domaine de la construction et de l'urbanisme. Prédire la durabilité d'un bâtiment en fonction de divers facteurs peut aider à planifier des interventions de maintenance et à améliorer la gestion des ressources.

Échantillon de Données

Pour cette étude, nous utiliserons un jeu de données fictif, comprenant les caractéristiques suivantes pour chaque bâtiment :

Age : Âge du bâtiment (en années).

Matériaux : Type de matériaux principaux utilisés (catégorique : "béton", "bois", "acier").

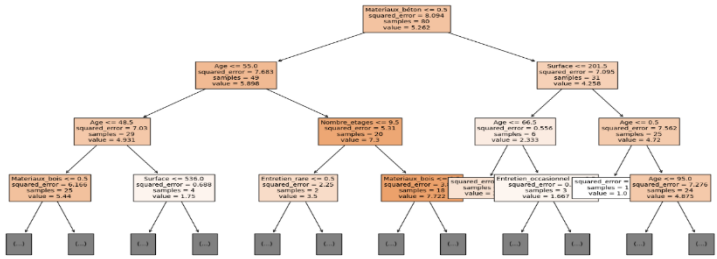
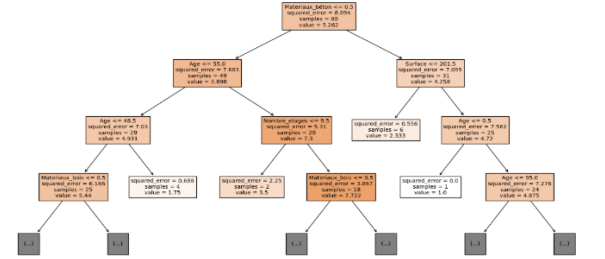
Surface : Surface totale du bâtiment (en mètres carrés).

Nombre d'étages : Nombre total d'étages.

Exposition aux éléments : Degré d'exposition aux conditions environnementales (échelle de 1 à 5).

Entretien : Fréquence des travaux d'entretien (catégorique : "régulier", "occasionnel", "rare").

Durabilité : Score de durabilité du bâtiment (variable cible, échelle de 1 à 10).

	
<p>CART</p>	<p>CPP</p>
<p>Une visualisation d'un arbre de décision qui a été entraîné en utilisant la méthode CART (Classification And Regression Trees).</p>	<p>1. Racine de l'Arbre : Le nœud racine commence avec la variable "Matériaux_béton <= 0.5" avec une erreur quadratique moyenne (squared_error) de 8.094, indiquant le début de la segmentation des données basée sur la présence de béton comme matériau de construction.</p>
<p>Nœud Racine : Le nœud supérieur (racine) indique que la première division (split) des données est basée sur l'âge des bâtiments, avec un seuil de 55 ans. Les bâtiments de 55 ans ou moins vont à la branche de gauche, et ceux de plus de 55 ans vont à la branche de droite.</p>	<p>2. Division Principale : À partir de la racine, l'arbre se divise en deux branches principales basées sur l'âge du bâtiment (Age <= 55.0 à gauche et Surface <= 201.5 à droite), avec chaque branche représentant les sous-ensembles de données résultants.</p>
<p>Importance des Caractéristiques : L'arbre montre que l'âge, le type de matériaux, la surface, le nombre d'étages et le niveau d'entretien sont des caractéristiques utilisées pour prédire la durabilité des bâtiments.</p>	<p>3. Nœuds et Branches : Chaque nœud suivant représente une décision supplémentaire basée sur différentes variables, comme le nombre d'étages (Nombre_etages <= 9.5) et l'âge (Age <= 48.5 ou Age <= 95.0). Les branches se terminent par des feuilles ou par des ellipses "...", indiquant d'autres divisions qui ne sont pas affichées.</p>
<p>Branches et Nœuds : Chaque nœud interne (les boîtes orange et marrons) représente un point de décision basé sur une caractéristique, et les branches indiquent le chemin suivi par les données en fonction des valeurs de ces caractéristiques. Par exemple, pour les bâtiments de 55 ans ou moins, la prochaine division est basée sur le type de matériaux (béton ou autre), et pour ceux de plus de 55 ans, la division suivante dépend de la surface du bâtiment.</p>	<p>4. CCP et Élagage : Le CCP vise à réduire la complexité de l'arbre en élaguant les branches qui contribuent peu à la précision du modèle. Cela est fait en ajustant le paramètre alpha pour équilibrer la complexité de l'arbre et la précision des prédictions. L'arbre présenté semble avoir été élagué à un certain niveau, comme indiqué par les nœuds terminaux et les valeurs de squared_error qui montrent le degré d'erreur à chaque décision prise par l'arbre.</p>

<p>Feuilles (Résultats Finaux): Les nœuds feuilles (non montrés, indiqués par '(...)') sont les résultats finaux de l'arbre, où les prédictions sont faites. Ils sont atteints après avoir suivi les branches de décision depuis le nœud racine.</p>	<p>5. Interprétation des Feuilles :</p> <p>Les feuilles de l'arbre (les nœuds finaux) présentent les valeurs prédites pour la variable cible, qui dans ce cas pourrait être le score de durabilité des bâtiments.</p> <p>Les feuilles affichent également le nombre d'échantillons qui tombent dans cette catégorie et l'erreur quadratique moyenne pour ces prédictions.</p>
<p>Évaluation des Splits: Chaque nœud montre l'erreur quadratique (squared_error) associée à la division à ce point, le nombre d'échantillons qui tombent dans ce nœud (samples), et la valeur moyenne de la durabilité pour ces échantillons (value). Par exemple, le nœud racine a une erreur quadratique d'environ 7.683, avec 49 échantillons aboutissant à ce nœud et une valeur moyenne de durabilité d'environ 5.898.</p>	<p>6. Prédictions Finales :</p> <p>Les valeurs dans les feuilles sont les prédictions finales de l'arbre pour les groupes de données qui suivent les chemins jusqu'à ces points. Elles sont basées sur les séquences de questions et de réponses illustrées par les branches et les nœuds.</p>
<p>En résumé, cet arbre de décision fournit un modèle de prédiction de la durabilité des bâtiments basé sur divers attributs. Le modèle peut être utilisé pour prédire la durabilité d'un nouveau bâtiment en suivant les décisions de l'arbre depuis la racine jusqu'à une feuille, en fonction des caractéristiques du bâtiment.</p>	<p>7. Performance du Modèle :</p> <ul style="list-style-type: none"> • La performance du modèle CCP peut être évaluée par les erreurs quadratiques moyennes indiquées dans chaque feuille. Une erreur faible indique que le modèle fait de bonnes prédictions pour ce groupe de données.
	<p>En résumé, ce graphe représente un modèle d'arbre de décision qui a été optimisé par le CCP pour améliorer la généralisation et réduire le risque de surajustement. Il fournit une vue visuelle des décisions prises par le modèle et montre comment le CCP a éventuellement conduit à un modèle plus simple et plus interprétable.</p>

PHASE 2 : CHAID MODEL

I. CHAID

CHAID (Chi-squared Automatic Interaction Detection) est une technique créée par Gordon V. Kass en 1980. Il s'agit d'une méthode de classification pour construire des arbres de décision en utilisant les statistiques du chi-carré et la valeur de p correspondante afin d'identifier des divisions optimales et de découvrir les relations entre une variable de réponse catégorielle et d'autres variables prédictives catégorielles.

Ce modèle est un outil utilisé pour découvrir des relations entre les variables, une technique d'arbre de décision basée sur un test de signification ajusté (test de Bonferroni). Il divise les répondants en plusieurs groupes en fonction de la relation entre la variable sous-jacente et la variable dépendante, puis subdivise chaque groupe en plusieurs sous-groupes. Il construit un modèle prédictif, ou arbre, pour aider à déterminer comment les variables se combinent le mieux pour expliquer le résultat dans la variable dépendante donnée.

CHAID est utile lors de l'exploration de motifs dans des ensembles de données comportant de nombreuses variables catégorielles. C'est également une manière pratique de résumer les données, car les relations peuvent être facilement visualisées. En fait, comme l'indique le nom de l'algorithme, le critère de base pour la division récursive d'une population hétérogène en groupes homogènes selon les catégories de la variable dépendante (y compris la formation de catégories de variables indépendantes et la sélection de variables indépendantes statistiquement significatives) est la statistique du test du chi-carré. En conséquence, l'algorithme minimise les variations de la variable dépendante au sein des groupes et les maximise entre les groupes.

L'algorithme CHAID peut effectuer différents types d'analyses, tels que la classification (lorsque la variable cible est catégorielle), la régression (lorsque la variable cible est continue et qu'il y a une seule variable cible) et l'analyse multivariée (lorsqu'il y a plus d'une variable cible). Dans la classification, il vise à assigner chaque observation à l'une des classes prédéfinies en fonction des valeurs des variables prédictives. En régression, il est utilisé pour estimer la valeur de la variable cible pour chaque observation. En analyse multivariée, il vise à identifier les relations et interactions entre les variables cibles et les variables prédictives.

C. Decision tree components in CHAID analysis

Dans l'analyse CHAID, les composants de l'arbre de décision sont les suivants :

Nœud racine (Root node) : Le nœud racine contient la variable dépendante ou cible. Par exemple, CHAID est approprié si une banque souhaite prédire le risque de carte de crédit en fonction d'informations telles que l'âge, le revenu, le nombre de cartes de crédit, etc. Dans cet exemple, le risque de carte de crédit est la variable cible, et les autres facteurs sont les variables prédictives.

Nœud parent (Parent's node) : L'algorithme divise la variable cible en deux catégories ou plus. Ces catégories sont appelées nœuds parents ou nœuds initiaux. Dans l'exemple de la banque, les catégories élevée, moyenne et faible sont les nœuds parents.

Nœud enfant (Child node) : Les catégories de variables indépendantes qui se trouvent sous les catégories parentes dans l'arbre d'analyse CHAID sont appelées nœuds enfants.

Nœud terminal (Terminal node) : Les dernières catégories de l'arbre d'analyse CHAID sont appelées nœuds terminaux. Dans l'arbre d'analyse CHAID, la catégorie qui a une influence majeure sur la variable dépendante vient en premier, et la catégorie moins importante vient en dernier. Ainsi, elle est appelée nœud terminal.

D. Chi_square

Le Chi-carré est une mesure statistique visant à trouver la différence entre les nœuds enfants et les nœuds parents. Pour le calculer, nous déterminons la différence entre les effectifs observés et attendus de la variable cible pour chaque nœud, et la somme carrée de ces différences standardisées nous donne la valeur du Chi-carré.

La formule du chi-carré est la suivante :

$$\sum ((y - y')^2 / y'),$$

Où y représente la valeur réelle et y' la valeur attendue.

E. CHAID Algorithm

L'idée centrale de l'algorithme CHAID est de diviser de manière optimale les échantillons en fonction de la variable cible donnée et de l'indice de la caractéristique sélectionnée (comme une variable prédictive), puis de regrouper le tableau de contingence pour juger automatiquement en fonction de la signification du test du chi-carré. La sélection des champs dans l'algorithme CHAID est réalisée en utilisant le test du chi-carré.

a) Sélection des variables cibles pour la catégorisation

La première étape de l'algorithme CHAID consiste à spécifier quelles variables dans l'ensemble de données seront utilisées comme base pour la classification et la segmentation. Ces variables, appelées variables cibles, sont les variables dépendantes ou de réponse que nous souhaitons expliquer ou prédire en utilisant les autres variables de l'ensemble de données. Les variables cibles peuvent être soit catégorielles (nominales ou ordinales) soit continues (intervalles ou ratios). En fonction du type et du nombre de variables cibles, l'algorithme CHAID peut effectuer différents types d'analyses, tels que la classification, la régression ou l'analyse multivariée.

b) Croiser les catégories, créer des tableaux 2D

La deuxième étape de l'algorithme CHAID consiste à effectuer une croix-tabulation pour chaque variable prédictive tour à tour avec la variable cible. Une croix-tabulation est une méthode statistique qui résume la fréquence ou la proportion d'observations qui tombent dans chaque combinaison de catégories de deux variables. Une croix-tabulation peut être représentée par un tableau de contingence bidimensionnel, qui est une matrice affichant les comptages ou pourcentages d'observations dans chaque cellule définie par les variables de ligne et de colonne. La croix-tabulation nous permet d'examiner la relation entre la variable cible et la variable prédictive, ainsi que de tester si la distribution de la variable cible est indépendante de la variable prédictive.

c) Calculer les valeurs du Chi-carré et comparer les valeurs de p

La troisième étape de l'algorithme CHAID consiste à évaluer la signification de la relation entre la variable cible et la variable prédictive pour chaque tableau 2D. La signification est mesurée par le test du chi-carré d'indépendance, qui est basé sur la statistique du chi-carré. La statistique du chi-carré est une valeur numérique qui quantifie la différence entre les fréquences observées et les fréquences attendues dans le tableau 2D. Les fréquences observées sont les décomptes réels des observations dans chaque cellule du tableau, tandis que les fréquences attendues sont les décomptes hypothétiques des observations qui se produiraient si la variable cible et la variable prédictive étaient indépendantes l'une de l'autre.

La formule pour calculer la fréquence attendue pour chaque cellule est la suivante :

$$E_{ij} = NR_i \times C_j / N$$

où E_{ij} is the expected frequency for the cell in the i th row and j th column, R_i is the row total for the i th row, C_j is the column total for the j th column, and N is the grand total of all observations. The formula for calculating the chi-square statistic is:

où E_{ij} est la fréquence attendue pour la cellule de la i -ème ligne et de la j -ème colonne, R_i est le total de la ligne pour la i -ème ligne, C_j est le total de la colonne pour la j -ème colonne, et N est le total global de toutes les observations. La formule pour calculer la statistique du chi-carré est la suivante :

$$X^2 = \sum_{i=1}^r \sum_{j=1}^c \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

où O_{ij} est la fréquence observée et E_{ij} est la fréquence attendue pour la cellule de la i -ème ligne et de la j -ème colonne, r est le nombre de lignes, et c est le nombre de colonnes. La statistique du chi-carré suit une distribution du chi-carré avec $(r-1)(c-1)$ degrés de liberté, où r et c sont le nombre de lignes et de colonnes dans le tableau 2D.

d) Sélectionner la table avec la plus faible valeur de p

La quatrième étape de l'algorithme CHAID consiste à choisir le meilleur séparateur pour le nœud parmi les variables prédictives qui ont été croisées avec la variable cible. Le meilleur séparateur est la variable prédictive qui a la plus petite valeur de p ajustée par rapport à la variable cible. La valeur de p ajustée est obtenue en appliquant une méthode de correction pour tenir compte du problème des tests multiples, qui se produit lorsque nous effectuons plusieurs tests statistiques sur les mêmes données et augmentons la chance de commettre une erreur positive (c'est-à-dire, rejeter une vraie hypothèse nulle). L'algorithme CHAID utilise la correction de Bonferroni, qui divise le niveau alpha original (c'est-à-dire, le niveau de signification) par le nombre de tests effectués, et compare les valeurs de p avec le niveau alpha ajusté. Par exemple, si le niveau alpha original est de 0,05 et que nous effectuons 10 tests, le niveau alpha ajusté est de $0,05/10 = 0,005$, et seules les valeurs de p inférieures ou égales à 0,005 sont considérées comme significatives. La variable prédictive résultante avec la plus petite valeur de p ajustée est sélectionnée comme le meilleur séparateur pour le nœud, et les catégories de la variable prédictive forment les branches ou nœuds enfants du nœud.

e) Utiliser la variable catégorielle associée comme variable de premier niveau

La cinquième étape de l'algorithme CHAID consiste à construire le premier niveau de l'arbre de décision en utilisant le meilleur séparateur pour le nœud qui a été sélectionné à l'étape précédente. Le meilleur séparateur est la variable prédictive qui a la plus petite valeur de p ajustée par rapport à la variable cible, ce qui indique la plus forte association entre les deux variables. La variable prédictive choisie comme meilleur séparateur devient la variable de premier niveau dans l'arbre de décision. Cette variable agit comme la division initiale dans l'arbre, divisant l'ensemble de données en sous-groupes en fonction de ses catégories. Chaque catégorie de la variable prédictive forme une branche ou un nœud enfant du nœud racine, qui contient l'ensemble des données. Les nœuds terminaux sont les nœuds qui n'ont pas d'autres divisions, et ils sont étiquetés avec la classe prédite basée sur la majorité de la variable cible dans le nœud.

f) Répéter le processus, classer jusqu'à ce que la valeur de p dépasse alpha ou que toutes les variables soient classées

La sixième étape de l'algorithme CHAID consiste à appliquer de manière récursive la même procédure à chaque nœud enfant créé par la division précédente, jusqu'à ce que l'arbre soit

pleinement développé ou que les critères d'arrêt soient atteints ; croiser les catégories, calculer les valeurs du chi-carré, choisir la meilleure table et déterminer la variable associée pour la prochaine division. L'itération se poursuit jusqu'à ce que l'une des deux conditions soit remplie : soit la valeur de p dépasse un niveau de signification prédéterminé (alpha), ce qui signifie que la relation n'est plus statistiquement significative, soit toutes les variables ont été classées.

F. Example: CHAID Algorithm

Pour cet exemple concret de modèle CHAID, nous avons sélectionné un ensemble de données comprenant différentes caractéristiques telles que la volatilité, le ratio cours/bénéfice (P/E), la croissance économique, le rendement des obligations à 10 ans moins le rendement des obligations à 2 ans (10Y 2Y Yield), et la direction du marché. L'objectif est de déterminer la caractéristique la plus pertinente par rapport à la direction du marché afin de créer un arbre de décision.

Nous débutons le processus en examinant la caractéristique de la volatilité. Cette caractéristique comporte deux classes : haute et basse. Pour chaque classe, nous calculons les valeurs du test du chi-deux en comparant les décisions observées (UP ou DOWN) avec les décisions attendues. Ces valeurs sont ensuite utilisées pour déterminer la pertinence de la volatilité par rapport à la direction du marché.

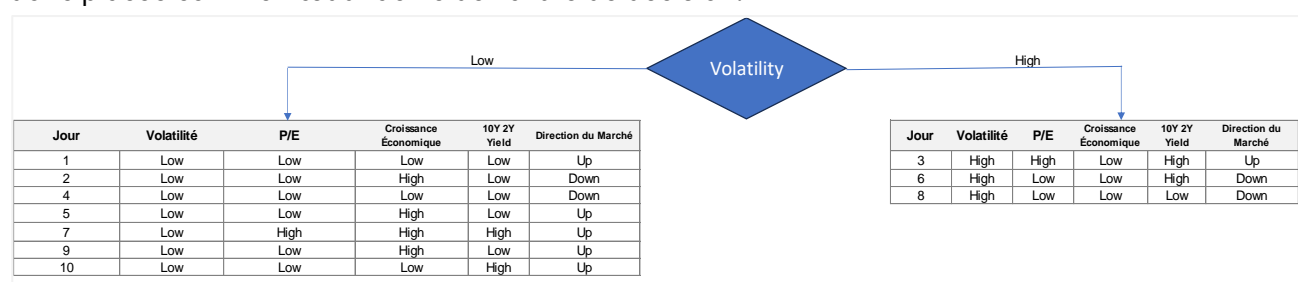
1 Volatilité

	Up	Down	Total	Expected	Chi-square - Yes	Chi-square - No
High	1	2	3	1.5	0.408	0.408
Low	5	2	7	3.5	0.802	0.802
Total					2.420	

Par exemple, si la volatilité est élevée, la valeur du test du chi-deux est calculée comme $\chi^2 = \frac{(1 - 1.5)^2}{1.5} = 0.408$, où 1 est la valeur observée et 1.5 est la valeur attendue. Ce processus est répété pour chaque instance, et les valeurs du test du chi-deux sont sommées pour obtenir le chi-carré total pour la volatilité.

Feature	Chi-square
Volatilité	2.420
P/E	2.000
Croissance économique	1.414
10Y 2Y Yield	1.414

Nous répétons ce processus pour chaque caractéristique, calculant ainsi le chi-carré total pour chacune. Dans cet exemple, nous observons que la colonne de volatilité présente le chi-carré le plus élevé, indiquant qu'elle est la caractéristique la plus importante. Cette caractéristique est donc placée comme nœud racine de l'arbre de décision.



Dans cette branche spécifique du modèle CHAID, nous continuons à évaluer la pertinence des différentes caractéristiques en calculant le test du chi-deux pour les variables P/E, la croissance économique et le rendement des obligations à 10 ans moins le rendement des obligations à 2 ans (10Y 2Y Yield). Ce processus nous permet de déterminer laquelle de ces caractéristiques est la plus significative pour la direction du marché.

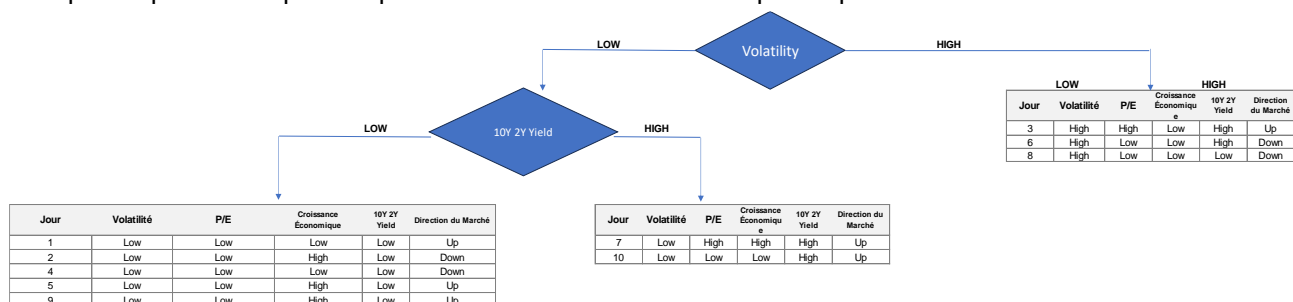
2 P/E

	Up	Down	Total	Expected	Chi-square - Yes	Chi-square - No
High	1	0	1	0.5	0.707	0.707
Low	4	2	6	3	0.577	0.577
Total					2.569	

Pour la caractéristique P/E, qui présente deux classes (Low et High), nous calculons les valeurs du test du chi-deux pour chaque classe en comparant les décisions observées (UP ou DOWN) avec les décisions attendues. Ces valeurs sont ensuite sommées pour obtenir le chi-carré total pour la caractéristique P/E qui est de 2.569.

Feature	Chi-square
P/E	2.569
Croissance économique	2.231
10Y 2Y Yield	2.632

De manière similaire, nous répétons ce processus pour la croissance économique et le 10Y 2Y Yield, évaluant ainsi la contribution de chaque caractéristique à la direction du marché. Dans cet exemple, le 10Y 2Y Yield émerge avec le chi-carré le plus élevé parmi les trois caractéristiques, indiquant qu'il est le plus important dans cette branche spécifique de l'arbre de décision.



En plaçant le 10Y 2Y Yield comme nœud racine de cette branche, nous continuons à construire l'arbre de décision de manière itérative, en identifiant les caractéristiques les plus pertinentes à chaque étape pour déterminer la direction du marché. Cette approche dynamique et basée sur des tests statistiques contribue à la création d'un modèle CHAID informatif et adapté aux caractéristiques spécifiques de l'ensemble de données.

Le processus se répète à chaque étape suivante, où la caractéristique avec le chi-carré le plus élevé est choisie comme nœud de division. Ce processus itératif conduit à la création d'un arbre de décision qui prend en compte les relations significatives entre les caractéristiques et la direction du marché. *Voir la figure 6.*

II. Comparaison de CHAID et CART

CHAID (Chi-square Automatic Interaction Detector)

Le modèle CHAID repose sur le principe de l'optimisation locale, privilégiant l'indépendance des nœuds. Il utilise le test du chi-carré pour identifier les variables indépendantes ayant le plus grand impact sur la variable dépendante. Contrairement à CART, CHAID génère un arbre multi-branché, créant un nœud distinct pour chaque catégorie d'une variable indépendante.

Un aspect clé du CHAID est son approche pré-élagage, une méthode de taille avant la division. Cela réduit le temps d'entraînement et de test, minimisant ainsi le risque de surajustement. Bien que le pré-élagage puisse initialement réduire les performances de généralisation, il contribue à une meilleure adaptation du modèle aux données.

Dans la Figure 1, le chi-carré est utilisé pour mettre en évidence la corrélation entre les caractéristiques et l'intention d'achat de voiture. Cette méthodologie permet de quantifier la force de la relation entre chaque caractéristique et la variable cible, offrant ainsi des informations précieuses pour la prédiction.

CART (Classification and Regression Tree)

Contrairement à CHAID, CART se concentre sur l'optimisation globale, utilisant des critères d'impureté tels que le coefficient de Gini. Le processus de segmentation est basé sur la sélection du meilleur test pour chaque nœud, résultant en un arbre binaire. Contrairement à CHAID, CART adopte la méthode de post-élagage, encourageant la croissance maximale de l'arbre avant d'élager les nœuds.

La Figure 2 illustre l'utilisation du coefficient Gini pour évaluer l'importance des caractéristiques. Cependant, la répétition des termes peut rendre difficile une évaluation intuitive de l'importance relative de chaque caractéristique.

CART a l'avantage de traiter les données manquantes en cherchant des alternatives, tandis que CHAID considère les valeurs manquantes comme une catégorie distincte. Cela donne à CART une certaine flexibilité dans la gestion des données manquantes, ce qui peut être crucial dans des situations réelles où les données ne sont pas toujours complètes.

PHASE 3 : Mise en pratique

Dans cette phase pratique, notre objectif est de construire un modèle de classification visant à prédire la direction du marché américain, à savoir s'il sera haussier ou baissier durant le mois suivant. Pour ce faire, nous avons choisi 9 variables explicatives pertinentes, chacune fournissant des informations cruciales sur différents aspects de l'économie américaine. Ces variables sont :

VIX (CBOE Volatility Index) :

Le VIX mesure la volatilité implicite du marché financier, reflétant la perception des investisseurs quant au niveau de risque.

10Y2Y (10-Year Treasury Minus 2-Year Treasury):

Cette variable représente la différence entre les rendements des obligations du gouvernement américain à 10 ans et à 2 ans. Elle offre des indications sur les attentes du marché en termes d'inflation et de rendements des investissements à long terme.

MS (M2 Money Supply) :

La masse monétaire M2 englobe la monnaie en circulation, les dépôts bancaires et certains instruments financiers. Son évolution peut influencer l'inflation et la santé de l'économie.

CPI (Consumer Price Index) :

Le CPI mesure la variation des prix des biens et services consommés par les ménages américains. Il est un indicateur clé de l'inflation et de la stabilité économique.

UMCS (University of Michigan Consumer Sentiment):

Le sentiment des consommateurs, tel que mesuré par l'indice de l'Université du Michigan, offre des perspectives sur la confiance des consommateurs, qui peut affecter les décisions d'achat et l'activité économique globale.

NFP (Total Nonfarm) :

Le nombre total d'emplois non agricoles fournit un indicateur essentiel de la santé du marché du travail aux États-Unis. Les variations dans ce chiffre peuvent influencer la confiance des investisseurs et la direction du marché.

HS (Housing Starts) :

Les données sur les mises en chantier de logements aux États-Unis sont indicatives de l'activité dans le secteur immobilier, offrant des insights sur la santé économique globale.

PMI (Purchasing Manufacturing Index) :

L'indice PMI mesure l'activité manufacturière et peut signaler les changements dans la croissance économique. Une valeur supérieure à 50 suggère une expansion.

PE (Shiller PE Ratio) :

Le Shiller PE Ratio évalue la valorisation du marché boursier en comparant le cours des actions à leurs bénéfices sur une moyenne mobile sur 10 ans. Il peut aider à identifier les périodes de surévaluation ou de sous-évaluation du marché.

Notre approche de modélisation impliquera la mise en place de trois modèles distincts : CHAID, Random Forest, et CART. Chacun de ces modèles sera évalué en fonction de ses performances, de sa capacité à gérer les données manquantes, et de ses caractéristiques d'optimisation.

Cette démarche méthodique permettra d'établir un modèle robuste de prédiction de la direction du marché américain, intégrant des variables économiques clés pour une prise de décision éclairée.

I. Modèle Random Forest

Dans cette phase de construction du deuxième modèle, le Random Forest, nous disposons de neuf variables explicatives (features) significatives, notamment le VIX, 10Y2Y, MS, CPI, UMCS, NFP, HS, PMI, et PE. Ces variables captent divers aspects de l'économie américaine, allant de la volatilité du marché financier à la confiance des consommateurs, en passant par des indicateurs clés tels que l'emploi, l'inflation, et la valorisation du marché boursier.

La variable cible que nous cherchons à prédire est la direction du S&P 500, catégorisée comme "Up" (haussière) ou "Down" (baissière). Cette orientation du marché servira de référence pour évaluer l'efficacité de notre modèle Random Forest dans la prévision de tendances sur le marché financier américain.

a) Le Choix des Variables :

Le choix des variables explicatives est fondamental pour la performance du modèle. Chacune des 9 variables a été sélectionnée en raison de son importance potentielle dans l'anticipation des mouvements du marché. Le VIX reflète la volatilité, le 10Y2Y offre des informations sur les anticipations des taux d'intérêt, le MS mesure la masse monétaire etc. Leur inclusion vise à fournir une représentation complète des facteurs économiques influençant le marché.

b) Prétraitement des Données :

Dans le contexte du modèle Random Forest, le prétraitement des données est simplifié par la nature de l'algorithme. Étant robuste aux valeurs aberrantes et capable de gérer les données manquantes, nous pouvons contourner ces étapes. De plus, la normalisation des données n'est pas requise car le Random Forest n'est pas sensible à l'échelle des variables.

La variable cible, la direction du S&P 500, est déjà catégorielle, simplifiant davantage le processus. Les données, s'étalant de 1995 à la période actuelle avec une fréquence mensuelle, offrent une base temporelle robuste pour la construction du modèle.

Sur l'ensemble des données, une brève visualisation de la fréquence des mouvements haussiers et baissiers du S&P 500 a été réalisée pour une meilleure compréhension de la distribution des classes.

Voir Figure 7 : Direction du S&P500

c) Division des Données :

Après l'importation des variables dans Python, une description statistique préliminaire et une vérification de l'absence de valeurs manquantes ont été effectuées. Par la suite, les données ont été divisées en un ensemble d'entraînement (80%) et un ensemble de test (20%). Cette division permettra au modèle Random Forest de s'entraîner sur une partie des données et d'être évalué sur des données non vues pour évaluer sa capacité de généralisation. Le modèle n'aura pas connaissance de l'ensemble de test pendant son processus d'apprentissage, garantissant une évaluation impartiale de ses performances.

d) Construction du Modèle Random Forest :

La construction du modèle Random Forest implique plusieurs étapes clés, allant du choix des hyperparamètres à l'ajustement du modèle sur les données d'entraînement, jusqu'à l'évaluation des performances sur l'ensemble de test.

1. Library RandomForestClassifier :

La bibliothèque utilisée pour construire le modèle Random Forest est RandomForestClassifier, une implémentation disponible dans la bibliothèque scikit-learn en Python. Scikit-learn fournit

une interface conviviale pour construire, entraîner et évaluer des modèles d'apprentissage automatique.

2. Hyperparamètres :

Les hyperparamètres, essentiels dans la configuration du modèle RandomForestClassifier, sont des paramètres du modèle qui ne sont pas appris à partir des données, mais doivent être spécifiés avant l'entraînement. Dans une première phase, nous avons choisi d'utiliser les hyperparamètres par défaut fournis par la bibliothèque de l'algorithme, offrant ainsi une base de référence pour notre modèle Random Forest.

Un hyperparamètre clé de RandomForestClassifier est le nombre d'estimateurs (n_estimators), représentant le nombre d'arbres dans la forêt. Dans l'exemple donné, nous avons fixé n_estimators=100, ce qui signifie que notre forêt est composée de 100 arbres de décision. Cette valeur peut être ajustée en fonction des besoins spécifiques du problème.

De plus, dans la création du modèle Random Forest, d'autres hyperparamètres tels que le critère de division (criterion), la profondeur maximale des arbres (max_depth), le nombre minimal d'échantillons requis pour diviser un nœud interne (min_samples_split), le nombre minimal d'échantillons requis pour être une feuille (min_samples_leaf), et le nombre maximal de variables à considérer pour la meilleure division (max_features) peuvent être spécifiés. Dans l'exemple ci-dessous, ces hyperparamètres sont explicitement définis :

```
# Create a Random Forest Classifier
rf = RandomForestClassifier(n_estimators = 100,
                           criterion = "gini",
                           max_depth = None,
                           min_samples_split = 2,
                           min_samples_leaf = 1,
                           max_features = "sqrt",
                           )
```

3. Choix du Splitting :

Le Random Forest utilise un processus de bagging (Bootstrap Aggregating) pour construire plusieurs arbres à partir de sous-ensembles aléatoires des données d'entraînement. Chaque arbre est construit sur un échantillon bootstrap des données (avec remplacement), et à chaque nœud, un sous-ensemble aléatoire de variables explicatives est considéré pour la division.

Cette approche de splitting aléatoire permet de réduire la corrélation entre les arbres, améliorant ainsi la performance globale du modèle. Cela rend le modèle plus robuste et moins susceptible de surajuster aux particularités des données d'entraînement (overfitting).

4. Ajustement du Modèle sur les Données d'Entraînement :

Une fois le modèle créé, il est ajusté sur les données d'entraînement à l'aide de la méthode fit. Cela implique que chaque arbre de la forêt est construit en utilisant un échantillon bootstrap des données, et le modèle apprend à partir de ces sous-ensembles pour optimiser les prédictions sur l'ensemble d'entraînement.

5. Prédiction sur l'Ensemble de Test :

Après l'entraînement, le modèle est évalué sur l'ensemble de test. Les prédictions sont générées à l'aide de la méthode predict.

6. Évaluation des Performances :

Enfin, les performances du modèle sont évaluées à l'aide de métriques telles que l'accuracy. L'accuracy mesure la proportion de prédictions correctes par rapport à l'ensemble de test.

La matrice de confusion fournit une vue détaillée des performances d'un modèle de classification en comparant les prédictions du modèle avec les vraies classes de l'ensemble de test.

Voir Figure 9 : Matrix de confusion

On remarque que le modèle a du mal à détecter les mois baissiers de manière satisfaisante, comme indiqué par le nombre plus élevé de faux positifs (20) par rapport aux faux négatifs (8). Cela suggère que le modèle a une tendance à surestimer les périodes haussières, ce qui peut être dû à une variété de facteurs tels que la complexité de la dynamique du marché ou le besoin d'ajustements dans les hyperparamètres du modèle.

L'arbre de décision généré par le modèle offre un aperçu visuel de la logique suivie par l'algorithme pour prendre des décisions. Chaque nœud de l'arbre représente une condition sur une variable explicative, et chaque feuille représente une prédiction.

Voir Figure 10 : L'arbre issu du modèle Random Forest

7. Analyse des Importances des Variables :

Après l'entraînement du modèle Random Forest, il est essentiel d'analyser l'importance de chaque variable explicative dans la prise de décision du modèle. Cette analyse fournit des informations sur les caractéristiques qui ont le plus contribué aux prédictions du modèle. Voici la figure suivante qui en résulte :

Voir Figure 8 : Visualisation de l'importance des variables explicatives

On observe que le VIX (CBOE Volatility Index) affiche le score d'importance le plus élevé dans le modèle Random Forest. Cette observation est significative, car l'importance d'une variable dans le contexte du Random Forest est évaluée en fonction de la contribution de cette variable aux décisions prises par l'ensemble des arbres de la forêt. La raison derrière le score élevé d'importance du VIX peut être attribuée à son rôle crucial en tant qu'indicateur de la volatilité implicite du marché financier. Une volatilité plus élevée peut indiquer une période d'incertitude ou de turbulence sur les marchés, ce qui peut influencer les mouvements du S&P 500.

Le sentiment des consommateurs, représenté par l'indice de l'Université du Michigan (UMCS), est également attribué à un score d'importance significatif.

e) Optimisation des Hyperparamètres par GridSearchCV :

```
# Define the parameter grid for hyperparameter tuning
param_grid = {
    'n_estimators': [50, 100, 120, 150, 200],
    'max_depth': [15, 20, 25, 30],
    'min_samples_split': [1, 2],
    'max_features': [2, 4, 6],
    'min_samples_leaf': [1, 2]
}

# Create a Random Forest Classifier
rf_classifier = RandomForestClassifier(random_state=0)

# Create a GridSearchCV object
grid_search = GridSearchCV(rf_classifier, param_grid, cv=10, scoring='accuracy')
```

Dans cette étape cruciale, nous avons cherché à optimiser les performances du modèle Random Forest en ajustant ses hyperparamètres. L'utilisation de la méthode GridSearchCV de scikit-learn a permis une exploration systématique d'un espace prédéfini d'hyperparamètres afin de trouver la combinaison la plus performante. Voici une décomposition détaillée de cette procédure :

La première étape consiste à définir une grille des paramètres à explorer. Dans notre cas, nous avons spécifié différentes valeurs pour des hyperparamètres clés tels que le nombre d'estimateurs (n_estimators), la profondeur maximale de l'arbre (max_depth), le nombre minimum d'échantillons requis pour diviser un nœud interne (min_samples_split), et le nombre minimum d'échantillons requis pour être une feuille (min_samples_leaf).

Nous avons instancié un classifieur Random Forest avec des paramètres par défaut. Ce classifieur sera utilisé comme base pour l'optimisation des hyperparamètres.

L'objet GridSearchCV est créé en prenant comme arguments le classifieur Random Forest, la grille des paramètres à explorer, le nombre de plis pour la validation croisée (cv=10), et la métrique de scoring (scoring='accuracy'). La validation croisée est une étape cruciale qui permet d'évaluer la performance du modèle sur différentes partitions de l'ensemble d'entraînement, assurant ainsi une évaluation robuste.

En ajustant l'objet GridSearchCV sur l'ensemble d'entraînement, le processus de recherche des meilleures combinaisons d'hyperparamètres est lancé. La validation croisée est effectuée pour chaque combinaison, évaluant ainsi la performance du modèle de manière exhaustive.

Une fois la recherche terminée, les meilleurs hyperparamètres sont extraits à l'aide de l'attribut best_params_ de l'objet GridSearchCV. Ces valeurs représentent la configuration qui a donné les meilleures performances sur l'ensemble d'entraînement.

Le modèle avec les meilleurs hyperparamètres est ensuite utilisé pour faire des prédictions sur l'ensemble de test. La métrique d'accuracy est calculée pour évaluer la précision du modèle sur cet ensemble.

Malgré l'optimisation, l'accuracy demeure à 0.528, soulignant les défis inhérents à la prédiction des mouvements du marché financier. Cette modestie des résultats incite à une réflexion continue et à l'exploration d'améliorations potentielles du modèle pour mieux saisir la complexité dynamique du marché. L'optimisation des hyperparamètres représente une étape cruciale dans cette quête d'amélioration constante des performances du modèle.

II. Modèle CART

1. Préparation et Séparation des Données.

Nous préparé l'ensemble de données en définissant les caractéristiques (X) pour les Variables Explicatives et (Y) pour la variable Expliqué ou cible, presumant la tendance de l'indice S&P 500.

Nous avons divisé les données en ensembles d'entraînement test, en maintenant une distribution de 80/20. Soit 80% Entraînement et 20% Test. **Les résultats sont dans la capture d'image ci-après →**

```
X_train shape: (276, 9)
X_test shape: (70, 9)
y_train shape: (276,)
y_test shape: (70,)
```

et de

2. Entraînement du Modèle.

Le modèle a été entraîné avec des hyperparamètres spécifiques et déterminants comme **la profondeur maximale de l'arbre (max_depth)** et **le nombre minimum d'échantillons par feuille (min_samples_leaf)**. La recherche des meilleurs hyperparamètres est réalisée via **GridSearchCV** qui utilise **la validation croisée pour évaluer les performances sur les données d'entraînement**. Les meilleurs paramètres résultants indiquent un modèle pas trop complexe pour éviter le surajustement, visant une meilleure généralisation sur les données de test. **Image ci-après →**

```
print("Meilleurs hyperparamètres trouvés:")  
print(best_params)
```

```
Meilleurs hyperparamètres trouvés:  
{ 'max_depth': 3, 'min_samples_leaf': 6 }
```

3. Évaluation du Modèle

L'évaluation du modèle formé a été réalisée en utilisant des métriques de performance telles que **la précision, le rappel et le score F1** pour différentes classes (Hausse ou Baisse). Ces métriques fournissent un aperçu de l'exactitude du modèle et de sa capacité à classer correctement la direction de la tendance du S&P 500.

4. Visualisation de l'Arbre de Décision

Nous Observons que chaque nœud dans l'arbre représente un point de décision, et cette représentation visuelle aide à comprendre la logique derrière les prédictions du modèle. *(Bien vouloir consulter le Fichier Python SVP)*

5. Métriques de Performance.

Les meilleurs hyperparamètres trouvés (profondeur maximale : 3, nombre minimum d'échantillons par feuille : 6) et les métriques de performance du modèle (**précision, le rappel et le score F1**).

La précision est d'environ 58 %, ce qui pourrait être considéré comme modeste dans le domaine difficile de la prévision des tendances financières.

```
Classe 0:  
precision: 0.5161  
recall: 0.5333  
f1-score: 0.5246  
support: 30.0000
```

```
Classe 1:  
precision: 0.6410  
recall: 0.6250  
f1-score: 0.6329  
support: 40.0000  
accuracy: 0.5857
```

```
Classe macro avg:  
precision: 0.5786  
recall: 0.5792  
f1-score: 0.5788  
support: 70.0000
```

```
Classe weighted avg:  
precision: 0.5875  
recall: 0.5857  
f1-score: 0.5865  
support: 70.0000
```

Conclusions :

Les indicateurs économiques comme **PMI, VIX, CPI et PE** s'avèrent pertinents pour prévoir les tendances boursières. Le VIX se distingue comme mesure fiable de la volatilité du marché.

Avec une précision de 58.57%, le modèle est un choix aléatoire, bien qu'il reste perfectible. La balance des prédictions suggère une performance homogène à travers les catégories.

Pour renforcer le modèle, il est envisageable de complexifier l'arbre décisionnel et d' étoffer les données. Ce modèle peut servir à l'analyse prédictive ou être intégré à des stratégies prédictives composites. Ce modèle CART, avec des ajustements, pourrait ainsi devenir un outil financier pratique.

III. Modèle CHAID (sur SPSS)

L'analyse CHAID (Chi-squared Automatic Interaction Detection) est une méthode d'analyse de données qui se distingue par sa capacité à construire des arbres de décision complexes basés sur des tests statistiques de chi-deux.

L'analyse CHAID (Chi-squared Automatic Interaction Detection) est une méthode d'analyse de données qui se distingue par sa capacité à construire des arbres de décision complexes basés sur des tests statistiques de chi-deux. Cette technique, implantée dans SPSS (Statistical Package for the Social Sciences), offre une approche puissante pour explorer les relations complexes entre les variables dans un ensemble de données.

Le processus de création d'un modèle CHAID sur SPSS commence par la sélection d'une variable cible, souvent catégorielle, que l'on cherche à expliquer ou prédire. Le modèle cherche ensuite à identifier les variables prédictives qui présentent des relations significatives avec la variable cible.

Le principal avantage de CHAID réside dans sa capacité à gérer des ensembles de données avec des variables catégorielles, offrant ainsi une solution robuste pour des problèmes complexes de classification. Contrairement à d'autres méthodes d'arbre de décision, CHAID ne se limite pas aux variables binaires, ce qui élargit son champ d'application.

Voir la figure 14.

La première étape cruciale de notre processus consistait à valider le modèle CHAID. Cette validation a été effectuée en procédant à une partition en échantillons de notre base de données. En divisant notre ensemble de données en deux ensembles distincts, à savoir un ensemble d'apprentissage représentant 70% des données et un ensemble de test représentant 30%, nous avons établi une base solide pour évaluer la performance du modèle. Cette approche garantit une évaluation rigoureuse du modèle sur des données non vues, permettant ainsi de mieux généraliser ses performances à des situations réelles. Expliquons maintenant la logique et la justification derrière ce choix.

Voir la figure 15.

Lors de la conception du modèle CHAID dans SPSS, nous avons soigneusement choisi les critères pour assurer une construction optimale du modèle. En fixant un nombre minimal d'observations par nœud à 3 pour le nœud parent et 2 pour les nœuds enfants, nous visons à garantir que chaque division du modèle est soutenue par un nombre suffisant d'observations. Cette décision repose sur le principe que des nœuds bien définis contribuent à une meilleure précision du modèle en évitant des subdivisions trop petites qui pourraient conduire à une instabilité ou à une suradaptation du modèle aux données d'apprentissage.

Parallèlement, nous avons fixé une profondeur maximale de 20 pour limiter la complexité du modèle. Limiter la profondeur contribue à éviter la création de branches excessivement nombreuses, ce qui pourrait conduire à une surcomplexité et nuire à la généralisation du modèle sur de nouvelles données. En maintenant une profondeur maximale raisonnable, nous cherchons à équilibrer la capacité du modèle à capturer des relations complexes tout en évitant un ajustement excessif aux particularités des données d'apprentissage.

Voir la figure 16.

Dans la configuration des hyperparamètres du modèle CHAID, nous avons pris des décisions stratégiques pour garantir la fiabilité et la pertinence des résultats. Le niveau de signification pour les nœuds de division a été fixé à 0.1, indiquant que seules les divisions ayant un impact statistiquement significatif sur le modèle seront considérées. Cette valeur joue un rôle essentiel

dans la détermination de la pertinence des décisions prises par le modèle lors de la construction de l'arbre de décision.

De même, le niveau de signification pour la fusion des catégories a été fixé à 0.05, suggérant que seules les fusions présentant une différence statistiquement significative seront retenues. Ce paramètre assure une attention particulière à la simplification du modèle, en favorisant la fusion des catégories qui ne présentent pas de différences substantielles en termes de contribution à la variable cible.

Le test statistique du khi-deux avec le rapport de vraisemblance a été utilisé comme métrique de base pour évaluer la pertinence des divisions et fusions. Ce choix repose sur sa sensibilité aux différences entre les distributions observées et attendues, faisant de lui un outil robuste pour détecter des relations significatives dans les données.

La méthode Bonferroni a été adoptée pour ajuster les valeurs de signification, un processus crucial pour contrôler le taux d'erreur global. En ajustant les seuils de signification, la méthode Bonferroni renforce la validité des résultats en minimisant les risques associés aux tests multiples, assurant ainsi une interprétation plus fiable des conclusions du modèle.

Voir la figure 17.

Notre approche pour le traitement des variables dans le modèle CHAID s'est concentrée sur la création d'intervalles pertinents afin de mieux représenter la complexité des relations entre les variables. Pour chaque variable, nous avons soigneusement sélectionné des divisions appropriées, contribuant ainsi à une représentation plus précise des schémas de comportement au sein du modèle.

En regroupant les variables en intervalles significatifs, nous cherchons à capturer de manière optimale les variations subtiles au sein des données. Cette stratégie permet au modèle CHAID d'appréhender les nuances dans les relations entre les variables, renforçant ainsi sa capacité à générer des divisions et des règles de décision plus informatives.

Le choix de divisions spécifiques pour chaque variable a été guidé par la recherche d'une balance entre la granularité nécessaire pour capturer les variations significatives et la simplicité requise pour garantir une interprétation aisée des résultats. Ainsi, chaque variable a été traitée individuellement pour garantir une représentation fidèle de sa contribution au modèle global.

Figure 1: Resultat du modèle CHAID

Classification				
Echantillon	Observé	Prévisions		
		DOWN	UP	Pourcentage correct
Apprentissage	DOWN	37	49	43,0%
	UP	15	142	90,4%
	Pourcentage global	21,4%	78,6%	73,7%
Test	DOWN	13	26	33,3%
	UP	10	54	84,4%
	Pourcentage global	22,3%	77,7%	65,0%
Méthode de croissance : CHAID				
Variable dépendante : SP500				

Performance du Modèle CHAID sur l'Ensemble d'Apprentissage :

Lorsqu'on plonge dans les méandres de l'ensemble d'apprentissage, le modèle CHAID démontre une certaine compétence, bien que nuancée, dans la prédiction des mouvements du marché. En ce qui concerne les périodes de baisse, le modèle s'est avéré être un oracle modeste avec une précision de prédiction de 43,0%. Cela indique une capacité du modèle à percevoir certains des signes annonciateurs des phases de déclin, bien que cette capacité soit encore perfectible.

Cependant, la vraie prouesse du modèle émerge lorsqu'il est confronté aux embruns de la hausse du marché. Dans ce domaine, le modèle CHAID se transforme en un visionnaire éclairé, capturant avec une précision étonnante 90,4% des mouvements à la hausse. Ces résultats optimistes suggèrent que le modèle peut capter efficacement les schémas et tendances associés aux phases de croissance du marché.

Performance du Modèle CHAID sur l'Ensemble de Test :

Lorsque le modèle CHAID est soumis à l'épreuve du feu de l'ensemble de test, la nature de ses prédictions se dessine de manière plus contrastée. Lors des phases de baisse, le modèle ne maintient qu'une précision de 33,3%, montrant une vulnérabilité à anticiper les périodes de déclin sur de nouveaux ensembles de données. Néanmoins, les résultats ne sont pas sans éclat, soulignant que le modèle peut encore identifier un tiers des mouvements à la baisse.

C'est dans les moments de hausse que le modèle CHAID conserve sa brillance, prédisant avec assurance 84,4% des mouvements à la hausse sur l'ensemble de test. Cette constance dans la capacité à anticiper les périodes de croissance est un point fort du modèle.

Le modèle CHAID, bien qu'ayant ses forces et faiblesses, se révèle être un outil puissant dans la prédiction des mouvements du marché. Ses performances élevées dans la prévision des hausses du marché suggèrent un potentiel d'application dans des contextes d'investissement où la croissance est cruciale. Cependant, des ajustements pour améliorer la détection des périodes de déclin pourraient être nécessaires pour garantir une performance équilibrée dans toutes les conditions du marché. [Voir l'arbre sur la figure 19.](#)

Conclusion générale

La conclusion de notre étude comparative entre les modèles CHAID (Chi-squared Automatic Interaction Detection) et Random Forest révèle des nuances intrigantes dans leurs performances respectives. Les résultats obtenus mettent en lumière la complexité des choix en matière de modélisation et soulignent l'importance de sélectionner le modèle le mieux adapté à chaque situation.

Tout d'abord, l'Accuracy de Random Forest, établie à 0.54, offre une perspective intéressante sur sa capacité à généraliser les données. Cependant, cette performance doit être interprétée avec prudence, car elle peut être influencée par divers facteurs tels que la taille de l'échantillon et la complexité des relations entre les variables. D'un autre côté, le modèle CHAID a affiché une Accuracy supérieure, atteignant 0.65, suggérant une meilleure adéquation aux spécificités du jeu de données.

Lorsqu'on se plonge dans les raisons de ces différences, on observe que Random Forest, en agrégeant les prédictions de multiples arbres de décision, peut souffrir de surajustement si la diversité entre les arbres est insuffisante. Parfois, le modèle peut être trop complexe pour des ensembles de données plus petits, conduisant à une baisse de la précision.

D'un autre côté, le modèle CHAID, en se basant sur le test du khi-deux pour créer des arbres de décision, peut mieux s'adapter à des jeux de données avec des relations plus simples et des

structures claires. Son approche intuitive de construction d'arbres basée sur des seuils de significativité statistique lui confère une robustesse particulière dans certains contextes.

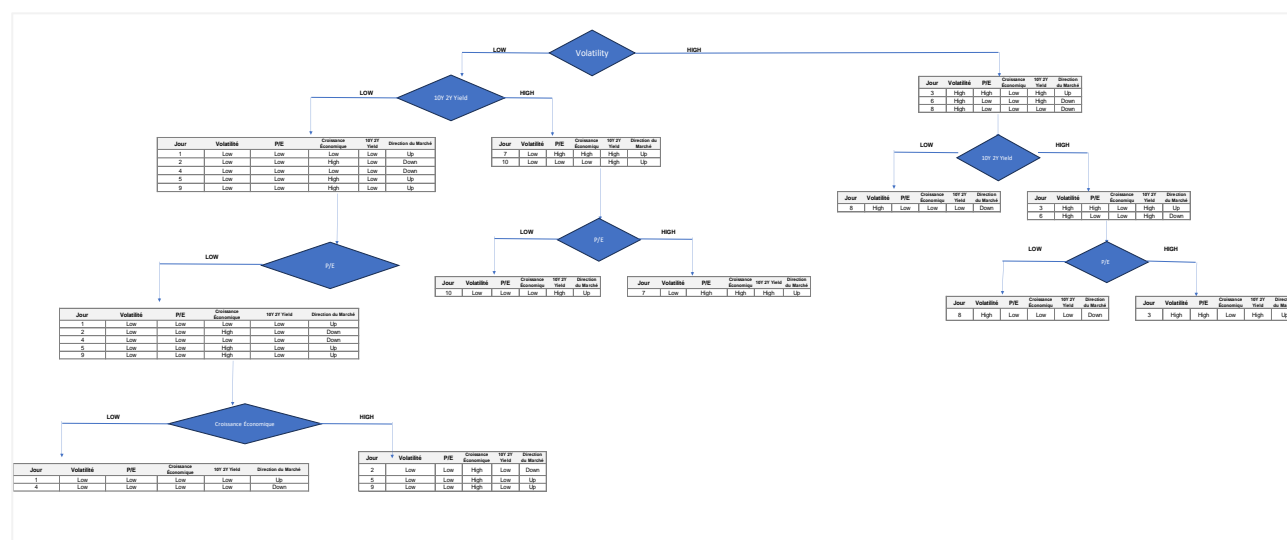
En somme, le choix entre CHAID et Random Forest dépend largement de la nature des données, de la complexité des relations entre les variables et des objectifs spécifiques de la modélisation. Alors que Random Forest excelle dans la gestion de la complexité, CHAID brille dans des situations où des relations plus claires sont nécessaires. Le chercheur doit donc prendre en considération ces nuances pour choisir le modèle qui servira au mieux les besoins de son analyse.

APPENDIX:

Table 1: Variables Explicatives

Code	Indicator
VIX	CBOE Volatility Index: VIX, Index, Monthly, Not Seasonally Adjusted
10Y2Y	10-Year Treasury Constant Maturity Minus 2-Year Treasury Constant Maturity, Percent, Monthly, Not Seasonally Adjusted
MS	M2, Percent Change from Year Ago, Monthly, Seasonally Adjusted
CPI	Consumer Price Index for All Urban Consumers: All Items in U.S. City Average, Percent Change from Year Ago, Monthly, Seasonally Adjusted
UMCS	University of Michigan: Consumer Sentiment, Percent Change from Year Ago, Monthly, Not Seasonally Adjusted
NFP	All Employees, Total Nonfarm, Percent Change from Year Ago, Monthly, Seasonally Adjusted
HS	New Privately-Owned Housing Units Started: Total Units, Percent Change from Year Ago, Monthly, Seasonally Adjusted Annual Rate
PMI	ISM Purchasing Manufacturing Index
PE	Shiller PE Ratio

Figure 2: Exemple CHAID



Phase III:

Random forest

Figure 3: Direction du S&P500

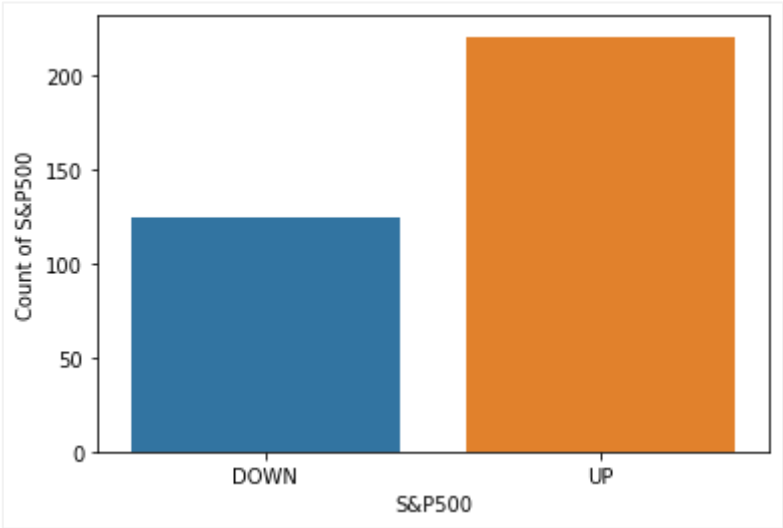


Figure 4: Visualisation de l'importance des variables explicatives

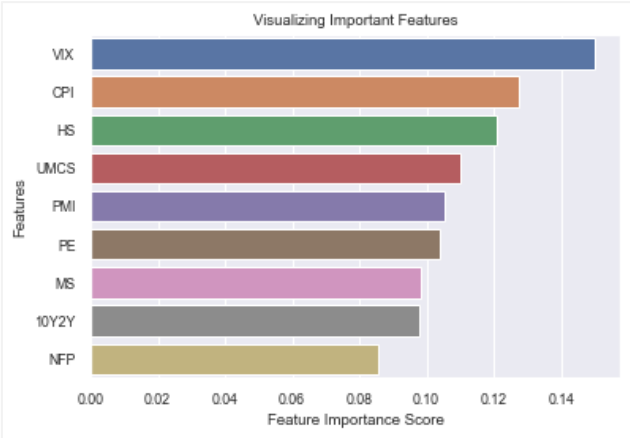


Figure 5: Matrix de confusion

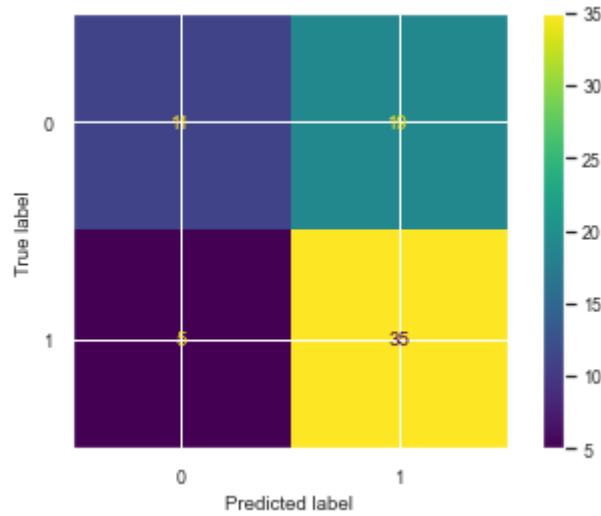
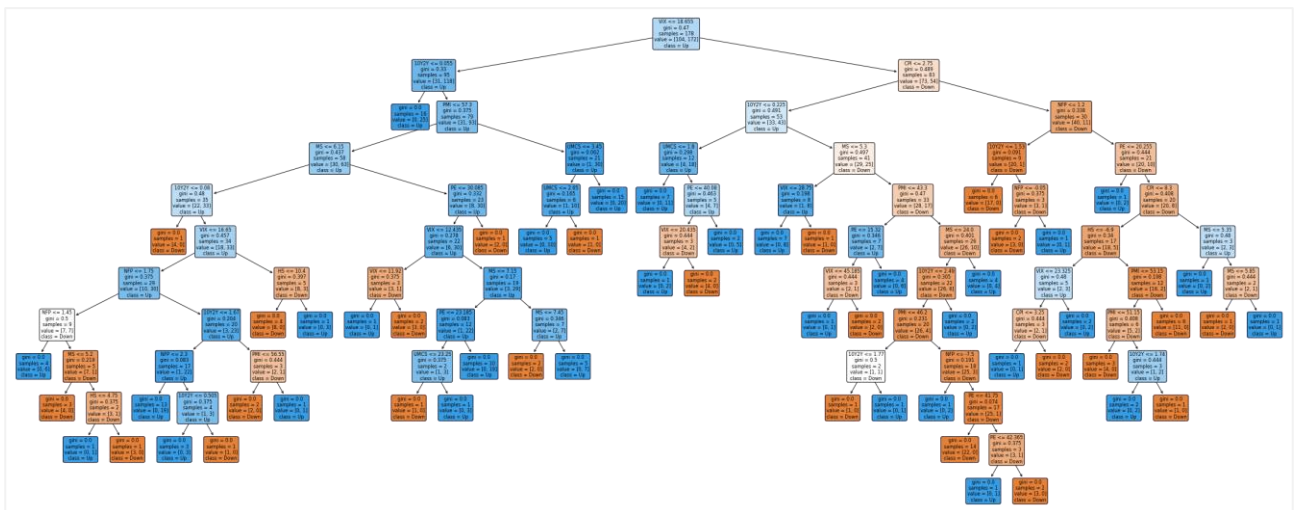


Figure 6: L'arbre issu du modèle Random Forest



Random forest with cross validation

Figure 7: Matrix de confusion

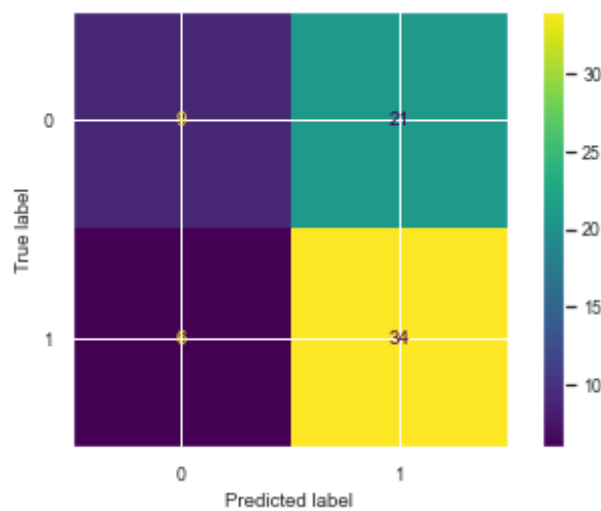


Figure 8: Visualisation de l'importance des variables explicatives

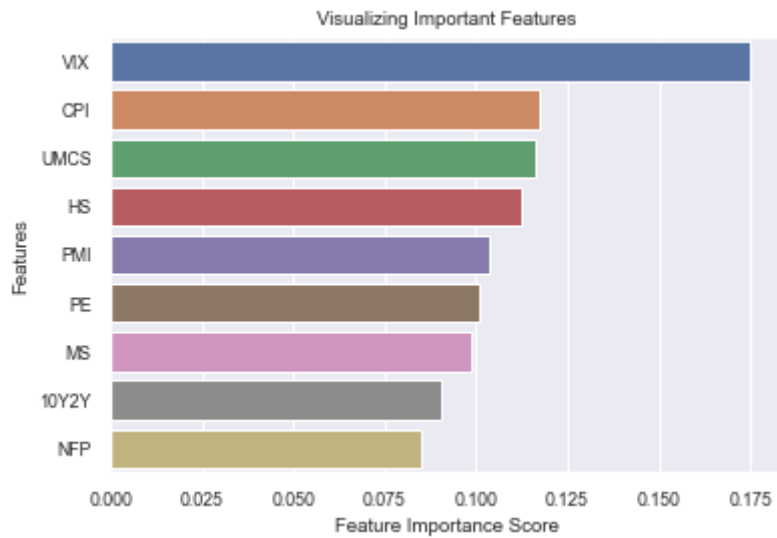
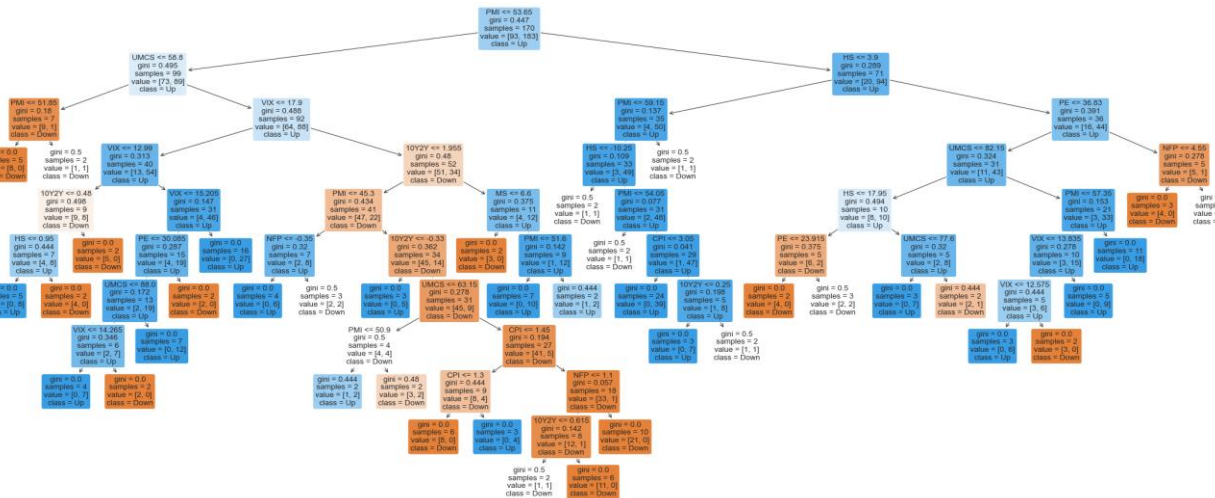


Figure 9: L'arbre issu du modèle Random Forest with cross validation



CHAID (SPSS)

Figure 10: Arbre de décisions : Validation

The dialog box 'Arbre de décisions : Validation' contains the following options and fields:

- ☐ **Aucun**
- ☐ **Validation croisée**
Nombre de niveaux d'échantillon :
La validation croisée n'est pas disponible pour les méthodes CRT et Quest si l'élagage est sélectionné
- ☒ **Validation par partition d'échantillon**
 - Allocation d'observation**
 - ☒ **Utiliser l'affectation aléatoire**
Echantillon d'apprentissage (%) : Echantillon de test : 30,0000%
 - ☐ **Utiliser une variable**
Variables : Echantillon scindé par :
Les observations avec une valeur de 1 sont affectées à l'échantillon d'apprentissage. Toutes les autres sont utilisées dans l'échantillon de test.
 - Afficher les résultats pour**
 - ☒ **Echantillons de test et d'apprentissage**
 - ☐ **Echantillon de test uniquement**

Buttons:

Figure 11: CHAID : Limites de croissance

The 'Arbre de décisions : Critères' dialog box, CHAID tab, contains the following settings:

- Limites de croissance** (selected tab)
- Profondeur maximale de l'arbre**
 - ☐ **Automatique**
Le nombre maximal de niveaux est de 3 pour CHAID ; 5 pour CRT et QUEST.
 - ☒ **Personnalisé**
Valeur :
- Nombre minimal d'observatio...**
 - Noeud parent :
 - Noeud enfant :

Buttons:

Figure 12: CHAID : les hyperparamètres

Arbre de décisions : Critères

Limites de croissance CHAID Intervalles

Niveau de signification pour

Noeuds de scission : 0,1

Fusion des catégories : 0,05

Statistiques du Khi-carré

☐ Pearson

☒ Rapport de vraisemblance

Estimation du modèle

Nombre maximum d'itérations : 200

Changement minimum dans les fréquences théoriques de cellule : 0,01

☒ Ajustement des valeurs de signification à l'aide de la méthode Bonferroni.

☒ Autoriser la scission des catégories fusionnées à l'intérieur d'un noeud

Poursuivre Annuler Aide

Figure 13: CHAID : Intervalles des variables

Arbre de décisions : Critères

Limites de croissance CHAID Intervalles

Intervalles des variables d'échelle indépendantes

☒ Nombre fixe Valeur : 5

☐ Personnalisé

Intervalles

Variable	Intervalles
10Y2Y	10
MS	10
CPI	10
UMCS	10
NFP	10
HS	10
PMI	10
PE	10

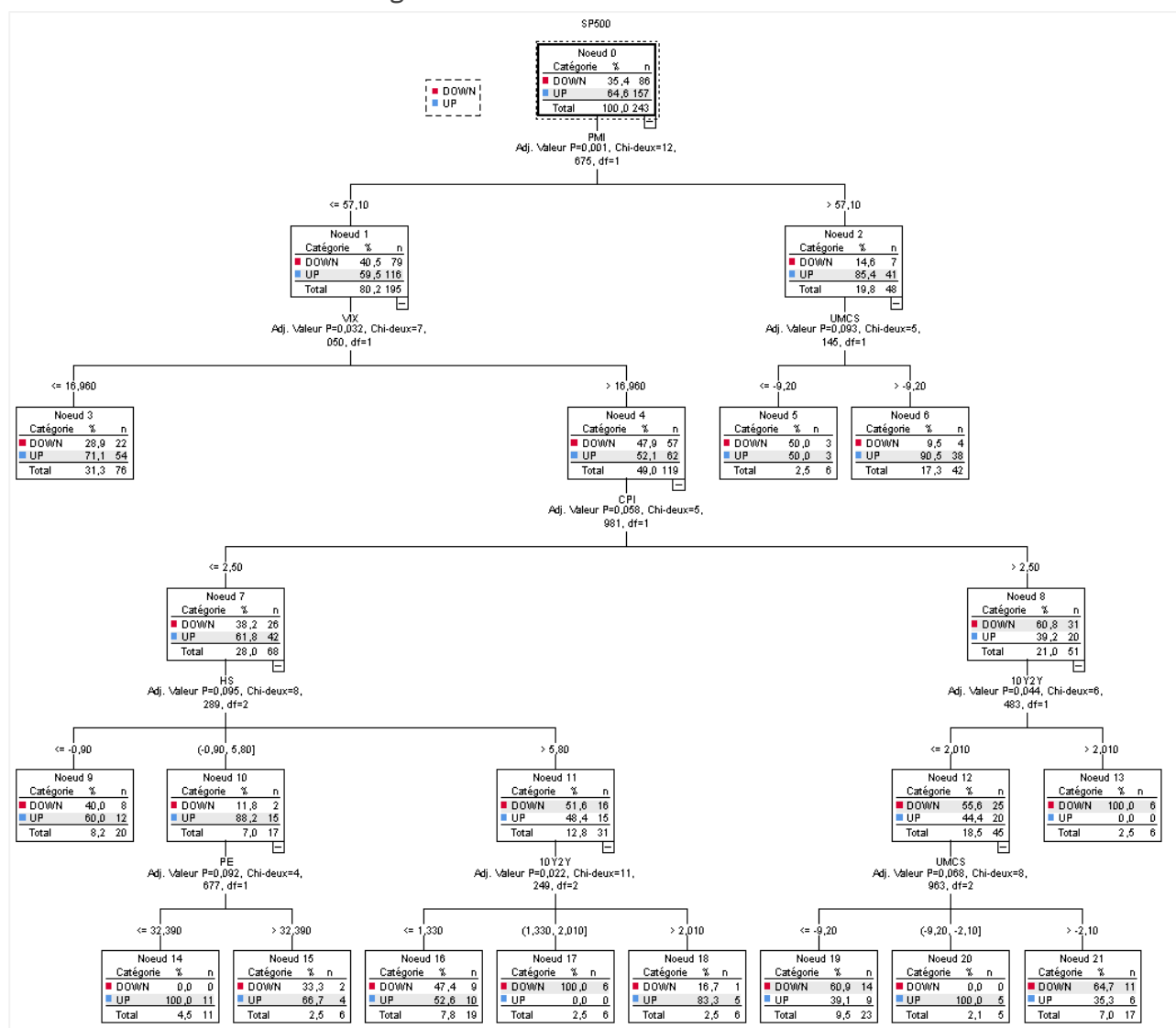
Poursuivre Annuler Aide

Figure 14: Resultat du modèle CHAID

Classification				
Echantillon	Observé	Prévisions		
		DOWN	UP	Pourcentage correct
Apprentissage	DOWN	37	49	43,0%
	UP	15	142	90,4%
	Pourcentage global	21,4%	78,6%	73,7%
Test	DOWN	13	26	33,3%
	UP	10	54	84,4%
	Pourcentage global	22,3%	77,7%	65,0%

Méthode de croissance : CHAID
Variable dépendante : SP500

Figure 15: L'arbre de décision : CHAID



Sources:

CHAID DECISION TREE: METHODOLOGICAL FRAME AND APPLICATION Marina Milanović Faculty of Economics, University of Nis, Serbia. *ECONOMIC THEMES* (2016) 54(4): 563-586

Improved CHAID Algorithm for Document Structure Modelling. Abdel Belaïd, Philippe Moinel, Yves Rangoni. HAL Id: inria-00579684

The Macroeconomy as a Random Forest - Philippe Goulet Coulombe. University of Pennsylvania. March 8, 2021

CHAID DECISION TREE: METHODOLOGICAL FRAME AND APPLICATION Marina Milanović Faculty of Economics, University of Nis, Serbia

Classification of intraday S&P500 returns with a Random Forest. Lohrmann Christoph, Luukka Pasi. *International Journal of Forecasting*

<https://sefiks.com/2020/03/18/a-step-by-step-chaid-decision-tree-example/><https://www.r-bloggers.com/2018/05/chaid-and-r-when-you-need-explanation-may-15-2018/>

<https://www.analyticsvidhya.com/blog/2021/05/implement-of-decision-tree-using-chaid/>

<https://ibecav.netlify.app/post/analyzing-churn-with-chaid/>

<https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/chaid/>

<https://www.adiance.com/blog/how-to/a-guide-to-chaid-a-decision-tree-algorithm-for-data-analysis/>

<https://mksingh0892.medium.com/chaid-decision-tree-30a4c7ba6efc>

<https://www.mdpi.com/2227-7390/11/11/2558>

<https://medium.com/@t0masGutierrez/how-to-beat-the-s-p-500-index-by-1-52e306cae59c>

<https://www.mdpi.com/2227-7390/11/11/2558>