MEMO-F524
MASTER THESIS

Machine Learning applied to stars

Auteur :

TALHAOUI Yassin

Section :

Computer science

Promoteur :

Defrance Matthieu

March 11, 2025

Contents

1 Introduction

L'objectif de cette thése de master est de comprendre et d'investiguer comment les techniques actuelles de machine learning peuvent nous aider à étudier les propriétés stellaires de étoiles simples et binaires. Au départ, l'astrophysique était un domaine scientifique trés pauvre en données mais au cours du temps de nombreuses missions spatiales ainsi que de grandes études en astonomies ont permis de réconlter de quantités massives de données provenant de centaines de millions de sources astronomiques. Dans l'ére GAIA, il est prévu que le volume des données augmente d'avantages, vers le domaine du pétaoctet. L'astrophysique est donc devenue une science intensivement axées sur les données. Toutes ces données stellaires ont permis l'émergence de nouveaux défis algorithmiques, informatiques et également statistiques. C'est dans ce context que les techniques de machines learning, utilisées dans divers domaines d'études scientifiques, pourraient s'avérer être d'une grande utilité pour extraire des informations à partir d'observations. En tirant parti des algorithmes de machine learning, les astrophysiciens peuvent s'attaquer plus efficacement à un large éventail de questions de recherches, ce qui nous permet de mieux comprendre le cosmos et de faire de nouvelles découvertes.

2 Intégration de techinques de machine learning aux méthodes astrophysiques traditionnelles

Pour explorer l'intégration des techniques de machine learning aux méthodes astrophysiques traditionnelles, nous allons nous lancer dans un défi où la combinaison des connaissances du domaine et des approches basées sur les données illuminera le cosmos avec une une grande clarté. Cet effort collaboratif exploite l'expertise des astronomes et des data scientists, réunissant le savoir des chercheurs et les connaissances informatiques pour explorer les propriétés stellaires. Dans ce qui suit une présentation des méthodologies utilisées dans plusieurs études dans le domaines sera faite.

Comprendre les méthodes astrophysiques traditionnelles

Pour commencer nous allons procéder par une exploration des méthodes astrophysiques traditionnelles, incluant la spectroscopie, la photométrie et la modélisation stellaire. Sans pour autant trop aller dans les subtilités des systèmes de classification stellaire, des trajectoires d'évolution et des analyses d'abondance chimique, afin d'établir une base de connaissances solide dans le domaine.

Identifier les Challenges et Opportunitiés

En examinant le domaine de la recherche astrophysique, les chercheurs identifient les zones dans lesquelles les méthodes traditionnelles se heurtent à des limites ou à des inefficacités. L'identification des possibilités d'amélioration telles que les techniques de machine learning peuvent nous apporter quelque chose en terme d'automatisation de tâches à forte intensité de main-d'œuvre ou la découverte de modèles cachés dans les données, afin de guider notre étude vers l'amélioration de la précision des prédictions et l'obtention de nouvelles connaissances à partir des données que nous détenons.

S'engager dans une collaboration interdisciplinaire

La collaboration entre les astronomes et les data scientists crée un environnement où les experts du domaine et les practiciens dans le domaine du machine learning convergent pour échanger des idées, des méthodologies et des points de vue. Ce dialogue interdisciplinaire facilite le partage des connaissances et comble le fossé entre l'astrophysique théorique et l'analyse informatique des données.

Acquisition et traitement des données

Parmi les tâches les plus importantes, il y a la gestion d'ensembles de données divers et représentatifs comprenant des spectres stellaires, des mesures photométriques et des informations auxiliaires, notamment des classifications stellaires, des âges et des métallicités. Des techniques rigoureuses de preprocessing traitent les problèmes de qualité des données, tels que le bruit, les erreurs d'étalonnage et les valeurs manquantes, garantissant ainsi l'intégrité des analyses.

Feature Engineering et sélection des caractéristiques

La collaboration avec les astrophysicien permet d'identifier des caractéristiques astrophysiques pertinentes qui incluent des propriétés stellaires importantes. En nous appuyant sur nos connaissances du domaine, nous concevons des caractéristiques informatives qui capturent les intensités des lignes spectrales, les formes du continuum et d'autres caractéristiques clés. Les techniques de réduction de la dimensionnalité distillent les données spectrales à haute dimension dans des espaces de caractéristiques interprétables, améliorant ainsi l'efficacité des calculs et l'interprétabilité.

Développement et validation de modèles

La co-conception de modèles de machine learning adaptés à des questions ou des tâches astrophysiques spécifiques, telles que l'estimation des paramètres stellaires ou la classification, fait avancer notre exploration. L'adoption d'une variété d'algorithmes, y compris les méthodes traditionnelles de régression et de classifica-

tion, ainsi que des architectures sophistiquées de deep learning telles que les réseaux de neurones convolutifs (CNN) ou les réseaux de neurones récurrents (RNN), nous permettent d'extraire des informations à partir de données complexes.

Interprétabilité et transparence

En mettant l'accent sur l'interprétabilité et la transparence des modèles, les chercheurs collaborent avec les astronomes pour développer des techniques d'interprétabilité a posteriori. L'analyse de l'importance des caractéristiques, les mécanismes d'attention et les explications diagnostiques des modèles mettent en lumière la logique sous-jacente des prédictions des modèles, facilitant ainsi la confiance et la compréhension.

Raffinement et évaluations itératives

En adoptant une approche itérative du perfectionnement et de l'évaluation des modèles, nous sollicitons le retour d'information des experts du domaine à chaque étape du processus de développement. En validant continuellement les performances des modèles par rapport aux observations de terrain, nous affinons les algorithmes et les méthodologies sur la base d'observations empiriques et de considérations spécifiques au domaine, garantissant ainsi la robustesse et la fiabilité des analyses.

Plus-value

Grâce à cette combinaison des techniques astrophysiques traditionnelles et des techniques de machine learning, de nouvelles perspectives s'ouvrent sur le domaine complexe des propriétés stellaires, ouvrant la voie à de futures avancées dans notre compréhension du cosmos.

3 Utilisation de données spectrales

Pour investiguer sur les techniques de machines learning qui peuvent nous aider, nous allons considérer de grands échantillons de spectres stellaires provenant d'enquêtes comme GALAH, Gaia-ESO survey.

3.1 Introduction aux sprectres

Importance des spectres

Notre compréhension des propriétés physiques des étoiles dépend en grande partie de l'analyse de leurs spectres. L'examen des raies d'absorption permet de déterminer la masse, la température et la composition des étoiles. La forme des raies nous renseigne sur les processus atmosphériques.

Compositions

Les spectres stellaires sont constitués d'un spectre continu sur lequel se superposent des lignes spectrales étroites, principalement des lignes d'absorption sombres, mais aussi parfois des lignes d'émission lumineuses.

Le spectre continu

Le spectre continu provient de la surface chaude de l'étoile. L'atmosphère absorbe des longueurs d'onde spécifiques, ce qui crée des zones sombres dans le spectre, indiquant différentes compositions chimiques

Classification

Les spectres stellaires sont classés en fonction de l'intensité de ces raies spectrales. Ce système de classification a été initié par Isaac Newton, puis perfectionné par Joseph Fraunhofer et d'autres.

Méthodes de mesure

Les spectres stellaires sont généralement obtenus à l'aide de prismes objectifs ou de spectrographes à fente. Ces méthodes permettent une analyse détaillée des lignes spectrales individuelles.

Analyse

Les spectres sont convertis en tracés d'intensité, révélant la densité du flux en fonction de la longueur d'onde. La forme des raies spectrales fournit des informations précieuses sur les atmosphères stellaires, tandis que l'intensité des raies permet de déterminer les compositions chimiques

Classification spectrale de Harvard

Développé à l'Observatoire de Harvard, ce système de classification classe les étoiles en fonction de leurs caractéristiques spectrales, principalement la température. Il comprend des lettres désignant les types spectraux et des nombres pour les sous-classes

Classification spectrale de Yerkes

Un système de classification plus précis introduit par l'Observatoire de Yerkes prend en compte à la fois la température et la luminosité. Il classe les étoiles en six catégories de luminosité, ce qui permet de mieux comprendre leurs propriétés.

Spectres particuliers

Certaines étoiles présentent des spectres particuliers en raison de facteurs tels que des vents stellaires puissants, une rotation ou des interactions binaires. Les étoiles Wolf-Rayet, les étoiles Be et les étoiles à coquille en sont des exemples.

3.2 Intéret des spectre stellaire

De plus, L'analyse des spectres stellaire nous permet de comprendre la nature et l'évolution des corps célestes comme les étoiles. Ils nous donnent des informations tels que :

- **La température** : La température de surface d'une étoile et celle de son enveloppe externe peuvent être déduites de la couleur de la lumière qu'elle émet. En effet, une étoile plus chaude paraîtra plus bleue, car la température élevée favorise l'émission de lumière à des longueurs d'onde plus courtes, conformément à la loi du rayonnement thermique. En analysant son spectre, il est possible d'estimer une température "effective" pour l'étoile, prenant en compte le transfert de radiation à travers les différentes couches de son atmosphère stellaire.
- **La composition chimique** : Les fréquences particulières des raies spectrales fournissent des informations distinctes sur les éléments qui absorbent ou émettent des photons. Des bases de données spectroscopiques ont été élaborées à partir de l'étude de spectres produits en laboratoire, ce qui facilite l'identification de l'origine des raies observées, qu'elles soient en absorption ou en émission, dans les spectres astronomiques. En analysant l'intensité relative des raies caractéristiques des éléments détectés et en se basant sur des modèles théoriques, il est possible d'inférer la composition chimique de l'atmosphère de chaque étoile.
- **La vitesse** : $\Delta\lambda$, le décalage des raies spectrales observées, est couramment utilisé pour mesurer les vitesses. Il permet de calculer la vitesse radiale $\mathbf{v}$ d'un objet céleste, qui correspond à sa composante

de vitesse d'éloignement le long de la ligne de visée. Cette vitesse est exprimée par :

$v = c\frac{\Delta\lambda}{\lambda}$.

En somme, l'analyse des spectres stellaires est un outil puissant pour sonder les secrets des étoiles. En révélant leur température, leur composition chimique et leur vitesse, ces spectres nous offrent une fenêtre fascinante sur la nature et l'évolution des astres célestes. Ils constituent ainsi un pilier essentiel de la recherche en astrophysique, nous permettant de mieux comprendre les mystères de l'univers.

4 Études antérieures sur l'analyse spectrale stellaire

4.1 Analyse de la composition chimique

Les chercheurs ont utilisé des techniques de machine learning pour analyser les spectres stellaires et en déduire la composition chimique. En entraînant des modèles sur des données spectrales avec des abondances chimiques connues, les algorithmes peuvent prédire la composition élémentaire des étoiles sur la base de leurs spectres. Des méthodes d'extraction de caractéristiques telles que **l'analyse en composantes principales (PCA)** ont été utilisées pour réduire la dimensionnalité des données spectrales tout en conservant les informations pertinentes. Des algorithmes de machine learning tels que **Support Vector Machines (SVM)** ou les **randomForest** ont été employés pour classer les étoiles dans différentes classes d'abondance chimique sur la base de leurs spectres.

4.2 Estimation de la température

Des techniques de Machine Learning ont été utilisées pour estimer la température effective des étoiles à partir de leurs caractéristiques spectrales. Les chercheurs ont utilisé des techniques de régression telles que la régression linéaire ou les réseaux de neurones pour prédire la température des étoiles à partir de leurs caractéristiques spectrales. Des méthodes d'ingénierie des caractéristiques, y compris la sélection des longueurs d'onde ou la normalisation du continuum, ont été appliquées pour améliorer la performance prédictive des modèles d'estimation de la température.

4.3 Prédiction de la luminosité

Des algorithmes de machine learning ont été utilisés pour prédire la luminosité des étoiles, qui est une mesure de leur luminosité intrinsèque. L'estimation de la luminosité est essentielle pour comprendre les propriétés des étoiles et leurs stades d'évolution. Des algorithmes de régression tels que les processus gaussiens ou des modèles de deep learning tels que les réseaux de neurones convolutifs (CNN) ont été utilisés pour prédire la luminosité stellaire à partir de données spectrales. Des techniques d'extraction de caractéristiques, telles que les indices d'intensité des raies ou les rapports de flux, ont été utilisées pour capturer les informations pertinentes des spectres stellaires pour la prédiction de la luminosité.

4.4 Méthodologies et algorithmes

Extraction des caractéristiques : Des méthodes telles que PCA, ont été utilisées pour extraire les caractéristiques pertinentes des spectres stellaires tout en réduisant la dimensionnalité. **Classification** : Des algorithmes tels que SVM, Random Forests ou k-nearest neighbors ont été utilisés pour classer les étoiles

dans différentes catégories sur la base de leurs caractéristiques spectrales. **Régression** : Des techniques telles que la régression linéaire, les processus gaussiens ou les réseaux de neurones ont été utilisées pour prédire les paramètres stellaires continus tels que la température ou la luminosité à partir des données spectrales.

5 The Payne: Self-consistent ab initio Fitting of Stellar Spectra

The Payne est une contribution significative au domaine de l'astrophysique stellaire, en particulier dans le domaine de l'analyse spectrale à l'aide de techniques de machine learning. L'étude présente une nouvelle approche de l'ajustement des spectres stellaires, connue sous le nom de « The Payne », qui combine les principes de la modélisation physique avec des algorithmes de machine learning pour parvenir à un ajustement spectral autoconsistant et précis.

5.1 Principaux points et résultats

The Payne utilise un cadre qui incorpore des principes physiques fondamentaux, tels que les modèles d'atmosphère stellaire et la physique atomique, dans le processus d'interpolation spectral. Contrairement aux méthodes empiriques traditionnelles, The Payne effectue un ajustement ab initio auto-consistant des spectres stellaires, ce qui signifie qu'il dérive des paramètres physiques directement à partir des données d'observation sans s'appuyer sur des modèles préexistants. En utilisant des algorithmes avancés de machine learning, tels que les réseaux neuronaux artificiels, The Payne est capable de modéliser efficacement et avec précision les relations complexes entre les paramètres stellaires et les caractéristiques spectrales. L'étude démontre l'efficacité de «The Payne» dans l'interpolation des spectres stellaires sur une large gamme de types stellaires et de stades d'évolution. La nature auto-consistante de The Payne permet une détermination robuste des paramètres stellaires clés, y compris la température effective, la gravité de surface, la métallicité et les abondances élémentaires, directement à partir des spectres observés.

5.2 Méthodologies et algorithmes

Payne utilise des réseaux de neurones artificiels comme algorithme de machine learning pour l'ajustement spectral. Ces réseaux de neurones sont entraînés sur un vaste ensemble de données de spectres synthétiques générés à partir de modèles d'atmosphère stellaire. Des techniques d'extraction de caractéristiques, telles que le regroupement des longueurs d'onde ou la normalisation du continuum, peuvent être utilisées pour prétraiter les données spectrales avant de les introduire dans le réseau de neurone. L'architecture du réseau de neurone est conçue pour capturer les relations non linéaires complexes entre les caractéristiques spectrales d'entrée et les paramètres stellaires de sortie. Au cours du processus d'apprentissage, le réseau de neurone

apprend à mettre en correspondance les spectres observés et les paramètres stellaires correspondants, ce qui permet d'obtenir une grande exactitude et une grande précision dans l'ajustement spectral.

5.3 Implications et contributions

The Payne représente une avancée significative dans les techniques d'analyse spectrale, offrant une approche plus robuste et physiquement motivée pour ajuster les spectres stellaires par rapport aux méthodes traditionnelles. En combinant les principes de la modélisation physique avec des algorithmes de machine learning, The Payne permet aux astronomes de dériver des paramètres stellaires précis et auto-consistants directement à partir des données d'observation. La nature auto-adaptative de The Payne le rend particulièrement adapté à l'analyse des relevés spectroscopiques à grande échelle, pour lesquels des méthodes automatisées et efficaces d'analyse spectrale sont essentielles. L'étude ouvre de nouvelles perspectives pour l'étude des populations stellaires, des abondances chimiques et de l'évolution stellaire en fournissant un outil puissant pour l'analyse des spectres stellaires avec une exactitude et une précision sans précédent.

5.4 Implémentation

The Payne est un framework d'interpolation de spectres stellaires développé par Ting-Yuan sen. Il vise à effectuer une d'interpolation auto-cohérent des spectres stellaires en utilisant une combinaison de techniques de modélisation physique et de machine learning.Le code source est accessible sur un dépot publique sur github à l'adresse `https://github.com/tingyuansen/The_Payne`.

The Payne se veut être un outil sophistiqué basé sur un réseau de neurone conçu pour analyser les spectres stellaires et déduire des paramètres stellaires tels que la température effective, la gravité de surface et l'abondance des éléments. La mise en œuvre de « The Payne » implique plusieurs composants, chacun servant un objectif spécifique dans le flux de travail global. Voici une description détaillée de chaque composant et de la manière dont ils contribuent collectivement à la construction et à l'utilisation de « The Payne » :

Architecture du réseau de neurone

L'architecture du réseau de neurone constitue la partie principale de « The Payne » et est responsable de l'apprentissage de la cartographie entre les spectres stellaires et les paramètres stellaires. Il comprend plusieurs couches, notamment des couches d'entrée, cachées et de sortie, avec diverses fonctions d'activation et techniques de régularisation pour faciliter l'apprentissage et éviter le surajustement. "The Payne" utilise une architecture de réseau de neurone (deep learning), souvent dotée de plusieurs couches cachées, pour capturer les relations complexes inhérentes aux spectres stellaires.

Données d'entraînement

Les données d'entraînement consistent en une vaste collection de spectres stellaires, chacun associé aux paramètres stellaires correspondants obtenus à partir de sources de référence telles que des études spectroscopiques ou des modèles théoriques. Ces spectres sont généralement prétraités pour éliminer le bruit, normaliser les intensités et gérer les valeurs manquantes avant d'être introduits dans le réseau neuronal pour la formation. La qualité et la diversité des données d'entraînement ont un impact significatif sur les performances et la capacité de généralisation de « The Payne ».

Phase d'entrainement

La phase d'entrainement consiste à introduire de manière itérative des ensembles de spectres prétraités dans le réseau de neurone et à ajuster ses poids et ses biais pour minimiser l'écart entre les paramètres stellaires prédits et réels. La phase d'entrainement est généralement effectuée à l'aide d'algorithmes d'optimisation tels que Adam rectifié (RAdam), qui mettent à jour efficacement les paramètres du réseau en fonction des gradients de la fonction de perte. Des hyperparamètres tels que le taux d'apprentissage, la taille de l'ensemble de données et le nombre d'époch sont réglés pour optimiser la convergence et les performances du réseau de neurone.

Optimisation Adam rectifié (RAdam)

RAdam est un algorithme d'optimisation avancé qui améliore l'optimiseur Adam traditionnel en rectifiant son taux d'apprentissage adaptatif. Il résout le problème de la mauvaise convergence et des dépassements dans les premiers stades de la phase d'entrainement en ajustant dynamiquement le taux d'apprentissage en fonction de la variance des gradients passés. La mise en œuvre de RAdam dans « The Payne » garantit une optimisation stable et efficace des paramètres du réseau de neurone, conduisant à une convergence plus rapide et à des performances améliorées du modèle.

Modèle spectral

Le composant du modèle spectral encapsule la formulation mathématique et les principes physiques qui sous-tendent la relation entre les paramètres stellaires et les caractéristiques spectrales. Il fournit des fonctions permettant de générer des spectres synthétiques basés sur des paramètres stellaires et des grilles de longueurs d'onde donnés, permettant la synthèse de spectres dans une large gamme d'atmosphères et de compositions stellaires. Le modèle spectral sert de vérité terrain par rapport à laquelle les prédictions de « The Payne » sont validées et calibrées.

Inférence de paramètres

Une fois le réseau de neurones formé et validé, il peut être utilisé pour déduire des paramètres stellaires à partir des spectres observés. Étant donné un nouveau spectre, « The Payne » utilise le réseau de neurones entraîné pour prédire les paramètres stellaires correspondants, tels que la température effective, la gravité de surface et les abondances chimiques. Ces paramètres déduits peuvent ensuite être comparés à des valeurs de référence ou utilisés pour des analyses plus approfondies, telles que des études de populations stellaires ou la caractérisation d'exoplanètes.

Post-traitement et estimation de l'incertitude

Les étapes de post-traitement peuvent impliquer d'affiner les paramètres stellaires déduits, d'effectuer des contrôles de qualité et d'estimer les incertitudes associées aux prédictions. L'estimation de l'incertitude est cruciale pour quantifier la fiabilité de l'inférence de paramètres et évaluer la robustesse du modèle de réseau de neurones. Des techniques telles que le rééchantillonnage bootstrap ou l'inférence bayésienne peuvent être utilisées pour caractériser l'incertitude des paramètres stellaires prédits.

Intégration avec les pipelines d'analyse spectrale

"The Payne" peut facilement être intégré aux pipelines d'analyse spectrale existants utilisés par les astronomes et les astrophysiciens pour étudier les populations stellaires, la dynamique galactique et la caractérisation des exoplanètes. Il fournit un outil puissant pour automatiser et accélérer l'analyse d'ensembles de données spectrales à grande échelle, permettant aux chercheurs d'extraire des informations précieuses sur les propriétés et l'évolution des étoiles et des galaxies.

Apport à la recherche

En résumé, « The Payne » représente une approche de pointe de l'inférence de paramètres stellaires utilisant des réseaux de neurone et des techniques d'optimisation avancées. Sa conception modulaire, associée à un processus de formation et à un modèle spectral robustes, permet aux astronomes et aux chercheurs de libérer tout le potentiel des spectres stellaires pour comprendre les complexités et les mystères de l'univers.

5.5 Tutoriel

Un Jupyter notebook ecrit en python sert de guide complet pour comprendre et utiliser les fonctionnalités du code « The Payne » pour ajuster les spectres stellaires. Voici une répartition détaillée des informations fournies dans le notebook :

Introduction et aperçu

Le notebook commence par présenter le code « The Payne » et ses principales fonctionnalités, en soulignant son rôle dans l'ajustement des spectres stellaires en utilisant une combinaison de techniques de modélisation physique et de machine learning. Il décrit les principaux objectifs du notebook, notamment la génération de spectres de modèles, l'interpolation des spectres observés et la formation de réseaux de neurones personnalisés.

Configuration et dépendances

le processus de configuration est également décrit, y compris l'importation des bibliothèques et des modules nécessaires à l'exécution du code "The Payne". Il définit également des paramètres essentiels tels que le réseau de longueurs d'onde et le masque APOGEE, qui sont cruciaux pour le traitement et l'analyse des spectres stellaires.

Génération de spectres de modèles

Le notebook montre comment générer des spectres modèles pour des étoiles individuelles en fonction de labels d'entrée telles que la température effective, la gravité de surface et l'abondance des éléments. Il explique le processus de mise à l'échelle des lables et d'utilisation d'un réseau de neurones pour prédire le spectre correspondant aux paramètres donnés.

Interpolation de Spectres

Ici, le notebook simule un spectre observé en ajoutant du bruit au spectre généré. Il montre ensuite comment le code « The Payne » s'adapte au spectre bruité à l'aide de ses algorithmes d'ajustement, récupérant finalement les labels d'entrée du spectre ajusté.

Téléchargement et installation de Spectra réels

Le notebook fournit des exemples pratiques en téléchargeant des spectres APOGEE réels et en les interpolants à l'aide du code « The Payne ». En appliquant le code à des données d'observation réelles, il démontre son efficacité et son applicabilité à la recherche astrophysique.

Entrainement de réseaux de neurones personnalisés

Le notebook propose des instructions pour entraîner des réseaux de neurones personnalisés à l'aide de données d'entraînement définies par l'utilisateur. Il explique comment spécifier des paramètres tels que le nombre de neurones, le taux d'apprentissage et la taille de l'ensemble, et fournit des visualisations de l'erreur d'entrainement et de validation pour surveiller le processus d'entrainement.

Notes pratiques

En conclusion du notebook, des conseils pratiques sont fourni et des considérations pour utiliser efficacement le code « The Payne ». Des aspects tels que l'efficacité informatique, l'optimisation des paramètres de formation et les défis potentiels que les utilisateurs peuvent rencontrer lors de l'ajustement spectral et de la formation sur les réseaux neuronaux sont abordés.

En suivant les exemples et les instructions fournis dans le Jupyter Notebook, les utilisateurs peuvent acquérir une compréhension complète du code « The Payne » et exploiter ses capacités d'ajustement et d'analyse des spectres stellaires dans leurs efforts de recherche astrophysique. De plus, le notebook offre des informations sur la personnalisation des réseaux de neurones et l'optimisation des paramètres de formation en fonction d'objectifs de recherche et d'ensembles de données spécifiques.

6 Défis et limites

Lors de l'application de techniques de machine learning à l'astrophysique stellaire, plusieurs défis et limitations doivent être pris en compte.

Qualité des données

Les échantillons de données de spectres stellaires peuvent contenir des artefacts, des effets instrumentaux ou des erreurs de calibrage qui peuvent affecter la qualité des données. De plus, les variations de la qualité des données entre différentes sources d'observation ou instruments peuvent introduire des biais ou des incohérences dans l'analyse.

Taille d'échantillons

L'obtention d'échanitillons de données importants et diversifiés de spectres stellaires pour l'entraînement de modèles de machine learning peut s'avérer difficile, en particulier pour les objets stellaires rares ou exotiques. La taille limitée des échantillons peut conduire à une couverture insuffisante de l'espace des paramètres, ce qui affecte la capacité du modèle à se généraliser à des données inédites.

Réduction du bruit

Les spectres stellaires sont souvent soumis à des bruits provenant de diverses sources, notamment le bruit des photons, le fond du ciel et les effets instrumentaux. Le développement de techniques robustes de réduction du bruit qui filtrent efficacement le bruit tout en préservant le signal sous-jacent est crucial pour une analyse spectrale précise.

Interprétation des modéles

Les modèles de machine learning, en particulier les modèles complexes tels que les modèles de deep learning, peuvent manquer d'interprétabilité, ce qui rend difficile la compréhension de la manière dont ils parviennent à leurs prédictions. Des modèles interprétables sont essentiels pour comprendre les processus physiques qui sous-tendent les phénomènes stellaires et pour valider la fiabilité des prédictions des modèles.

Performance de généralisation

L'overfitting se produit lorsqu'un modèle apprend à capturer du bruit ou des modèles non pertinents dans les données d'apprentissage, ce qui conduit à une mauvaise performance de généralisation sur des données non vues. Les techniques de régularisation, la validation croisée et le contrôle de la complexité du modèle sont essentiels pour atténuer le surajustement et garantir la robustesse du modèle.

Introduction de biais

Ceci peut se produire lorsque l'ensemble de données d'entraînement n'est pas représentatif de la population sous-jacente d'intérêt, ce qui conduit à des prédictions de modèle faussées. II faut veiller à ce que l'ensemble de données d'entraînement couvre de manière adéquate toute la diversité des propriétés stellaires et évite les biais introduits par les méthodes d'observation ou d'échantillonnage.

Généralisation à des données non vues

Les modèles de machine learning formés sur un ensemble de données d'observation peuvent ne pas bien se généraliser à des données inédites provenant de différents télescopes, instruments ou conditions d'observation. Les techniques d'apprentissage par transfert, qui exploitent les connaissances acquises sur un ensemble de données pour améliorer les performances sur un autre, peuvent aider à résoudre les problèmes de généralisation.

Motivation

Il est essentiel de relever ces défis et ces limites pour appliquer avec succès les techniques de machine learning à l'astrophysique stellaire. En développant des méthodologies robustes, en intégrant la connaissance du domaine et en évaluant soigneusement les performances des modèles, les chercheurs peuvent surmonter ces obstacles et libérer tout le potentiel des techniques de machine learning pour faire progresser notre compréhension du cosmos.

7 Avancées et innovations récentes

Les progrès récents des méthodologies de machine learning ont considérablement amélioré l'analyse des spectres stellaires, en offrant des approches innovantes pour l'ingénierie des caractéristiques, la réduction de la dimensionnalité et l'optimisation des modèles. En voici quelques exemples :

Feature Engineering

- Les techniques de deep learnig ont permis l'extraction automatique de caractéristiques à partir de données spectrales, éliminant ainsi la nécessité d'une ingénierie manuelle des caractéristiques.
- Les réseaux de neurones convolutifs (CNN) ont été utilisés pour apprendre directement les représentations hiérarchiques des caractéristiques spectrales, en capturant à la fois les modèles locaux et globaux.
- Les réseaux de neurones récurrents (RNN) ont été appliqués pour capturer les dépendances temporelles dans les séries de données spectrales, permettant une modélisation plus précise des phénomènes dynamiques tels que la variabilité stellaire.

Réduction de la dimensionnalité

Les autoencodeurs variationnels (VAE) et les réseaux adversariaux génératifs (GAN) ont été utilisés pour la réduction non supervisée de la dimensionnalité des données spectrales. Ces techniques apprennent des représentations à faible dimension des spectres tout en préservant les informations essentielles, ce qui facilite un traitement et une analyse plus efficaces.

Optimisation des modèles

Les méthodes d'optimisation bayésiennes, telles que les processus gaussiens et les réseaux neuronaux bayésiens, ont été utilisées pour le réglage des hyperparamètres et l'optimisation des modèles. Ces techniques permettent une exploration plus efficace de l'espace des hyperparamètres et une meilleure convergence des modèles de machine learning.

Architectures de deep learning

- Les réseaux de neurones convolutifs (CNN) ont été appliqués aux données spectrales pour des tâches telles que la classification des objets stellaires, l'identification des caractéristiques spectrales et l'estimation des paramètres stellaires. Les réseaux neuronaux convolutifs peuvent apprendre automatiquement des modèles spatiaux dans les données spectrales, ce qui les rend bien adaptés aux tâches qui impliquent l'analyse d'informations structurées dans l'espace.

- Les réseaux de neurones récurrents (RNN) ont été utilisés pour modéliser les dépendances temporelles dans les séries chronologiques de données spectrales, ce qui permet de prédire la variabilité stellaire et les événements transitoires.
- Long Short-Term Memory (LSTM) networks, un type de RNN, se sont révélés prometteurs pour capturer les dépendances à long terme dans les données spectrales séquentielles, permettant une modélisation plus précise de la dynamique stellaire au fil du temps.

Ces récentes avancées dans les méthodologies de machine learning ont révolutionné l'analyse des spectres stellaires, permettant un traitement plus précis et plus efficace des données d'observation. En tirant parti des architectures d'apprentissage de type deep learning et des techniques innovantes d'ingénierie des caractéristiques et d'optimisation des modèles, les chercheurs peuvent obtenir de nouvelles informations sur les processus physiques complexes qui se produisent dans les étoiles et les galaxies.

8 Orientations futures et tendances émergentes

Alors que les possibilités en matière de machine learning continuent de s'étendre et que notre compréhension de l'astrophysique stellaire s'approfondit, les orientations futures de la recherche promettent d'ouvrir de nouvelles perspectives et de faire progresser notre connaissance du cosmos. Les tendances émergentes dans les deux domaines offrent des possibilités passionnantes d'innovation et de découverte.

Perspectives d'avenir

À l'avenir, la recherche dans le domaine du machine learning et de l'astrophysique stellaire devrait permettre d'explorer des questions de plus en plus complexes et interdisciplinaires. L'intégration de techniques avancées de machine learning aux méthodes astrophysiques traditionnelles permettra aux chercheurs de s'attaquer aux mystères fondamentaux de l'astrophysique avec plus de précision et d'efficacité.

Identifier les tendances émergentes

Identifier les tendances émergentes :

Plusieurs tendances émergentes sont sur le point de façonner le paysage du machine learning et de l'astrophysique stellaire :

- **Analyse de données multimodales** : Avec l'avènement de l'astronomie multi-longueurs d'onde et multi-messagers, la recherche future se concentrera sur l'intégration de données provenant de diverses sources, telles que les observations optiques, infrarouges, radio et d'ondes gravitationnelles. Les algorithmes de machine learning capables d'analyser des flux de données multimodales joueront un rôle essentiel dans la découverte de synergies et de corrélations entre les différentes longueurs d'onde et les messagers cosmiques.
- **Apprentissage par transfert** : Les techniques d'apprentissage par transfert, qui exploitent les connaissances acquises dans un domaine pour améliorer les performances dans un autre domaine, deviendront de plus en plus courantes en astrophysique stellaire. En transférant les représentations apprises de populations stellaires bien étudiées vers des régions sous-explorées de l'espace des paramètres, l'apprentissage par transfert permet une exploration et une caractérisation plus efficaces de diverses populations stellaires.

- **Méthodes d'ensemble** : Les approches d'apprentissage d'ensemble, qui combinent les prédictions de plusieurs modèles afin d'améliorer la précision et la robustesse, seront mises à profit pour traiter les incertitudes et les complexités inhérentes aux phénomènes astrophysiques. Les méthodes d'ensemble offrent un cadre puissant pour l'intégration de divers modèles, sources de données et incertitudes d'observation, permettant des prédictions et des déductions plus fiables.

Discussion sur les applications potentielles

Le Machine Learning offre un immense potentiel pour révolutionner les enquêtes et les missions astronomiques à venir, en proposant de nouvelles approches pour l'analyse, l'interprétation et la découverte des données :

- **Analyse automatisée des observations** : Les algorithmes de machine learning rationaliseront l'analyse des observations astronomiques à grande échelle, permettant la détection et la caractérisation automatisées des objets célestes, des événements transitoires et des phénomènes astrophysiques. Le traitement des données en temps réel et la classification des événements amélioreront notre capacité à identifier des phénomènes cosmiques rares et insaisissables.
- **Cosmologie de précision** : Les techniques de machine learning faciliteront les analyses cosmologiques de précision en extrayant des signaux subtils des ensembles de données cosmologiques, tels que les cartes du fond diffus cosmologique, les observations de structures à grande échelle et les observations d'ondes gravitationnelles. Les méthodes statistiques avancées et les techniques de sélection de modèles permettront une estimation plus précise des paramètres et des tests d'hypothèses dans les modèles cosmologiques.
- **Caractérisation des exoplanètes** : Les algorithmes de machine learning feront progresser le domaine de la caractérisation des exoplanètes en permettant la détection et la classification des systèmes exoplanétaires à partir de spectres stellaires et d'observations photométriques. De nouvelles méthodes d'extraction de caractéristiques et des modèles basés sur les données amélioreront notre capacité à identifier les exoplanètes, à caractériser leurs atmosphères et à évaluer leur potentiel d'habitabilité.

En résumé, les orientations futures de la recherche sur les techniques de machine learning et l'astrophysique stellaire exploreront les tendances émergentes telles que l'analyse de données multimodales, l'apprentissage par transfert et les méthodes d'ensemble, ouvrant ainsi la voie à des avancées transformatrices dans notre compréhension de l'univers. En exploitant la puissance des techniques de machine learning, les astronomes ouvriront de nouvelles perspectives sur le cosmos, élucideront ses mystères et repousseront les frontières de la connaissance humaine.

Objectif du projet

L'objectif de ce projet est de construire un modèle de machine learning capable de prédire le type d'une étoile sur la base de ses propriétés physiques, telles que la température, la luminosité, le rayon, la magnitude absolue, la couleur et la classe spectrale. Cette tâche de classification suit le diagramme de Hertzsprung-Russell (HR), un outil fondamental en astrophysique qui catégorise les étoiles en fonction de leur température et de leur luminosité.

Nous commençons par un problème de classification binaire, en sélectionnant deux types d'étoiles distincts pour faciliter l'entraînement et l'évaluation des modèles. Ensuite, nous l'étendons progressivement à la classification multi-classes avec les six types d'étoiles.

À propos du jeu de données

Ce jeu de données comprend 240 étoiles, classées en six types d'étoiles différents :

- 0 → Brown Dwarf
- 1 → Red Dwarf
- 2 → White Dwarf
- 3 → Main Sequence
- 4 → Supergiant
- 5 → Hypergiant

Les propriétés de chaque étoile sont mesurées par rapport au Soleil :

- Température (K) - Température de surface en kelvins
- Luminosité (L/Lo) - Luminosité par rapport au soleil
- Rayon (R/Ro) - Rayon par rapport au soleil
- Magnitude absolue (Mv) - Luminosité intrinsèque
- Couleur de l'étoile - Couleur observée après analyse spectrale
- Classe spectrale - Classification basée sur les raies spectrales (O, B, A, F, G, K, M)

Collecte et préparation des données

L'ensemble des données a été créé en utilisant des équations astrophysiques réelles et des sources de données telles que :

- la loi de Stefan-Boltzmann - pour calculer la luminosité
- Loi de déplacement de Wien - Pour estimer la température de surface
- Relations de magnitude absolue - Pour déterminer la luminosité intrinsèque
- Méthodes de parallaxe - Pour dériver les valeurs du rayon

L'ensemble des données a été compilé à partir de plusieurs sources en ligne, ce qui a nécessité environ trois semaines de collecte et de prétraitement, en veillant à ce que les données manquantes soient calculées à l'aide de formules astrophysiques.

Importance de l'étude

Comprendre comment les étoiles sont classées et comment elles évoluent dans le temps est fondamental pour l'astronomie et l'astrophysique. Cette étude aide à :

- Valider le diagramme HR à l'aide du machine learning.
- Développer des modèles prédictifs pour la classification des étoiles.
- Explorer l'importance des caractéristiques dans la différenciation des étoiles.

Ce projet s'appuie sur des classificateurs de machine learning pour analyser les performances de différents modèles dans la classification des étoiles, depuis la simple classification binaire jusqu'à la classification multi-classes à 6 classes. Les résultats peuvent fournir des indications précieuses sur les relations entre les propriétés stellaires et sur la manière dont les étoiles s'inscrivent dans le cadre de l'évolution stellaire.

Exploration et analyse du dataset

Avant de construire un modèle de classification, il est essentiel d'explorer et de comprendre l'ensemble de données afin d'identifier les modèles, les relations et les besoins potentiels de prétraitement. Cette section se concentre sur l'analyse de la structure de l'ensemble de données, la vérification des valeurs manquantes, la visualisation des distributions et la compréhension des corrélations entre les caractéristiques.

1. Aperçu de l'ensemble de données

Nous commençons par charger l'ensemble de données et par afficher ses premières lignes afin de comprendre sa structure et le type de données qu'il contient.

```
# Display basic information about the dataset
df.info()
df.head()
```

Sortie :

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 240 entries, 0 to 239
Data columns (total 7 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Temperature (K)         240 non-null    int64
 1   Luminosity(L/Lo)        240 non-null    float64
 2   Radius(R/Ro)            240 non-null    float64
 3   Absolute magnitude(Mv)  240 non-null    float64
 4   Star type               240 non-null    int64
 5   Star color              240 non-null    object
 6   Spectral Class          240 non-null    object
dtypes: float64(3), int64(2), object(2)
memory usage: 13.2+ KB
Temperature (K) Luminosity(L/Lo)        Radius(R/Ro)    Absolute magnitude(Mv)  Star type
    Star color      Spectral Class
0       3068    0.002400        0.1700  16.12   0       Red     M
1       3042    0.000500        0.1542  16.60   0       Red     M
2       2600    0.000300        0.1020  18.70   0       Red     M
3       2800    0.000200        0.1600  16.65   0       Red     M
4       1939    0.000138        0.1030  20.06   0       Red     M
```

Nous remarquons que :

- Le nombre de variables est de 7 et que nous avons 240 échantillons dans l'ensemble de données.
- Les types de données sont numériques, sauf **Star color** et **Spectral Class** qui sont catégorielles.

2. Vérification des valeurs manquantes

Les données manquantes peuvent avoir un impact sur la performance du modèle. Nous vérifions les valeurs manquantes afin de déterminer si des étapes de prétraitement, telles que l'imputation ou la suppression, sont nécessaires.

```
# Check for missing values
df.isnull().sum()
```

Sortie ;

```
Temperature (K)           0
Luminosity(L/Lo)          0
Radius(R/Ro)              0
Absolute magnitude(Mv)    0
Star type                 0
Star color                0
Spectral Class            0
dtype: int64
```

Nous n'avons pas de valeurs manquantes.

3. Résumé statistique

La génération de statistiques sommaires permet de connaître l'étendue, la moyenne et la distribution des variables numériques.

```
# Check for missing values
df.isnull().sum()
```

Sortie :

```
        Temperature (K) Luminosity(L/Lo)        Radius(R/Ro)    Absolute magnitude(Mv)  Star type
count   240.000000      240.000000      240.000000      240.000000      240.000000
mean    10497.462500    107188.361635   237.157781      4.382396        2.500000
std     9552.425037     179432.244940   517.155763      10.532512       1.711394
min     1939.000000     0.000080        0.008400        -11.920000      0.000000
25%     3344.250000     0.000865        0.102750        -6.232500       1.000000
```

```
50%     5776.000000     0.070500        0.762500        8.313000        2.500000
75%     15055.500000    198050.000000   42.750000       13.697500       4.000000
max     40000.000000    849420.000000   1948.500000     20.060000       5.000000
```

Cela permet de :

- détecter les valeurs aberrantes (par exemple, des températures ou des luminosités anormalement élevées).
- Comprendre l'échelle et la variance des caractéristiques.

4. Distribution des classes (vérification de l'équilibre)

Pour nous assurer que notre ensemble de données n'est pas fortement déséquilibré, nous visualisons la distribution des différents types d'étoiles.

```
# Count the occurrences of each star type
sns.countplot(x=df['Star type'], palette='viridis')
plt.xlabel("Star Type")
plt.ylabel("Count")
plt.title("Class Distribution of Star Types")
plt.show()
```

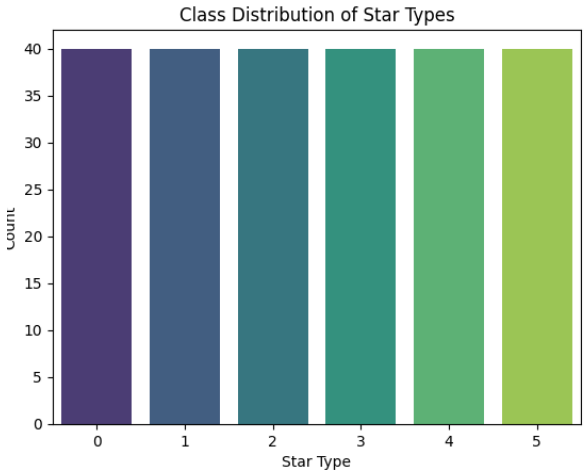

Figure 1: Distribution des classes

L'ensemble de données est parfaitement équilibré entre les différentes classes, avec 40 échantillons par classe.

5. Analyse de corrélation des variables

Matrice de Correlation

Cette matrice de corrélation donne un aperçu des relations entre les différentes caractéristiques de l'ensemble de données. Voici quelques observations clés :

Forte corrélation avec le type d'étoile :

La luminosité (0,68), le rayon (0,66) et la température (0,41) présentent une forte corrélation positive avec le type d'étoile.

La magnitude absolue (-0,96) est fortement corrélée négativement avec le type d'étoile, ce qui est logique puisque les étoiles plus brillantes (valeurs de magnitude plus faibles) ont tendance à se situer plus haut dans la hiérarchie de classification.

Dépendances des variables :

Luminosité et rayon (0.53) : Les étoiles plus grandes ont tendance à être plus lumineuses.

Magnitude absolue et luminosité (-0,69) : Une luminosité plus élevée se traduit par des valeurs de magnitude absolue plus faibles (relation inverse par définition).

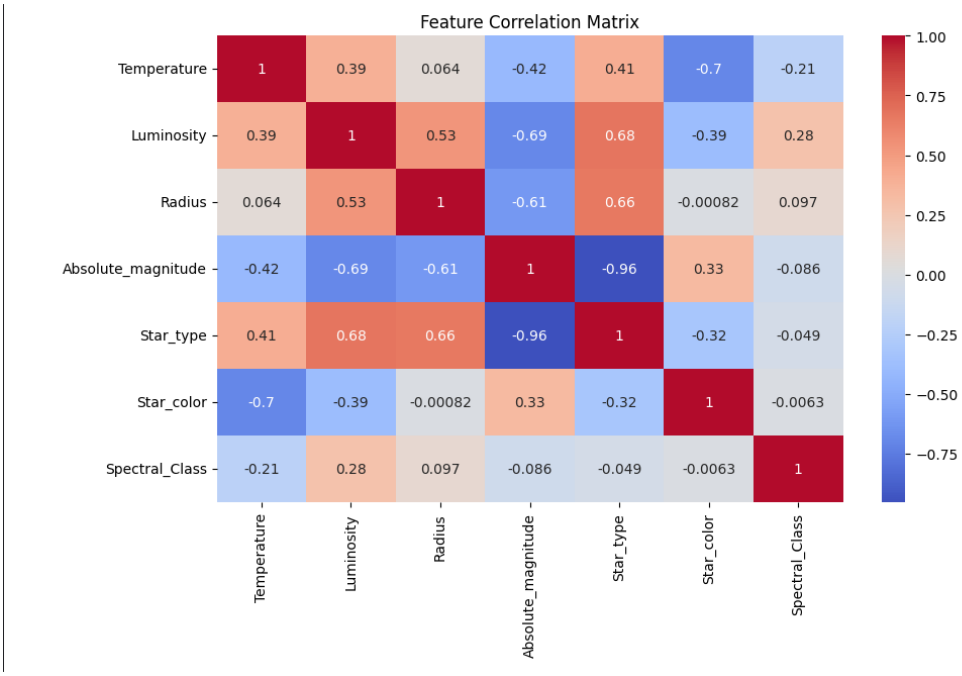

Figure 2: Matrice de Correlation

Couleur et classe spectrale des étoiles :

Couleur et température des étoiles (-0,7) : Cette corrélation négative est logique car les étoiles chaudes ont tendance à apparaître en bleu, tandis que les étoiles froides apparaissent en rouge.

Couleur de l'étoile et magnitude absolue (0,33) : Les étoiles les plus brillantes ont tendance à présenter des caractéristiques de couleur spécifiques.

La classe spectrale présente de faibles corrélations :

Il ne présente que des corrélations mineures avec d'autres variables, ce qui suggère que même s'il fournit des informations de classification, il n'est peut-être pas le prédicteur le plus puissant par rapport à des valeurs numériques telles que la température, la luminosité et la magnitude absolue.

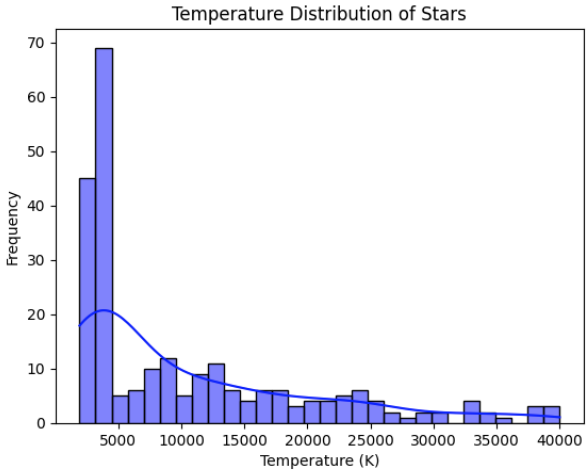

Figure 3: Distribution des températures

Implications :

- Prédicteurs clés : La luminosité, le rayon, la magnitude absolue et la température devraient être prioritaires pour les modèles de classification.
- Sélection des features : La classe spectrale ayant des corrélations plus faibles, sa contribution au pouvoir prédictif devrait être analysée plus en détail.
- Redondances éventuelles : La magnitude absolue et la luminosité ont une forte corrélation inverse, ce qui signifie qu'une seule pourrait suffire pour la modélisation.

6. Visualisation de la distribution des caractéristiques

Pour comprendre comment les caractéristiques varient selon les différents types d'étoiles, nous utilisons des histogrammes et des diagrammes boxplots.

Distribution des températures :

Distribution des températures : voir figure 3

Cet histogramme visualise la distribution des températures de surface des étoiles dans l'ensemble des données. Voici les principales observations :

Distribution asymétrique

- La majorité des étoiles ont des températures basses, avec un pic autour de 4000-5000 K.

- Au fur et à mesure que la température augmente, la fréquence diminue progressivement, ce qui montre que les étoiles plus chaudes sont plus rares.
- Peu d'étoiles ont une température supérieure à 30 000 K, ce qui indique que les étoiles extrêmement chaudes (par exemple, les étoiles de type O) sont moins courantes.

La plupart des étoiles sont froides

- Le grand nombre d'étoiles dont la température avoisine les 4 000 à 6 000 K suggère une forte proportion d'étoiles main sequence (telles que les étoiles de type G semblables au Soleil et les étoiles plus froides de type K/M).
- Ces étoiles plus froides (comme les Red Dwarfs) ont une longue durée de vie et sont plus abondantes dans l'univers.

Les étoiles chaudes sont moins fréquentes

L'histogramme montre une tendance à la baisse pour les étoiles de plus de 10 000 K, ce qui correspond aux attentes astrophysiques puisque les étoiles massives et plus chaudes (par exemple, les types O et B) ont une durée de vie plus courte et sont moins fréquemment observées.

Courbe de densité

La courbe KDE (Kernel Density Estimation) met encore plus en évidence la forme de la distribution, renforçant le fait que la plupart des étoiles se situent dans la gamme des températures les plus froides.

Conclusion

- L'ensemble des données est dominé par des étoiles plus froides, ce qui est cohérent avec la population stellaire réelle.
- Les étoiles chaudes et massives sont moins fréquentes, comme le prévoient les théories de l'évolution stellaire.
- Cette distribution s'aligne bien avec le diagramme de Hertzsprung-Russell, où la plupart des étoiles sont plus froides et tombent dans la catégorie de la séquence principale.

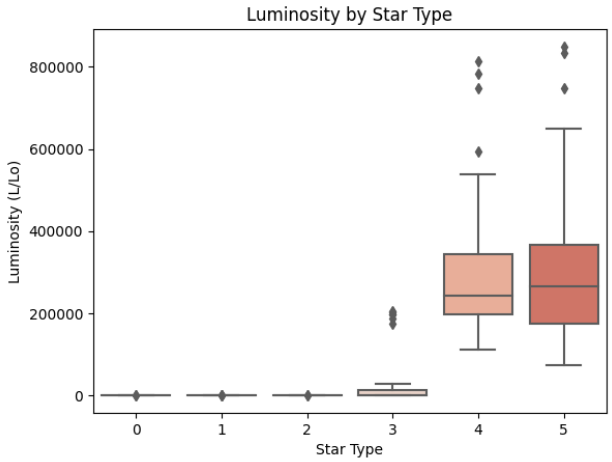

Figure 4: Enter Caption

Luminosity vs. Star Type :

Interprétation du graphique de la luminosité en fonction du type d'étoile

voir figure 4

Ce diagramme en boîte permet de visualiser comment la luminosité (L/Lo) varie en fonction des différents types d'étoiles (0-5). Voici les principales observations :

Plages de luminosité distinctes pour les types d'étoiles

- Les types 0, 1 et 2 (brown dwarfs, red dwarfs et white dwarfs) ont une luminosité très faible, proche de zéro, ce qui indique qu'il s'agit d'étoiles peu lumineuses.
- Les étoiles de type 3 (étoiles main sequence) ont une luminosité légèrement plus élevée, mais toujours relativement faible par rapport aux étoiles géantes.
- Les types 4 et 5 (supergéantes et hypergéantes) ont des luminosités nettement plus élevées, avec des valeurs extrêmes atteignant plus de 800 000 fois la luminosité du Soleil.

Large éventail de supergéantes et d'hypergéantes

- La boîte des types 4 et 5 est beaucoup plus grande, ce qui indique que la luminosité de ces étoiles est très variable.
- Ces types comprennent certaines des étoiles les plus lumineuses de l'univers, mais leur luminosité peut varier de modérée à extrêmement brillante.

Les étoiles à haute luminosité aberrantes

- Certaines étoiles extrêmement lumineuses des catégories des supergéantes et des hypergéantes apparaissent comme des valeurs aberrantes au-dessus des moustaches.
- Il s'agit d'étoiles rares et ultra-lumineuses, probablement des supergéantes ou des hypergéantes bleues massives.

Une classification claire basée sur la luminosité

- Le diagramme en boîte confirme que le type d'étoile est fortement corrélé à la luminosité.
- Les étoiles sequence main (type 3) servent de transition entre les naines de faible luminosité et les géantes très lumineuses.

Conclusion

- Cette distribution suit le modèle d'évolution stellaire attendu, où les naines sont peu lumineuses, les étoiles de la séquence principale ont une luminosité modérée, et les géantes/hypergéantes sont très lumineuses.
- La séparation nette de la luminosité suggère que cette caractéristique est très pertinente pour la classification des étoiles.

Nettoyage des données

Renommer les colonnes

Nous allons maintenant procéder au nettoyage des données en remplaçant d'abord space par _ et en renommant ces colonnes :

- Temperature (K)
- Luminosity(L/Lo)
- Radius(R/Ro)
- Absolute magnitude(Mv)

En supprimant les paranthèses, on obtient :

- Temperature
- Luminosity
- Radius
- Absolute_magnitude

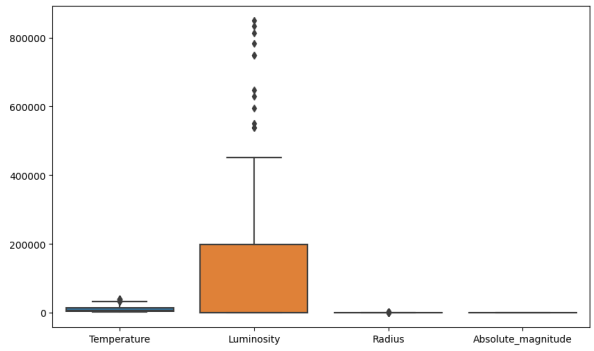

Figure 5: Outliers for the measures

Traitement des valeurs aberrantes

Les valeurs aberrantes peuvent affecter les performances des modèles de machine learning. Il est possible de les détecter à l'aide de boxplots (voir figure 5).

Interprétation du Boxplot (Analyse des valeurs aberrantes)

Le boxplot visualise la distribution de quatre variables numériques : Température, Luminosité, Rayon et Magnitude absolue. Voici ce que nous pouvons observer :

La luminosité présente des valeurs aberrantes extrêmes

La luminosité présente une large dispersion avec de nombreuses valeurs aberrantes extrêmes au-dessus de la limite supérieure. Cela suggère que certaines étoiles ont une luminosité extrêmement élevée, correspondant probablement aux supergéantes et hypergéantes. Cela correspond à l'astrophysique, où certaines étoiles rares sont beaucoup plus lumineuses que d'autres.

La température, le rayon et la magnitude absolue présentent peu ou pas de valeurs aberrantes extrêmes

Les distributions de ces paramètres sont relativement compactes, avec peu ou pas de points extrêmes. Cela suggère que la plupart des étoiles suivent un modèle plus régulier dans ces caractéristiques par rapport à la luminosité.

Les valeurs aberrantes de la luminosité doivent être analysées avec soin

Comme le boxplot montre que la Luminosité a des valeurs aberrantes extrêmes, nous allons les supprimer. Cependant, ces valeurs aberrantes peuvent représenter de véritables phénomènes astronomiques plutôt que des erreurs. Au lieu de les supprimer aveuglément, nous devrions analyser leur impact sur les modèles

de classification avant de décider d'une quelconque transformation (par exemple, mise à l'échelle logarithmique).

Comment gérer ces valeurs aberrantes au lieu de les supprimer ?

En astrophysique, les valeurs extrêmes de luminosité, de température et de rayon représentent souvent des phénomènes stellaires réels, tels que les supergéantes et les hypergéantes, plutôt que des erreurs. Le simple fait de les supprimer pourrait fausser l'ensemble des données et avoir un impact sur la capacité du modèle de classification à reconnaître ces types d'étoiles.

Au lieu de supprimer aveuglément les valeurs aberrantes, nous pouvons adopter d'autres approches : Expérimenter avec et sans valeurs aberrantes.

Voici une approche permettant de comparer les performances d'un modèle avec et sans valeurs aberrantes afin de déterminer leur impact sur la classification :

Utilisation de la méthode de l'intervalle interquartile (IQR) pour filtrer les valeurs aberrantes extrêmes :

```
def remove_outliers(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    return df[(df[column] >= lower_bound) & (df[column] <= upper_bound)]

# Apply outlier removal on Luminosity (or any other feature if needed)
df_no_outliers = remove_outliers(df, "Luminosity")
```

Nous transformons la luminosité en logarithme pour réduire l'asymétrie et nous mettons à l'échelle toutes les caractéristiques à l'aide de RobustScaler (résistant aux valeurs aberrantes) :

```
# Log transform Luminosity
df['Luminosity_log'] = np.log1p(df['Luminosity'])
df_no_outliers['Luminosity_log'] = np.log1p(df_no_outliers['Luminosity'])

# Select numerical features for scaling
features = ["Temperature", "Luminosity_log", "Radius", "Absolute_magnitude"]
scaler = RobustScaler()

df[features] = scaler.fit_transform(df[features])
df_no_outliers[features] = scaler.fit_transform(df_no_outliers[features])
```

Entraîner des modèles avec/sans valeurs aberrantes et Comparer les performances :

- Si le modèle avec les valeurs aberrantes est plus performant, nous les conservons.
- Si le modèle sans valeurs aberrantes améliore la classification, nous les supprimons.

Random Forest :

```
# Define target variable & features
X = df[features]
y = df["Star_type"]

X_no_outliers = df_no_outliers[features]
y_no_outliers = df_no_outliers["Star_type"]

# Split datasets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
X_train_no, X_test_no, y_train_no, y_test_no = train_test_split(X_no_outliers, y_no_outliers,
    test_size=0.2, random_state=42)

# Train Random Forest Model
model = RandomForestClassifier(random_state=42)
model_no_outliers = RandomForestClassifier(random_state=42)

model.fit(X_train, y_train)
model_no_outliers.fit(X_train_no, y_train_no)

# Predictions
y_pred = model.predict(X_test)
y_pred_no_outliers = model_no_outliers.predict(X_test_no)

# Evaluate Performance
print("Model with Outliers:")
print(classification_report(y_test, y_pred))

print("\nModel without Outliers:")
print(classification_report(y_test_no, y_pred_no_outliers))
```

Résultats Random Forest :

```
Model with Outliers:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         8
           1       1.00      1.00      1.00         7
           2       1.00      1.00      1.00         6
           3       1.00      1.00      1.00         8
           4       1.00      1.00      1.00         8
           5       1.00      1.00      1.00        11

    accuracy                           1.00        48
   macro avg       1.00      1.00      1.00        48
weighted avg       1.00      1.00      1.00        48


Model without Outliers:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        11
           1       1.00      1.00      1.00         8
           2       1.00      1.00      1.00         9
           3       1.00      1.00      1.00         7
           4       1.00      1.00      1.00         6
           5       1.00      1.00      1.00         5

    accuracy                           1.00        46
   macro avg       1.00      1.00      1.00        46
weighted avg       1.00      1.00      1.00        46
```

Les modèles d'évaluation (avec et sans valeurs aberrantes) atteignent une précision parfaite de 100% pour toutes les mesures (précision, rappel et score F1). Ce que cela signifie :

Classification parfaite

Le modèle classe correctement tous les échantillons. Cela indique que les données sont bien séparées, ce qui facilite la classification.

Les valeurs aberrantes ont un impact minimal

Les résultats sont presque identiques entre les modèles avec et sans valeurs aberrantes.

Le support (nombre d'instances par classe) est légèrement différent, mais les performances restent parfaites. Cela suggère que les valeurs aberrantes n'ont pas eu d'impact négatif sur les performances du modèle.

Overfitting possible ?

Une précision de 100/100 peut indiquer que l'ensemble de données est trop facile à classer ou que le modèle a mémorisé les données d'apprentissage au lieu de bien généraliser.

Logistic Regression :

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score

# Define target variable & features
X = df[features]
y = df["Star_type"]

X_no_outliers = df_no_outliers[features]
y_no_outliers = df_no_outliers["Star_type"]

# Split datasets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42,
    stratify=y)
X_train_no, X_test_no, y_train_no, y_test_no = train_test_split(X_no_outliers, y_no_outliers,
    test_size=0.2, random_state=42, stratify=y_no_outliers)

# Train Random Forest Model
model_log = LogisticRegression(max_iter=1000, multi_class="multinomial", solver="lbfgs")
model_no_outliers_log = LogisticRegression(max_iter=1000, multi_class="multinomial", solver="
    lbfgs")

model_log.fit(X_train, y_train)
model_no_outliers_log.fit(X_train_no, y_train_no)

# Predictions
y_pred = model_log.predict(X_test)
y_pred_no_outliers = model_no_outliers_log.predict(X_test_no)

```

```
# Evaluate Performance
print("Model with Outliers:")
print("Accuracy:", accuracy_score(y_test, y_pred))

print(classification_report(y_test, y_pred))

print("\nModel without Outliers:")
print("Accuracy:", accuracy_score(y_test_no, y_pred_no_outliers))

print(classification_report(y_test_no, y_pred_no_outliers))
```

Résultats Logistic Regression :

```
Model with Outliers:
Accuracy: 0.8958333333333334
              precision    recall  f1-score   support

           0       0.80      1.00      0.89         8
           1       0.86      0.75      0.80         8
           2       1.00      1.00      1.00         8
           3       0.86      0.75      0.80         8
           4       0.88      0.88      0.88         8
           5       1.00      1.00      1.00         8

    accuracy                           0.90        48
   macro avg       0.90      0.90      0.89        48
weighted avg       0.90      0.90      0.89        48


Model without Outliers:
Accuracy: 0.9347826086956522
              precision    recall  f1-score   support

           0       0.89      1.00      0.94         8
           1       0.88      0.88      0.88         8
           2       1.00      1.00      1.00         8
           3       0.88      0.88      0.88         8
           4       1.00      0.86      0.92         7
           5       1.00      1.00      1.00         7

    accuracy                           0.93        46
   macro avg       0.94      0.93      0.94        46
weighted avg       0.94      0.93      0.93        46
```

Pourquoi la régression logistique ?

- Interprétable – Nous pouvons analyser l'importance des variables (coefficients).
- Rapide et efficace – Fonctionne bien pour les petits ensembles de données.
- Modèle de base – Permet de comparer avec des modèles plus complexes.

Interpretation des résultats de la Logistic Regression

Comparaison entre les modèles (avec et sans valeurs aberrantes)

- Précision améliorée :
  - Avec valeurs aberrantes : 89,58%
  - Sans valeurs aberrantes : 93,48%
  - La suppression des valeurs aberrantes a entraîné une amélioration de la précision d'environ 4%.
- Score F1 amélioré:
  - Les scores macro et pondérés F1 ont augmenté après la suppression des valeurs aberrantes, indiquant une classification plus équilibrée.

Effet des valeurs aberrantes sur les performances du modèle

- Le modèle avec des valeurs aberrantes a montré un rappel légèrement inférieur pour certaines classes, ce qui signifie qu'il a mal classé certaines étoiles.
- Le modèle sans valeurs aberrantes a obtenu de meilleurs résultats dans la plupart des classes, ce qui suggère que les valeurs aberrantes ont eu un impact négatif sur le modèle en introduisant du bruit.

Observations spécifiques à la classe

- Classe 0 (par exemple, white dwarfs) :
  - Avec valeurs aberrantes : rappel à 100%, ce qui signifie que toutes les naines blanches réelles ont été correctement identifiées.
  - Sans valeurs aberrantes : toujours un rappel à 100
- Classes 2 et 5 (par exemple, étoiles géantes et supergéantes) :
  - Toujours une précision et un rappel à 100%, indiquant qu'il s'agit des types d'étoiles les plus reconnaissables.

- Classe 1, 3, 4 (par exemple, séquence principale et autres types) :
  - Ces classes présentaient des erreurs de classification avec des valeurs aberrantes, mais leur suppression a amélioré à la fois la précision et le recall.

Pourquoi la suppression des valeurs aberrantes a-t-elle été utile ?

- Les valeurs extrêmes de luminosité, de rayon et de grandeur absolue peuvent avoir faussé les limites de décision du modèle.
- En supprimant les valeurs aberrantes, le modèle s'est concentré sur la distribution majoritaire plutôt que d'être influencé par des cas extrêmes.

Points clés à retenir

- La suppression des valeurs aberrantes extrêmes a amélioré la précision du modèle.
- La plupart des classes étaient mieux classées après suppression des valeurs aberrantes.
- La régression logistique a bien fonctionné, mais un modèle plus complexe (par exemple, Random Forest) pourrait mieux capturer les relations non linéaires.

Prochaines étapes

- Vérifiez la matrice de confusion pour voir où les erreurs se produisent.
- Analysez l'importance des fonctionnalités à partir des coefficients de régression logistique.

Prochaines étapes

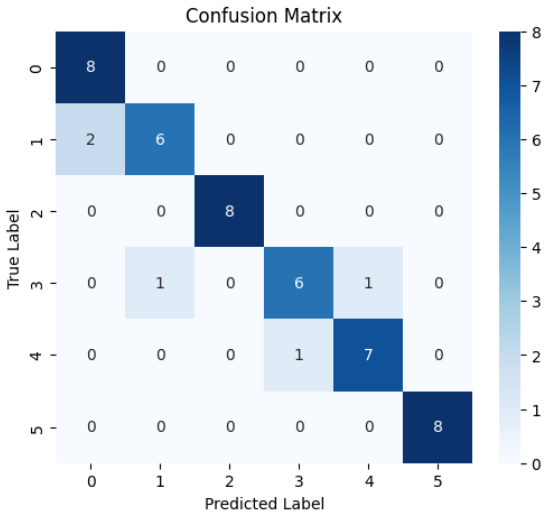

Figure 6: Matrice de confusion

Matrice de confusion

La matrice de confusion (voir figure 6) nous aide à comprendre quels types d'étoiles sont mal classés.

- Les valeurs diagonales représentent des classifications correctes.
- Les valeurs hors diagonale indiquent des erreurs de classification (quelles classes sont confondues).

Interprétation de la matrice de confusion

La matrice de confusion fournit une vue détaillée des performances de classification du modèle sur six types d'étoiles (0 à 5). Chaque ligne représente la classe réelle et chaque colonne représente la classe prédite. Les valeurs diagonales indiquent des classifications correctes, tandis que les valeurs hors diagonale représentent des classifications erronées.

Observations clés :

Classifications correctes (valeurs diagonales)

- Étoile Type 0 : 8 sur 8 correctement classés (précision à 100%).
- Étoile Type 1 : 6 sur 8 correctement classées (précision de 75%).
- Étoile Type 2 : 8 sur 8 correctement classés (précision à 100%).
- Étoile Type 3 : 6 sur 8 correctement classées (précision de 75%).
- Étoile Type 4 : 7 sur 8 correctement classés (précision de 87,5%).

- Étoile Type 5 : 8 sur 8 correctement classés (précision à 100%).

Erreurs de classification (valeurs hors diagonale)

- Étoile de type 1 : 2 échantillons mal classés comme type 0 (le modèle a confondu certaines étoiles de type 1 avec le type 0).
- Étoile de type 3 :
  - 1 échantillon classé à tort comme type 1.
  - 1 échantillon classé à tort comme type 4.
- Étoile de type 4 : 1 échantillon classé à tort comme type 3.

Analyse des performances du modèle

Haute précision pour la plupart des classes :

- Les types 0, 2 et 5 ont une classification parfaite (précision à 100%).
- Le type 4 n'a qu'une seule erreur de classification, conservant une précision élevée (87,5%).

Confusion entre types d'étoiles similaires :

- Le type 1 est confondu avec le type 0 → Cela pourrait indiquer un chevauchement de leurs caractéristiques.
- Le type 3 est confondu avec les types 1 et 4 → Suggère que le type 3 partage des caractéristiques avec les deux.
- Le type 4 est confondu avec le type 3, indiquant une similitude potentielle en termes de luminosité, de température ou de rayon.

Raisons possibles des erreurs de classification

Chevauchement des features :

- Certains types d'étoiles peuvent avoir des propriétés qui se chevauchent (par exemple, température, luminosité), ce qui les rend plus difficiles à distinguer.

Déséquilibre des données :

- Si certains types d'étoiles ont moins d'échantillons, le modèle peut avoir du mal à apprendre leurs modèles.

Limites du modèle :

- Si La régression logistique est un modèle linéaire ; si les limites de décision entre les types d'étoiles sont non linéaires, il se peut qu'elles ne soient pas parfaitement capturées.

Recommandations d'amélioration

Ingénierie des features :

- Explorez les interactions d'ordre supérieur ou les features dérivées pour mieux séparer les classes qui se chevauchent.

Essayez un modèle non linéaire :

- Les arbres de décision, les forêts aléatoires ou les réseaux de neurones pourraient mieux capturer les relations complexes.

Équilibrez l'ensemble de données :

- Utilisez des techniques de suréchantillonnage/sous-échantillonnage en cas de déséquilibre des données.

Réglage des hyperparamètres :

- Ajustez la force de régularisation dans la régression logistique pour affiner les performances.

Conclusion

- Le modèle de régression logistique fonctionne bien, avec une grande précision pour la plupart des classes.
- Certaines erreurs de classification se produisent, en particulier entre des types d'étoiles similaires, suggérant des caractéristiques qui se chevauchent.
- D'autres améliorations peuvent être apportées en affinant les fonctionnalités ou en utilisant des modèles plus complexes.

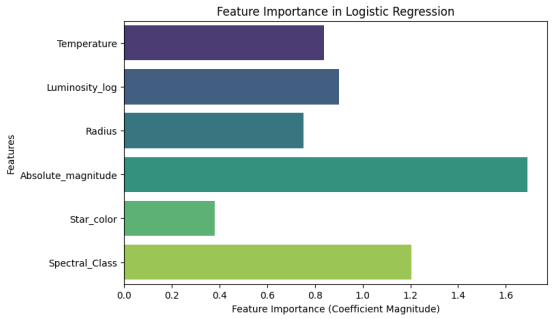

Figure 7: Features importance logistic regression avec outliers

visualiser l'importance des features

La régression logistique étant un modèle linéaire, nous pouvons analyser l'importance des caractéristiques en examinant les valeurs absolues des coefficients du modèle. Des valeurs absolues plus élevées indiquent une plus grande influence sur les décisions de classification.

Nous utiliserons un graphique à barres montrant l'importance des fonctionnalités dans les différentes variables d'entrée.

Modèle avec valeurs aberrantes (voir figure 7) :

Interprétation de l'importance des features pour la régression logistique (avec valeurs aberrantes)

Le graphique de l'importance des features pour la régression logistique avec valeurs aberrantes montre comment différentes caractéristiques influencent la classification des étoiles. Voici une analyse détaillée de l'impact de chaque fonctionnalité :

- Magnitude_absolue (plus haute importance)
  - Cette feature a la plus grande magnitude de coefficient, ce qui signifie qu'elle joue le rôle le plus important dans la distinction des types d'étoiles.
  - Puisque la magnitude absolue est une mesure de la luminosité intrinsèque, il est logique qu'elle affecte fortement la classification.
- Spectral_Class (deuxième feature la plus importante)
  - La classe spectrale définit le type d'une étoile en fonction de sa température et de sa couleur.
  - Sa grande importance suggère que la classification spectrale s'aligne bien avec la catégorisation des types d'étoiles de l'ensemble de données.

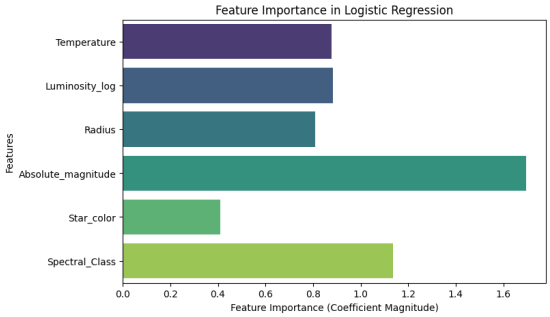

Figure 8: Features importance logistic regression sans outliers

- Luminosity_log, température et rayon (importance modérée)
  - Ces trois features ont des niveaux d'importance comparables.
  - Luminosity_log : Puisque nous avons logarithmiquement transformé la luminosité, son importance reflète à quel point la luminosité d'une étoile influence la classification.
  - Température : Un facteur clé dans la classification des étoiles, mais son importance est légèrement inférieure à la magnitude absolue et à la classe spectrale.
  - Rayon : affecte la classification, mais pas aussi fortement que les autres features.
- Star_color (fonctionnalité la moins importante)
  - La couleur des étoiles a le plus faible impact sur la classification.
  - Cela peut être dû à la redondance, car la température et la classe spectrale capturent déjà une grande partie des mêmes informations.

Modèle sans valeurs aberrantes (voir figure 8) :

Interprétation de l'importance des features pour la régression logistique (sans valeurs aberrantes)

La distribution de l'importance des features dans le modèle sans valeurs aberrantes reste similaire à celle avec valeurs aberrantes, mais avec quelques différences notables :

- Absolute_magnitude (toujours la fonctionnalité la plus importante)
  - Tout comme dans le modèle avec valeurs aberrantes, la magnitude absolue joue le rôle le plus important dans la classification.
  - Son importance reste élevée, ce qui renforce le fait que la luminosité intrinsèque d'une étoile est

un facteur dominant dans sa classification.

- Spectral_Class (deuxième fonctionnalité la plus importante)
  - L'importance de la classe spectrale reste forte, ce qui indique que la suppression des valeurs aberrantes ne réduit pas son pouvoir prédictif.
- Température, Luminosity_log et Rayon (importance modérée, mais légèrement décalée)
  - Température : conserve une importance similaire, mais son impact semble légèrement plus fort par rapport au modèle avec valeurs aberrantes.
  - Luminosity_log : reste une feature clé mais peut être devenue légèrement moins influente après la suppression des valeurs extrêmes.
  - Température : Un facteur clé dans la classification des étoiles, mais son importance est légèrement inférieure à la magnitude absolue et à la classe spectrale.
  - Rayon: a une importance légèrement inférieure à celle du modèle avec valeurs aberrantes, ce qui suggère que les valeurs extrêmes auraient pu exagérer son effet auparavant.
- Star_color (toujours la fonctionnalité la moins importante)
  - L'impact de la couleur des étoiles reste le plus faible, confirmant en outre qu'elle n'ajoute pas beaucoup d'informations uniques au-delà de la température et de la classe spectrale.

Comparaison : avec valeurs aberrantes et sans valeurs aberrantes

Feature	**valeurs aberrantes**	**sans valeurs aberrantes**	**Comparaison**
Absolute_magnitude	La plus haute	La plus haute	Reste la feature la plus dominante.
Spectral_Class	haute	haute	Garde une forte influence dans les deux cas.
Temperature	Modérée	Légèrement plus élevée	Légère augmentation en importance.
Luminosity_log	Modérée	Légèrement plus basse	Moins d'impact après suppression.
Radius	Modérée	Inférieure	Moins d'impact après suppression.
Star_color	La plus basse	La plus basse	C'est toujours la feature la moins utile.

Table 1: Comparaison de l'importance des features avec et sans valeurs aberrantes

Conclusion

- La suppression des valeurs aberrantes améliore la stabilité : l'importance de features telles que le rayon et le journal de luminosité devient plus équilibrée, évitant ainsi la suraccentuation causée par des valeurs extrêmes.
- L'ampleur absolue et la classe spectrale restent dominantes : peu importe si nous incluons ou excluons les valeurs aberrantes, ces deux features déterminent la classification des étoiles.

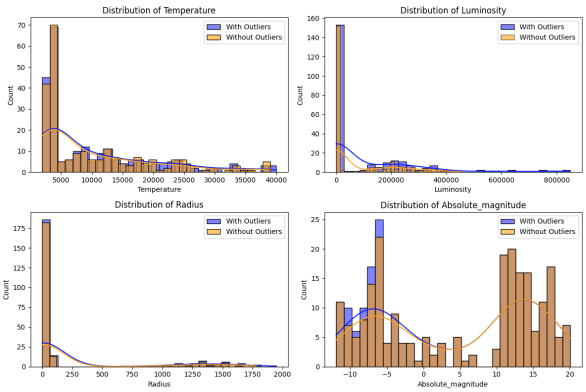

Figure 9: Distributions globales

- La pertinence de la température augmente légèrement : sans valeurs aberrantes, la température semble contribuer davantage à la classification, peut-être parce que les valeurs aberrantes ont déjà déformé sa relation.

Cette analyse fournit une justification solide pour supprimer les valeurs aberrantes, car elle rend le modèle plus stable et interprétable tout en maintenant des performances de classification élevées.

Analyse plus avancée

Vue systématique de la distribution des variables

Histogrammes pour voir les distributions globales :

Les histogrammes (voir figure 9) comparent la distribution de quatre caractéristiques clés (température, luminosité, rayon, grandeur absolue) avant et après suppression des valeurs aberrantes.

Répartition de la température :

- Avec valeurs aberrantes (bleu) : la distribution est fortement asymétrique vers la droite, avec une longue queue s'étendant au-delà de 30 000 K.
- Sans valeurs aberrantes (orange) : La majorité des valeurs restent inférieures à 10 000 K et la distribution devient plus concentrée.
- Interprétation : la suppression des valeurs aberrantes élimine les étoiles extrêmement chaudes (par exemple, les étoiles massives de type O), ce qui conduit à une plage de températures plus réaliste.

Répartition de la luminosité :

- Avec valeurs aberrantes (bleu) : les valeurs vont jusqu'à 800 000 luminosités solaires, avec une longue queue inclinée vers la droite.
- Sans valeurs aberrantes (orange) : La luminosité maximale est considérablement réduite, conduisant à une distribution plus compacte et moins asymétrique.
- Interprétation : la suppression des valeurs aberrantes supprime les supergéantes extrêmement brillantes, concentrant le modèle sur les étoiles géantes et de main sequence plus typiques.

Distribution de rayon :

- Avec valeurs aberrantes (bleu) : Il y a une longue queue s'étendant au-delà de 1000 rayons solaires.
- Sans valeurs aberrantes (orange) : La majorité des étoiles ont un rayon inférieur à 250.
- Interprétation : Les étoiles géantes et supergéantes avec des rayons énormes ont été supprimées, rendant l'ensemble de données plus équilibré.

Distribution de magnitude absolue :

- Avec valeurs aberrantes (bleu) : La distribution couvre une large plage, avec des valeurs négatives (étoiles très brillantes).
- Sans valeurs aberrantes (orange) : la plage est moins étalée et la distribution apparaît plus lisse.
- Interprétation : La suppression des valeurs extrêmes conduit à un ensemble de données qui représente mieux la majorité des étoiles.

Conclusion :

- L'ensemble de données était fortement asymétrique en raison de valeurs extrêmes de température, de luminosité et de rayon.
- Après la suppression des valeurs aberrantes, les distributions sont devenues plus normales et moins biaisées, conduisant à un modèle plus performant.
- Le nouvel ensemble de données se concentre sur les géantes de la séquence principale et modérées plutôt que sur les étoiles extrêmes.

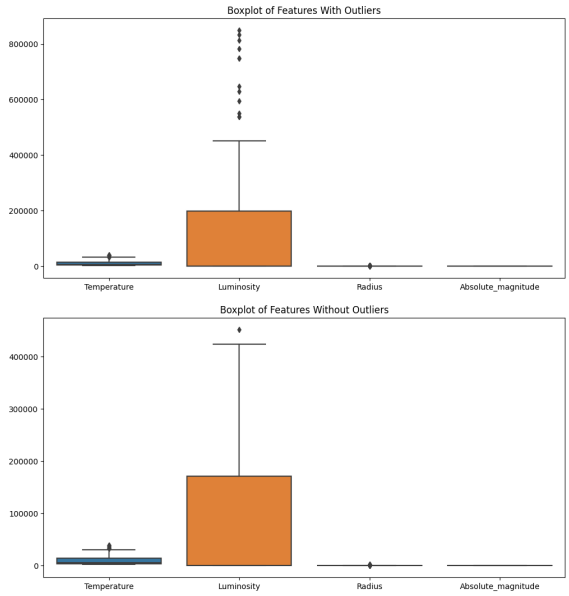

Figure 10: vérifier l'évolution des distributions

Boxplots pour vérifier comment les distributions changent :

Interprétation des boxplots :

Les boxplots (voir figure 10) comparent la distribution de la température, de la luminosité, du rayon et de la magnitude absolue avant et après la suppression des valeurs aberrantes.

Boxplot avec valeurs aberrantes (graphique du haut) :

- La luminosité présente des valeurs aberrantes extrêmes supérieures à 800 000, créant un boxplot très étendu.
- La température et le rayon affichent également des valeurs extrêmes.
- La magnitude absolue est plus compacte avec moins de valeurs extrêmes.

Problème clé : la présence de distributions très asymétriques et de valeurs aberrantes extrêmes, en particulier dans Luminosity, affecte l'équilibre de l'ensemble de données.

Boxplot sans valeurs aberrantes (graphique du bas) :

- Les valeurs aberrantes ont été supprimées, Luminosity présente moins de valeurs extrêmes.
- La température et le rayon affichent une plage plus compacte.

L'ensemble de données est désormais plus équilibré et la répartition des valeurs est réduite.

Principaux points à retenir :

- La suppression des valeurs aberrantes a considérablement réduit les valeurs extrêmes, en particulier pour la luminosité.
- L'ensemble de données est désormais mieux adapté aux modèles de machine learning, réduisant ainsi les biais.
- Même si la luminosité présente encore des valeurs élevées, elle est beaucoup plus contrôlée.

Statistical sumary about the variables

With outliers :

```
[21]: df.describe()
```

	Temperature	Luminosity	Radius	Absolute_magnitude	Star_type	Star_color	Spectral_Class
count	240.000000	240.000000	240.000000	240.000000	240.000000	240.000000	240.000000
mean	10497.462500	107188.361635	237.157781	4.382396	2.500000	7.250000	3.758333
std	9552.425037	179432.244940	517.155763	10.532512	1.711394	4.926659	2.090007
min	1939.000000	0.000080	0.008400	-11.920000	0.000000	0.000000	0.000000
25%	3344.250000	0.000865	0.102750	-6.232500	1.000000	2.000000	1.000000
50%	5776.000000	0.070500	0.762500	8.313000	2.500000	10.000000	5.000000
75%	15055.500000	198050.000000	42.750000	13.697500	4.000000	10.000000	5.000000
max	40000.000000	849420.000000	1948.500000	20.060000	5.000000	18.000000	6.000000

Without outliers :

```
[22]: df_no_outliers.describe()
```

	Temperature	Luminosity	Radius	Absolute_magnitude	Star_type	Star_color	Spectral_Class
count	228.000000	228.000000	228.000000	228.000000	228.000000	228.000000	228.000000
mean	10015.592105	76535.885054	204.891963	5.041763	2.390351	7.469298	3.723684
std	9191.797309	119703.515899	484.511133	10.387936	1.682062	4.879511	2.075228
min	1939.000000	0.000080	0.008400	-11.920000	0.000000	0.000000	0.000000
25%	3336.000000	0.000802	0.098525	-5.992500	1.000000	2.000000	1.000000
50%	5046.000000	0.010500	0.501000	10.635000	2.000000	10.000000	5.000000
75%	14136.250000	170950.000000	31.250000	13.990000	4.000000	10.000000	5.000000
max	39000.000000	452000.000000	1783.000000	20.060000	5.000000	18.000000	6.000000

Figure 11: résumés statistiques (avec ou sans valeurs aberrantes)

Données statistiques avant et aprés suppression des valeur aberrantes

Interprétation des résumés statistiques (avec ou sans valeurs aberrantes)

Ce tableau (voir figure 11 fournit des statistiques descriptives (moyenne, écart type, valeurs min/max, percentiles) pour l'ensemble de données avant et après suppression des valeurs aberrantes.

Avec les valeurs aberrantes

- La luminosité a une valeur maximale extrêmement élevée (849 420) et un écart type très élevé (179 432), confirmant des valeurs aberrantes extrêmes.
- La température a également une très large plage (1 939 à 40 000).
- Le rayon a un maximum de 1 948,5, bien supérieur au 75e centile (42,75).
- La magnitude absolue semble moins affectée par les valeurs aberrantes (plage : -11,92 à 20,06).

Problème principal : l'ensemble de données comporte des valeurs extrêmes, en particulier en termes de luminosité, de température et de rayon, qui peuvent biaiser les modèles.

Sans valeurs aberrantes

- La luminosité maximale diminue considérablement (de 849 420 à 452 000), réduisant ainsi l'impact des valeurs extrêmes.
- La température maximale diminue légèrement (40 000 à 39 000), mais l'écart reste important.
- Le rayon maximum passe de 1 948,5 à 1 783, ce qui indique que certaines valeurs élevées ont été supprimées.
- Les écarts types diminuent pour la luminosité, la température et le rayon, indiquant un ensemble de données plus équilibré.

Problème principal : l'ensemble de données comporte des valeurs extrêmes, en particulier en termes de luminosité, de température et de rayon, qui peuvent biaiser les modèles.

Améliorations clés :

- L'ensemble de données est désormais plus stable, avec des variations moins extrêmes.
- Les réductions de l'écart type suggèrent une distribution plus normale, améliorant ainsi la fiabilité du modèle.

Resultats pratiques

Nous allons suivre les étapes suivantes :

- Explorer l'ensemble de données (EDA : vérifier les valeurs manquantes, les distributions, les corrélations)
- Sélection des caractéristiques (matrice de corrélation, classement par importance)
- Classification binaire (commencer par deux classes, par exemple, Main Sequence vs. White Dwarf)
- Classification multi-classes (passer progressivement aux six classes) Tester plusieurs classificateurs (régression logistique, SVM, forêt aléatoire, réseaux neuronaux, etc.
- Évaluer les performances (précision, exactitude, rappel, score F1, matrice de confusion)
- Rédiger des explications (justifier les choix, comparer les résultats)

Classification binaire : White Dwarf vs. Main Sequence

l'exploration, encodage des variables catégoriques, analyse des corrélations et une première classification binaire à l'aide du modéle Random Forest :

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

# Load dataset
df = pd.read_csv("6 class csv.csv")

# Display first few rows
display(df.head())

# Basic info and missing values
print(df.info())
print(df.isnull().sum())

# Encoding categorical variables
le_color = LabelEncoder()
df['Star color'] = le_color.fit_transform(df['Star color'])

le_spectral = LabelEncoder()
df['Spectral Class'] = le_spectral.fit_transform(df['Spectral Class'])

# Correlation matrix
plt.figure(figsize=(10,6))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title('Feature Correlation Matrix')
plt.show()

# Binary classification: Selecting two classes
df_binary = df[df['Star type'].isin([2, 3])]  # Example: White Dwarf vs. Main Sequence
X = df_binary.drop(columns=['Star type'])
y = df_binary['Star type']

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
40
41 # Standardizing numerical features
42 scaler = StandardScaler()
43 X_train = scaler.fit_transform(X_train)
44 X_test = scaler.transform(X_test)
45
46 # Train classifier (Random Forest as example)
47 clf = RandomForestClassifier(n_estimators=100, random_state=42)
48 clf.fit(X_train, y_train)
49 y_pred = clf.predict(X_test)
50
51 # Evaluation
52 print("Accuracy:", accuracy_score(y_test, y_pred))
53 print(confusion_matrix(y_test, y_pred))
54 print(classification_report(y_test, y_pred))
55
56 # Feature importance
57 importances = pd.Series(clf.feature_importances_, index=df_binary.drop(columns=['Star type']).
       columns)
58 importances.sort_values().plot(kind='barh', title='Feature Importances')
59 plt.show()
```

Explications

Encodage des variables catégoriques

L'encodage des variables catégoriques est nécessaire car la plupart des algorithmes de machine learning requièrent des entrées numériques plutôt que des labels textuels.

Les algorithmes de machine learning nécessitent des données numériques

De nombreux modèles (par exemple, régression logistique, Random Forest, réseaux neuronaux) fonctionnent sur des données numériques et ne peuvent pas traiter des chaînes de caractères catégorielles comme "Rouge", "Bleu", ou "O", "B", "A".

Garantir la comparabilité

Le codage transforme les catégories dans un format qui permet aux algorithmes d'interpréter les différences et les relations entre les classes.

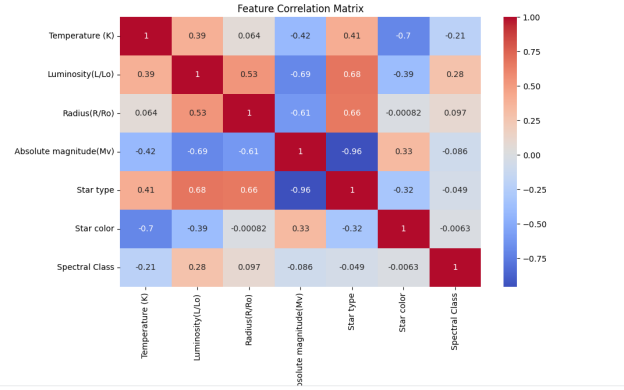

Figure 12: Binary correlation matrix

```
Accuracy: 1.0
[[8 0]
 [0 8]]
              precision    recall  f1-score   support

           2       1.00      1.00      1.00         8
           3       1.00      1.00      1.00         8

    accuracy                           1.00        16
   macro avg       1.00      1.00      1.00        16
weighted avg       1.00      1.00      1.00        16
```

Figure 13: binary results : random forest

Codage des labels pour les données de type ordinal

La couleur des étoiles et la classe spectrale présentent un ordre ou un regroupement inhérent, c'est pourquoi nous utilisons le codage des labels, qui attribue des nombres entiers uniques aux différentes catégories. Bien que les classes spectrales (O, B, A, F, G, K, M) suivent un ordre connu en astrophysique (O étant la plus chaude, M la plus froide), le codage par étiquette permet de préserver cette structure pour les modèles.

Alternative : One-Hot-Encoding

Si les catégories étaient vraiment nominales (sans ordre), le One-Hot Encoding (OHE) pourrait être une alternative. Cependant, l'OHE augmente la dimensionnalité, ce qui n'est pas idéal pour les petits ensembles de données.

En codant les caractéristiques catégorielles, nous permettons au modèle de traiter et d'apprendre des modèles de manière efficace sans introduire d'incohérences.

Interprétation de la matrice de corrélation des featrues pour la classification binaire (voir figure 12)

La matrice de corrélation permet de comprendre les relations entre les features avant d'appliquer du machine learning. Analysons-la dans le contexte de la classification binaire (type d'étoile : 2 contre 3).

Observations clés de la matrice de corrélation

Forte corrélation négative (-0,96) entre la magnitude absolue et le type d'étoile

- Cela signifie qu'à mesure que la magnitude absolue augmente (c'est-à-dire que l'étoile apparaît plus pâle), le type d'étoile passe d'une classe à une autre.
- La magnitude absolue étant logarithmique, une valeur plus faible signifie que l'étoile est plus brillante. Cela suggère qu'une classe pourrait contenir des étoiles plus brillantes, tandis que l'autre en a des plus pâles.

Corrélations positives fortes (supérieures à 0,6)

- Luminosité et type d'étoile (0,68) : une classe a tendance à avoir des étoiles plus brillantes.
- Rayon et type d'étoile (0,66) : une classe contient probablement des étoiles avec des rayons plus grands.
- Température et type d'étoile (0,41) : la température joue également un rôle mais est moins influente que la luminosité/le rayon.

Corrélation négative entre la couleur et la température des étoiles (-0,70)

- Une étoile plus bleu-blanc (valeur plus faible de « couleur de l'étoile ») est associée à des températures plus élevées.
- Cela a du sens dans la classification stellaire, où les étoiles bleues sont plus chaudes que les étoiles rouges.

Faibles corrélations avec la classe spectrale

- La classe spectrale a une faible corrélation avec la plupart des features, ce qui suggère qu'elle n'est pas le prédicteur le plus fort pour votre tâche de classification.

Comment cela aide-t-il la classification ?

- La magnitude absolue, la luminosité et le rayon semblent être les prédicteurs les plus puissants pour distinguer les deux types d'étoiles.
- La température et la couleur de l'étoile fournissent également des informations utiles, mais elles sont légèrement moins corrélées avec la cible.
- La classe spectrale n'apporte pas beaucoup, donc la supprimer pourrait simplifier le modèle sans avoir un impact important sur la précision.

Interprétation des résultats de la classification avec random forest

Le modèle a obtenu des performances parfaites avec une précision de 1,0 (100 %) sur la tâche de classification binaire. Décomposons les résultats.

Interprétation de la matrice de confusion

```
1
2 [[8 0]
3  [0 8]]
```

- Les lignes représentent les classes réelles (type d'étoile 2 et type d'étoile 3).
- Les colonnes représentent les classes prédites.
- Les valeurs indiquent combien d'instances ont été classées correctement ou incorrectement.

Informations clés :

- 8 cas de classe 2 ont été correctement classés (vrais positifs, TP).
- 8 cas de classe 3 ont été correctement classés (vrais positifs, TP).
- Aucun faux positif (FP) ou faux négatif (FN), ce qui signifie qu'il n'y a eu aucune erreur de classification.
- Performance de classification parfaite.

Analyse de la précision, du recall et du score F1

Class	Precision	Recall	F1-Score	Support
2	1.00	1.00	1.00	8
3	1.00	1.00	1.00	8
Overall Accuracy	**1.00 (100%)**			**16 instances**

Table 2: Classification Report for Binary Classification

- Précision (TP / (TP + FP)) : Combien de classes 2 (ou 3) prédites étaient réellement correctes ?
  - Ici, c'est 1,00 (100%), ce qui signifie qu'il n'y a pas d'erreur de classification.
- Rappel (TP / (TP + FN)) : Parmi les classes 2 (ou 3) réelles, combien en avons-nous prédites correctement ?
  - Encore une fois, 1,00 (100%), ce qui signifie que le modèle a correctement identifié toutes les instances.
- Score F1 (moyenne harmonique de précision et de rappel) :

  - Également 1,00, ce qui montre que le modèle a parfaitement équilibré les deux mesures.

- Moyenne macro et moyenne pondérée :
  - Étant donné que les deux classes sont équilibrées (8 instances chacune), les deux moyennes sont également de 1,00.

Points clés à retenir

- Précision parfaite (100%) : le modèle a correctement classé tous les exemples de test.
- Aucune erreur de classification : aucun faux positif (FP) ou faux négatif (FN).
- Classes bien séparées : les caractéristiques (comme la grandeur absolue, la luminosité, le rayon) sont probablement très discriminantes.

Préoccupations potentielles :

- Overfitting ? Si l'ensemble de données est petit, le modèle a peut-être mémorisé des modèles au lieu de généraliser.
- Taille de l'ensemble de test ? Il n'y avait que 16 échantillons de test, les résultats pourraient donc ne pas être généralisables à un ensemble de données plus vaste.
- Essayer la validation croisée : pour confirmer la robustesse, il faudrait tester le modèle sur différentes divisions.

Vérification du surapprentissage (overfitting) :

Vérifions les performances avec une cross-validation: :

- La K-Fold cross-validation permet d'évaluer les performances sur différents sous-ensembles de données.
- Si le modèle fonctionne bien sur certains plis mais mal sur d'autres, cela peut indiquer un surajustement (overfitting).

Résultats :

```
Cross-Validation Scores: [1. 1. 1. 1. 1.]
Mean CV Accuracy: 1.0
```

Les scores de la cross-validation sont [1. 1. 1. 1. 1.], ce qui signifie que pour chaque pli, le modèle a atteint une précision de 100%. La précision moyenne du CV est également de 1,0 (100 %), ce qui suggère que le modèle est systématiquement parfait dans différents sous-ensembles des données d'entraînement.

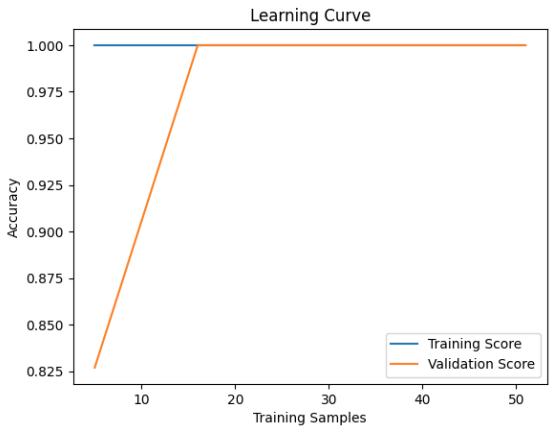

Figure 14: learning curve

8.0.1 Points clés à retenir :

Généralisation parfaite sur les données d'entraînement

- le modèle fonctionne de manière identique sur tous les plis de validation croisée, ce qui signifie qu'il n'y a aucune variation de précision entre les différentes divisions de données.
- Cela peut indiquer que les données sont trop faciles à classer ou que le modèle a appris des modèles très distincts pour les features sélectionnées.

Fortes Indications d'Overfitting

- Il est très rare d'obtenir une précision de 100 % dans tous les cas, à moins que l'ensemble de données soit très simple ou qu'il présente une forte séparabilité des features.
- Si l'ensemble de test (données invisibles) affiche également une précision de 100 %, cela peut signifier que le modèle mémorise plutôt que généralise.

8.1 Interprétation de la courbe d'apprentissage (voir figure 14) :

La courbe d'apprentissage montre que la précision de la formation et de la validation atteint très rapidement 100 % et y reste. Voici ce que cela suggère :

- Signes de surapprentissage
  - La précision de l'apprentissage est constamment à 100% → le modèle mémorise probablement les données au lieu de les généraliser.
  - Aucun écart entre les courbes d'apprentissage et de validation → En général, un petit écart est attendu en raison des problèmes de généralisation. Ici, les deux courbes convergent parfaitement,

ce qui est rare dans les problèmes du monde réel.

- Causes Possibles
  - Trop peu d'échantillons d'entraînement : le modèle peut voir les mêmes modèles de manière répétée, ce qui facilite la mémorisation.

8.2 Réflexions finales

Étant donné que la cross-validation et la courbe d'apprentissage affichent une précision de 100%, le modèle est probablement trop parfait pour un scénario réel. Il faudrait vérifier si le ensemble de données est trop simple, car mmême en condsidérant le dataset entier (avec toutes les classes), nous obtenons une accuraccy de 100% (voir sections précedentes) et une accurracy autour des 90% avec une régressionlogistique. En tout cas, l'accurraccy parfait obtenu avec une random forest est trés probablement dû a :

- Un dataset trop simple pour être classifié.
- La puissance de classification des ensemble learners comme la random forest.

9 Ensemble de données de classification stellaire - SDSS17

9.1 Aperçu de l'ensemble de données

L'ensemble de données utilisé dans cette étude est le Stellar Classification Dataset - SDSS17, provenant du Sloan Digital Sky Survey (SDSS). Il contient 100 000 observations astronomiques, chacune classée comme galaxie, étoile ou quasar en fonction de ses caractéristiques spectrales. L'objectif de cette tâche de classification est de développer un modèle de machine learning capable de distinguer ces trois catégories en fonction des features fournies.

Chaque observation comprend 17 features d'entrée et 1 variable cible (classe), qui indique le type de l'objet. Les features englobent les données photométriques (u, g, r, i, z), les coordonnées spatiales (alpha, delta) et les mesures de décalage vers le rouge (redshift), entre autres. De plus, l'ensemble de données contient plusieurs colonnes d'identification, telles que obj_ID et spec_obj_ID, qui ne sont pas pertinentes pour la classification et seront supprimées lors du prétraitement.

Colonnes :

- obj_ID = Identifiant d'objet, la valeur unique qui identifie l'objet dans le catalogue d'images utilisé par le CAS
- alpha = Angle d'ascension droite (a J2000 epoch)
- delta = Angle de déclinaison (a J2000 epoch)
- u = Filtre ultraviolet dans le système photométrique
- g = Filtre vert dans le système photométrique
- r = Filtre rouge dans le système photométrique
- i = Filtre proche infrarouge dans le système photométrique
- z = Filtre infrarouge dans le système photométrique
- run_ID = Numéro d'exécution utilisé pour identifier l'analyse spécifique
- rereun_ID = Numéro de réexécution pour spécifier comment l'image a été traitée
- cam_col = Colonne de caméra pour identifier la ligne de balayage dans la course
- field_ID = Numéro de champ pour identifier chaque champ
- spec_obj_ID = ID unique utilisé pour les objets spectroscopiques optiques (cela signifie que 2 observations différentes avec le même spec_obj_ID doivent partager la classe de sortie)
- class = classe d'objet (galaxie, étoile ou objet quasar)
- redshift = valeur de décalage vers le rouge basée sur l'augmentation de la longueur d'onde
- plate = ID de plaque, identifie chaque plaque dans SDSS
- MJD = Date julienne modifiée, utilisée pour indiquer quand une donnée SDSS a été prise
- fiber_ID = ID de fibre qui identifie la fibre qui a pointé la lumière vers le plan focal dans chaque observation

Informations clés

- Variable cible : class (galaxie, étoile ou quasar).
- u, g, r, i, z (données photométriques)
- alpha, delta (informations de position, peuvent être utiles)

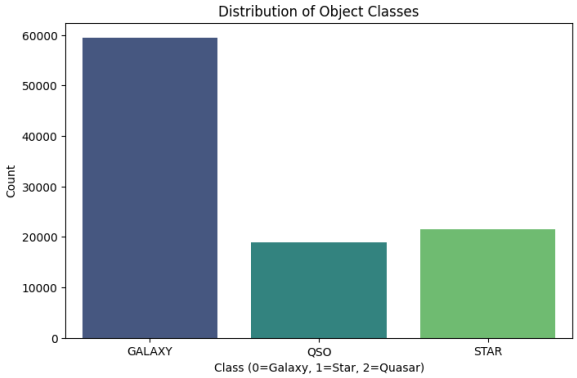

Figure 15: Distribution des classes

10 Analyse exploratoire des données (EDA)Analyse exploratoire des données (EDA)

10.1 Analyse des données

- Toutes les features sont numériques, sauf **class** qui catégorique.
- Pas de valeur manquantes.
- Pas de dupliquées.
- Le jeu de données se composent de 100000 lignes et 18 features

10.2 Visualisation de la distribution des classes

Diagramme à barres (voir figure 15) du nombre d'objets (étoiles, galaxies, quasars), afin de vérifier si les classes sont équilibrées.

Le graphique à barres illustre la distribution des classes d'objets dans l'ensemble de données, qui se compose de trois catégories : galaxies, étoiles et quasars (QSO).

Observations clés:

- Déséquilibre de classe :
  - Les galaxies sont la catégorie la plus courante, avec un nombre significativement plus élevé par rapport aux deux autres classes.
  - Les étoiles et les quasars (QSO) apparaissent dans des proportions relativement plus petites.
- Impact potentiel sur les performances du modèle :

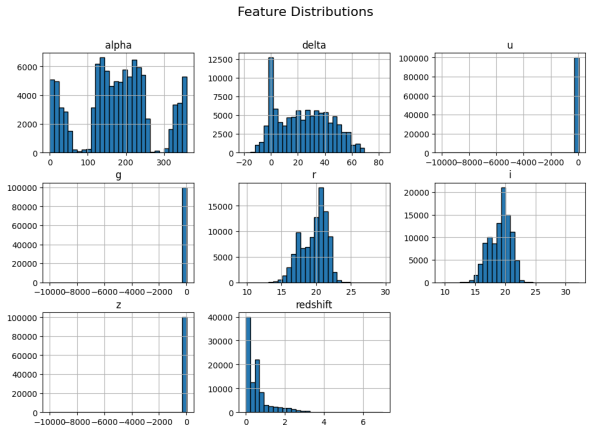

Figure 16: Distributions des features principales

  - Le déséquilibre peut affecter la formation du modèle, car les modèles de machine learning ont tendance à favoriser la classe majoritaire (Galaxies).
  - Si cette situation n'est pas résolue, cela pourrait conduire à une précision et à un rappel inférieurs pour les classes sous-représentées (Quasars et Étoiles).

- Stratégies d'atténuation :
  - Envisager d'utiliser la pondération des classes dans la formation du modèle pour équilibrer l'influence des classes.
  - Explorer l'augmentation des données ou le suréchantillonnage (par exemple, SMOTE) pour les quasars et les étoiles afin d'améliorer la généralisation du modèle.
  - Utilisez des mesures d'évaluation telles que le score F1 et le recall pour garantir des performances équitables dans toutes les classes.

Cette analyse est essentielle pour comprendre les biais des ensembles de données et concevoir un modèle qui fonctionne bien sur tous les objets astronomiques.

10.3 Distributions des features principales

Les histogrammes ci-dessus (voir figure 16) fournissent des informations sur la distribution des features clés dans l'ensemble de données, qui sont essentielles pour comprendre la nature des données.

Observations clés :

- Ascension droite (alpha) :
  - Les valeurs varient de 0 à 360 degrés, correspondant à la longitude céleste.

  - La distribution semble multimodale, suggérant différentes régions d'observation ou des modèles de regroupement dans le ciel.

- Déclinaison (delta) :
  - Les valeurs sont principalement concentrées entre -20 et 60 degrés, ce qui correspond à la région du ciel observable dans SDSS.
  - On observe un pic autour de 0 degré, indiquant une concentration plus dense d'objets dans certaines régions du ciel.

- Magnitudes photométriques (u, g, r, i, z) :
  - Ces features représentent la luminosité mesurée dans différents filtres de longueur d'onde.
  - Les distributions u, g et z semblent biaisées avec des valeurs extrêmement négatives, ce qui pourrait indiquer un problème de mise à l'échelle des données ou des valeurs aberrantes.
  - Les bandes r et i présentent une distribution normale, ce qui est attendu pour les mesures basées sur la magnitude en astronomie.

- Décalage vers le rouge (redshift) :
  - La distribution est fortement asymétrique vers la droite, ce qui signifie que la plupart des objets ont de faibles valeurs de décalage vers le rouge, avec une majorité concentrée près de 0.
  - Quelques objets présentent des décalages vers le rouge très élevés, correspondant probablement à des quasars ou à des galaxies lointaines s'éloignant à grande vitesse.

Problèmes potentiels et prochaines étapes :

- Les valeurs aberrantes dans les features photométriques (u, g, z) doivent être étudiées et éventuellement corrigées.
- La mise à l'échelle (scaling) des features (par exemple, la standardisation ou la normalisation) doit être envisagée pour améliorer les performances du modèle.
- Les distributions multimodales en alpha et delta pourraient suggèrer la nécessité d'une analyse de clustering plus poussée.

Cette analyse exploratoire permet de garantir que l'ensemble de données est bien préparé pour la classification tout en mettant en évidence les étapes de prétraitement potentielles pour améliorer la précision du modèle.

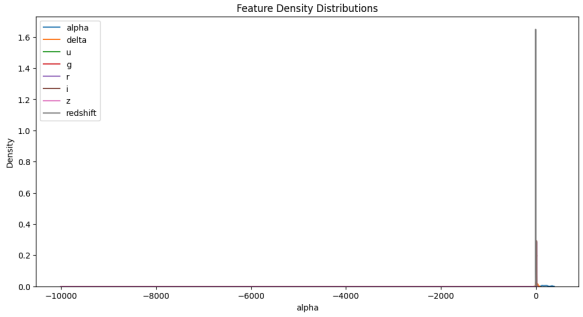

Figure 17: Distributions de densité

10.4 Distributions de densité des features principales

Le graphique de densité (voir figure 17) fournit un aperçu de la distribution des principales features de l'ensemble de données. Cependant, d'après la visualisation, il semble y avoir un problème avec l'échelle de certaines features, ce qui rend difficile la distinction de leurs modèles de densité.

Observations clés :

- Valeurs extrêmement négatives
  - Le graphique montre des valeurs extrêmement négatives pour certaines features (par exemple, g, u, z), ce qui est très inhabituel.
  - Cela suggère des anomalies potentielles dans les données, une mise à l'échelle incorrecte ou des valeurs aberrantes qui doivent être traitées.
- Très concentré, proche de zéro
  - La majeure partie de la densité est regroupée près de zéro, ce qui rend difficile l'observation de distributions significatives.
  - Cela indique que certaines features ont des échelles sensiblement différentes par rapport à d'autres, ce qui peut potentiellement affecter la formation du modèle.
- Problème de mise à l'échelle des features
  - Étant donné que les données astronomiques couvrent souvent plusieurs magnitudes, certaines features (par exemple, le décalage vers le rouge, les magnitudes) nécessitent probablement une transformation logarithmique ou une normalisation pour être correctement visualisées.
  - La visualisation actuelle suggère qu'une mise à l'échelle est nécessaire avant que des modèles significatifs puissent être observés.

Prochaines étapes :

- Recherchez les valeurs aberrantes potentielles dans l'ensemble de données, en particulier les valeurs négatives dans les entités où elles ne devraient pas se produire.
- Appliquer la normalisation ou la standardisation pour amener toutes les features à une même échelle.

10.5 Heatmap de corrélation des principales features

La heatmap de corrélation (voir figure 18) représente visuellement les relations entre les différentes entités de l'ensemble de données, avec des valeurs allant de -1 à 1 :

- 1 (rouge) → Corrélation positive parfaite : lorsqu'une feature augmente, l'autre augmente également.
- -1 (bleu) → Corrélation négative parfaite : lorsqu'une feature augmente, l'autre diminue.
- 0 (bleu foncé) → Aucune corrélation : les features sont indépendantes.

Observations clés :

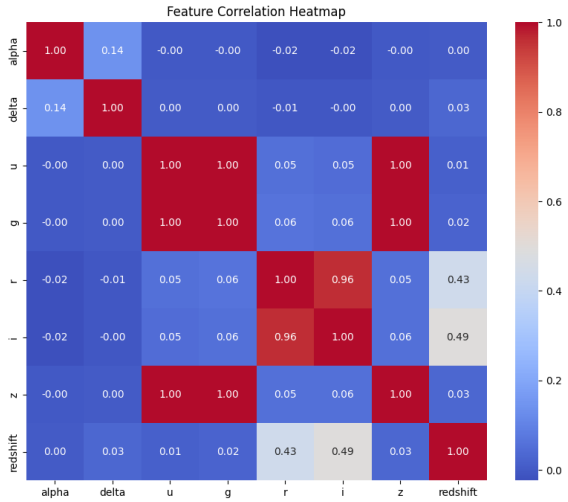

Figure 18: features correlation heatmap

- Fortes corrélations positives :
  - r et i (0,96) : Ces deux features sont fortement corrélées, ce qui indique qu'elles fournissent presque les mêmes informations. L'une d'elles pourrait être redondante.
  - u, g et z (corrélation 1) : ces trois caractéristiques semblent être presque identiques, ce qui suggère une duplication potentielle ou une forte dépendance.
  - r et redshift (0,43) et i et redshift (0,49) : Il existe une corrélation modérée entre ces features et le redshift, ce qui signifie qu'elles pourraient être utiles pour le prédire.
- Corrélation faible ou inexistante :
  - alpha et delta (0,14) : Faible corrélation, ce qui signifie que les coordonnées célestes n'influencent pas fortement les autres features mesurées.
  - La plupart des autres paires de features ont une corrélation proche de zéro, ce qui indique qu'elles sont indépendantes et contribuent de manière unique à l'ensemble de données.
- Implications pour le machine learning :
  - Sélection des features : Étant donné que u, g et z sont presque identiques, il peut être inutile de conserver les trois. Une technique de réduction de dimensionnalité (par exemple, PCA) pourrait s'avérer utile.
  - Multicolinéarité : une forte corrélation entre r et i suggère que l'un d'eux pourrait être supprimé pour éviter la redondance dans certains modèles (par exemple, la régression linéaire).

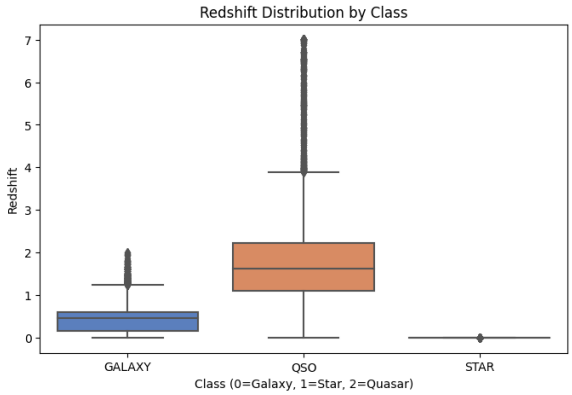

Figure 19: Distribution de redshift en fonction de la target

- Pouvoir prédictif : r et i ont une corrélation modérée avec redshift, ce qui signifie qu'ils pourraient être utiles pour le prédire.

Etapes envisageables :

- Étudiez la redondance des features et envisagez de supprimer ou de transformer les features hautement corrélées.
- Analysez l'importance de chaque feature pour déterminer celles qui contribuent le plus à la classification.
- Si nécessaire, appliquez des techniques de réduction de dimensionnalité pour éviter les problèmes de multicolinéarité.

Cette heatmap permet d'affiner le processus d'ingénierie des features, en garantissant que le modèle est formé avec les variables les plus pertinentes et les plus indépendantes.

10.6 Interprétation de la distribution du redshift par classe (analyse des features par rapport a la cible)

Ce boxplot (voir figure 19) montre comment le redshift varie selon les différentes classes d'objets astronomiques : galaxie, quasar (QSO) et étoile. Les principaux points à retenir sont les suivants :

Distribution du redshift par classe :

- Galaxies (case de gauche, bleue) :
  - Ils ont un redshift relativement faible, généralement compris entre 0 et 1.

  - Certaines valeurs aberrantes existent au-delà de 1, mais elles ne sont pas extrêmes.

- Quasars (QSO) (case du milieu, orange) :
  - Affiche une gamme beaucoup plus large de valeurs de redshift, de près de 0 jusqu'à plus de 4.
  - Ils ont un redshift médian le plus élevé par rapport aux galaxies et aux étoiles.
  - Il existe un grand nombre de valeurs aberrantes au-dessus de 4, indiquant un sous-ensemble de quasars très éloignés.
- Étoiles (case de droite, bleu foncé) :
  - Ils ont un redshift presque nul, ce qui signifie qu'ils sont beaucoup plus proches de nous que les galaxies et les quasars.
  - Il n'y a pas de dispersion significative dans les valeurs de redshift, ce qui confirme que les étoiles ne présentent pas de redshift élevé.

Relation entre le redshift et la classe :

- Le redshift est une feature distinctive forte :
  - Les étoiles ont un redshift presque nul.
  - Les galaxies ont un faible redshift.
  - Les quasars ont le redshift le plus élevé, ce qui signifie qu'ils sont les objets les plus éloignés observés.

Implications pour les modèles de classification :

- Le redshift est une feature clé pour distinguer les quasars des galaxies et des étoiles.
- Les étoiles peuvent être facilement identifiées en raison de leur redshift proche de zéro.
- Les galaxies et les quasars présentent un certain chevauchement, mais les quasars ont généralement un redshift plus élevé.
- Un modèle de classification peut exploiter le redshift pour améliorer la précision, en particulier pour séparer les quasars des galaxies.

Conclusion :

Ce graphique confirme que le redshift joue un rôle crucial dans la distinction des objets astronomiques. Il est particulièrement utile pour identifier les quasars, qui sont beaucoup plus éloignés que les galaxies et les étoiles.

11 Feature Engineering

Pour préparer l'ensemble de données à la modélisation, les étapes de prétraitement suivantes ont été appliquées :

- Suppression des colonnes non informatives: plusieurs fonctionnalités liées à l'ID (obj_ID, spec_obj_ID, run_ID, etc.) ont été exclues car elles ne contribuent pas à la tâche de classification.
- Encodage de la variable cible : la colonne de classe, contenant des étiquettes catégorielles (Galaxy, Star, Quasar), a été codée en valeurs numériques pour les modèles de machine learning.
- Création de features d'index de couleur pour modéliser les différences de magnitude :
  - df["u-g"] = df["u"] - df["g"]
  - df["g-r"] = df["g"] - df["r"]
  - df["r-i"] = df["r"] - df["i"]
  - df["i-z"] = df["i"] - df["z"]
- Mise à l'échelle des fonctionnalités : les features numériques ont été standardisées à l'aide de StandardScaler pour garantir l'uniformité et améliorer les performances du modèle.
- Transformation logarithmique pour le redshift (réduction de l'asymétrie) et suppression de la colonne redshift d'origine (puisque nous utilisons maintenant log_redshift)
- Répartition train-test : l'ensemble de données a été divisé en 80% de training set et 20% de test set, garantissant une représentation équilibrée des classes.

12 Entraînement de modèles

12.1 Random Forest

Dans un premier temps, une RandomForest a été utilisée sur l'ensemble de données prétraitées. Ce modèle a été choisi en raison de sa capacité à gérer des modèles complexes et de sa robustesse face à l'overfitting. Le modèle a été évalué à l'aide de matrices de précision, de confusion et de rapports de classification pour évaluer ses performances initiales.

Résultats de la randomForest :

```
Random Forest Accuracy: 0.9788
Classification Report:
               precision    recall  f1-score   support

           0       0.98      0.99      0.98     11889
           1       0.97      0.93      0.95      3792
           2       0.99      1.00      1.00      4319

    accuracy                           0.98     20000
   macro avg       0.98      0.97      0.98     20000
weighted avg       0.98      0.98      0.98     20000
```

Le modèle Random Forest a atteint une précision impressionnante de 97,88 %, ce qui signifie qu'il a correctement classé près de 98 % des objets de l'ensemble de tests. Analysons plus en détail les résultats :

- Précision (valeur prédictive positive) : proportion de prédictions correctes parmi toutes les instances prédites pour chaque classe.
  - 0,98 pour les galaxies → Lorsque le modèle prédit « Galaxie », il est correct à 98 %.
  - 0,97 pour les étoiles → 97 % des « étoiles » prédites sont en réalité des étoiles.
  - 0,99 pour les quasars → Extrêmement précis dans l'identification des quasars.
- Recall (sensibilité) : la proportion d'instances réelles qui ont été correctement prédites.
  - 0,99 pour les galaxies → Le modèle a identifié correctement 99 % des galaxies réelles.
  - 0,93 pour les étoiles → Certaines étoiles ont été mal classées (confusion potentielle avec les galaxies).
  - 1,00 pour les quasars → Le modèle a correctement identifié tous les quasars (aucun faux négatif).
- Score F1 : la moyenne harmonique de la précision et du recall (équilibre les deux mesures).
  - Idéal pour les quasars (1,00), suivi des galaxies (0,98) et des étoiles (0,95).
  - Les étoiles (1) ont le rappel le plus faible (0,93), ce qui suggère une certaine erreur de classification.
- Support : le nombre d'instances réelles par classe dans l'ensemble de données.
  - La plupart des échantillons appartiennent à la classe des Galaxies (11 889), suivie des Quasars (4 319) et des Étoiles (3 792).

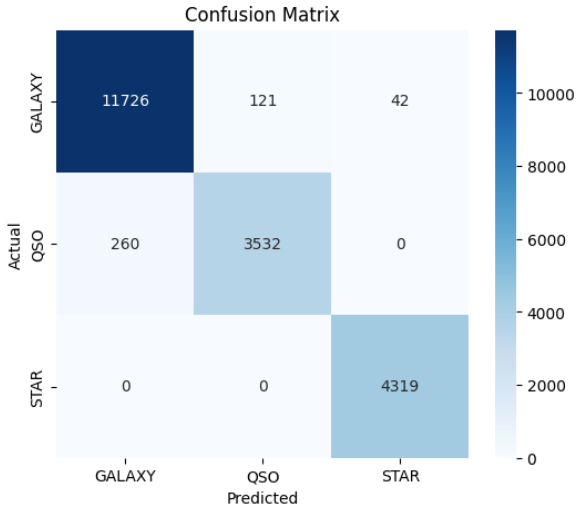

Figure 20: Matrice de confusion

Observations clés :

- Précision globale élevée (97,88 %) → Le modèle fonctionne exceptionnellement bien.
- Excellentes performances sur les quasars (classe 2) → Un rappel de 100 % signifie qu'aucun quasar n'a été mal classé.
- Légère faiblesse dans les étoiles (classe 1) → Un rappel de 0,93 suggère que certaines étoiles sont confondues avec des galaxies ou des quasars.

Matrice de confusion (figrure 20) :

- Les galaxies sont très bien classées → Seulement 1,37 % sont mal classées (163/11889).
- Les quasars présentent une légère erreur de classification (260 mal classés comme galaxies, 121 comme QSO) → 93 % de recall (3532/3792).
- Les étoiles sont parfaitement classées → 100 % de précision pour les étoiles (4319/4319) !

Analyse de l'imortance des features (figure 21) :

- Le redshift logarithmique est la caractéristique la plus importante (score d'importance 0,5) : Cela signifie que le log_redshift joue un rôle dominant dans la distinction entre galaxies, quasars et étoiles. Le redshift est un facteur crucial en astronomie, car il indique la vitesse et la distance d'un objet par rapport à la Terre, ce qui le rend très pertinent pour la classification.
- Les indices de couleur sont importants :Les indices r-i, g-r, i-z ont une importance modérée. Ces indices de couleur (différences de magnitudes dans différents filtres) aident à distinguer les types d'objets en

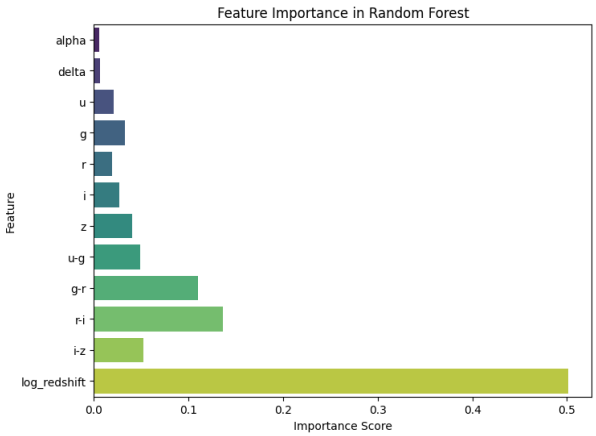

Figure 21: Features importance

fonction de leurs caractéristiques spectrales.

- Les magnitudes individuelles ont une importance plus faible : Les magnitudes u, g, r, i, z sont moins importantes que les indices de couleur. Cela suggère que les différences relatives de magnitude (indices de couleur) fournissent une information plus significative que les magnitudes absolues.
- Les features de position (alpha, delta) sont les moins importantes : L'ascension droite (alpha) et la déclinaison (delta) ne contribuent presque pas à la classification. Cela est logique, car la classification des objets repose principalement sur leurs propriétés spectrales plutôt que sur leur position spatiale.

Features Sélection avec forêt aléatoire

Pour cela, nous utilisons un seuil d'importance des fonctionnalités (à l'aide de l'importance de la random forest précedentes) Nous supprimons les feature ayant une importance très faible : Inférieure à 0.02.

Résultats :

```
Random Forest Accuracy with Feature Selection: 0.98015
Classification Report:
               precision    recall  f1-score   support

           0       0.98      0.99      0.98     11889
           1       0.97      0.93      0.95      3792
           2       1.00      1.00      1.00      4319

    accuracy                           0.98     20000
   macro avg       0.98      0.97      0.98     20000
weighted avg       0.98      0.98      0.98     20000
```

Nous avons une légère amélioration au niveau de la précision qui passe à 0.98015. Mais le modéle de base était déjà assez performant.

12.2 XGBoost

Résultats :

```
XGBoost Accuracy: 0.9781
Classification Report:
               precision    recall  f1-score   support

           0       0.98      0.99      0.98     11889
           1       0.97      0.94      0.95      3792
           2       0.99      1.00      0.99      4319

    accuracy                           0.98     20000
   macro avg       0.98      0.97      0.97     20000
weighted avg       0.98      0.98      0.98     20000
```

Model	Accuracy	Galaxy F1	QSO F1	Star F1
Random Forest	0.9788	0.98	0.95	1.00
XGBoost	0.9781	0.98	0.95	0.99

Table 3: Comparison de performance de classification entre Random Forest et XGBoost

Observations clés :

- Les deux modèles fonctionnent de manière similaire (Random Forest est légèrement meilleur).
- XGBoost peut être plus efficace pour les grands ensembles de données, tandis que Random Forest est plus interprétable.

Matrice de confusion (figure 22):

- Très grande précision dans la détection des galaxies, avec une classification erronée minimale.
- La plupart des erreurs se produisent lorsque les quasars sont classés comme des galaxies, mais le recall global reste élevé.
- les étoiles sont classées avec une précision presque parfaite, avec seulement 24 classées à tort comme galaxies.

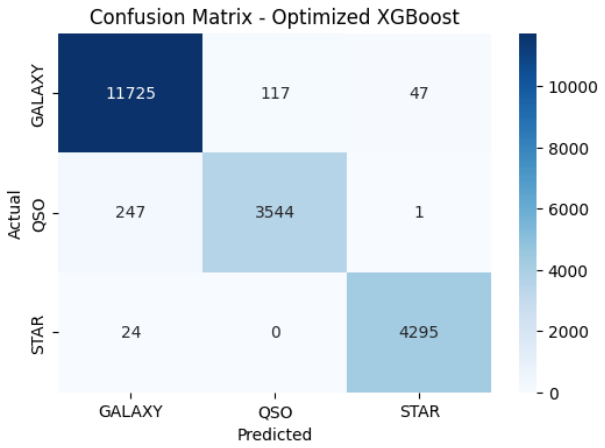

Figure 22: conf-mat-xgb

12.3 Logistic Regression

Entraînons un modèle de régression logistique et évaluons ses performances. Étant donné que la régression logistique est un modèle linéaire plus simple, elle risque de ne pas être aussi performante que Random Forest ou XGBoost.

Résultats :

```
Logistic Regression Accuracy: 0.9608
Classification Report:
               precision    recall  f1-score   support

           0       0.96      0.97      0.97     11889
           1       0.95      0.89      0.92      3792
           2       0.96      1.00      0.98      4319

    accuracy                           0.96     20000
   macro avg       0.96      0.95      0.95     20000
weighted avg       0.96      0.96      0.96     20000
```

- Galaxie (Classe 0) :Haute précision (0,96) et rappel (0,97), ce qui signifie que la plupart des galaxies sont correctement classées.
- Quasar (QSO, Classe 1) : Le recall est plus faible (0,89) par rapport aux autres classes. Cela signifie que certains quasars sont mal classés (peut-être comme des galaxies). La précision reste élevée (0,95), donc lorsque le modèle prédit un QSO, il est généralement correct.
- Étoile (Classe 2) : Le recall le plus élevé (1,00), ce qui signifie que presque toutes les étoiles ont été

correctement identifiées. Cela suggère que les étoiles sont bien séparées dans l'espace des features.

12.4 Réseau neuronal (Perceptron multicouche - MLP)

Essayons un réseau neuronal (perceptron multicouche - MLP) pour la classification. J'utiliserai un réseau simple à propagation directe avec des couches entièrement connectées. Je l'entraînerai et évaluerai sa précision.

Résulats :

```
Epoch 1/20
2500/2500                                                    5s 2ms/step - accuracy:
    0.8969 - loss: 0.3089 - val_accuracy: 0.9610 - val_loss: 0.1275
Epoch 2/20
2500/2500                                                    4s 1ms/step - accuracy:
    0.9628 - loss: 0.1393 - val_accuracy: 0.9650 - val_loss: 0.1152
Epoch 3/20
2500/2500                                                    4s 1ms/step - accuracy:
    0.9641 - loss: 0.1180 - val_accuracy: 0.9629 - val_loss: 0.1242
Epoch 4/20
2500/2500                                                    4s 2ms/step - accuracy:
    0.9662 - loss: 0.1121 - val_accuracy: 0.9674 - val_loss: 0.1073
Epoch 5/20
2500/2500                                                    4s 2ms/step - accuracy:
    0.9666 - loss: 0.1054 - val_accuracy: 0.9690 - val_loss: 0.1028
Epoch 6/20
2500/2500                                                    4s 2ms/step - accuracy:
    0.9674 - loss: 0.1050 - val_accuracy: 0.9636 - val_loss: 0.1256
Epoch 7/20
2500/2500                                                    4s 2ms/step - accuracy:
    0.9700 - loss: 0.0999 - val_accuracy: 0.9673 - val_loss: 0.1085
Epoch 8/20
2500/2500                                                    4s 2ms/step - accuracy:
    0.9696 - loss: 0.0994 - val_accuracy: 0.9706 - val_loss: 0.1003
Epoch 9/20
2500/2500                                                    4s 2ms/step - accuracy:
    0.9699 - loss: 0.0964 - val_accuracy: 0.9718 - val_loss: 0.0976
Epoch 10/20
2500/2500                                                    4s 2ms/step - accuracy:
    0.9701 - loss: 0.0977 - val_accuracy: 0.9699 - val_loss: 0.1022
Epoch 11/20
2500/2500                                                    4s 2ms/step - accuracy:
    0.9704 - loss: 0.0961 - val_accuracy: 0.9714 - val_loss: 0.0981
Epoch 12/20
```

```
2500/2500                                                    4s 2ms/step - accuracy:
    0.9716 - loss: 0.0932 - val_accuracy: 0.9712 - val_loss: 0.0954
Epoch 13/20
2500/2500                                                    4s 2ms/step - accuracy:
    0.9719 - loss: 0.0919 - val_accuracy: 0.9720 - val_loss: 0.0941
Epoch 14/20
2500/2500                                                    4s 1ms/step - accuracy:
    0.9726 - loss: 0.0905 - val_accuracy: 0.9726 - val_loss: 0.0918
Epoch 15/20
2500/2500                                                    4s 1ms/step - accuracy:
    0.9718 - loss: 0.0932 - val_accuracy: 0.9735 - val_loss: 0.0917
Epoch 16/20
2500/2500                                                    4s 2ms/step - accuracy:
    0.9723 - loss: 0.0898 - val_accuracy: 0.9718 - val_loss: 0.0976
Epoch 17/20
2500/2500                                                    4s 2ms/step - accuracy:
    0.9726 - loss: 0.0890 - val_accuracy: 0.9728 - val_loss: 0.0960
Epoch 18/20
2500/2500                                                    4s 2ms/step - accuracy:
    0.9731 - loss: 0.0872 - val_accuracy: 0.9717 - val_loss: 0.0980
Epoch 19/20
2500/2500                                                    4s 2ms/step - accuracy:
    0.9736 - loss: 0.0887 - val_accuracy: 0.9740 - val_loss: 0.0905
Epoch 20/20
2500/2500                                                    4s 2ms/step - accuracy:
    0.9731 - loss: 0.0880 - val_accuracy: 0.9740 - val_loss: 0.0890

Final Accuracy : 0.9739500284194946
```

- Amélioration de la précision.
  - La précision de validation finale a atteint 97,40 %, ce qui est légèrement meilleur que la régression logistique (96,08 %) et comparable à Random Forest (97,88 %) et XGBoost (97,81 %).
  - La précision de d'entrainement est également très proche de la précision de la validation, ce qui suggère que le modèle se généralise bien sans overfitting sévère.
- Réduction de l'erreur (Loss).
  - L'erreur diminue constamment au fil des epoch, ce qui indique que le modèle apprend bien.
  - L'erreur de validation finale est de 0,0890, ce qui suggère un modèle bien optimisé avec des erreurs minimales.

- Convergence et stabilité.
  - La précision s'améliore rapidement au cours des premières epoch, atteignant plus de 96 % à l'époque 2, puis s'améliore progressivement jusqu'à 97,40 %.
  - Il n'y a pas de fluctuations drastiques, ce qui signifie que le taux d'apprentissage et le processus d'optimisation sont stables.
- Comparaison avec les autres modèles
  - Le MLP est plus performant que la régression logistique en raison de sa capacité à capturer des relations complexes.
  - Il atteint des performances comparables à Random Forest et XGBoost, démontrant que le deep learning est une approche efficace pour cette tâche de classification.
  - Le coût de calcul supplémentaire lié à la formation du réseau neuronal ne justifie peut-être pas le gain de précision mineur par rapport aux méthodes basées sur les arbres.

Conclusion

Le modèle de réseau neuronal fonctionne bien, atteignant une grande précision avec une bonne généralisation. Bien qu'il offre de légères améliorations, les modèles basés sur des arbres comme Random Forest et XGBoost offrent des performances similaires avec des temps de formation potentiellement plus courts et des avantages en termes d'interprétabilité.

13 Comparaison des modèles

Voici une comparaison des quatre modèles en fonction de leur précision et de leurs scores F1 par classe:

Model	**Accuracy**	**Galaxy F1**	**QSO F1**	**Star F1**
Logistic Regression	0.9608	0.97	0.92	0.98
Random Forest	0.9788	0.98	0.95	1.00
XGBoost	0.9781	0.98	0.95	0.99
Neural Network (MLP)	0.9740	0.98	0.94	0.99

Table 4: Comparaison des performances des modèles

Analyse et principaux points à retenir :

- Random Forest atteint la précision la plus élevée (97,88 %), surpassant légèrement XGBoost (97,81 %) et le réseau neuronal (97,40 %).
- La régression logistique obtient les moins bons résultats (96,08 %), mais fournit néanmoins des résultats raisonnables, démontrant son efficacité en tant que modèle simple.

- Le réseau neuronaux (MLP) offre des performances comparables à XGBoost, mais à un coût de calcul plus élevé.
- Pour l'interprétabilité et l'efficacité, Random Forest et XGBoost sont préférables.
- Si du deep learning est nécessaire, le réseau neuronal fonctionne bien mais n'offre pas d'avantage significatif par rapport aux méthodes basées sur les arbres.

14 Recommandation finale

- Si l'interprétabilité et l'efficacité sont des priorités → Random Forest ou XGBoost.
- Si une approche de deep learning est souhaitée → le réseau de neurones MLP est un bon choix.
- Si la simplicité est nécessaire → La régression logistique est un bon modèle de base.

References

[1] A. Antoniadis-Karnavas, S. Sousa, E. Delgado-Mena, N. Santos, G. Teixeira, and V. Neves. Odusseas: a machine learning tool to derive effective temperature and metallicity for m dwarf stars. *Astronomy & Astrophysics*, 636:A9, 2020.

[2] A. Behmard, E. A. Petigura, and A. W. Howard. Data-driven spectroscopy of cool stars at high spectral resolution. *The Astrophysical Journal*, 876(1):68, 2019.

[3] G. M. De Silva, K. C. Freeman, J. Bland-Hawthorn, S. Martell, E. W. De Boer, M. Asplund, S. Keller, S. Sharma, D. B. Zucker, T. Zwitter, et al. The galah survey: scientific motivation. *Monthly Notices of the Royal Astronomical Society*, 449(3):2604–2617, 2015.

[4] S. Fabbro, K. Venn, T. O'Briain, S. Bialek, C. Kielty, F. Jahandar, and S. Monty. An application of deep learning in the analysis of stellar spectra. *Monthly Notices of the Royal Astronomical Society*, 475(3):2978–2993, 2018.

[5] H. Karttunen, P. Kröger, H. Oja, M. Poutanen, and K. J. Donner. *Stellar Spectra*, pages 227–239. Springer Berlin Heidelberg, Berlin, Heidelberg, 2017.

[6] G. Longo, E. Merényi, and P. Tiňo. Foreword to the focus issue on machine intelligence in astronomy and astrophysics. *Publications of the Astronomical Society of the Pacific*, 131(1004):1–6, 2019.

[7] R. Olney, M. Kounkel, C. Schillinger, M. T. Scoggins, Y. Yin, E. Howard, K. Covey, B. Hutchinson, and K. G. Stassun. Apogee net: Improving the derived spectral parameters for young stars through deep learning. *The Astronomical Journal*, 159(4):182, 2020.

[8] J.-V. Rodríguez, I. Rodríguez-Rodríguez, and W. L. Woo. On the application of machine learning in astronomy and astrophysics: A text-mining-based scientometric analysis. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 12(5):e1476, 2022.

[9] Y.-S. Ting, C. Conroy, and H.-W. Rix. Accelerated fitting of stellar spectra. *The Astrophysical Journal*, 826(1):83, 2016.

[10] Y.-S. Ting, C. Conroy, H.-W. Rix, and P. Cargile. The payne: Self-consistent ab initio fitting of stellar spectra. *The Astrophysical Journal*, 879(2):69, jul 2019.