Projet IAS : Classification étoiles, galaxies et quasars

2022-2023

Membres du groupe :

- **FERGUI**
- **ACHOUR**
- **NEDDAF**

Lien du dataset :

https://www.kaggle.com/datasets/fedesoriano/stellar-classification-dataset-sdss17/code?datasetId=1866141&sortBy=voteCount

Sommaire

Introduction :

La classification précise des objets célestes revêt une importance cruciale dans le domaine de l'astronomie et de la recherche spatiale. L'étude des étoiles, astres et quasars permet de mieux comprendre la composition de l'univers, l'évolution des galaxies et les processus astrophysiques complexes qui y sont associés. Cependant, compte tenu du grand nombre d'objets célestes et de leurs caractéristiques distinctes, la classification manuelle de ces entités devient fastidieuse et sujette à des erreurs potentielles.

Dans ce contexte, l'application des techniques de machine learning offre une approche prometteuse pour automatiser la classification des étoiles. À travers ce rapport, nous présenterons les différentes étapes de notre approche, les résultats obtenus et les implications de notre étude. Nous discuterons également des opportunités de recherche futures et des pistes d'amélioration pour optimiser davantage les performances des modèles de classification utilisés.

Exploration préliminaire des données :

Nous avons commencé par analyser les données disponibles collectées à partir d'observations astronomiques. Pour visualiser et comprendre ces données, nous avons utilisé différentes techniques graphiques et statistiques descriptives. Nous avons créé un diagramme circulaire pour visualiser la répartition de nos trois classes (étoile, galaxie, quasar), nommées respectivement en STAR, GALAXY et QSO, dans l'ensemble de données pour comprendre l'équilibre de nos classes.

Nous avons également utilisé des techniques d'analyse de corrélation pour évaluer les relations linéaires entre les variables. Cela nous a permis de détecter des variables fortement corrélées ce qui peut avoir un impact sur les performances des modèles de classification.

Choix des métriques de performance :

La tâche principale de notre projet est la classification précise des étoiles, astres, quasars à partir des caractéristiques observables. Pour évaluer les performances de nos modèles de

classification, il est essentiel de choisir certaines métriques de performance appropriées pour quantifier l'exactitude de nos précision. Nous avons opté pour 3 métriques :

- L'accuracy:

Accuracy = (Vrais positifs + Vrais négatifs) / (Vrais positifs + Vrais négatifs + Faux positifs + Faux négatifs)

l'accuracy évalue la performance globale du modèle en tenant compte de toutes les classes.

- La balanced accuracy

Balanced accuracy = ½ * (Vrais positifs / (Vrais positifs + Faux négatifs)) + ½ * (Vrais négatifs / (Vrais négatifs + Faux positifs))

La balanced accuracy peut s'avérer très utile pour des classes inégalement répartis dans le dataset (ce qui est notre cas, 60% pour la classe galaxy, 21% pour la classe STAR et 19% pour QSO)

- La précision

Précision = Vrais positifs / (Vrais positifs + Faux positifs)

Il s'agit aussi d'un indicateur de performance pour un modèle de classification.

Pour résumer la performance de nos modèles, on visualise une matrice de confusion. Il s'agit d'une matrice résumant les résultats de prédiction pour une classification. Sur la diagonale s'affichent les résultats biens prédits (vrais positifs et vrais négatifs) et sur les autres cases, on a les erreurs (faux positifs et faux négatifs)

Travail sur le fond :

Preprocessing:

Nettoyage, encodage, Normalisation et PCA :

Dans ce volet, nous avons analysé les données pour un éventuel nettoyage. Nous avons constaté qu'une de nos variables est de type Objet ce qui bloque le traitement de données. Nous avons numérisé cette variable en un type int. Nous avons encoder la variable catégorielle classe de notre ensemble de données(0 pour les galaxies, 1 pour les quasars et 2 pour les étoiles). Nous avons constaté au préalable que certaines variables n'apportent rien comme information importante et cela concerne les variables ID.

Nous avons enfin normalisé les variables numériques pour les ramener à une échelle commune.

Afin d'avoir plus d'informations sur les variables, on a utilisé une matrice de corrélation qui nous montre la présence de nombreuses relations fortes entre certaines variables.

Pour cela, on a décidé d'utiliser une PCA sur le jeu de donnée d'un côté pour l'utilisation du modèle avec les données transformées, puis en parallèle d'un autre côté utiliser le modèle avec

les données sans PCA, cela afin de comparer les 2 approches et de voir si la PCA s'avère efficace ou le contraire. Les conclusions sont citées à la fin de ce rapport.

Une fois qu'on a nettoyé et préparé nos données, on les a séparées en ensembles d'entraînement et de test. l'ensemble d'entraînement est utilisé pour entraîner notre modèle, tandis que l'ensemble de test est utilisé pour évaluer les performances de notre modèle sur des données non connues par le modèle.

Architecture et choix de modéle

Dans le cadre de notre projet, nous avons travaillé sur deux modèles : arbre de décision, Random forest : l'architecture du modèle Random forest repose sur l'ensemble d'arbre de décision. Il s'agit d'un modèle d'apprentissage supervisé qui combine les prédictions de plusieurs arbres de décision pour prendre une décision finale. chaque arbre de décision est construit de manière aléatoire en utilisant un sous ensemble des données d'entraînement et des variables d'entrée. Ensuite, les prédictions de chaque arbre sont agrégées pour obtenir la prédiction finale. Le choix de Random Forest pour notre projet de classification était justifié par sa capacité à gérer des ensembles de données complexes, notamment avec des variables fortement corrélées.
Arbre de décision : l'arbre de décision est un modèle d'apprentissage supervisé qui utilise une structure d'arbre pour prendre des décisions en fonction des valeurs des différentes variables d'entrée. chaque nœud de l'arbre représente une valeur possible de cette variable. l'arbre est construit en utilisant des critères de division pour maximiser la pureté des classes dans chaque nœud. l'arbre de décision est simple et facile à interpréter, ils sont également efficaces pour capturer des relations non linéaires entre les variables d'entrées et les classes cibles.

Choix d'hyper paramètres

Nous avons choisi d'utiliser une recherche d'hyper paramètres par une grille combinée à une cross validation à 5 plis. Cette technique robuste divise notre ensemble de données en plusieurs sous-ensembles d'entraînement et de validation et recherche les meilleurs hyper paramètres. Chaque sous-ensemble de validation est utilisé tour à tour pour évaluer les performances du modèle. Pour l'arbre de décision, nous avons utilisé la grille de recherche pour calculer les hyperparamètres les plus optimaux (profondeur maximale, critère de division : gini ou entropie,

nombre min d'échantillons requis pour une division, nombre min d’échantillons requis dans une feuille). Cette méthode consiste à évaluer les performances de modèle pour chaque combinaison de paramètres. Ajuster ces hyperparamètres permet de trouver le meilleur compromis entre la complexité de l’arbre et ses performances prédictives.

PCA ou sans PCA?

Les observations que nous avons tiré sur la matrice de corrélation affichent que certaines variables sont corrélés, nous avons donc décidé d’appliquer une analyse par composante principale pour éliminer les corrélations et le bruit qui peut influer sur nos modèles et pour avoir moins de redondances des données.

Nous observons que 4 composantes principales se dégagent et expliquent la variance de notre jeu de données. Pour voir si la PCA influe sur la performance des modèles, nous allons donc appliquer un modèle avec PCA et l’un sans pour voir si elle optimise notre modèle et est donc nécessaire à l’apprentissage.
Lorsque nous avons appliqué la PCA, notre objectif initial était de réduire la dimension du jeu de données tout en conservant un pourcentage élevé de la variance.
Cependant, les résultats obtenus ont révélé une dégradation de performances par rapport à l’utilisation des données sans PCA: - 0.5 à -0.7 % de balanced accuracy en moins.
La PCA peut introduire un certain degré de bruit dans les données réduites. Les composantes principales sont une combinaison linéaire des variables d'origine, ce qui peut entraîner une perte de précision et une augmentation du bruit dans les données transformées. Cette augmentation du bruit peut affecter négativement les performances du modèle de classification.

Résultats et Conclusion :

Métriques	Arbre de décision avec PCA	Arbre de décision sans PCA	RandomForest avec PCA	RandomForest sans PCA
Accuracy	0.934	0.976	0.957	0.978
Balanced Accuracy	0.917	0.968	0.948	0.971
Précision	0.934	0.975	0.957	0.977

Au vu des indicateurs, le “meilleur” modèle pour ce jeu de données est le RandomForest sans PCA.

Pour finir, ce projet a été un apprentissage pratique riche qui nous a permis de chercher et de raisonner sur la manière d’aborder un problème de classification de A à Z.