LEPL1109 - Statistics and Data Sciences
HACKATHON 2 - Classification: Stars, Galaxies and Quasars

Deadline: December 4, 2023

Lastname	Firstname	Noma
Antonutti	Adrien	31202100
Arnaud	Batiste	89492100
Delsart	Mathis	31302100
Lebrun	Léa	42072100
Michaux	Bastien	83982100
Remacle	Louis	82162100

Please, read carefully the following guidelines:

- Answer in English or French.
- **Do not modify the layout of the document.** Tour answers will be imported into gradescope for correction. Each answer must be contained in the "zone" predefined by the template.
- Clearly cite every source of information (even for pictures!);
- Whenever possible, use the `.pdf` format when you export your images: this usually makes your report look prettier[1];
- Do not forget to also submit your code on Moodle.

Image source [1].

The dataset [2] consists of 100,000 observations of space taken by the SDSS (Sloan Digital Sky Survey). Every observation is described by 17 feature columns and 1 class column which identifies it to be either a star, galaxy or quasar.

The project aims at building a ternary classifier for the following 3 classes: star, galaxy or quasar.

[1] This is because `.pdf` is a vector format, meaning that it keeps a perfect description of your image, while `.png` and other standard formats use compression. In other words, this means you can zoom as much as you want on your figure without decreasing image resolution. For simple plots, vector formats can also save a lot of memory space. On the other hand, we recommend using `.png` when you are plotting many data points: large scatter plots, heatmap, etc.

Question 1.1

Describe, briefly, your dataset (size, variables type, missing values, etc.).

Answer to 1.1

Le dataset contient 18 features différentes d'une taille de 100000 éléments chacune. Il y a donc 1.8 million d'éléments dans tout le dataset. Les différents types de variables présents dans le dataset sont les "int64_t" et "float64_t". Il y a aussi le type "object" pour la feature "class" qui sont les objects "Galaxy", "Stars" et "Quasar". Il n'y a aucune valeur manquante (NaN) dans le dataset.

Question 1.2

Based on a study of the features distribution (variance, number of unique values, number of missing values, etc.), can you identify some features that do not provide useful information for the classification task? Explain your analysis and remove those features from the dataset.

Answer to 1.2

Nous avons enlevé du dataset la feature "rerun_ID" en raison de sa singularité (= possède une seule valeur unique). Cette feature a donc une variance nulle et est inintéressante pour un algorithme de classification. Etant donné que la variance n'est pas une mesure normalisée, il est difficile d'évaluer la pertinence d'une variable en se basant sur un seuil défini. Vu que notre dataset ne contient aucune valeur manquante, on ne peut pas se baser sur ce critère. Nous avons des doutes sur l'utilité de la feature "cam_col" car elle possède uniquement 6 valeurs mais nous l'avions laissée car ces doutes ne sont pas fondés sur des principes théoriques. La feature "redshift" nous a aussi interpelé car sa variance est très faible (= 0.05339) mais nous préférons réaliser plus de tests (prochaines questions) pour vérifier son importance.

Question 1.3

What are the drawbacks (if any) of choosing a small test set (in proportion)? On the contrary, what are the consequences (if any) of a relatively large testing set (in proportion)?

Answer to 1.3

Choisir un ensemble de test trop petit par rapport à la taille du dataset présente des inconvénients. Cela peut rendre l'évaluation des performances du modèle moins fiable, car une quantité limitée de données de test peut ne pas être représentative de la globalité du dataset. De plus, cela peut aussi entraîner une évaluation biaisée du modèle, car il pourrait bien performer sur un petit ensemble de données mais être mal généralisé sur des données plus vastes. Cependant, un ensemble de données trop important présente le risque de surajustement, où le modèle apprend des détails spécifiques au jeu de données d'entraînement qui ne généralisent pas bien sur de nouvelles données, conduisant à une performance médiocre. Opter pour un ensemble de test relativement grand, proportionnellement à la taille du dataset, a des avantages en termes d'évaluation fiable des performances du modèle. Cependant,cela diminue aussi la taille de l'ensemble d'entraînement, affectant négativement la capacité du modèle à apprendre des motifs complexes. Trouver un équilibre entre la taille du jeu de test et celle du jeu d'entraînement est essentiel. Un compromis de 25 pourcents pour le jeu de test semble être une solution équilibrée, offrant des évaluations fiables sans sacrifier la qualité de l'apprentissage du modèle.

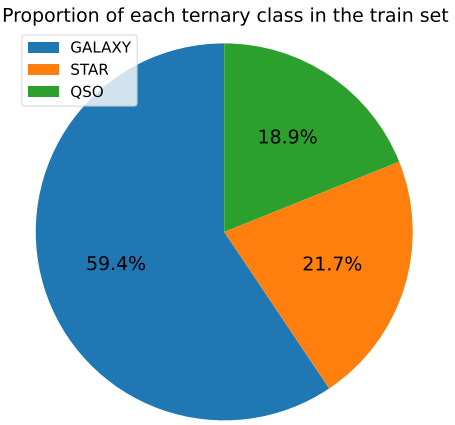
Question 2.1

Are the ternary classes balanced? What are the proportions of data in each class? Briefly, justify your answer and add a visualization.

Answer to 2.1

Non,la répartition entre les trois classes dans le dataset n'est pas homogène. Elles ne sont donc pas équilibrées. Il y a 59.4 pourcents de Galaxie, 21.7 pourcents d'étoiles et 18.9 pourcents de quasars.

Voici une visualisation à l'aide d'un "pie chart" :



Question 2.2

What would be the expected performance of a random classifier on this dataset?

Answer to 2.2

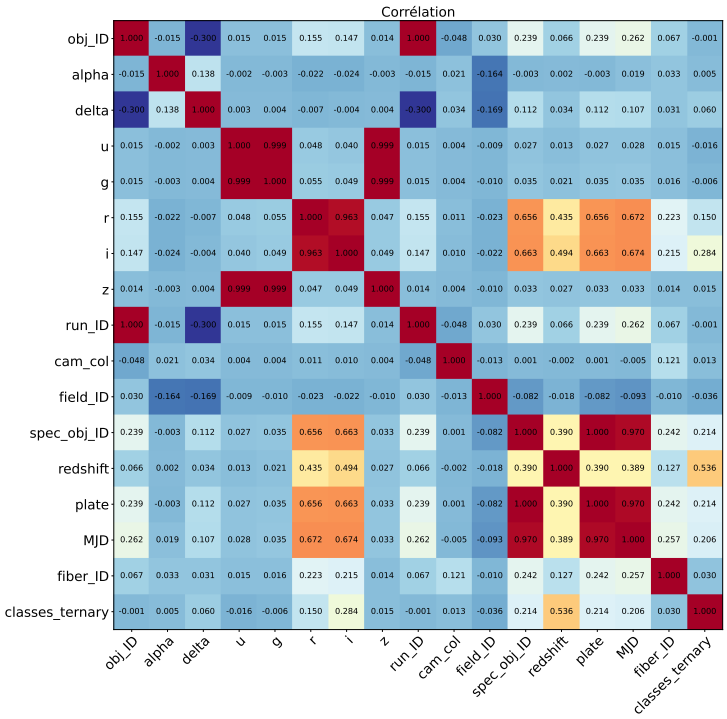
Ici on se demande qu'elle serait la performance d'un classificateur aléatoire. Dans ce cas, nous avons trois classes. Une façon de connaitre cette performance est de passer par la précision moyenne attendue. Dans le dataset d'entrainement, le pourcentage de GALAXY est trois fois plus grand que celui de STAR ou de QSO. Ce qui signifie que la précision pour détermine les GALAXY sera plus grande car il est mieux entrainé.

Question 2.3

Compute the correlation matrix of the dataset and plot it. Do you want to discard features based on this observation? Write clearly you decision rule.

Answer to 2.3

Voici la matrice de corrélation :



Il est judicieux de retirer certaines variables selon leur corrélation avec la variable classes_ternary. Premièrement, si la corrélation entre deux variables autres que classes_ternary est proche de 1, l'une des deux variables est redondante par rapport à l'autre. En effet, cela signifie que ces deux variables ont une relation parfaitement linéaire entre elles et n'apportent donc pas d'informations supplémentaires. Nous pouvons donc supprimer la variable obj_ID ou run_ID (corrélation de 1) et enlever deux variables parmi MJD, spec_obj_ID et plate (corrélation supérieure ou égale à 0.970). Parmi ces variables, nous gardons seulement run_ID et MJD. Deuxièmement, on remarque que la corrélation en valeur absolue entre la variable classes_ternary et d'autres variables est assez faible. Cela signifie que la relation positive ou négative entre la variable classes_ternary et ces autres variables est assez faible. Ces variables pourront donc être supprimées de notre base de données. Pour déterminer les variables à supprimer, il faut identifier les variables dont la valeur absolue de la corrélation avec la variable classes_ternary est inférieure à un certain seuil. Nous avons choisi le seuil à : 0.1. L'ensemble des variables supprimées est : ['obj_ID' 'alpha' 'delta' 'u' 'g' 'z' 'run_ID' 'cam_col' 'field_ID' 'fiber_ID']

Question 2.4

Why do we scale data? Justify properly, whether it is necessary or not for your feature set (X) and which scaler did you use.

Answer to 2.4

Standardiser les données est un processus important pour diverses raisons. Premièrement, les algorithmes basés sur les distances, comme K-NN, sont sensibles à l'échelle des features. La mise à l'échelle aide à garantir que les distances sont calculées de manière significative. Ensuite, standardiser permet de réduire la sensibilité à la "magnitude". Cela empêche la domination des grande magnitude par rapport aux petite magnitude. Pour nos données, la standardisation est nécessaire car la différence de magnitude entre les différentes features est très grande. Nous utilisons le StandardScaler(). Il met à l'échelle les features pour avoir une moyenne de 0 et un écart-type de 1. Il convient idéalement aux features qui suivent une distribution normale.

Question 3.1

Explain the idea of K-fold cross-validation and why it is useful. How the choice of K (in the cross-validation) impacts the bias and the variance of the scores obtained on the different folds? Choose and justify the number of folds you consider in this project.

Answer to 3.1

La K-fold cross-validation est une méthode permettant d'estimer la performance d'un modèle de machine learning. Elle consiste à diviser au hasard notre dataset disponible en K groupes de taille équivalente. Ensuite, nous allons itérer pour chaque groupe. Durant cette itération, notre classificateur est fit sur les K-1 autres groupes puis utilisé calculer l'erreur de prédiction au moyen d'un score, sur le groupe pour lequel est faite l'itération. Enfin, en prenant une moyenne de ce score sur chacune des itérations, nous arrivons à une estimation de l'erreur pas trop mauvaise.
Le nombre de groupes, K dans lequel nous divisons notre dataset lors de la K-fold cross-validation, a un impact sur le temps de calcul ainsi que sur les biais et la variance des scores obtenus sur les différents folds.
Premièrement, Ce nombre défini directement le nombre d'itérations nécessaire pour effectuer la méthode, ce qui signifie que plus, il est grand, plus la méthode prend du temps à calculer.
Deuxièmement, plus K est petit, plus le nombre de données pour fit le domaine est petit et moins le score final est basé sur un grand nombre de valeurs. Des biais pourraient apparaitre si la taille du train set est trop petite par rapport au dataset, et si le score final ne se base que sur peu de valeurs. De plus, plus K est petit, plus grande est la variance et moins la valeur obtenue est précise.
En ce qui concerne le nombre que l'on a choisit pour notre projet, ce nombre est de 5. Ce nombre est un bon compromis entre le temps nécessaire pour effectuer la méthode et le nombre de valeur sur lequel le score va se baser, qui est suffisamment grand pour minimiser le biais.

Question 3.2

Explain your methodology of model evaluation. More precisely, explain which hyperparameters you tune and the values you test for each of them. Next, provide the best hyperparameters configuration for each of the three models as well as their CV F1 score.

Answer to 3.2

Pour chaque modèle que nous voulons tester, nous effectuons une K-fold cross validation, en leur donnant différentes valeurs pour leurs hyperparamètres. Nous testons 3 modèles différents : une régression linéaire, une régression logistique et une K nearest neighbors.

Pour la régression linéaire, nous ne faisons varier aucun hyperparamètre. Nous recevons comme CV F1 score, 0.182 pour la classe STAR, 0.85 pour la classe GALAXY et 0.847 pour la classe QSO, ce qui nous fait en moyenne 0.626.

Pour la régression logistique, nous faisons varier l'hyperparamètre C qui est l'inverse de la force de régularisation. Nous avons testé le modèle pour les valeurs de C suivantes, 10^{-3}, 1, 10^{3}. Celle qui nous a donné le plus haut score est 10^{3}. Et le score que nous recevons est 0.987 pour la classe STAR, 0.873 pour la classe GALAXY et 0.899 pour la classe QSO, ce qui nous fait en moyenne 0.92.

Pour la K nearest neighbors, nous faisons varier les hyperparamètres n neighbors, weights et p. n_neighbors est le nombre de voisins qu'utilisera la méthode. Nous avons testé le modèle pour les valeurs de n_neighbors suivantes, 1, 5, 10 et 100. weights est la pondération, le poids donné à chaque voisin dans le modèle. Nous testons le modèle avec weights = 'uniform' et 'distance'. 'uniform' signifie que nous donnons le même poids à chaque voisin. 'distance' signifie qu'ils reçoivent comme poids l'inverse de leur distance. p est la manière dont nous calculons la distance dans le modèle. Nous testons le modèle pour p valant 1 et 2. Quand p = 1, la distance Manhattan est utilisée. Et quand p = 2, le modèle utilise la distance euclidienne.
Les valeurs des hyperparamètres qui nous ont donné le plus haut score sont n_neighbors = 5, weights = 'distance' et p = 1. Et le score que nous recevons est 0.963 pour la classe STAR, 0.965 pour la classe GALAXY et 0.931 pour la classe QSO, ce qui nous fait en moyenne 0.953.

Question 3.3

Based on your answers to previous questions, select a final model that you will keep as classifier. Justify.

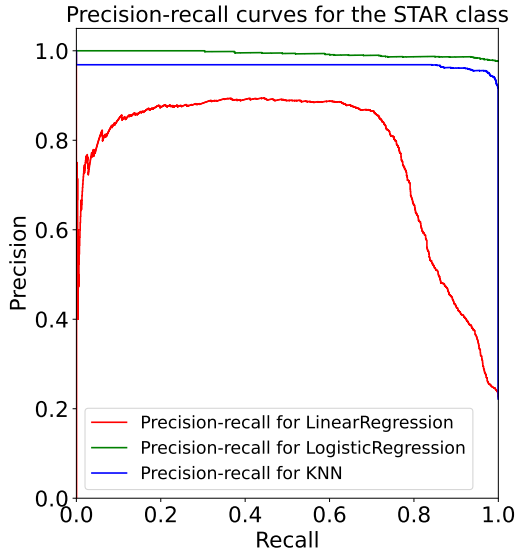
Answer to 3.3

Nous allons garder le modèle, K nearest neighbors avec les hyperparamètres, n neighbors = 5, weights = 'distance' et p = 1, parce qu'en moyenne, il a un plus au score CV F1 sur les trois classes.

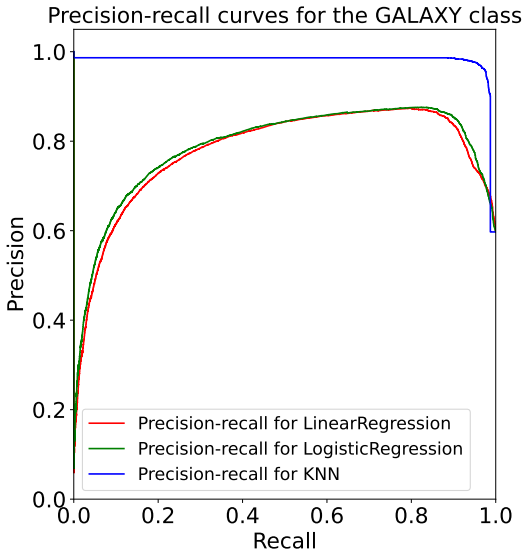
Question 3.4

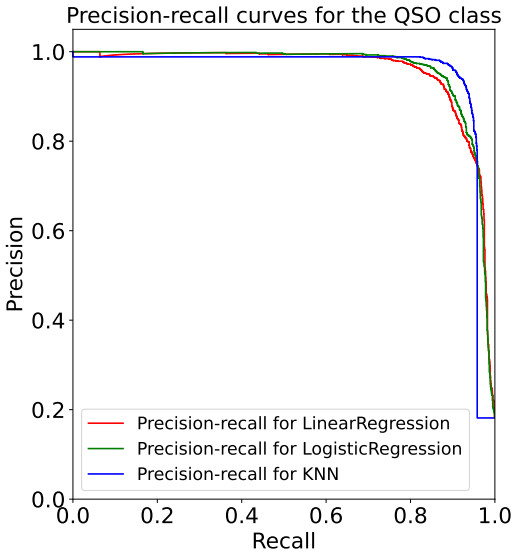
Plot the precision-recall curve for the three methods, one figure for each class. What happens to the precision and recall when the threshold tends to 0? And when it tends to 1? Explain and, if possible, establish a link with Question 2.1. For each class, for each method: what threshold would you use?

Answer to 3.4



Quand le seuil tend vers 0, le modèle aura tendance à prédire tout le temps 1, et donc, que la donnée appartient toujours à la classe. En conséquent, la precision tend vers la proportion de données appartenant à la classe dans le dataset, et le recall tend vers 1.
Quand le seuil tend vers 1, le modèle aura tendance à prédire tout le temps 0, et donc, que la donnée n'appartient jamais à la classe. En conséquent, le recall tend vers 0, et la precision tend vers 1 ou 0.





Question 4.1

Use the test set to estimate the precision, recall and F1 score of your final model and validate its performance on unseen data. Observe if the scores are similar to the ones estimated with your cross-validation. Are you satisfied by the performance of your classifier, in view of the task for which it will be used?

Answer to 4.1

Performance				
class	STAR	GALAXY	QSO	Mean
precision	0.942	0.967	0.959	0.956
recall	0.988	0.966	0.907	0.954
F1	0.964	0.967	0.932	0.954

Les valeurs obtenues ci-dessus sont très proches de celles obtenues lors de la cross-validation. Par exemple, la moyenne sur les 3 classes de F1 vaut 0.954, et pour la cross-validation, elle valait 0.953.
Le modèle a l'air de bien prédire les classes des objets spatiaux. Cependant, il produit une erreur non négligeable. Il faut donc ne pas lui laisser effectuer cette tâche seul.

References

[1] Stellar classification dataset - sdss17. https://www.kaggle.com/code/satoru90/stellar-classification-dataset-sdss17. Accessed: 2023-11-24.

[2] Nuno Ramos Carvalho. Sdss galaxy subset, March 2022.