# **INTRODUCTION**

<div style="text-align: justify">

Lorsqu’il est question de classification automatique d’images de galaxies ou de classification de courriels indésirables, une multitude d’algorithmes peuvent être utilisés pour résoudre ce problème. Par exemple, les arbres de décision, Bayes naïf et les K plus proches voisins (*KNN*). Seulement, ces algorithmes présentent des caractéristiques et des performances de classification différentes. Sans compter que la nature des données à classifier doit être prise en compte lors du choix de l’algorithme de classification. C’est-à-dire que non seulement certains algorithmes performent mieux que d’autres, mais certains algorithmes performent mieux avec des données d’une nature spécifique.

Parallèlement, les méthodes d’évaluation de l’apprentissage des algorithmes ont des performances différentes en terme de temps d’exécution et de variabilité des résultats. Partant de ces faits, l'objectif du laboratoire actuel est de comparer les performances de ces trois algorithmes de classification sur l’ensemble de données des images des galaxies et sur celui des courriels indésirables. D’abord, l’approche utilisée pour manipuler les données et pour créer les ensembles de données sera décrite. Par la suite, une discussion présentera une comparaison des méthodes d’évaluation utilisées. D’ailleurs, l’impact des hyperparamètres et des prétraitements sur les performances des modèles sera étudié. De plus, un parallèle sera établi entre les deux ensembles de données. Delà une recommandation sera émise concernant le type de classifieur à utiliser pour chaque ensemble de données. Enfin, une courte discussion portera sur les améliorations qui pourraient possiblement être appliquées aux classifieurs présentés dans ce rapport.

</div>

# **SUJET #1**

**1.1 - Approche de manipulation de données**

**1.2 – Création des ensembles de données**

**1.3 - Résultats**

**1.4 - Ensembles produits et leur forme. Représentation graphique.**

<div style="text-align: justify">

Avant tout, les primitives des galaxies extraites au laboratoire 1 ont été combinées à celles données pour le laboratoire. Pour ce faire, un script a été programmé pour extraire les quatre primitives du laboratoire 1 des plus de 16 000 images. Ce script a enregistré les primitives dans un fichier csv et ce, dans le même ordre que le fichier contenant les primitives données. Ensuite, le contenu du fichier des primitives du laboratoire 1 a manuellement été inséré dans le fichier des primitives données.

<br/><br/>

En ce qui concerne la lecture des données avant l'entraînement, le contenu de chacun des fichiers de primitives a été chargé dans deux listes; une pour les primitives et une pour la classe des observations. Ainsi, à la suite de la lecture des fichiers contenant les primitives de chaque ensemble, les listes suivantes ont été chargées :

</div>

|Liste|Contenu|Forme du vecteur|

|--|--|--|

|gala\_features|Primitives des galaxies|(16908, 79)|

|gala\_labels|Classes des observations des galaxies|(16908,)|

|pourriel\_features|Primitives des courriels|(2760, 57)|

|pourriel\_labels|Classes des observations des courriels|(2760,)|

<div style="text-align: justify">

Les listes pour les galaxies contiennent 8776 galaxies spirales et 8132 galaxies smooth. Quant aux listes pour les courriels, elles contiennent 1105 observations classées comme pourriel et 1655 observations qui ne sont pas des pourriels.

<br/><br/>

C’est important de noter que le type des valeurs des classes dans ces ensembles de données a été changé afin de faciliter l’interprétation des prédictions. De ce fait, dans les fichiers les classes des galaxies sont dénotées par les valeurs numériques 0 et 1. Lorsque ces données sont lues, les valeurs des classes sont changées pour les chaînes de caractère « spiral » et « smooth ». Par conséquent, les données contenues dans gala\_labels et pourriel\_labels doivent être encodées avec un LabelEncoder. Cet encodeur donne à nouveau une valeur numérique à chaque classe. De ce fait, lorsqu’une prédiction est effectuée, le LabelEncoder peut être utilisé pour obtenir la représentation en chaîne de caractères à partir de la prédiction sous forme numérique.

</div>

# **SUJET #2**

**2.1 - Méthodes de validation : Leave-one-out cross-validation, Leave-p-out cross-validation, k-fold cross-validation, holdout**

**2.2 - Présentation des approches de validation utilisées**

**2.3 - Résultats de comparaison des méthodes**

<div style="text-align: justify">

La première méthode d'évaluation de l'apprentissage utilisée dans ce rapport est le *Holdout*. Cette méthode sépare l’ensemble de données en deux sous-ensembles; un pour l'entraînement et un pour la validation. De ce fait, un *spit* 80/20 est utilisé pour la validation des modèles présentés dans ce rapport. C’est-à-dire que 80 % des données sont utilisés dans le sous-ensemble d’entraînement, alors que 20 % constitue le sous-ensemble de validation. D'ailleurs, la séparation de ces sous-ensembles est faite d’une façon stratifiée. Cette façon permet de conserver un ratio des différentes classes des observations presque équivalent entre les deux sous-ensembles. Parallèlement, avant la séparation de ces sous-ensembles, les données sont mélangées. Cela permet d’éviter d'entraîner un modèle sur un sous-ensemble d'entraînement qui n’est pas représentatif de la distribution des données. Par exemple, seulement les premiers 60 % des données d'un ensemble sont ordonnées. Dans cette situation, lors de la séparation 80/20 de cet ensemble, les données du sous-ensemble d'entraînement seront ordonnées contrairement à celles du sous-ensemble de validation. Ainsi, le modèle sera entraîné sur des données non représentatives de toutes les données. De ce fait, puisque les distributions des ensembles des galaxies et des courriels ne sont pas connues, c’est plus prudent d’assumer qu’elles ne sont pas stochastiques et de mélanger les données. Toutefois, la conséquence de la méthode *Holdout* est que le sous-ensemble de validation n’est pas utilisé pour l'entraînement. Dans le cas présent, 20 % des données sont sacrifiées pour pouvoir effectuer la validation. La méthode *K-folds* considère ce problème.

<br/><br/>

À cet égard, la méthode *K-folds* permet de réduire la quantité de données qui n’est pas utilisée pour l’entraînement. Pour se faire, l’ensemble original est séparé en k sous-ensembles. Chacun de ceux-ci sont utilisés à tour de rôle comme sous-ensemble de validation alors que les autres k - 1 sous-ensembles sont utilisés pour l'entraînement. Par exemple, dans ce rapport k = 10. Ainsi, dix entraînements ont lieux. Lors de chacun de ces entraînements, 10 % des données sont utilisées pour la validation. Par la suite, les moyennes des métriques des k entraînements sont calculées. La méthode *K-folds* permet donc d’assurer que toutes les observations sont prises en compte dans les métriques cumulatives des entraînements. Ce dernier fait suggère que les modèles entraînés lors de ce laboratoire sous une validation *K-folds* où k = 10 devraient avoir des résultats plus représentatif de la vérité que ceux entraînés avec un *Holdout* 80/20. Cette affirmation est expliquée dans le prochain paragraphe. Par ailleurs, comme pour le *Holdout*  les ensembles présentés dans ce rapport sont aussi mélangés avant d'être séparés par *K-folds* de façon stratifiée. Or, en dépit ces séparation des données, la qualité du résultat d’une validation dépendent des données choisie pour les sous-ensembles d’apprentissage et de validation.

<br/><br/>

Compte tenu de ce qui précède, la qualité de la représentation du résultat d’une validation, c’est-à-dire le fait que ce résultat représente réellement le résultat attendu, dépend entre autres de quelles données sont présentes dans les sous-ensembles d’apprentissage et de validation. Par exemple, deux validations sont effectuées sur le même ensemble. Cependant, pour chaque validation, les données ont un ordre différent au moment de les séparer en sous-ensemble d’apprentissage et de validation. Cet ordre différent pourrait être une conséquence du fait de mélanger les données telles qu’expliquées plus tôt lors de l’introduction de la méthode *Holdout*. Dans ce cas, puisque les sous-ensembles d’apprentissage sont différents, les résultats de ces deux validations seront différents. D’après ces faits, comment garantir que le résultat d’une validation n’est pas isolé; c’est-à-dire qu’il est représentatif de tous les résultats que l’on peut obtenir en considérant toutes les combinaisons de sous-ensemble d’apprentissage et de validation possibles? Obtenir la distribution des résultats possible serait très coûteux pour déterminer si un résultat prédit est représentatif. Or, intuitivement un résultat est inversement proportionnellement représentatif à la variance d’un sous-ensemble des résultats en considérant que le biais de ce sous-ensemble est très faible ou nul. Autrement dit, ce résultat est la moyenne d’une distribution sans biais et avec une variance très faible. Une façon de réduire cette variance est de calculer les moyennes des métriques de plusieurs exécutions de la validation. Ainsi, les métriques cumulatives sont le résultat attendu à une prochaine validation d’après la distribution des résultats observés à ce jour. Cette technique revient à utiliser la méthode de validation *K-folds*. Ainsi, a priori les résultats obtenus par la validation *K-folds* ont une variance plus faible et sont donc plus représentatifs que ceux obtenus par la méthode *Holdout*. Néanmoins, ce n’est pas parce qu’un résultat est plus représentatif qu’il est supérieur à un autre. C’est-à-dire qu’en prenant en compte que chaque méthode de validation a sa propre distribution de résultats possibles, hypothétiquement ceci implique la présence d’une certaine probabilité sous laquelle les résultats les plus forts d’un *Holdout* soient supérieurs aux résultats les plus faibles d’un *K-folds*. Cela dit, comment est-ce que les résultats de ces deux méthodes de validations se comparent pour ce laboratoire?

<br/><br/>

En ce qui concerne la différence entre les résultats obtenus par les méthodes Holdout et K-folds, celle-ci est étonnamment très faible. Que cela soit pour le score F1 ou pour la précision, la différence entre le Holdout et le K-folds est d’environ 0.01 et moins. De façon générale, les résultats du Holdout sont légèrement supérieurs à ceux du K-folds. Plus précisément, pour le score F1 et la précision, 5/8 des résultats obtenus par Holdout sont supérieurs à ceux obtenus par K-folds. Malgré tout, théoriquement les résultats du K-folds sont plus représentatifs des résultats attendus si ces algorithmes étaient entraînés sur toutes les données disponibles.

</div>

|Algorithme|Paramêtres|Prétraitements|Holdout (F1)|K-folds (F1)|Holdout (Accuracy)|K-folds (Accuracy)|

|--|--|--|--|--|--|--|

|tree\_pourriels|max\_depth=10|aucun|0.908301209133805|0.914205829307939|0.909420289855073|0.914492753623188|

|tree\_galaxies|max\_depth=10|aucun|0.918360252607762|0.913400423280966|0.9183914843288|0.913413768630234|

|bayes\_gaussian\_pourriels|aucun|aucun|0.801877184600297|0.813957301704396| 0.800724637681159|0.81268115942029|

|bayes\_gaussian\_galaxies|aucun|aucun|0.8037068645517|0.802685181544932|0.805736250739208|0.80476697421339|

|bayes\_multinominal\_pourriels|aucun|MinMaxScaler|0.910982065021301|0.900368426280921|0.911231884057971|0.901811594202899|

|bayes\_multinominal\_galaxies|aucun|MinMaxScaler|0.809501045429831|0.802830012974806|0.809580130100532|0.802933522592855|

|knn\_pourriels|n\_neighbors=20 weights=distance|MinMaxScaler|0.906917461740411|0.898690577528921|0.907608695652174|0.89963768115942|

|knn\_galaxies|n\_neighbors=48 weights=uniform|MinMaxScaler|0.899208199221601|0.905635426056412|0.899172087522176|0.905606813342796|

# 

# **SUJET #3**

**3.1 - Pour chacun des modèles d’apprentissage élaborés, explication de l’impact des hyperparamètres et des prétraitements sur les performances des modèles.**

<div style="text-align: justify">

Pour ce qui est de l’arbre de décision, la profondeur maximale qui a obtenu la meilleure performance est de 10. Curieusement, ceci s’applique aux deux ensembles de données, soit celui sur les galaxies et celui concernant les courriels. D’ailleurs, pour ces deux ensembles de données, la profondeur la moins performante est de 3. Ainsi, la profondeur maximale optimale pour ces ensembles doit être de 10 ou avoisinant une valeur près de 10.

<br/><br/>

Concernant les prétraitements appliqués à l’arbre de décision, la discrétisation non supervisée diminue la performance de l’arbre sur l’ensemble des courriels d’environ 0.04 pour le score F1. Pour ce qui est du MinMaxScaler, la différence entre les résultats obtenus avec son utilisation et sans prétraitement est non significative.

<br/><br/>

Par ailleurs, pour Bayes Gaussian, aucune comparaison n’est possible pour cet algorithme. Cela est dû au fait que celui-ci n’a pas de paramètres. De plus, Bayes Gaussian n’est pas évalué avec un prétraitement. D’ailleurs, Bayes Multinomial est utilisé avec un seul prétraitement sur l’ensemble des galaxies. Par contre, Bayes Multinomial est testé avec les deux prétraitements sur l’ensemble des pourriels. À ce sujet, les résultats de l’utilisation de MinMaxScaler et de KBinsDiscretizer sont respectivement 0.911 et 0.873 pour le score F1. Ainsi, le prétraitement MinMaxScaler est préférable pour Bayes Multinomial sur l’ensemble de données des pourriels.

<br/><br/>

Ensuite, pour KNN sur l’ensemble des pourriels, la valeur distance pour le paramètre weights est définitivement plus appropriée que la valeur uniforme. Dans les faits, toutes les exécutions où le paramètre weights est distance ont obtenu des résultats supérieurs aux exécutions où la valeur de ce paramètre est uniforme. Au contraire, pour l’ensemble de données des galaxies, aucune différence significative ne réside entre la performance de classification conformément à chacune de ces deux valeurs du paramètre distance.

<br/><br/>

D’autre part, les prétraitements appliqués sur KNN apportent un gain important sur les scores de classification. En effet, le résultat F1 sans prétraitement et avec MinMaxScaler pour l’ensemble des galaxies est respectivement 0.685 et 0.899. Cela représente un gain de 0.214. Pour l’ensemble des pourriels, le résultat F1 sans ce prétraitement et avec celui-ci est respectivement 0.802 et 0.907. Cette fois le gain est de 0.105. Enfin, sur ce même ensemble de données, le résultat F1 avec le KBinsDiscretizer est de 0.839; ce qui représente un petit gain de 0.037.

</div>

# **SUJET #4**

**4.1 - Discussion mettant en parallèle la nature des données.**

**4.2 - Un ensemble de données se démarque par rapport à un algorithme de classification?**

**4.3 - Creuser sur la nature intrinsèque des données, sur le processus d’extraction des primitives, sur les distributions des données.**

<div style="text-align: justify">

Pour ce qui est de l’ensemble de données représentant les courriels indésirables, les données sont directement extraites à partir de plusieurs séries de mots. Dans ce cas-ci, l’extraction des primitives se fait relativement facilement. À l’inverse, l’ensemble de données représentant les images de galaxies requiert plusieurs étapes de prétraitement avant de pouvoir extraire les primitives. En d’autres mots, l’extraction des primitives discriminantes est naturellement plus difficile à faire sur des images (galaxies) que sur du texte (courriels indésirables). À haut niveau, la majeure partie des résultats portant sur la précision de classification se retrouve à être plus élevé pour les courriels indésirables que les images de galaxies. Cette observation s’explique simplement avec ce qui a été souligné en début de paragraphe. La classification des résultats provenant d’une série de mots est plus simple et plus homogène à traiter que des images de types jpg par exemple.

<br/><br/>

Pour ce qui est des algorithmes de classification, il est logique de dire que l’arbre de décision est idéal pour traiter un ensemble de données comme les courriels indésirables. Il serait même possible de tirer une telle conclusion sans même regarder les résultats portant sur la précision de classification. En effet, il est non seulement plus facile de donner un seuil à l’arbre de décision pour classifier au mieux les données, l’arbre de décision a aussi un plus grand pouvoir sur la classification des données qui ne varient pas grandement comme celles des courriels indésirables.

<br/><br/>

Pour ce qui est de l’ensemble des données portant sur les images de galaxies, il est plus difficile de choisir un type de classificateur idéal sans regarder aux résultats obtenus. Il est peut-être difficile de choisir le bon classificateur, mais il est plus facile de dire quel classificateur est à éviter. Par exemple, avec la grande quantité d’échantillons obtenus pour les images de galaxies, KNN devient ridiculement lent. Aussi, il est reconnu pour être plus sensible au bruit. Du côté du classificateur Bayes, il a un plus de pouvoir sur la classification des données de type texte ou des données générales. L’hypothèse d’utiliser l’arbre de décision qui est résistant au bruit semble prometteuse. Dans ce cas-ci, malgré le grand nombre d’échantillons obtenus pour les images de galaxies, on constate que la quantité n’est pas nécessairement synonyme de précision de classification. En effet, 16908 échantillons d’images de galaxies contre 2760 échantillons de courriels électroniques montrent bien que la qualité des primitives joue un rôle très important sur la performance de classification. Cependant, lorsque la quantité d’échantillons et la qualité des primitives sont au rendez-vous, la performance de classification augmente grandement.

<div/>

# **SUJET #5**

**5.1 - Classificateur recommandées pour les ensembles de données. Quelles conditions (nombre de données privilégié)?**

<div style="text-align: justify">

À partir du tableau illustrant tous les meilleurs résultats obtenus des classificateurs, plusieurs observations peuvent être faites. Entre autres, le tableau permet d’énumérer l’ordre des algorithmes les plus prometteurs aux algorithmes les moins prometteurs. À haut niveau, on retrouve en première position les arbres de décisions, en deuxième position les K plus proches voisins (KNN) et en troisième position l’algorithme Bayes.

<br/><br/>

Pour ce qui est du classificateur de type arbre de décision, plusieurs facteurs appuient la raison pour laquelle il est l’algorithme le plus prometteur pour les deux ensembles de données. Dans un premier temps, sachant que les arbres de décisions sont par exemple moins sensibles au bruit que KNN, il est logique que pour le cas des galaxies, les arbres de décisions soient un meilleur choix. Aussi, la forte ressemblance retrouvée au niveau de la précision pour les deux ensembles de données avec l’arbre de décision permet d’indiquer que le taux de variation des deux ensembles de données est très bas.

<br/><br/>

Pour ce qui est du classificateur de type KNN, on constate qu’avec l’ensemble de données des courriels indésirables, le niveau de précision est nettement supérieur que l’ensemble de données des galaxies. Plus précisément, on retrouve une différence de précision entre 15% et 20% pour les deux ensembles de données. Puisque les galaxies ont un taux de bruits beaucoup plus élevé que les courriels indésirables, il est logique que KNN soit plus performant avec les courriels indésirables. La raison est simple, parmi les limitations de KNN, on retrouve une forte sensibilité au bruit.

<br/><br/>

Pour ce qui est du classificateur de type Bayes, deux types de distribution a été utilisée, Bayes Gaussian et Bayes Multinomial.

<br/><br/>

En se penchant sur les résultats obtenus avec Bayes Gaussian en F1 et au niveau de la précision, on constate que les résultats pour les deux ensembles de données sont très similaires. Avec une différence de plus ou moins 2%, de tels résultats s’expliquent par l’utilisation de la même formule de probabilité pour toutes les valeurs et classes retrouvées avec Bayes Gaussian. En fait, lorsque la quantité d'échantillons se met à être assez élevé, ce qui actuellement la cas, la distribution Multinomial à tendance à pencher vers une distribution Gaussienne, ce qui vient expliquer une fois de plus les résultats similaires obtenus.

<br/><br/>

Ensuite, en se penchant sur les résultats obtenus avec Bayes Multinomial, le résultat portant sur la précision est environ 10% plus élevé avec l’ensemble de données des courriels indésirables. Cette différence s’explique par le simple fait que Bayes Multinomial est reconnu et spécialement conçu pour l’analyse de texte ainsi que la classification générale. Il va donc de soi qu’utiliser Bayes Multinomial pour analyser des données provenant des images de galaxies soit moins intéressant à utiliser.

<br/><br/>

Enfin, avec ce qui a été discuté, l’arbre de décision est très prometteur autant pour l’ensemble de données des courriels indésirables que celui des galaxies.

</div>

# 

# **SUJET #6**

**6.1 - Pistes d’amélioration des classificateurs.**

<div style="text-align: justify">

Différentes améliorations pourraient être appliquées pour améliorer la performance des classificateurs. Avant tout, certains paramètres du prétraitement KBinsDiscretizer peuvent être optimisés, plus spécifiquement le paramètre n\_bins. Aucune expérimentation n’a été conduite dans ce rapport pour trouver la valeur optimale de ce paramètre.

<br/><br/>

Par la suite, lors de la recherche des meilleurs hyperparamètres, la méthode de validation *K-folds* devrait être utilisée à la place de la méthode *Holdout*. Comme expliqué à la question 2, les résultats de la méthode *Holdout* sont plus variables que ceux de la méthode *K-folds*. De ce fait, le choix des hyperparamètres est peut-être basé sur un modèle qui n’a pas appris aussi bien qu’attendu. Ainsi, utiliser la méthode *K-folds* permettrait de choisir les hyperparamètres en fonction d’un modèle qui est plus représentatif des performances attendues.

<br/><br/>

Par ailleurs, les valeurs testées pour le paramètre max\_depth des arbres de décision sont un ensemble et non un intervalle. En effet, les valeurs testées font partie de l’ensemble {None, 3, 5, 10, 30, 50}. Par conséquent, la valeur optimale de ce paramètre n’est peut-être pas celle identifiée, mais un voisin de celle-ci. Afin de choisir la meilleure valeur possible pour ce paramètre, les valeurs testées devraient être l'intervalle [3, 50] en plus de la valeur None.

</div>

# 

# **CONCLUSION**

<div style="text-align: justify">

En conclusion, lors de la mise en place de système d’apprentissage machine il existe une grande variété d’algorithmes et de paramètres qui peuvent être utilisés ou appliquer aux données pour la tâche de classification. Pour ce laboratoire il a été question de trouver parmi une liste de prétraitement, d’algorithme et des paramètres associés aux algorithmes pour trouver la meilleure combinaison qui permettrait d’avoir le meilleur résultat de classification de nos données.

<br/><br/>

En effet, pour nos données les classificateurs en arbre semblent être le meilleur choix suivi de près par KNN et ceci pour les galaxies et pour les pourriels. Pour les deux ensembles, la profondeur optimale semble être de 10 pour l’arbre de décision. La classification utilisant Bayes (Gaussien et Multinomial) a donné des résultats inférieurs aux autres classifications dans le cas des galaxies et des courriels. Le Bayes Gaussian a montré une baisse de précision sur nos données et le Bayes Multinominal montré une légère augmentation de la précision de l’apprentissage. Pour les prétraitements, celui du MinMaxScaler semble apporter une précision légèrement meilleure tandis que le KBinsDiscretizer réduit la précision observée.

<br/><br/>

De plus, certains aspects pourraient être améliorés. Par exemple, les différents paramètres de prétraitement n’ont pas été réellement étudiés (n\_bins par exemple). Une simple valeur ne retournant pas d’erreur a été utilisée. Aucune recherche du paramètre optimal n’a été effectuée pour ces valeurs. Pour les algorithmes d’apprentissage, des ensembles de valeur possible pour les paramètres ont été utilisés au lieu de tester toutes les valeurs de l’intervalle. Une précision supplémentaire aurait surement été atteinte si on avait été plus granulaire pour trouver les hyperparamètre optimaux.

</div>

***Information à mettre sous les tableaux:***

NOTE:

Une cellule ‘’nan’’ (rouge lorsque exécutée) représente pour une section en particulier un algorithme qui n’a pas été traité (non demandé pour ce laboratoire). Une cellule qui a la valeur 1 est une valeur bidon pour remplir le tableau. Pour afficher correctement le tableau final qui illustre tous les résultats, il faut ajouter des lignes vides pour obtenir le même nombre d’algorithme pour chaque section.

Il faut aussi souligner que le prétraitement indiqué dans ce tableau n’est pas nécessairement celui utilisé pour le K-fold.

Dans le tableau ci-dessous les lignes 3 et 4 sont à ignorer. Celles-ci sont présentes pour corriger un problème de nature technique.