Classification des données de feuilles

par
Jean Paul Latyr FAYE et Mingxuan SUN

Projet IFT712 2019

11 décembre 2019

Table des matières

Liste de acronymes

SVM	*Support Vector Machine* Machine à vecteurs de support
DTC	*Decision Tree Classifier* Classificateur d'arbre de Décision
KNN	*K-Nearest Neighbour* K-plus proches voisins
LDA	*Linear Discriminant Analysis* Analyse Discriminante Linéaire
NN	*Neural Networks* Réseaux de Neurones
LR	*Logistic Regression* Régression logistique

1 Introduction

L'apprentissage automatique est devenu aujourd'hui une partie intégrante de notre vie que nous soyons chercheurs, praticiens etc.. Indépendamment de leur domaine, les utilisateurs de l'apprentissage automatique ont presque un seul but qui est de faire de bonnes prédictions. La classification est l'une des techniques de l'apprentissage automatique dont l'objectif principale est de prédire la classe d'appartenance de toute donnée d'entrée dans le processus de classification. Les méthodes de classifications comportent généralement trois phases : une première phase d'entrainement, une deuxième phase de validation et une troisième phase dédiée à la prédiction. Les données dont on dispose dans la classification sont séparées aléatoirement en deux parties. Une partie dite de données d'entrainement et une autre partie pour la validation du modèle. Le modèle de classification est entrainé dans la phase d'entrainement en utilisant les données d'entrainement. Le but ici est de trouver surtout les paramètres et les hyper-paramètres du modèle. Ce processus se fait généralement en minimisant une fonction de perte. Après cette phase d'entrainement, on cherche à savoir comment le modèle se généralise sur des données jamais vues, c'est-à-dire les données de validation. Ainsi, on espère qu'on modèle qui parvient à bien généraliser sur des données de validation aura tendance à conduire à une prédiction acceptable dans la troisième phase de la classification. Comme déjà évoqué précédemment, le but final de la classification est de pouvoir faire de bonnes prédictions sur des données qui sont inconnues du modèle. Cependant, comment pouvons-nous s'assurer que le modèle ne mémorise pas juste les données d'entrainement pour conduire à

de mauvaises prédictions ? Quel modèle devons nous sélectionner pour notre problème en question ? Quelles sont les modifications nécessaire à apporter aux données dans le but d'améliorer les résultats de prédiction ? Comment devons nous trouver les hyper-paramètres de notre modèle pour une convergence rapide mais surtout pour pouvoir mieux généraliser dans le futur ? Ce sont là les questions que nous allons essayer de donner des réponses en appliquant des modèles de classification sur des données de feuilles.

Dans la première partie de ce projet, nous allons essayer de se familiariser avec les données en essayant de les exploiter le plus claire possible. Ceci nous permettra de savoir si les données nécessitent une transformation et quels sont les modèles qui pourraient performer mieux. En effet, cette étude pourra donner une information quant à la forme de la distribution des données, c'est-à-dire une distribution gausssienne etc.. Compte tenu des résultats d'exploration, nous allons faire une mise en échelle des données suivie d'une validation croisée pour augmenter la performance des modèles et trouver leurs hyper-paramètres. Après avoir présenté les résultats obtenues, nous allons finalement conclure en donnant le score obtenu lors de la soumission, dans le site de Kaggle, des résultats de notre meilleur modèle de classification sur les données de teste.

2 Étude préliminaire des données

Dans cette section, nous allons se concentrer sur la compréhension des données de feuilles à classifier. On s'intéressera d'abord à le statistique des caractérsitiques. En effet, l'exploration des données nous permettra d'avoir une bonne idée sur les modèles de classification à choisir due sa nécessité.

2.1 Description statistique

Nous allons utiliser les fonctions descriptives statistiques telles que la moyenne, la déviation standard etc. mais aussi une description graphique. Cependant nous commencerons d'abord par une observation plus proche de nos données et nous chercherons à savoir les différentes types d'attributs et la dimension de ces caractéristiques. Le but ici est simplement de mieux comprendre nos données avant d'appliquer les techniques de classification.

2.1.1 observation rapide des données

Une étape simple mais importante consiste à jeter un coup d'œil sur les données brutes. Ceci permet en fait d'avoir une bonne idée de chacune des

id	species	margin20	...	shape20	...	texture20
1	Acerr_Opalus	0.025391	...	0.000430	...	0.022461
2	Pterocaryar_Stenoptera	0.007812	...	0.000460	...	0.006836
3	Quercusr_Hartwissiana	0.005859	...	0.000507	...	0.027344
5	Tiliar_Tomentosa	0.003906	...	0.000404	...	0.012695
6	Quercusr_Variabilis	0.007812	...	0.001110	...	0.008789
...	...	...	...	...	...	...
1575	Magnoliar_Salicifolia	0.019531	...	0.000340	...	0.009766
1578	Acerr_Pictum	0.007812	...	0.000650	...	0.012695
1581	Alnusr_Maximowiczii	0.001953	...	0.000455	...	0.006836
1582	Quercusr_Rubra	0.003906	...	0.001181	...	0.027344
1584	Quercusr_Afares	0.011719	...	0.000562	...	0.000000

TABLE 1 – *Les caractéristiques margin, shape et texture des données brutes. Les espèces correspondent aux différentes classes. Note l'ordre de grandeur différente entres les caractéristiques. La première colonne représente l'identité des espèces.*

variables mais surtout de pouvoir mieux indexer si on cherche à s'adresser à une caractéristique spécifique. Il suffit ici de visualiser les quelques lignes des données comme le montre le Tableau 2.1. Une inspection des données permet de constater que la première colonne décrit les identités des classes alors que la deuxième colonne correspond aux classes. Nous avons en gros trois différentes caractéristiques que sont les *margins*, les *shapes* et les *texture*. Cependant on notera que chacune de ces caractéristiques est composée de plusieurs sous-caractéristiques comme par exemple, *margins* comporte *margin1* jusqu'à *margin64*.

2.1.2 Dimension et type de caractéristique

Connaître la dimension des données, c'est-à-dire combien de lignes et de colonnes comporte les données est important dans le choix des méthodes de classification. Il est aussi important de prendre connaissance des types des différentes variables. Ceci devra permettre d'unifier ou de transformer certain type de données en d'autres types plus convénientes aux modèles de classification qu'on devra tester. On peut constater rapidement que les données sont composées de 990 lignes et de 194 caractéristiques. Ces derniers sont toutes des variables continues à l'exception de la caractéristique classe qui est de type catégorique. Le Tableau 2.1 donne certaines de ces informations.

2.1.3 Étude statistique des caractéristiques

Nous pouvons avoir une bonne idée sur la distribution des variables en regardant en détail des fonctions statistiques comme la moyenne, la déviation standard, les percentiles, le minimum, le maximum etc.. On montre dans le Tableau 2.1.3 les résultats obtenues en appliquant la fonction *describe*() de la librairie Pandas sur certaines caractéristiques. On découvre que les variables *textures* ont une variation beaucoup plus importante comparées aux caractéristiques *margin* et *shape*. Cependant, sa valeur moyenne est presque comparable à celle du *margin* mais beaucoup plus élevé que la moyenne de la variable *shape*. Toutes ces observations révèlent déjà qu'une transformation des données brutes pourrait être importante avant l'application des modèles de classification. Beaucoup de méthodes d'apprentissage automatique sup-

	margin20	shape20	texture20
count	792.000000	792.000000	792.000000
mean	0.013154	0.000549	0.014582
std	0.009694	0.000363	0.016474
min	0.000000	0.000061	0.000000
25%	0.005859	0.000334	0.002930
50%	0.011719	0.000449	0.009766
75%	0.019531	0.000611	0.020508
max	0.048828	0.002300	0.099609

TABLE 2 – *Statistique de quelques variables. Noter la variation de la caractéristique shape trop faible comparée aux autres variables margin et texture*

posent en général que les données suivent une distribution gaussienne ce qui n'est pas toujours vérifiée. Cependant, même si cette hypothèse est vérifiée il peut arriver que la distribution soit balancée à gauche ou à droite. Une détection préalable de ce comportement informe sur une nécessité de transformation des données avant l'applications des modèles de classification. Cette transformation vise à mieux centrer la distribution conduisant à une meilleure justesse des modèles de classification qui admettant au début une distribution gaussienne. Une application de la fonction *skew* permet de conclure que les données de feuilles présentent une skew à gausse surtout avec les variables *margins*.

2.1.4 Distribution des classes

Une informations nécessaire dans les méthodes de classifications est la connaissance du nombre de classes dans les données d'entrainement. Cependant, lorsque nous voulons utiliser plusieurs méthodes de classifications nous chercherons en général à savoir si les classes sont biaisées ou distribués équitablement dans chaque classe. En effet, certains teste de performance d'un modèle de classification utilise l'information sur la distribution des points dans les classes. Un exemple généralement utilisé est les courbes de la fonction d ?efficacité du récepteur (ROC) qui fonctionnent mieux dans le cas de données biaisées dans les classes. L'utilisation de la fonction *groupby*() nous

spèces	nombre de points
Acer_Capillipes	10
Acer_Circinatum	10
Acer_Mono	10
Acer_Opalus	10
Acer_Palmatum	10
...	...
Tilia_Tomentosa	10
Ulmus_Bergmanniana	10
Viburnum_Tinus	10
Viburnum_x_Rhytidophylloides	10
Zelkova_Serrata	10

TABLE 3 – *Distribution des points dans les différentes classes.*

permet de constater rapidement que dans les données feuilles, les objets sont distribués d'une manière équitable dans les 99 classes. Dans le Tableau 2.1.4, on montre que dans chaque classe nous avons dix points. Ainsi, l'utilisation d'un moyen autre que les courbes de ROC, dans le but de tester la généralisation des classificateurs serait clairement un atout. Nous utiliserons les courbe en bars pour montrer la performance de nos modèles en fonctions des changement apportés aux données et de la validation croisée.

2.1.5 Étude de la correlation entre les caractéristiques

Une autre caractéristique des variables à regarder avant l'application des méthodes de classification est de savoir s'il existe une corrélation entre les caractéristiques des données. Très souvent, dans les méthode de classification basées sur les probabilités, on fait l'hypothèse qu'il n'y a pas de corrélation

entre les variables. Il est ainsi important de vérifier combien cette hypothèse est vraie dans les données feuilles. Dans le Tableau 2.1.5, on montre le degré de corrélation entre les variables des données feuilles. On peut constater une *forte* correlation entre les mêmes caractéristiques et une baisse de la correlation pour des variables différentes. Ainsi, l'hypothèse que les variables doivent être indépendantes n'est pas totalement fausse si on s'intéresse seulement à la forte corrélation. En effet, la corrélation est négative entre les trois groupes de caractéristiques. Ainsi, si on admet qu'on a seulement les variables *margins*, les *shapes* et les *texture* l'hypothèse que les variables sont indépendantes devient même vraie. Nous avons vu qu'une exploration statistique des

	margin10	shape10	texture10	margin20	shape20	texture20
margin10	1.000	-0.009	0.101	0.620	0.026	-0.124
shape10	-0.009	1.000	-0.022	0.004	0.809	0.059
texture10	0.101	-0.022	1.000	0.210	-0.004	-0.253
margin20	0.620	0.004	0.210	1.000	0.053	-0.155
shape20	0.026	0.809	-0.004	0.053	1.000	0.014
texture20	-0.124	0.059	-0.253	-0.155	0.014	1.000

TABLE 4 – *Corrélation entre les caractéristiques. On observe une forte corrélation entre les mêmes variables seulement.*

données permet d'apprendre plusieurs propriétés importantes des données brutes. Cependant cette exploration statistique ne peut sans doute remplacer une description graphique qui nous permettra de visionner les données dans l'espace. Dans ce qui suit, on passera à la description graphique telle que l'histogramme etc..

3 Description graphique

La description statistique est clairement importante pour une compréhension préliminaire des données. Cependant, une description graphique permet une vision plus claire surtout pour ce qui est de la distribution des différentes caractéristiques.

3.0.1 Histogramme

Comme la plupart des méthodes de classification supposent une distribution gaussienne des données d'entrainement, il est important d'avoir une

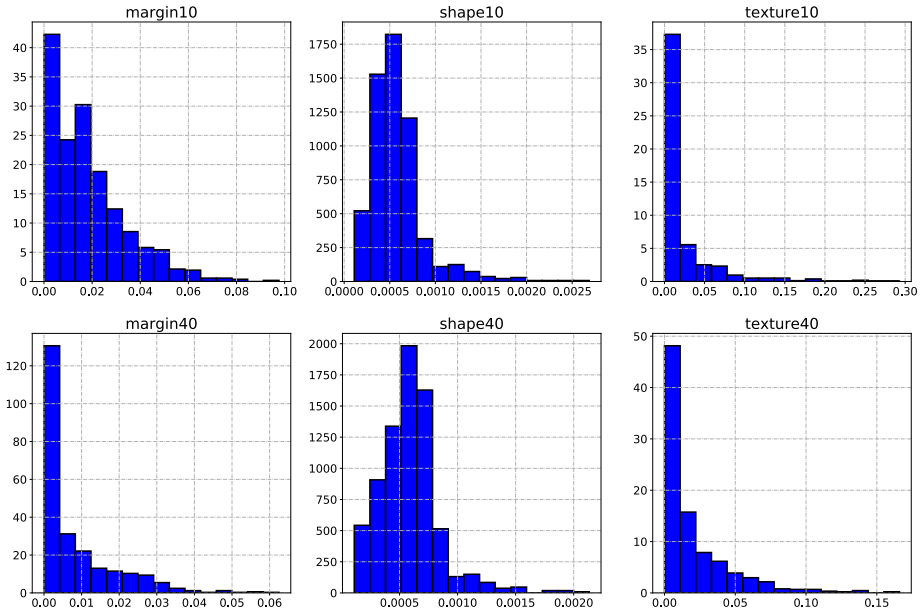

FIGURE 1 – *Histogramme de quelques caractéristiques bien sélectionnées sur une vision globale de la distribution des variables. Les variables shapes ont presque une distribution gaussienne*

idée plus claire de la distribution de nos données. On utilise la représentation en histogramme en considérant qu'il y a trois grandes caractéristiques que sont *margin*, *shape* et *texture*. Aussi nous allons toujours sélectionner aléatoirement deux sous-caractéristiques dans chacune des groupes de caractéristique *margin*, *shape* et *texture*. La Fig : 1 montre que la distribution des caractéristiques *shape* a la forme gaussienne mais il est cependant difficile de conclure la même chose pour les autres variables.

4 Projection des données

Dans cette section, nous allons projeter les données dans un espace en deux dimension. Pour la réduction de dimension, nous avons utiliser même la méthode de classification LDA qui permet de projeter des données brutes dans une dimension plus petite pour pouvoir faire une figure montrant la

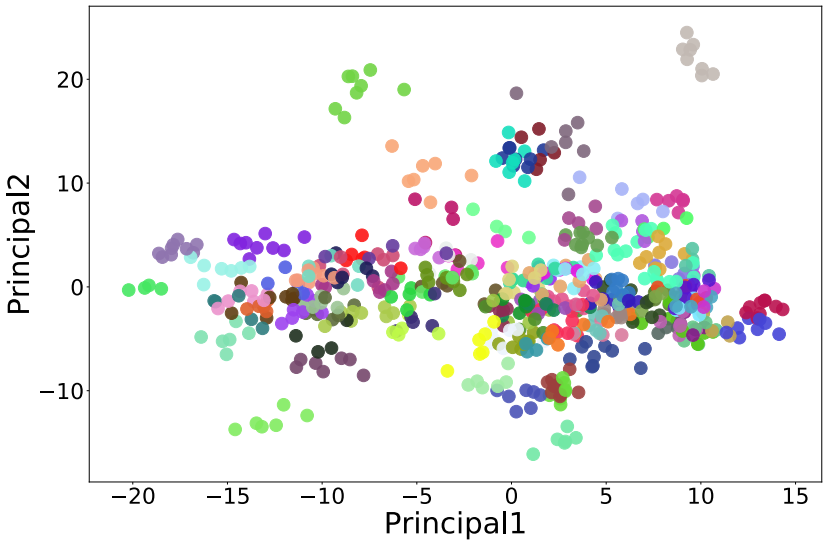

FIGURE 2 – *Dispersion des points dans les différentes classes.*

dispersion des données dans l'espace en deux dimension. La Fig : 2 montre clairement comment les points sont dispersées dans chacune des classes.

5 Présentation des résultats

Dans le but de trouver les hyper-paramètres des différentes modèles utilisés dans la classification, nous avons commencer par la validation croisée. Ainsi, chaque fonction de classificateur fait appelle à une fonction qui permet de fixer les valeurs de ses hyper-paramètres. Nous avons aussi utilisé la validation croisée mais cette fois-ci dans le but d'améliorer la justesse des modèles. Dans le cas de la recherche des hyper-paramètres, les meilleures valeurs sont celles où l'erreur sur les données de validation est la plus petite possible. Pour le deuxième cas, nous présenterons les résultats avec et sans la validation croisée dans le but de voir si la performance des modèles a bien augmenté. Nous avons vu d'après notre exploration des données qu'une transformation des données serait un atout pour l'application des modèles de classification. Nous regarderons, en plus de la validation croisé l'effet de la mise en échelle des données sur la performance des modèles.

5.1 Résultats sans la validation croisé

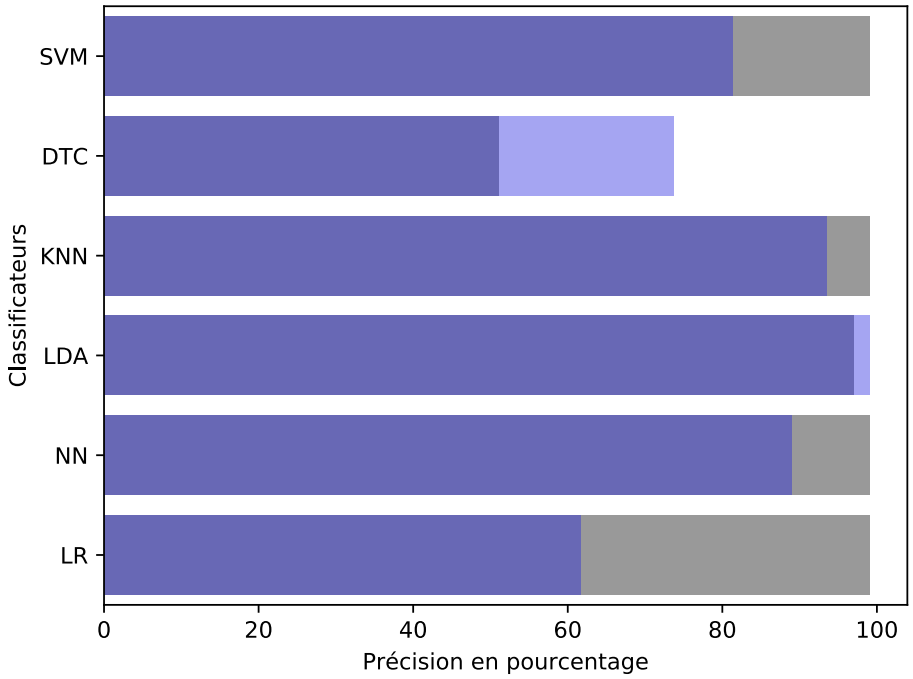

FIGURE 3 – *justesse des modèles de classification en pourcentage sans la validation croisée avec et sans mise en échelle respectivement en couleur noire et bleue.*

Nous présenterons ici les résultats obtenus sur la performance des données de validation qui sont une parties des données d'entrainement. La Fig : 3 montre la justesse obtenue pour chacune des modèles de classification sans utilisation de la validation croisée. Les bars en bleues et noires correspondent respectivement au cas où les données ont été mise ou non en échelle. Cette transformation force les caractéristiques de prendre des valeurs entre zéro et un (0 et 1). Dans ces deux cas, on n'a pas fait une validation croisée dans le but d'augmenter la justesse des modèles. Noter cependant qu'une validation croisé a été faite pour trouver les meilleurs hyper-paramètres de chacun des modèles. On constate que la mise en échelle a diminué la justesse des classificateurs DTC et LDA. Cependant, la transformation a été favorable pour tous les autres classificateurs vue une augmentation important de leur justesse. On prendra le modèle le plus performant celui dont la justesse est

plus proche de la moyenne des justesses des six modèles. Le meilleur modèle qui performe mieux, dans ces conditions, sur les données de validation est le réseau de neurones NN.

5.2 Résultats avec la validation croisé

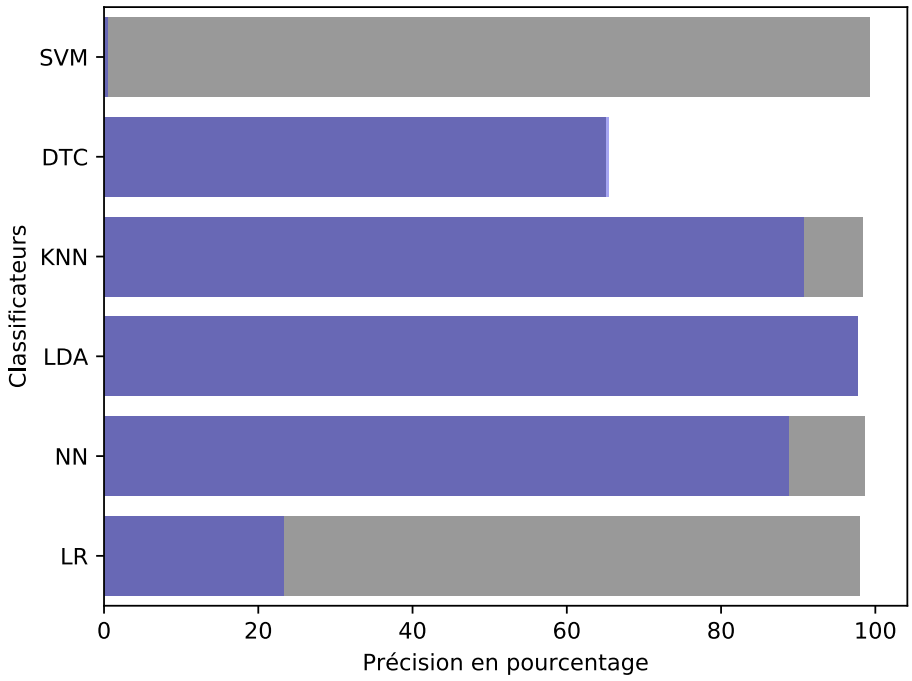

FIGURE 4 – *justesse des modèles de classification en pourcentage avec validation croisée avec et sans mise en échelle respectivement en noire et bleue*

Dans la sous-section précédente, nous avons vu que la mise en échelle des données permet d'augmenter la performance pour la plupart des modèles de classification. Dans cette sous-section nous verrons aussi l'effet de cette mise en échelle mais en plus de la validation croisée. Le nombre de sous ensemble dans cette validation croisée est fixé à dix pour tous les modèles. Le résultat est représenté dans la Fig. 4 où les bars en bleues correspondent à la justesse des modèles sans la mise en échelle et en noirs avec la mise en échelle des données suivie d'une validation croisée dans les deux cas. On remarque rapidement que la performance du modèle SVM est très mauvaise même si

on fait une validation croisée mais sans la transformation des données. Cependant lorsqu'il ya une mise en échelle on remarque une augmentation de la performance des modèles en général sauf le LDA qui n'est pas affecter par cette transformation. Ici encore, d'après notre définition du modèle le plus juste, c'est-à-dire celle dont la justesse est plus proche de la moyenne des justesse est toujours le modèle NN. Comment peut-on expliquer que la justesse du SVM reste trop faible en faisant une validation croisée sans mise en échelle ? En effet, l'exploration des données montre que certains vecteurs ont des cordonnées nulles pour la plupart des caractéristiques. Ainsi, ces vecteurs restent trop biaisés pour certaines dimensions. De ce fait, si on effectue pas une transformation sur les données il peut arriver que le sous ensemble choisit comme validation tombe exactement sur ces vecteurs trop biaisés ce qui va affecter sans doute la justesse du modèle. Une autre explication qu'on peut donner est que l'exploration montre qu'il y a en gros trois différentes caractéristiques (*margin*, *shape*, *texture*) comme le montre la corrélation entre ces derniers. Une validation croisée sans la mise en échelle entre les sous-caractéristiques affecterait sans doute la résultat du SVM qui se base seulement sur les vecteurs plus proche de la ligne de décision.

5.3 Combinaison de modèles

En fin, nous avons aussi tester les méthodes combinant plusieurs modèles tels que le Gradient Boosting Classifier, Random Forest Classifier et AdaBoost Classifier. Cependant nous n'avons pas obtenu des résultats meilleurs. Ainsi, nous avons décider de ne pas montrer ces résultats ici bien que le python pour ces modèles d'ensemble existe dans le code.

6 Conclusion

Dans ce projet, nous avons appliqué trois modèles de classification sur les données de feuilles tirées sur le cite Kaggle. Nous avons constaté que la performance de certains modèles a augmente après une transformation des données (mise en échelle) suivie d'une validation croisée. Le modèle de réseaux de neurones, dont ls justesse est plus proche de la moyenne des justesses des six modèle est considérer comme la plus performant. La meilleure performance est obtenue avec une mise en échelle et une validation croisée aussi bien pour retrouver ses hyper-paramètres mais aussi pour augmenter sa performance. Nous avons ainsi soumis les résultats obtenus avec ce modèle sur le site de Kaggle pour obtenir le score, c'est-à-dire une perte de 0.06124. Cette score semble être raisonnable compte tenu de la justesse du modèle.