Projet 8 : Participez à une compétition Kaggle !

Claire Gayral

Août 2021

Contents

Introduction

Kaggle est une platerforme qui propose des compétitions de datascience. Parmis ces compétitions, la compétition "Prédiction du type de couverture forestière" propose un ensemble de données cartographiques, avec pour objectif de classifier les différents sols des forêts observées. Les variables sont toutes définies dans la page de présentation du projet kaggle. Les données sont téléchargeables depuis cette même page.

Pour répondre à la problématique, plusieurs modélisations issues d'une exploration des données vont être proposées, et plusieurs modèles de classification seront testés.

1 Analyse exploratoire

La première étape, quelque soit le modèle de classification, est de nettoyer les données, et d'en extraire des variables pertinentes.

1.1 Description des données

Chaque ligne des tables représente une cellule donnée de 30 x 30 mètres de Les données d'entraînement sont composées de 15120 observations, avec le type de couverture forestière associée, et les données de sortie de 565892 cellules.

Il s'agit donc de construire un modèle supervisé à partir :

- des 2 variables catégorielles d'entrée, qui ont respectivement 4 et 40 modes, avec un unique mode par observation.
- des 10 variables numériques
- de la classification des sols donnée sur un certain ensemble d'entraînement. Il y a 7 types de sols à déterminer, et la sortie n'est pas multi-label.

Les données d'entraînement seront divisées en train et test à raison de 20 pour comparer les différentes modélisations. Néanmoins, une fois la meilleure modélisation et le meilleur modèle de classification choisis, les pré-traitements seront relancés sur l'intégralité du jeu d'entraînement pour prédire la classification finale.

La variable de sortie est une variable catégorielle encodée de façon ordinale. Elle est distribuée uniformément sur l'ensemble d'entraînement.

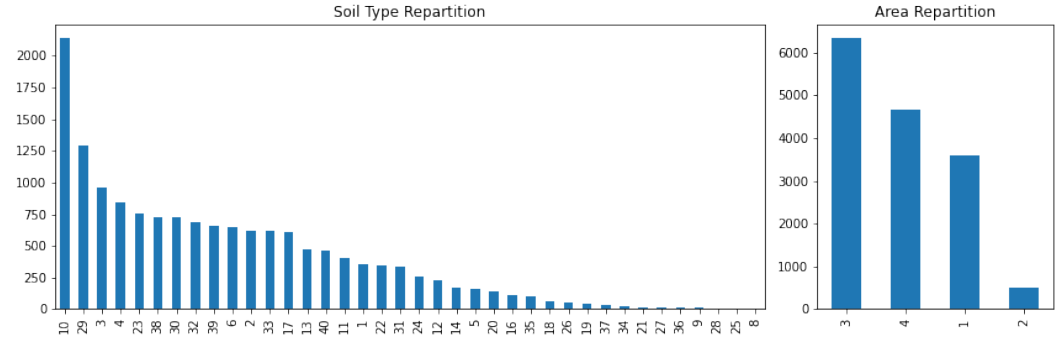
L'analyse exploratoire a été divisée en deux pour séparer le traitement des variables catégorielles et numériques.

1.2 Variables catégorielles

Les deux variables catégorielles sont encodée en "one-hot", et les classes sont bien séparées. La première variable correspond au type de sol, la deuxième à la zone géographique. Il n'est pas très clair que la première variable soit bien une variable "cartographique" et non "télé-détectée", mais comme cela n'est pas explicité, nous allons la garder.

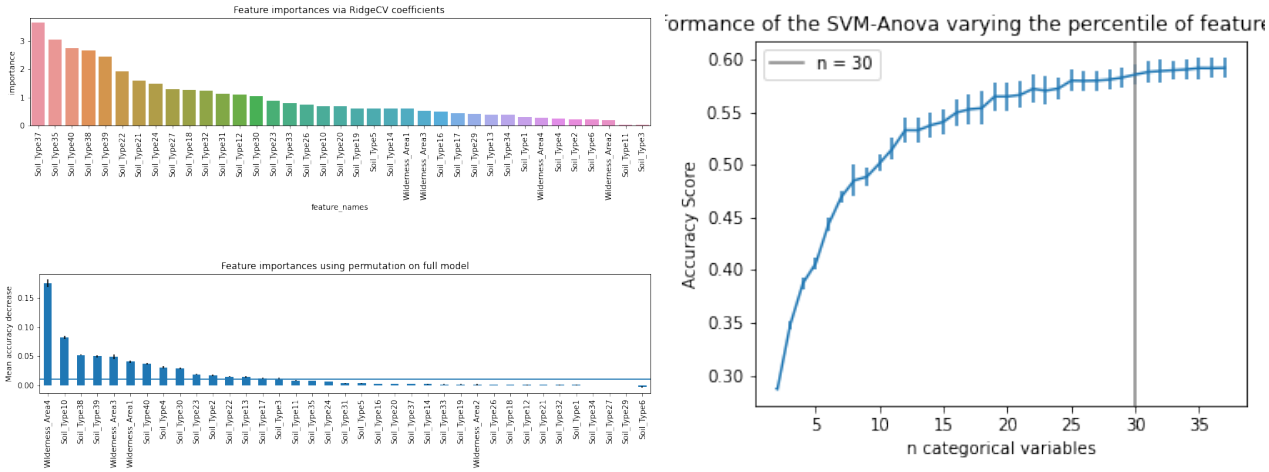
Analyse des variables

Les variables catégorielles ne sont pas réparties uniformément :

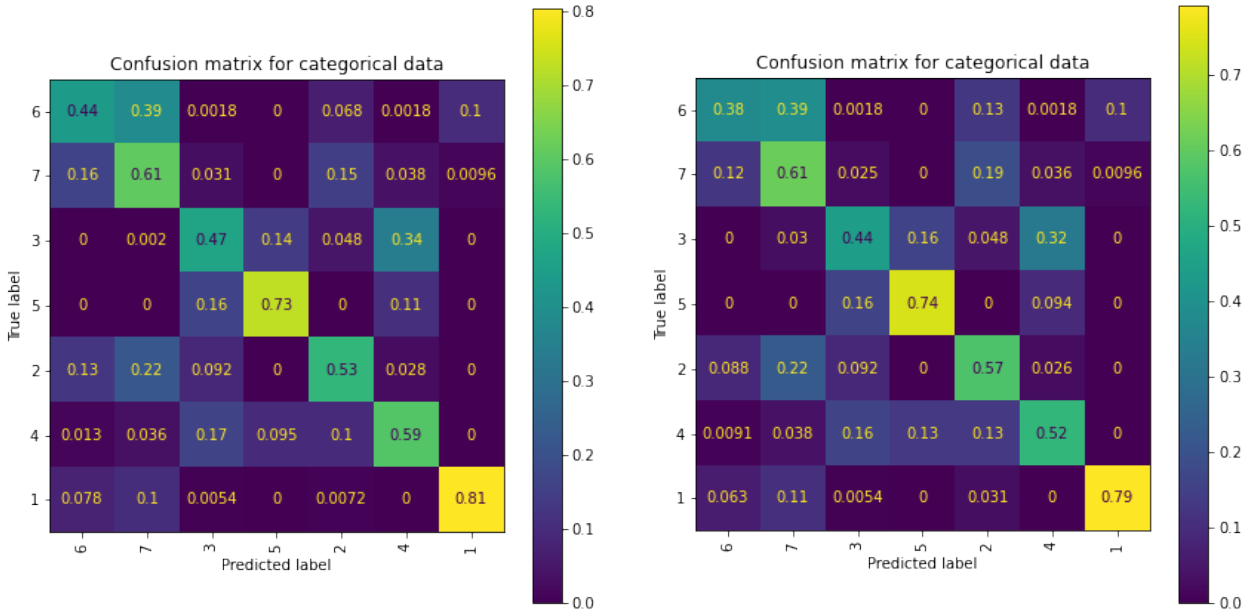


La variable catégorielle correspondant aux types de sols a beaucoup de catégories. Deux modes de type de sols ne sont pas représentés, il faudra retirer les colonnes associées.

L'importance de chaque mode pour la prédiction de la couverture forestière peut alors permettre de choisir les variables les plus pertinentes. Plusieurs méthodes comme le ridgeCV ou les randomForestClassifiers permettent de classer ainsi les modes par importance :

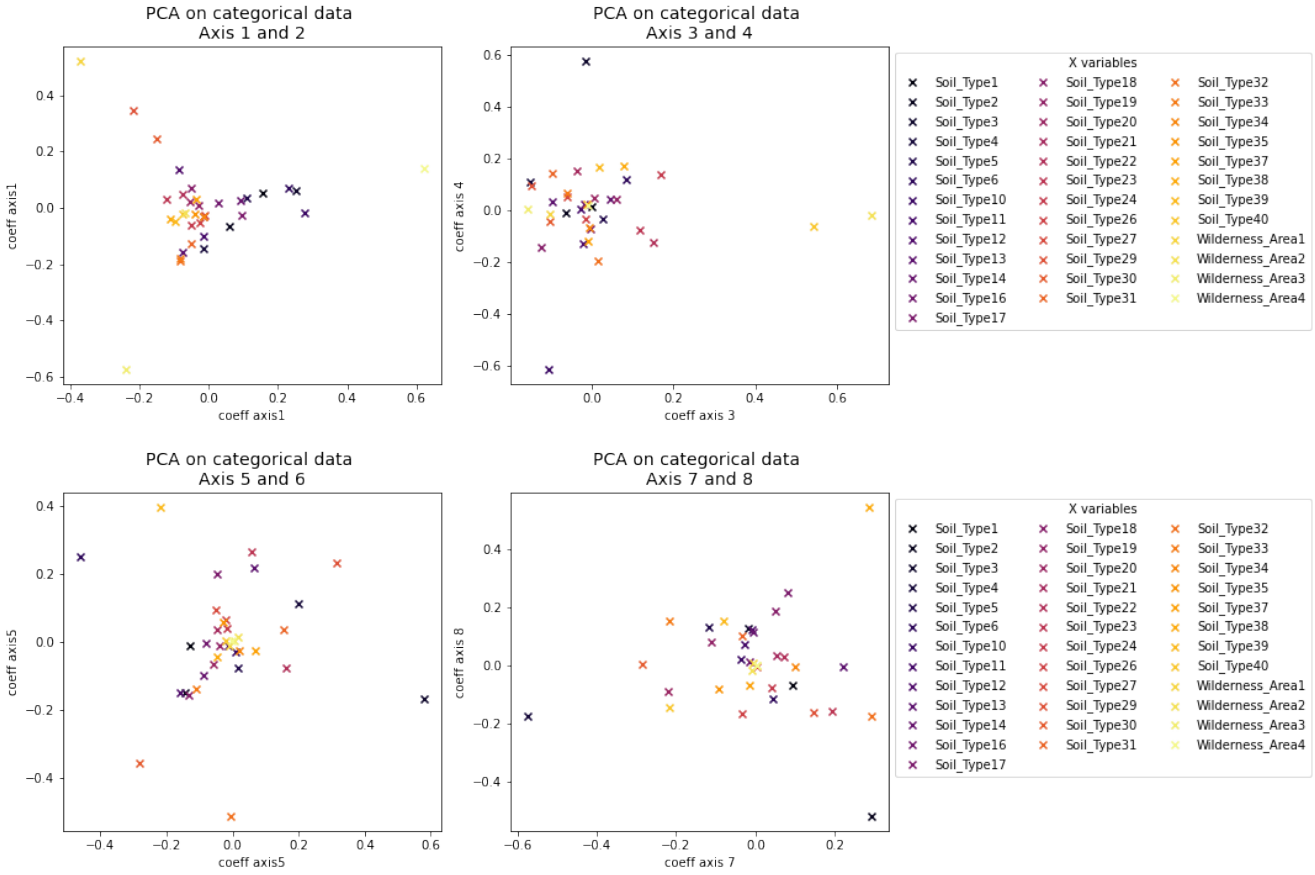


Selon les classifieurs, l'importance n'est pas la même, ce qui limite l'automatisation de ces procédés. De plus, en comparant la matrice de confusion avant et après le retrait des variables les moins importantes, on voit que l'on a juste rendu plus feignant le modèle. En effet, les modes les moins importants coïncident avec les modes les moins représentés, les retirer n'est pas forcément pertinent.



Une autre façon de réduire la dimension de ces variables catégorielles est de les projeter dans un espace de dimension inférieure. Ainsi, la PCA maximise l'accuracy pour 20 composantes, et les 8 premiers axes sont donnés

par la pondération suivante :

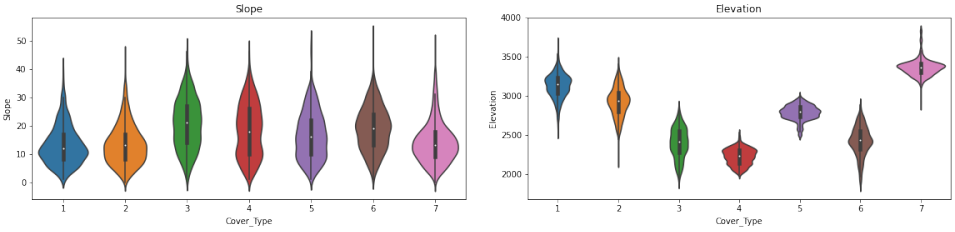


Les données catégorielles ainsi projetées dans des axes orthonormés de dimension 20 semblent donner des prédictions aussi bonnes qu'en dimension.

1.3 Variables numériques

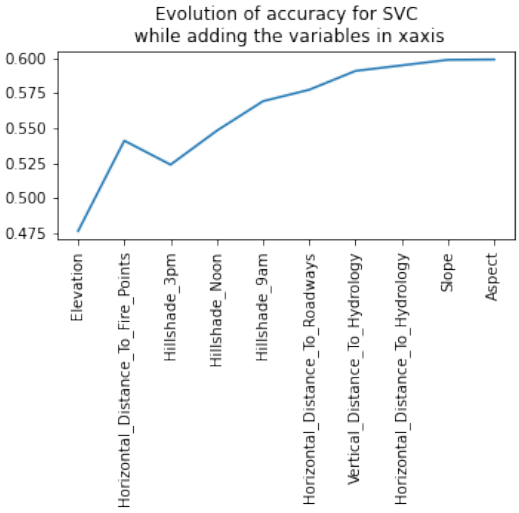
Ensuite, les 10 variables numériques ne séparent pas toutes aussi bien les classes de forêt. En effet, les valeurs de η^2 sont très différentes, il s'agit un critère statistique qui mesure à quel point une variable catégorielle est séparée par une variable numérique.

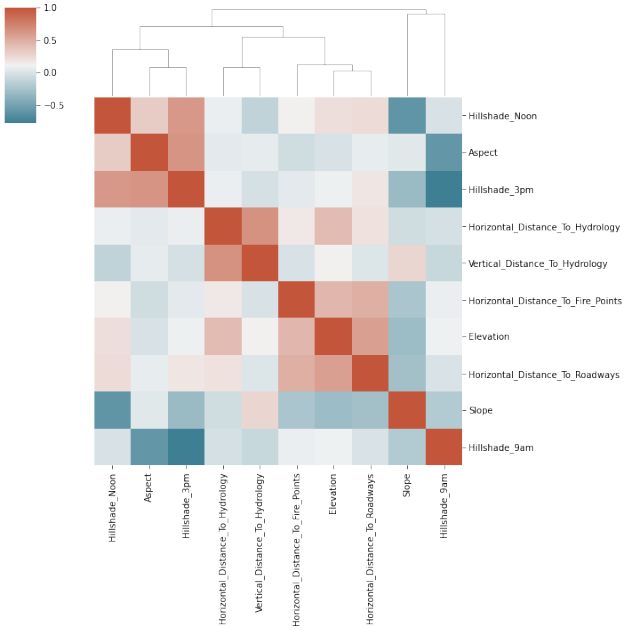
Hillshade_Noon	0.044745
Aspect	0.018814
Hillshade_3pm	0.072812
Horizontal_Distance_To_Hydrology	0.129856
Vertical_Distance_To_Hydrology	0.028816
Horizontal_Distance_To_Fire_Points	0.228525
Elevation	0.865734
Horizontal_Distance_To_Roadways	0.326858
Slope	0.107013
Hillshade_9am	0.130554



Ainsi, les classes à prédires sont plutôt bien séparées par la variable numérique "Elevation" et très mal par la variable "Aspect".

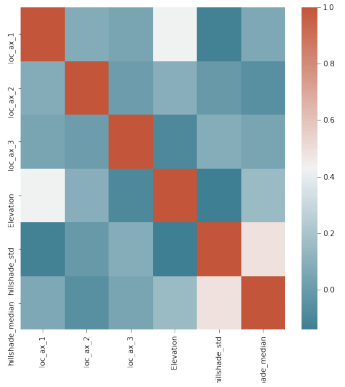
Comme pour les variables catégorielles, une sélection de variable est faite. On retrouve bien l'ordre d'importance donnée par les η^2, où la variable "Elevation" est prépondérante, quand "Aspect" est très peu (voir pas du tout) significative. En ajoutant la variable "Hillshade 3pm", l'accuracy chute. Cela veut peut-être dire que cette variable est corrélée aux première, et bruite plus les dnnées qu'elle n'apporte d'informations.





Les variables numériques sont corrélées les unes aux autres. Par exemple, la variable "Vertical_Distance_To_Hydrology" est fortement liée à la variable "Horizontal_Distance_To_Hydrology", et la variable "Hillshade_3pm" est inversement corrélée à "Hillshade_9am". La plupart des modèles de classification font l'hypothèse d'indépendance entre les variables : il vaudra mieux projeter les données dans un espace où les variables ne sont pas corrélées.

Avant de retirer des variables numériques, il est important d'extraire des variables plus pertinentes à partir de ces variables numériques. Par exemple, ce n'est pas tant l'ombre aux différentes heures de la journée qui compte, mais plutôt la variation d'ensoleillement. C'est ainsi que les 3 variables concernant l'ombre sont résumées en deux variables la variation d'ombre et l'ombre moyenne.



Des variables concernant l'hydrologie, seule la distance horizontale sera conservée, car la distance totale sépare moins bien les classes de forêt, et la distance verticale est corrélée à la distance horizontale. Enfin, les 3 variables de localisation 5loin des routes, des points de feux, des points d'eau) sont corrélées. Pour limiter le biais de cela, les variables sont projetées dans un espace orthonormé. Avec ces prétraitements, la matrice de corrélation indique que cette représentation des donnée

1.4 Les jeux d'entrainement

A partir des différents pré traitements explicités ci-dessus, différents jeux d'entraînement sont extraits :

- dataset 0 : le jeu de données initial, seules les variables constantes ont été retirées
- dataset 1 : les variables catégorielles et numériques non importantes sont retirées,

- dataset 2 : les variables catégorielles sont sélectionnées, et les variables numériques sont prétraitées
- dataset 3 : les variables catégorielles sélectionnées sont projetées dans un espace orthonormé adapté (de dimension 20), et les variables numériques sont prétraitées

Les datasets 4, 5, 6 et 7 correspondent aux datasets précédents orthonormalisés. L'orthonormalisation permet de supprimer les corrélations entre les variables, ce qui donne de meilleurs résultats, néanmoins, la projection rend compliqué l'interprétation du rôle dans la classification de chaque variable initiale.

2 Classification

Dans un premier temps, différents modèles classiques de classification sont lancés avec les paramètres par défaut, sur les modèles définis ci-dessus On obtient ainsi les tableaux de score (accuracy) et de temps d'entraînement, avec les différents datasets en lignes.

Score Accuracy

	Nearest Neighbors	Linear SVM	RBF SVM	Decision Tree	Random Forest	Neural Net	AdaBoost	Naive Bayes
0	0.799735	0.724074	0.741005	0.733069	0.674603	0.774868	0.749471	0.480688
1	0.807143	0.710053	0.751323	0.730952	0.690741	0.753968	0.737831	0.464021
2	0.552116	0.608201	0.648677	0.571958	0.555026	0.646296	0.503704	0.454233
3	0.557672	0.633598	0.665079	0.663228	0.612169	0.673810	0.539418	0.593386
4	0.826190	0.722751	0.715608	0.716402	0.698942	0.713757	0.763757	0.593386
5	0.818254	0.710317	0.713757	0.725132	0.718519	0.692328	0.768783	0.630423
6	0.605820	0.618783	0.628571	0.628042	0.657407	0.630159	0.625926	0.638360
7	0.655026	0.643122	0.631746	0.596032	0.639418	0.618783	0.599471	0.631481

Temps d'entrainement (en seconde)

	Nearest Neighbors	Linear SVM	RBF SVM	Decision Tree	Random Forest	Neural Net	AdaBoost	Naive Bayes
0	14.784027	0.191472	8.091215	0.138105	0.230682	4.571458	0.745726	0.145967
1	8.262091	0.149591	4.523373	0.129786	0.224175	2.888409	0.650367	0.132854
2	7.925421	0.142454	4.089891	0.128564	0.214091	2.539925	0.608353	0.130020
3	8.094473	0.199251	4.549675	0.133181	0.208550	2.879220	0.656897	0.133263
4	15.136565	0.314168	9.311963	0.215078	0.272194	3.598449	2.085076	0.146518
5	8.427706	0.182357	5.858842	0.159040	0.247225	2.492870	1.348735	0.131550
6	8.065719	0.169573	4.893156	0.157488	0.248050	2.503127	0.920149	0.131746
7	8.255832	0.158323	5.160444	0.157020	0.221932	3.020187	0.988942	0.131386

Ensuite, les modèles les plus pertinents (qui ont le meilleur score et n temps d'entrainement raisonnable) sont testés avec plusieurs paramètres, par validation croisée :

Score sur l'ensemble test
Accuracy

	Nearest Neighbors	SVM_rbf	SVM_linear	Decision Tree	Neural Net	AdaBoost
0	0.799735	0.841799	0.724339	0.789418	0.779365	0.762169
1	0.807143	0.842857	0.711905	0.782011	0.772222	0.749735
2	0.552116	0.432011	0.607143	0.540741	0.578042	0.503968
3	0.557672	0.438624	0.636772	0.589683	0.637037	0.551852
4	0.826190	0.846032	0.722751	0.785714	0.765344	0.790476
5	0.818254	0.843651	0.709259	0.809259	0.743651	0.796296
6	0.633069	0.662434	0.618519	0.573545	0.629101	0.627513
7	0.655820	0.689418	0.643386	0.587037	0.683598	0.579365

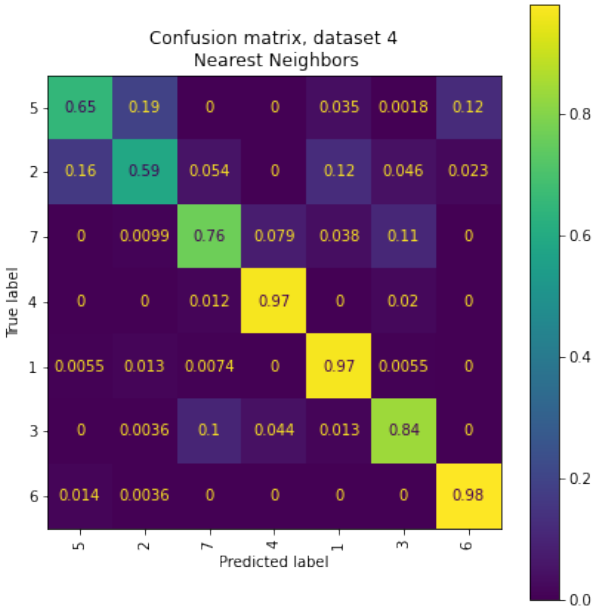
Temps de validation
(dépend du nombre de paramètres testés)

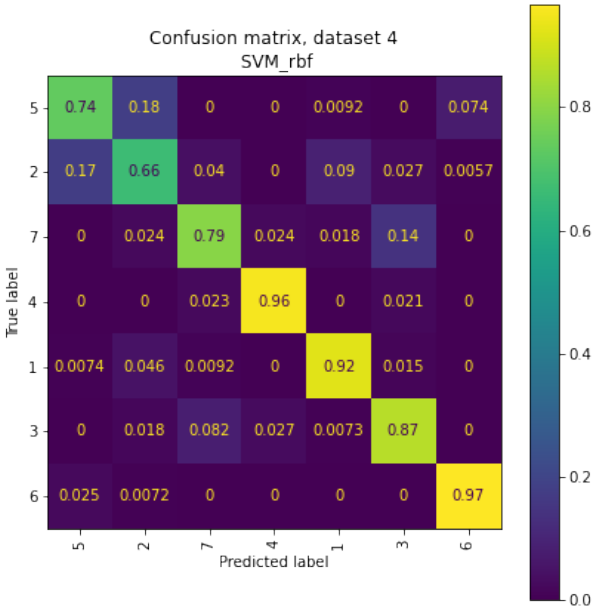
	Nearest Neighbors	SVM_rbf	SVM_linear	Decision Tree	Neural Net	AdaBoost
0	0.799735	0.841799	0.724339	0.789418	0.779365	0.762169
1	0.807143	0.842857	0.711905	0.782011	0.772222	0.749735
2	0.552116	0.432011	0.607143	0.540741	0.578042	0.503968
3	0.557672	0.438624	0.636772	0.589683	0.637037	0.551852
4	0.826190	0.846032	0.722751	0.785714	0.765344	0.790476
5	0.818254	0.843651	0.709259	0.809259	0.743651	0.796296
6	0.633069	0.662434	0.618519	0.573545	0.629101	0.627513
7	0.655820	0.689418	0.643386	0.587037	0.683598	0.579365

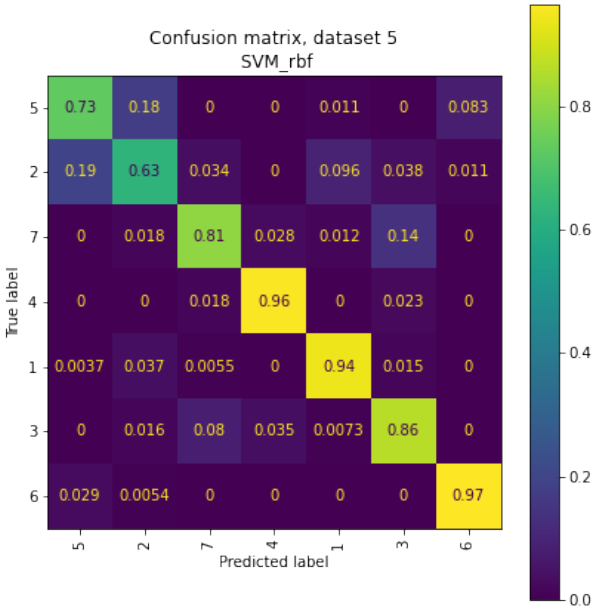
De ces tableaux, sont extraits les 5 modèles les plus performants, qui sont relancés pour analyser les résultats :

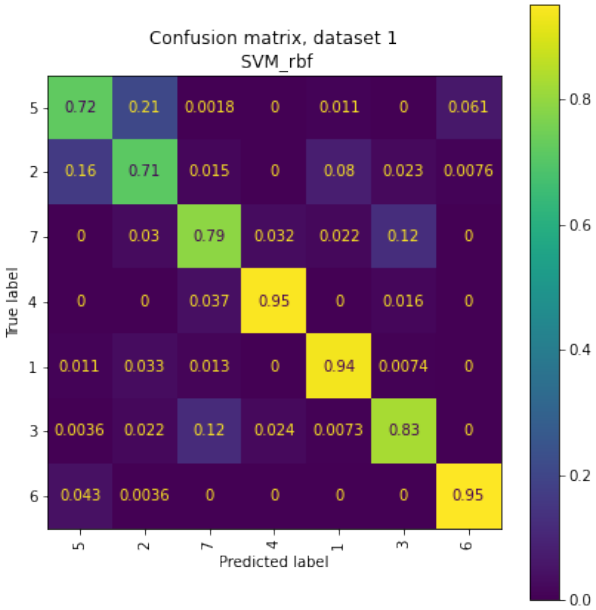
	model_name	dataset	AUC	balanced_accuracy	kappa_score	matthews_corrcoef	hinge_loss	params
0	Nearest Neighbors	4	0.897077	0.823105	0.797108	0.798199	0.347619	3.0
1	SVM_rbf	4	0.908989	0.843602	0.820310	0.820458	0.307937	10000.0
2	SVM_rbf	5	0.907559	0.841151	0.817516	0.817785	0.312698	10000.0
3	SVM_rbf	0	0.906730	0.839780	0.815395	0.815489	0.316402	1000.0
4	SVM_rbf	1	0.907340	0.840820	0.816631	0.816706	0.314286	1000.0

Et voilà les matrices de confusion associées au 4 premiers:


Confusion matrix, dataset 4
Nearest Neighbors
True label
Predicted label
5 2 7 4 1 3 6
5 0.65 0.19 0 0 0.035 0.0018 0.12
2 0.16 0.59 0.054 0 0.12 0.046 0.023
7 0 0.0099 0.76 0.079 0.038 0.11 0
4 0 0 0.012 0.97 0 0.02 0
1 0.0055 0.013 0.0074 0 0.97 0.0055 0
3 0 0.0036 0.1 0.044 0.013 0.84 0
6 0.014 0.0036 0 0 0 0 0.98
0.0 0.2 0.4 0.6 0.8


Confusion matrix, dataset 4
SVM_rbf
True label
Predicted label
5 2 7 4 1 3 6
5 0.74 0.18 0 0 0.0092 0 0.074
2 0.17 0.66 0.04 0 0.09 0.027 0.0057
7 0 0.024 0.79 0.024 0.018 0.14 0
4 0 0 0.023 0.96 0 0.021 0
1 0.0074 0.046 0.0092 0 0.92 0.015 0
3 0 0.018 0.082 0.027 0.0073 0.87 0
6 0.025 0.0072 0 0 0 0 0.97
0.0 0.2 0.4 0.6 0.8


Confusion matrix, dataset 5
SVM_rbf
True label
Predicted label
5 2 7 4 1 3 6
5 0.73 0.18 0 0 0.011 0 0.083
2 0.19 0.63 0.034 0 0.096 0.038 0.011
7 0 0.018 0.81 0.028 0.012 0.14 0
4 0 0 0.018 0.96 0 0.023 0
1 0.0037 0.037 0.0055 0 0.94 0.015 0
3 0 0.016 0.08 0.035 0.0073 0.86 0
6 0.029 0.0054 0 0 0 0 0.97
0.0 0.2 0.4 0.6 0.8


Confusion matrix, dataset 1
SVM_rbf
True label
Predicted label
5 2 7 4 1 3 6
5 0.72 0.21 0.0018 0 0.011 0 0.061
2 0.16 0.71 0.015 0 0.08 0.023 0.0076
7 0 0.03 0.79 0.032 0.022 0.12 0
4 0 0 0.037 0.95 0 0.016 0
1 0.011 0.033 0.013 0 0.94 0.0074 0
3 0.0036 0.022 0.12 0.024 0.0073 0.83 0
6 0.043 0.0036 0 0 0 0 0.95
0.0 0.2 0.4 0.6 0.8

Références

Annexe