

Projet 8 : Participez à une compétition Kaggle !

Claire Gayral

Août 2021

Contents

1	Analyse exploratoire	2
1.1	Description des données	2
1.2	Variables catégorielles	2
1.3	Variables numériques	4
1.4	Les jeux d'entraînement	5
2	Classification	6

Introduction

Kaggle est une plateforme qui propose des compétitions de datascience. Parmi ces compétitions, la [compétition "Prédiction du type de couverture forestière"](#) propose un ensemble de données cartographiques, avec pour objectif de classer 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. 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.

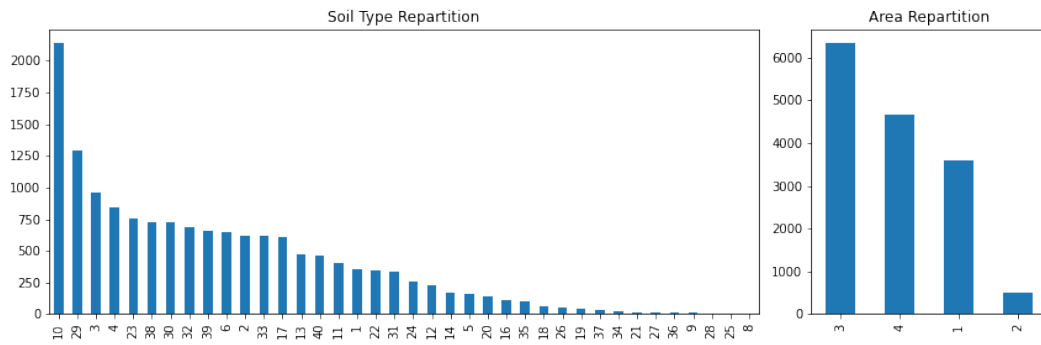
L'analyse exploratoire a été divisée en deux pour séparer le traitement des variables catégorielles et numériques.

1.2 Variables catégorielles

Les deux variables catégorielles sont encodées 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.

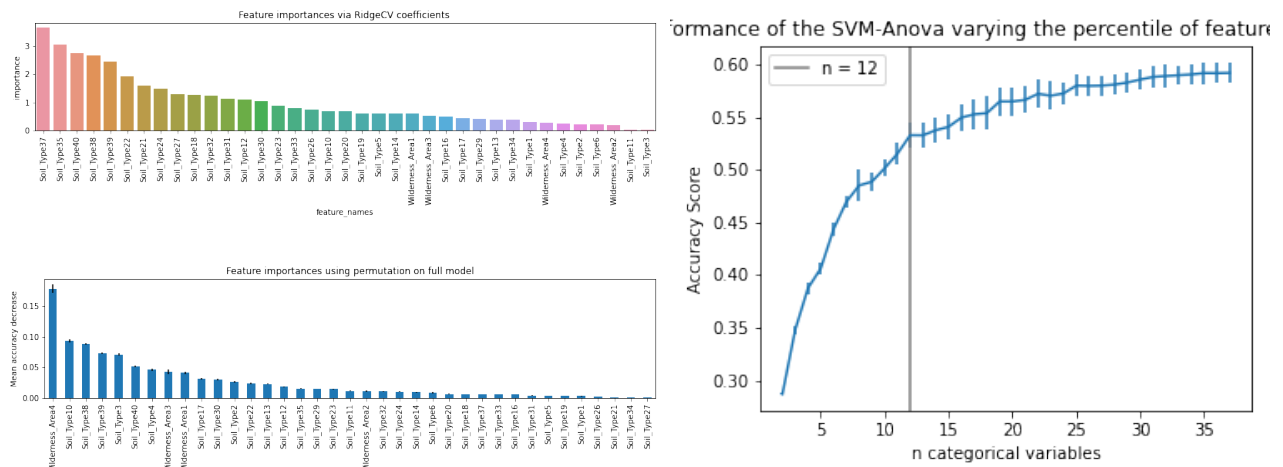
Analyse des variables

Les variables catégorielles ne sont pas réparties uniformément :

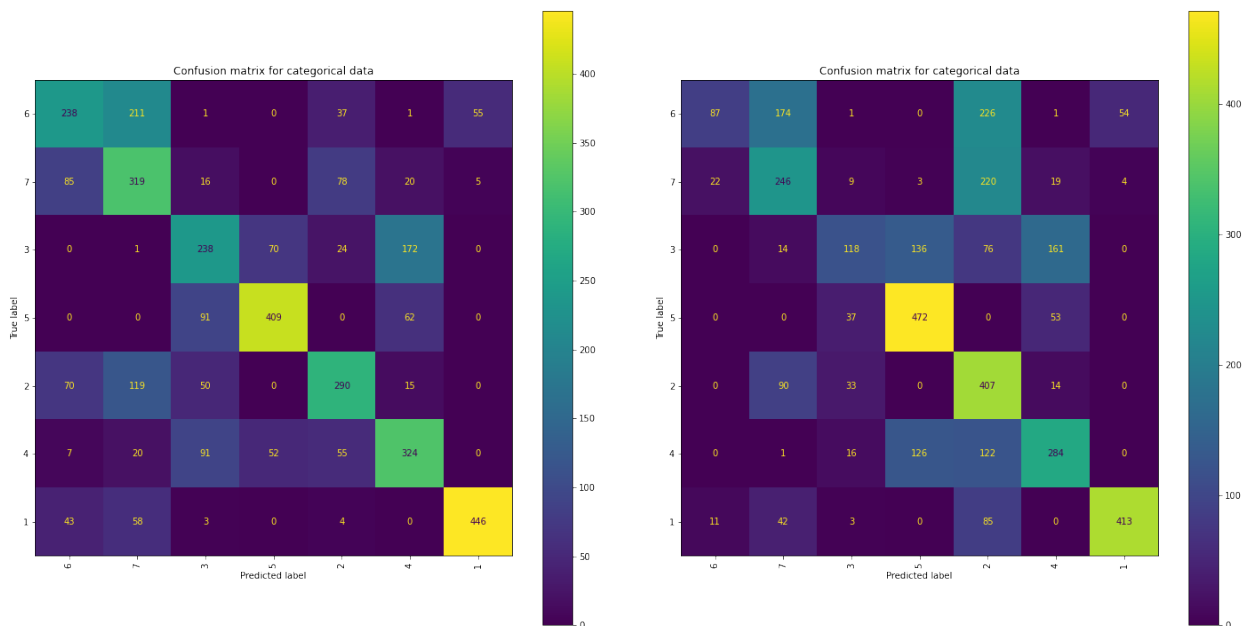


La variable catégorielle correspondant aux types de sols a beaucoup de catégories. Deux modes de type de sols ne sont pas représentés, il faudra retirer les colonnes associées.

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 :

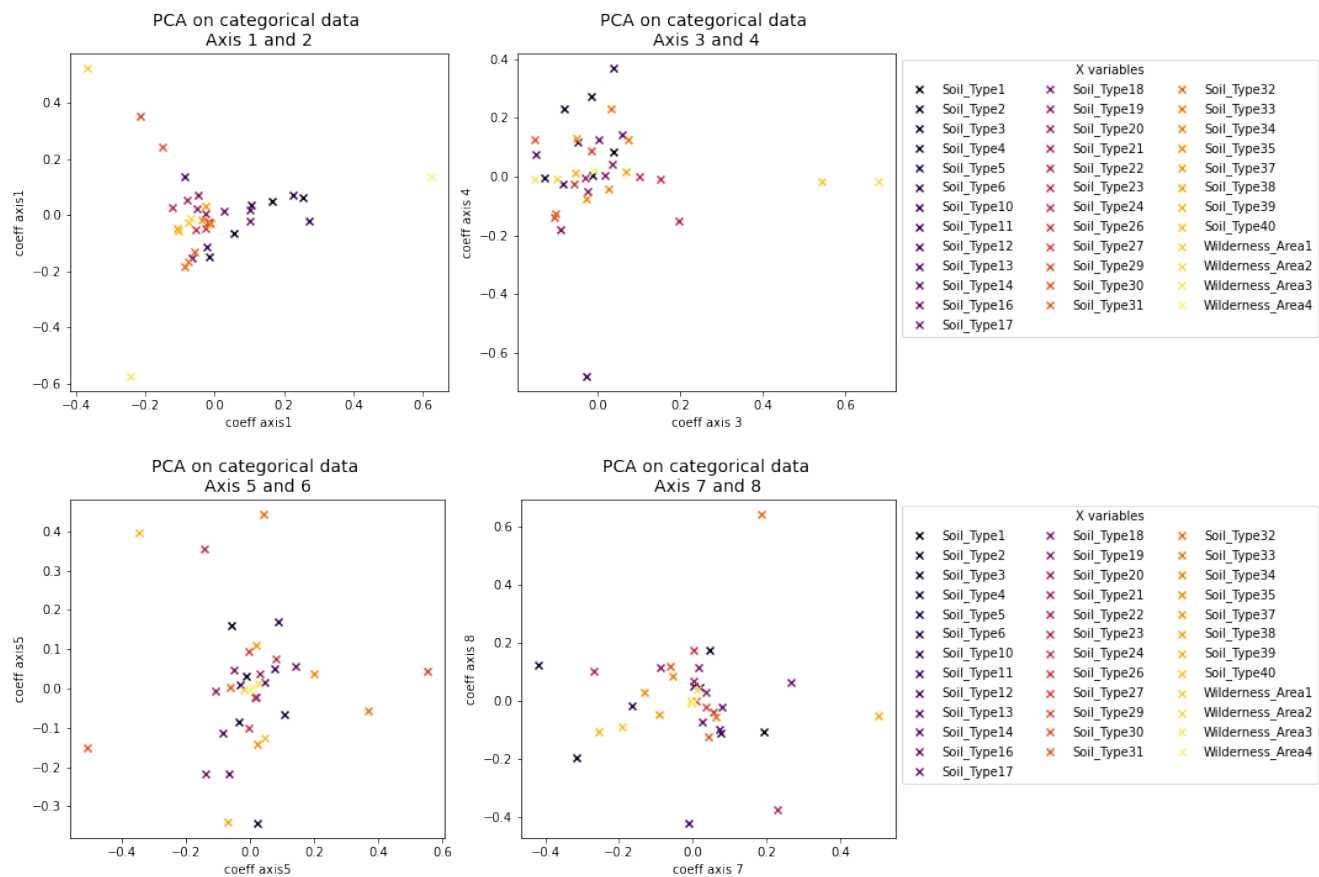


Selon les classifieurs, l'importance n'est pas la même, ce qui limite l'automatisation de ces procédés. De plus, en comparant la matrice de confusion avant et après le retrait des variables les moins importantes, on voit que l'on a juste rendu plus feignant le modèle. En effet, les modes les moins importants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

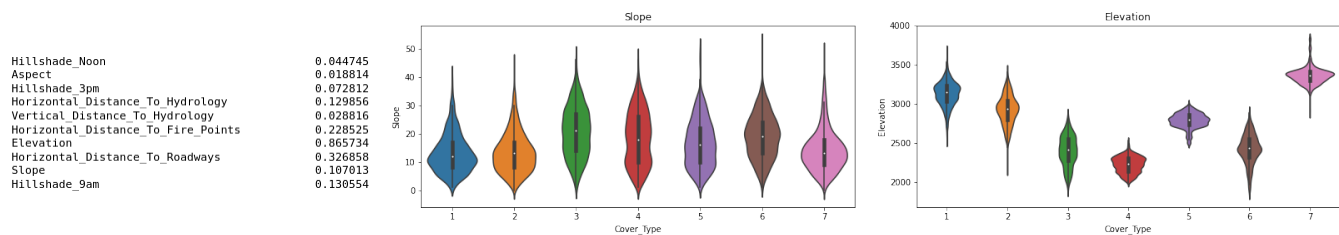
par la pondération suivante :



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

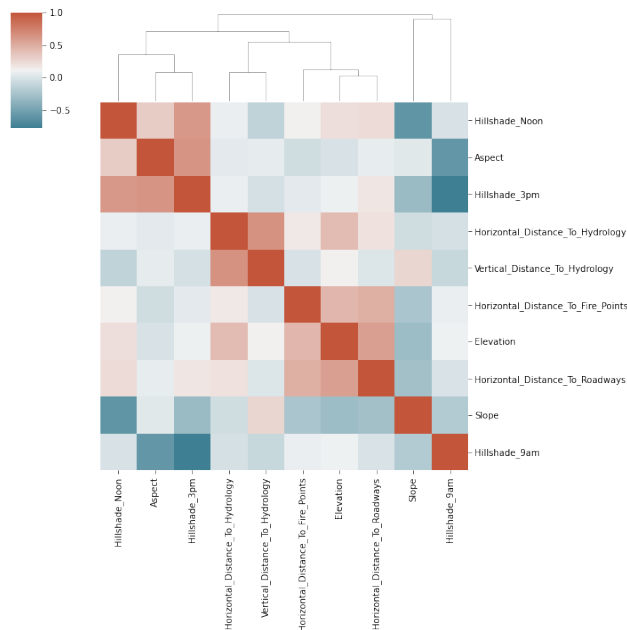
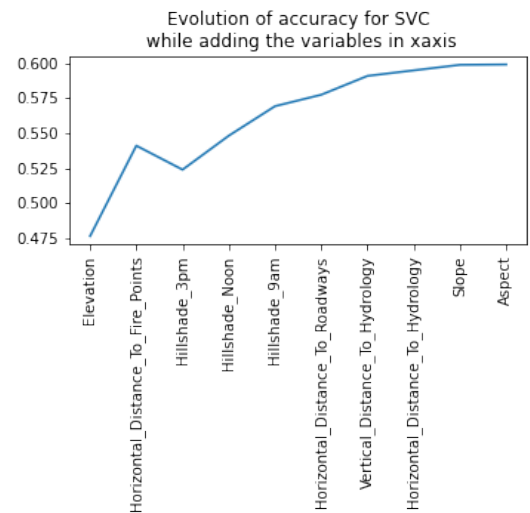
1.3 Variables numériques

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.



Ainsi, les classes à prédire 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 premières, et bruite plus les données 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 (loin 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 données

1.4 Les jeux d'entraînement

A partir des différents prétraitements 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 non importantes sont retirées, et les variables numériques sont prétraitées

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

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

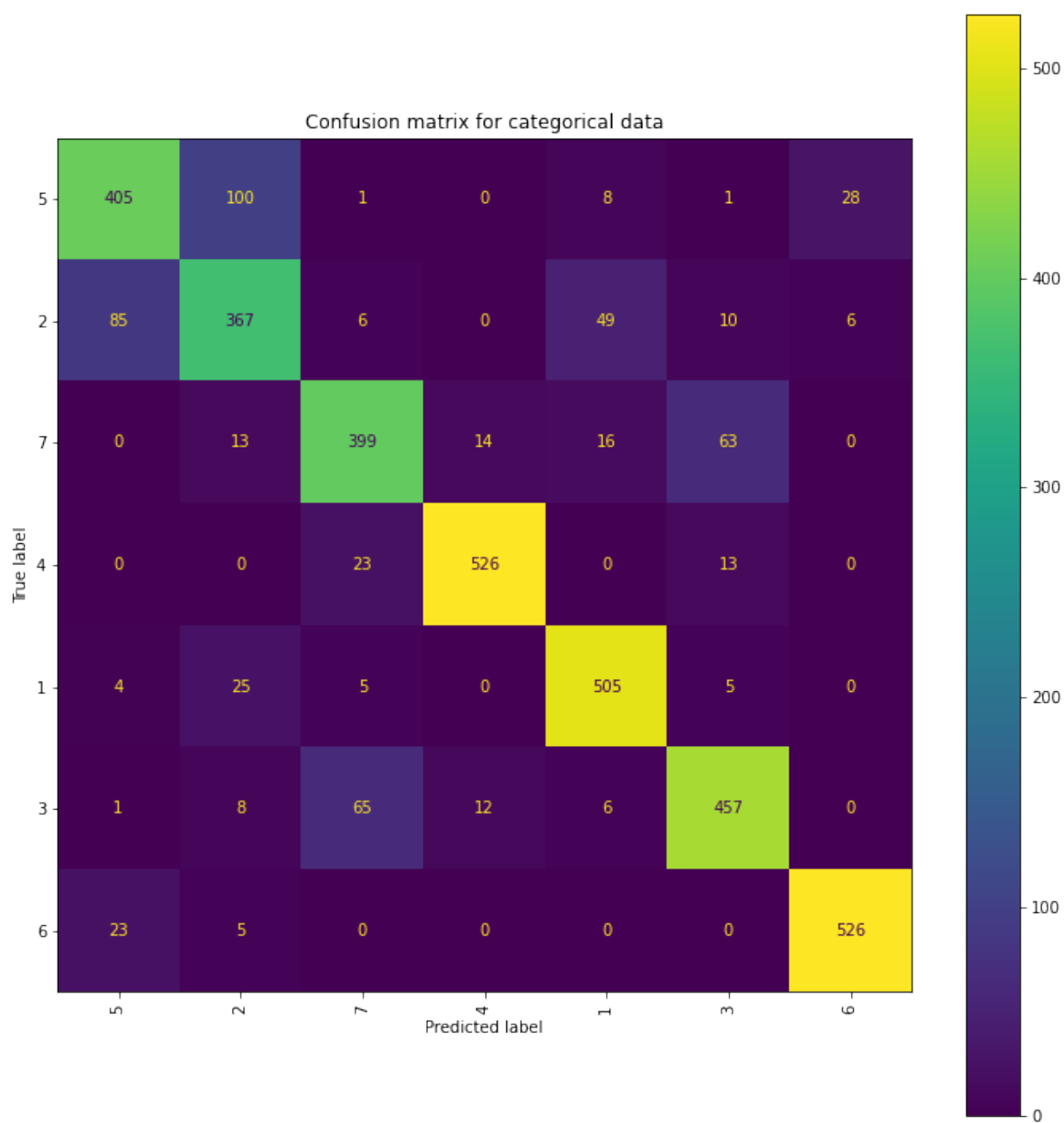
	Nearest Neighbors	Linear SVM	RBF SVM	Decision Tree	Random Forest	Neural Net	AdaBoost	Naive Bayes	QDA
0	0.799735	0.701587	0.773016	0.732804	0.689683	0.775132	0.749206	0.490212	0.441005
1	0.791799	0.687831	0.802910	0.731481	0.698148	0.768783	0.742857	0.466138	0.404762
2	0.794974	0.703439	0.777513	0.740212	0.693915	0.780423	0.771693	0.633069	0.576984
3	0.813492	0.702381	0.775397	0.696561	0.682804	0.785450	0.737566	0.593122	0.409524
4	0.826190	0.719577	0.133598	0.714286	0.700265	0.710317	0.755026	0.596032	0.427778
5	0.791005	0.696561	0.807143	0.703704	0.675397	0.756349	0.739683	0.596032	0.406878
6	0.797354	0.703439	0.754233	0.705026	0.661376	0.774339	0.739947	0.607937	0.577249
7	0.791005	0.697354	0.807143	0.697884	0.677513	0.754233	0.741005	0.593122	0.405820

	Nearest Neighbors	Linear SVM	RBF SVM	Decision Tree	Random Forest	Neural Net	AdaBoost	Naive Bayes	QDA
0	17.019813	6.468931	59.618168	0.462922	0.687727	14.107391	3.533867	0.378033	0.706198
1	14.893226	5.541974	45.061024	0.398874	0.676596	12.019682	3.369611	0.343076	0.791143
2	14.247006	5.540740	48.553928	0.701491	0.758192	13.865766	6.933675	0.358742	0.992963
3	14.885334	5.190781	90.084564	0.931312	0.788404	14.192776	7.847549	0.363959	0.712773
4	17.430017	732.697043	74.762680	1.205033	0.987755	12.481229	12.279679	0.408645	1.078117
5	15.605935	6.514596	45.221306	0.934420	0.890597	12.406370	9.829463	0.332906	0.781618
6	15.199486	5.379707	51.099873	0.781930	0.843552	14.861792	9.541407	0.327263	0.617705
7	15.685208	6.191312	44.658276	0.857266	0.847621	13.177748	9.057373	0.364178	0.619402

	Nearest Neighbors	Decision Tree	Neural Net	AdaBoost	SVM_rbf	SVM_linear	SVM_sigmoid
0	0.799735	0.790212	0.774074	0.760582	0.783862	0.732540	0.325661
1	0.791799	0.780423	0.773280	0.751058	0.806349	0.715873	0.308466
2	0.794974	0.797090	0.763228	0.788624	0.786508	0.726984	0.302116
3	0.814286	0.785714	0.796296	0.776190	0.816931	0.723016	0.283069
4	0.826190	0.789683	0.777778	0.782011	0.133598	0.726720	0.193651
5	0.791005	0.750794	0.787831	0.772222	0.809524	0.714815	0.311640
6	0.797354	0.761905	0.777513	0.762963	0.761905	0.726984	0.281481
7	0.804233	0.774603	0.773280	0.757672	0.787037	0.722751	0.260053

	Nearest Neighbors	Decision Tree	Neural Net	AdaBoost	SVM_rbf	SVM_linear	SVM_sigmoid
0	42.717831	2.703397	260.574387	48.780626	87.938108	444.921498	31.316665
1	36.485941	2.271381	197.072983	44.361669	58.521966	234.759884	25.200056
2	35.936939	7.693201	197.123844	110.268960	57.799640	202.923640	26.891430
3	35.364199	7.707678	186.278281	109.292224	55.984566	357.543533	24.856390
4	41.902307	16.223135	239.357029	214.816663	113.423352	15550.490489	30.000337
5	36.461801	11.719659	183.074093	162.381018	58.193633	74.529723	29.252898
6	36.015026	10.841118	183.170215	150.953557	61.400271	238.143942	25.600023
7	35.006364	9.547890	172.476833	136.462530	56.014080	472.914845	24.570857

	model_name	dataset	AUC	balanced_accuracy	kappa_score	matthews_corrcoef	hinge_loss	params
0	Nearest Neighbors	4.0	0.897077	0.823105	0.797108	0.798199	0.347619	3.0
1	SVM_rbf	5.0	0.909441	0.844411	0.820943	0.821033	0.306878	100.0
2	SVM_rbf	3.0	0.905777	0.838094	0.813839	0.813983	0.319048	1000.0
3	SVM_rbf	0.0	0.907178	0.840544	0.816320	0.816401	0.314815	1000.0
4	SVM_rbf	7.0	0.905558	0.837703	0.813524	0.813678	0.319577	1000.0
5	SVM_rbf	1.0	0.906514	0.839393	0.815080	0.815176	0.316931	1000.0



Références

Annexe