POLYTECHNIQUE MONTRÉAL

LE GÉNIE
EN PREMIÈRE CLASSE

INF8215
Groupe 01

TP3

Classifications multiclasses : légumes secs

Par

Brando, Tovar **1932052**

Vega, Estefan **1934346**

Équipe : **BrandiniStifini**

Le 15 avril 2022

Table des matières

1 Contexte

Dans ce travail pratique, il nous était demander de classifier des légumes secs dans leur catégorie respectives. Il y en avait 7 en tout ; Sira, Horoz, Dermason, Barbunya, Cali, Bombay, Seker et nous devions déterminer la catégorie à l'aide de 16 *features*. Nous avions donc à résoudre un problème de classification multiclasses. Nous avons décider d'utiliser la librairie *scikit-learn* et le modèle que nous avons utilisé se base sur les machines à vecteurs de support (*SVC OneVsOneClassifier*). Nous avons aussi exploré d'autres modèle telles que celui basée sur la descente de gradient stochastique (*SGDClassifier*) et celle basée sur les forêts aléatoires (*RandomForestClassifier*)

2 Prétraitement

Avant de commencé à résoudre le problème, il est utile de se familiariser avec les données. Nos données étaient constitué de 16 *features*. Il y avait 6000 données de test. Nous avons commencé par voir s'il manquait des valeurs dans certaines de no données test ce qui n'était pas le cas. Nous avons ensuite regardé si nos donées étaient balancées.

Figure 1 – Diagramme à bandes des catégories

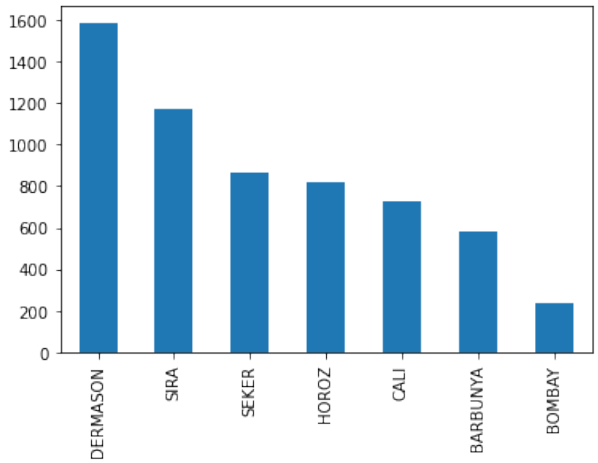
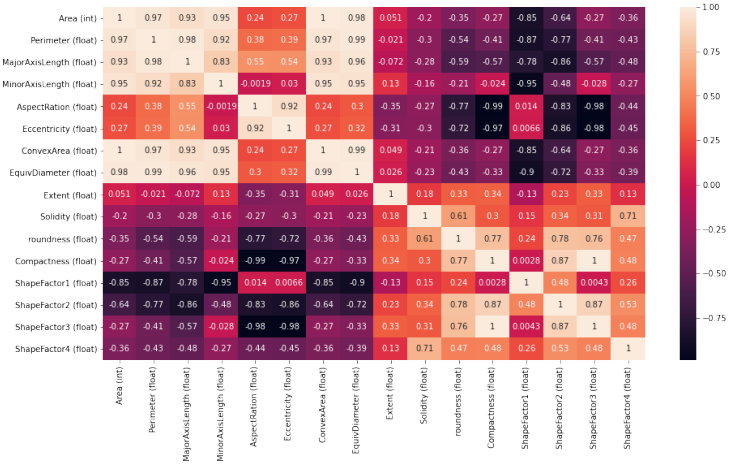


Figure 2 – Matrice de corrélation



Il a été intéressant de voir qu'il y a présence de débalancement (figure 1) et que certains attributs étaient très corrélés entre eux soit > 0.9 ou < -0.9 (figure 2). Nous avons essayer de retirer les attributs corrélées, mais il n'y a pas vraiment eu de gain et nous avons donc décider de garder tous les attributs.

Une fois les analyse de données terminées, nous avons dû faire quelques changement dans les données afin de pouvoir utiliser notre modèle. Nous avons en premier lieu, dû transformer les valeurs de X_train (attributs) en float. Nous avons ensuite retirer les valeurs de *ID* dans X_train et y_train. Nous pouvions ainsi entrainer notre modèle. Étant données les mauvais résultats initiaux, nous avons dû faire appel à la normalisation. Nous sommes donc passées d'une précision de 0.268 à une de 0.933 sur le classificateur SVM.

3 Méthodologie

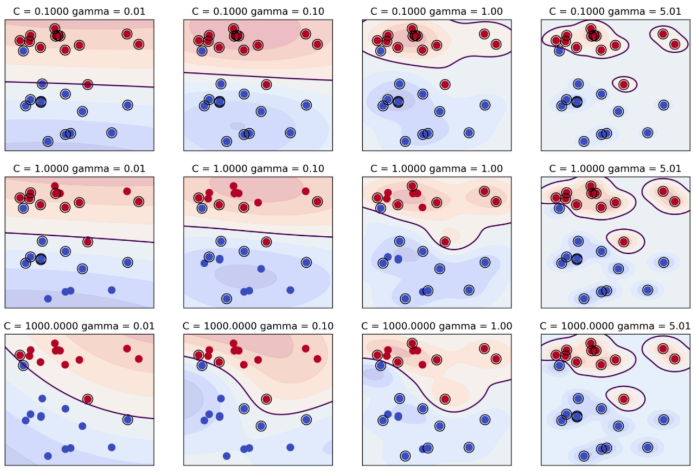
Le classificateur que nous avons choisis (*SVC*) est un classificateur binaire. Il était donc nécessaire de combiner ce classificateur avec une autre méthode. Nous avons donc choisi la méthode OneVsOne (/OneVsOneClassifier). En combinant ces deux algorithmes, nous obtenons un classificateur multiclasses qui est en fait composées de plusieurs classificateurs binaire. Dans notre cas nous avons 7 * 6 / 2 = 21 classificateurs binaires aux total.

En ce qui concerne la répartition des données, nous avons utilisé l'intégralité des données pour tester notre modèle. Nous avons pris cette décision, car nous avons utilisé en parallèle la méthode de vérification *k-fold cross-validation* avec k = 10. Ainsi, les données sont diviser en 10 parties dont 9 sont utilisées pour l'entraînement et la dernière comme partie de validation.

Comme nous pouvons le voir à la figure figure 1, nos données étaient assez débalancées. Nous avons fait des recherches à ce sujet et il y avait plusieurs solutions possibles. Il y avait la possibilité de faire du *oversampling* et du *undersampling*. Nous n'avons malheureusement pas eu de succès avec ces deux méthodes. Étant donnée que nous avons utilisés le modèle *SVC* de scikit-learn, nous avions accès a l'hyperparamètre $class_{weight}$ avec comme valeur *balanced* qui associe un poids à chaque classe selon sa fréquence dans les données pour gérer les débalancements. Malheureusement cet hyperparamètre n'a pas impacté positivement nos résultats.

Le modèle que nous avons utilisé est basé sur les machines à vecteurs de support et il y a deux paramètres principaux à tenir en compte, *C* et *gamma*. D'abord il est utile de comprendre que ce modèle classifie les données de façon linéaire et que pour classifier des données de non linéaires, la méthode utilise une fonction applé noyau (*kernel*). La valeur du gamma détermine à quel point les points proche et loin de la délimitation ont de l'importance. En d'autre mots, des valeurs élevé de gamma auront une meilleur délimitation, avec plus de curvatures, entre les données et donc le modèle aura plus tendance à souffrir de sur-apprentissage. À l'inverse, des valeurs faibles de gamma nous mèneront plus vers un problème de sous-apprentissage. La valeur de C nous indique plutôt notre niveau de tolérance aux erreurs et sa valeur impactera sur la généralisation de notre modèle. Une grande valeur de C augmente la precision et reduit les erreurs. Il fallait donc trouver un équilibre entre la précision de notre de modèle sur le données d'entrainement et sa capacité a généraliser. Nous avons trouver un équilibre optimale avec le valeur *gamma* = 0.19 et *C* = 2.8. Les impacts de ses paramètres peuvent être observés sur la figure 3.

Figure 3 – SVM paramètres C et gamma



4 Résultats

5 Discussion

6 stuff

N	N^2	N^3	N^4	sqrt(n)	sqrt[4](N)
1	1	1	1	1	1
2	4	8	16	1.4142	1.1892
3	9	27	81	1.7321	1.3161

Table 1 – A table

N	N^2	N^3	N^4	sqrt(n)	sqrt[4](N)
1	1	1	1	1	1
2	4	8	16	1.4142	1.1892
3	9	27	81	1.7321	1.3161

Table 2 – Another table

Table 3 – Regular table

N	N^2	N^3	N^4	sqrt(n)	sqrt[4](N)
1	1	1	1	1	1
2	4	8	16	1.4142	1.1892
3	9	27	81	1.7321	1.3161

```
1 ls -l
```

Listing 1 – Source code

```
total 740
-rw-rw-r-- 1 bndo bndo   1406 Apr 13 21:12 config.tex
drwxrwxr-x 2 bndo bndo   4096 Apr 14 23:52 img
drwxrwxr-x 2 bndo bndo   4096 Apr 13 23:34 _minted-report
-rw-rw-r-- 1 bndo bndo    459 Apr 13 21:12 packages.tex
-rw-rw-r-- 1 bndo bndo   1378 Apr 13 21:12 README.org
-rw-rw-r-- 1 bndo bndo   2978 Apr 15 00:12 report.bbl
-rw-rw-r-- 1 bndo bndo  10426 Apr 15 00:13 report.org
-rw-rw-r-- 1 bndo bndo 700441 Apr 15 00:12 report.pdf
-rw-rw-r-- 1 bndo bndo   9287 Apr 15 00:12 report.tex
-rw-rw-r-- 1 bndo bndo    738 Apr 13 21:12 template.bib
```

7 Section 2

selon une etude (Lemieux et al., 2021)

Références

Lemieux, V. L., Mashatan, A., Safavi-Naini, R. & Clark, J. (2021). A Cross-Pollination of ideas about Distributed Ledger Technological Innovation through a Multidisciplinary and Multisectoral lens : Insights from the Blockchain Technology Symposium '21. *Technology Innovation Management Review*, 58-66. https://doi.org/10.22215/timreview/1445