First Work
-
Artificial Intelligence and Smart Systems

Igor Varejão

Vitória, Brazil

Universidade Federal do Espírito Santos

Abstract

Machine Learning is taking an important role nowadays, with models been applied to a vary of tasks such as weather prediction, credit analysis and so forth. Therefore, it becomes a priority to study the algorithms on this area and evaluate theirs performance. This paper aims at the study of basics algorithms on classifications problems as also a new one, the KMC, KMeans centroids. This is an open source work that can be seen at github repository

1. Introduction

The paper consists in evaluate the performance of 5 machine learning algorithms in the wine database, they are:

- ZeroR (ZR)
- Gaussian Naive Bayes (GNB)
- K Nearest Neighbours (KNN)
- Decision Tree (DT)
- KMeans Centroids (KMC)

The experiments are divided into two parts:

1. First step: Evaluation of ZeroR and Gaussian Naive Bayes as they are non parametric, and shall not need to be applied a grid search

2. Second step: Evalutation of KNN, DT and KMC

In order to maintain the realiability of the results, the cross validation was applied in both steps, where in first step a 3 round stratified cross validation with 10 fold were done and, in the second, a stratified cross validation with 4 fold in the grid search (inner cycle) with a 3 round stratified cross validation with 10 fold in outer cycle. All of the classifiers were implemented using the sklearn framework [1], as the statics tests were using the scipy framework [2] and the boxplot were made with seaborn framework [3].

2. The Dataset

The wine dataset is a classic and very easy multi-class classification dataset. These data are the results of a chemical analysis of wine grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wine. In a classification context, this is a well posed problem with "well behaved" class structures.

2.1. Domain Description

All features are continuous real and positive.

2.2. Features and Targets

The features are attributes of the wine and the target are the wine's cultivar.

- **Features**: Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium, Total phenols, Flavanoids, Nonflavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines, Proline
- **Targets**: $cultivar_0$, $cultivar_1$, $cultivar_2$

2.3. Numbers of Instance

There are 178 instances

3. KMC - Kmeans Centroids

This method is a parametric classifier which applies a kmeans algorithm into the training set, computes k centroids of each class and then for prediction find the nearest neighbor of the test set, otherwise, applies a KNN with $k = 1$ to predict the class of sample.

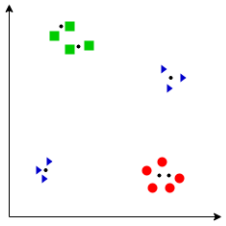

Figure 1: KMC with $k = 2$

4. Experiments on dataset and it results

To evaluate the models, the accuracy metric was applied and for the comparison between them, a boxplot, as well as the mean, standard deviation and the 95% confidence-interval of the metric recordings of all the predictions made in the execution of the model was generated. The graph 2 and table 4 are shown below.

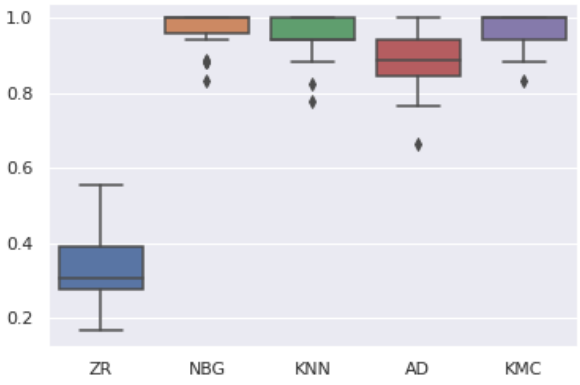

Figure 2: Boxplot of the results of each model

Método	Média	Desvio Padrão	Limite Inferior	Limite Superior
ZR	0.31	0.14	0.26	0.36
GNB	0.97	0.05	0.96	0.99
KNN	0.95	0.06	0.93	0.97
DT	0.89	0.07	0.87	0.92
KMC	0.97	0.05	0.95	0.98

Table 1: Statics of each model results

Furthermore, the p-values o the t-test and the non parametric wilcoxon test are scores statics that were computed between all models. If the p-value is smaller than the threshold(0.05), then we reject the null hypothesis of equal averages, and are presented in bold in table below 4 .

ZR	**0.0**	**0.0**	**0.0**	**0.0**
0.000002	GNB	**0.002293**	**0.0**	0.264618
0.000002	**0.004509**	KNN	**0.000002**	0.080042
0.000002	**0.000006**	**0.000057**	DT	**0.0**
0.000002	0.417413	0.058782	**0.000032**	KMC

Table 2: Hyposthesis tests

5. Conclusions

5.1. Results Analysis

As expected the ZeroR performance was much worse compared to the others as turns out it generates predictions that ignores the input features. With respect to the others models, they had pretty good results approaching the ideal model, evidencing that the dataset in question are easy to classify, also expected, given the dataset provider has classified the dataset as *"a well posed problem with "well behaved" class structures"*.

Regarding the hypothesis tests, we can see that the KMC might have equal averages of the KNN and GNB, while the others comparisons reject the null hypothesis of equal averages.

5.2. Work Contributions

The work presented does not make any relevant contributions to the academic world. Although, it contribution to machine learning comprehension is immeasurable. It taught the basics classifications algorithms used in nowadays and how to compare models, an essential skill in this area. Another skill learned were to create new classifier given that KMC is not implemented in the sklearn framework. And last but not least, taught how to interpreter the p-values of hypothesis tests.

5.3. Improvements and future works

In order to note difference between those classifiers in matter of performance it shall be an interest work to analyse theirs results in a more challenging dataset.

References

[1] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, Journal of machine learning research 12 (Oct) (2011) 2825–2830.

[2] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods 17 (2020) 261–272. doi:10.1038/s41592-019-0686-2.

[3] M. L. Waskom, seaborn: statistical data visualization, Journal of Open Source Software 6 (60) (2021) 3021. doi:10.21105/joss.03021.
URL https://doi.org/10.21105/joss.03021