Performance Comparison of Learning Algorithms on Spambase dataset

Assignment 2 DV2542

Mohit Vellanki
Department of Computer Science and Engineering
BTH
Karlskrona, Sweden
move15@student.bth.se

I. Introduction

This report gives the details about the performance comparison of three supervised learning algorithms on the spambase dataset in Python using the Scikit-learn library. The comparison between the algorithms is done using the computational performance metric – training time and predictive performance metrics – accuracy of the model and the F-measure score also known as F1 score. The metrics are calculated on the stratified folds of the data generated using the stratified cross validation. Friedman test is conducted on the three metrics and then it is checked if there is a significance difference between the results obtained using different algorithms.

The dataset consists of 4601 instances and 58 attributes out of which 1 attribute (spam) is the target attribute. Initially in this project, the dataset is split into ten folds using the stratified cross validation technique. The model is then trained and then the instances are classified using the three supervised learning algorithms. The algorithms chosen are - Gaussian Naive Bayes, Decision Tree Classifier and Logistic Regression. Once the instances of the classifier models were created and the results are generated. Friedman test is conducted on the results manually

II. Results

In the first table, the accuracy of each algorithm is given along with the ranks, average and the Friedman statistic for the ten folds.

Folds	Naive Bayes	Decision Tree	Logistic Regression
1	0.8438(3)	0.9241(2)	0.9306(1)
2	0.859(3)	0.9219(2)	0.9241(1)
3	0.8829(3)	0.9024(2)	0.9176(1)
4	0.8652(3)	0.9109(2)	0.9457(1)
5	0.8848(3)	0.9261(2)	0.9326(1)
6	0.8283(3)	0.9239(2)	0.9348(1)
7	0.8326(3)	0.937(2)	0.9565(1)
8	0.8674(3)	0.9152(2)	0.9391(1)
9	0.634(3)	0.8301(2)	0.8497(1)
10	0.719(3)	0.8192(2)	0.8562(1)
avg	3	2	1
	Friedman statistic = 20		

In the second table, the training time of each algorithm is given along with the ranks, average and the Friedman statistic for the ten folds.

Folds	Naive Bayes	Decision Tree	Logistic Regression
1	0.007(1)	0.073(3)	0.06(2)
2	0.014(1)	0.093(2)	0.095(3)
3	0.011(1)	0.111(2)	0.143(3)
4	0.019(1)	0.122(3)	0.069(2)
5	0.016(1)	0.096(2)	0.163(3)
6	0.012(1)	0.102(3)	0.077(2)
7	0.011(1)	0.109(2)	0.133(3)
8	0.012(1)	0.115(3)	0.079(2)
9	0.013(1)	0.094(3)	0.054(2)
10	0.012(1)	0.124(3)	0.066(2)
avg	1	2.6	2.4
	Friedman statistic = 15.2		

In the last table, the F-measure of each algorithm is given along with the ranks, average and the Friedman statistic for the ten folds.

Folds	Naive Bayes	Decision Tree	Logistic Regression
1	0.8294(3)	0.9003(2)	0.908(1)
2	0.8456(3)	0.8971(2)	0.9003(1)
3	0.8683(3)	0.8739(2)	0.8939(1)
4	0.8517(3)	0.8864(2)	0.9307(1)
5	0.8658(3)	0.9071(2)	0.9141(1)
6	0.8192(3)	0.903(2)	0.9194(1)
7	0.8205(3)	0.9169(2)	0.9429(1)
8	0.8479(3)	0.8949(2)	0.9218(1)
9	0.6693(3)	0.803(2)	0.8179(1)
10	0.7127(3)	0.7855(2)	0.8136(1)
avg	3	2	1
	Friedman statistic = 20		

III. Analysis

The Friedman statistic for the three metrics - accuracy, training time and the F-measure is 20, 15.2 and 20. It is significantly more than the critical value (7.8). So the algorithms are not similar in performance (significantly different).

For n = 10 and k = 3, the Nemenyi critical difference is 1.047. Hence when the accuracy and F-measure are considered, Logistic regression algorithm is significantly better than the Naïve Bayes. When the training time is taken into consideration, Naive Bayes is significantly faster than the other two algorithms.