Supervised Machine Learning Algorithms Analysis

Jungsoo Lee jul273@ucsd.edu
University of California
San Diego, CA 92023, USA

Abstract

This paper is to replicate 2006 paper by Caruana and Niculescu-Mizil, focusing on analysis of performance of several supervised machine learning algorithms. The algorithms are performed on various datasets through multiple trials with a range of hyperparameter search. All of the datasets are binary classification problems. The performance of each algorithm is measured with various performance metrics. Additionally, the effect of training sampling size and the time complexity of performance are analyzed, as well. The results are investigated in detailed level offering multiple tables and appendices in between texts.

1. Introduction

As numerous machine learning algorithms have been developed since the mid 20th century, performance of each algorithm varies greatly depending on its principle and characteristics. The CNM06 explores ten supervised machine learning algorithms on eleven binary classification problems with eight performance metrics. While this paper is intended to use the same methodology, it is a small version exploring three algorithms on four binary classification problems with five performance metrics. The goal of this paper is to compare the performance of algorithms at a detailed level and analyze them.

2. Method

2.1. Learning Algorithms

Three different classifiers used in this paper: Logistic Regression, Random Forest, and Naive Bayes. All of the classifiers are used with varying parameters. MinMaxScaler from sklearn is used to normalize the attributes in all datasets, although Random Forest and Naive Bayes have no need of.

Logistic Regression predicts the probability of an event using a logistic function. Since it predicts the probability, its output values range from 0 to 1. The model is trained with ridge parameters ranging from 10^-8 to 10^4.

Random Forest is an ensemble method of decision trees. The model is trained with parameters of varying maximum depth of 5, 6, 7, and None.

Naive Bayes is based on Bayes' theorem, considering each feature independently to predict the probability. The model is trained with parameters of varying variance smoothing of np.logspace(0, -9, num=5).

2.2. Performance Metric

The algorithms are measured in five different performance metrics: accuracy (ACC), F-score (FSC), precision (PRC), recall (RCL), and the area under the ROC curve (AUC). Overall, the models performed well having most of the metrics from 0.70 ~ 0.99.

ACC is the ratio of correct predictions to all predictions. It is a straightforward and intuitive metric which the cost of misclassification is especially critical for heavily imbalanced datasets. FSC is the weighted average of the PRC and RCL. It may be a better model than ACC but it is less interpretable because it is a metric PRC and RCL combined. PRC is the ratio of correct positive predictions to all positive predictions. It is preferred when false positive is crucial. RCL is the ratio of correct positive predictions to all real positives. It is preferred when false negative is crucial. AUC, the most widely used metric, is the area under the curve where the false positive rate and true positive rate meet on the graph.

2.3. Datasets

All of the three datasets are binary classification problems and collected from UCI Machine Learning Repository.

The purpose of ADULT dataset is to predict the income of an individual using fourteen attributes such as age, workclass, and education. Individuals with income greater than 50K are labeled as a positive class and among 30162 observations, 25% of the dataset represents a positive class.

The purpose of OCCUPANCY dataset is to predict the occupancy of an office using six attributes such as light, temperature, and humidity. Offices with occupied status are labeled as a positive class and among 8143 observations, 21% of the dataset represents a positive class.

The purpose of CREDIT CARD dataset is to predict probability of default of a client using twenty attributes such as amount of credit, gender and education. Clients with default payment are labeled as positive class and among 30000 observations, 22% of the dataset represents positive class.

The purpose of MAGIC dataset is to predict the image of primary gamma/ hadronic showers using ten attributes such as number of major/minor axis of ellipse and sum of content of all pixels. Images of primary gamma are labeled as a positive class and among 19020 observations, 65% of the dataset represents positive class. For technical reasons, the number of negative class is underestimated, but in real data, it represents the majority of the observations.[2]

Table 1. Description of Problems

Problems	# ATTR	TRAIN SIZE	TEST SIZE	% POZ
ADULT	14/105	5000	25162	25%
OCCUPANCY	6	5000	3143	21%
CREDIT CARD	23	5000	25000	22%

MAGIC	10	5000	14020	65%

3. Experiment

All of the factors are equally used for every algorithm except for parameters, since each algorithm has different parameters. 5000 of training samples are randomly collected from each dataset and all of the rest of the dataset are used as test sets. Through grid search, the optimal hyperparameter is selected. This setting is kept the same for five trials.

4. Discussion

The results of performance by algorithms are shown below in two tables. Table 2 shows the mean test set performance by algorithm/ data set combo and the best performance for each dataset is indicated as bold font. Table 3 shows the test set performance by algorithm/ metric combo and the best performance for each metric is indicated as bold font. * indicates a performance that is not statistically significant from the best performance and it is investigated through t-tests. The threshold for t-test is set p=0.05 as CNM05 indicates.

Overall, all of the algorithms perform relatively well on the datasets. The mean of test set performance of the Random Forest algorithm is the highest in ADULT, OCCUPANCY, and MAGIC datasets, but it is not significantly different from the highest in CREDIT CARD dataset, as well. The mean of test set performance of the Random Forest algorithm is the highest in every performance metrics except for RCL, in which Gaussian Naive Bayes is the highest.

Table 2. Test Set Performance Algorithm/ Data Set Combo

MODEL	ADULT	OCCUPANCY	CREDIT CARD	MAGIC
LogisticRegression	0.679876*	0.700777	0.379210	0.809530
RandomForestClassifier	**0.743908**	**0.991897**	0.607837*	**0.902381**
GaussianNB	0.699049*	0.964431	**0.611524**	0.793949

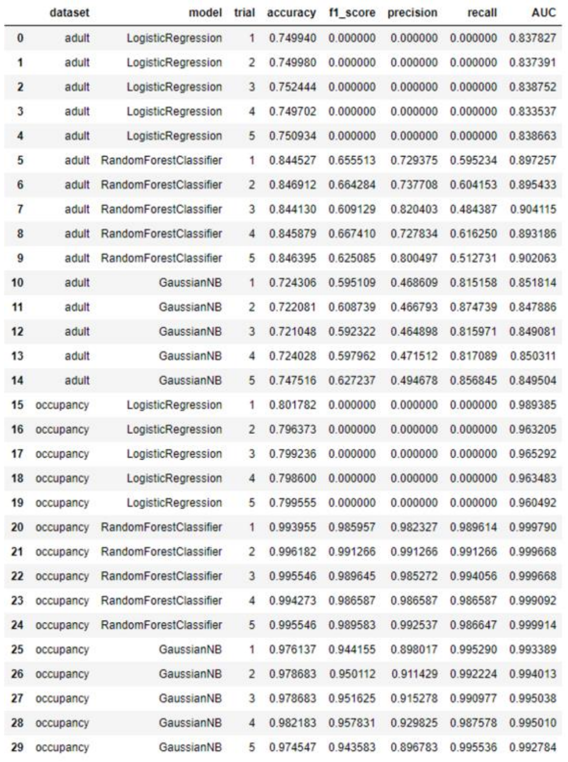
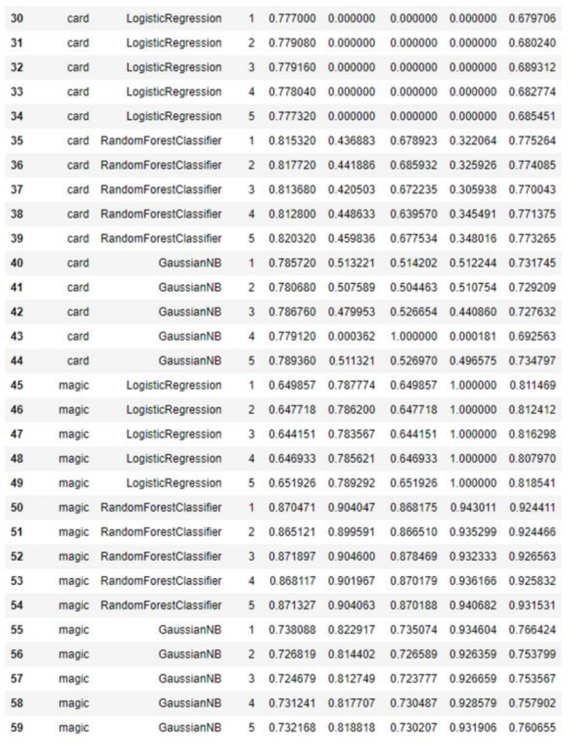
Table 3. Test Set Performance Algorithm/ Data Set Combo.

Model	ACC	FSC	PRC	RCL	AUC
LogisticRegression	0.794524	0.456736	0.703023	0.422209	0.835249

RandomForestClassifier	**0.882063**	**0.746023**	**0.821220**	0.708413	**0.899811**
GaussianNB	0.806772	0.720590*	0.661287	**0.812775**	0.834768

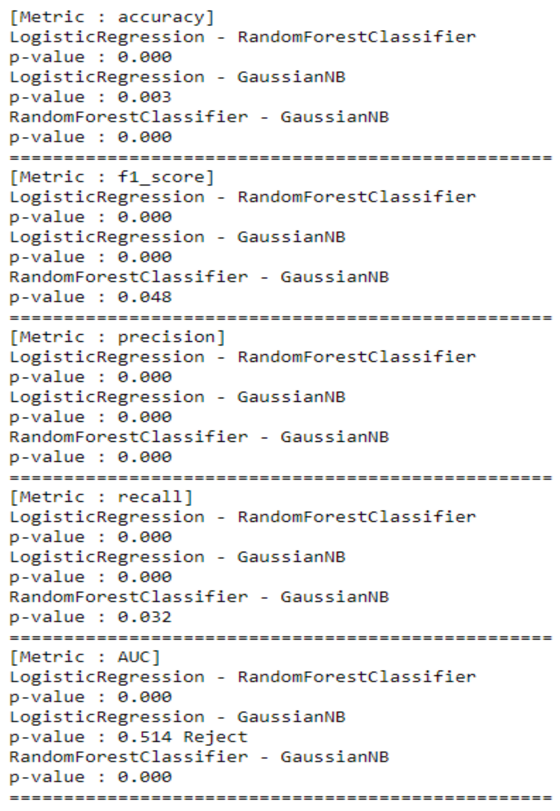
Appendix 1. Training Set Performance Algorithm/ Data Set Combo				
MODEL	ADULT	OCCUPANCY	CREDIT CARD	MAGIC
LogisticRegression	0.682557	0.7034434	0.3992838	0.8099464
RandomForestClassifier	**0.949233**	**1.00000**	**0.702199**	**1.00000**
GaussianNB	0.6995912	0.966424	0.6119244	0.7929318

Appendix 2. Raw Test Scores

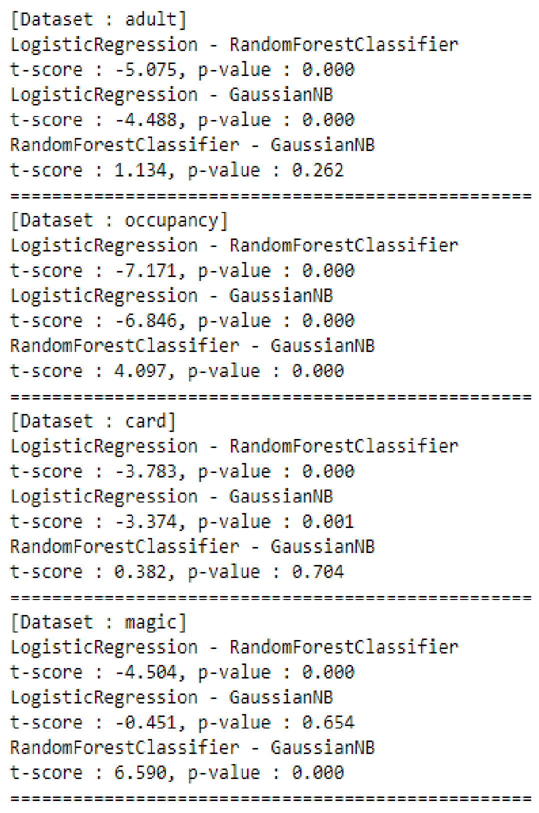



	dataset	model	trial	accuracy	f1_score	precision	recall	AUC
0	adult	LogisticRegression	1	0.749940	0.000000	0.000000	0.000000	0.837827
1	adult	LogisticRegression	2	0.749980	0.000000	0.000000	0.000000	0.837391
2	adult	LogisticRegression	3	0.752444	0.000000	0.000000	0.000000	0.838752
3	adult	LogisticRegression	4	0.749702	0.000000	0.000000	0.000000	0.833537
4	adult	LogisticRegression	5	0.750934	0.000000	0.000000	0.000000	0.838663
5	adult	RandomForestClassifier	1	0.844527	0.655513	0.729375	0.595234	0.897257
6	adult	RandomForestClassifier	2	0.846912	0.664284	0.737708	0.604153	0.895433
7	adult	RandomForestClassifier	3	0.844130	0.609129	0.820403	0.484387	0.904115
8	adult	RandomForestClassifier	4	0.845879	0.667410	0.727834	0.616250	0.893186
9	adult	RandomForestClassifier	5	0.846395	0.625085	0.800497	0.512731	0.902063
10	adult	GaussianNB	1	0.724306	0.595109	0.468609	0.815158	0.851814
11	adult	GaussianNB	2	0.722081	0.608739	0.466793	0.874739	0.847886
12	adult	GaussianNB	3	0.721048	0.592322	0.464898	0.815971	0.849081
13	adult	GaussianNB	4	0.724028	0.597962	0.471512	0.817089	0.850311
14	adult	GaussianNB	5	0.747516	0.627237	0.494678	0.856845	0.849504
15	occupancy	LogisticRegression	1	0.801782	0.000000	0.000000	0.000000	0.989385
16	occupancy	LogisticRegression	2	0.796373	0.000000	0.000000	0.000000	0.963205
17	occupancy	LogisticRegression	3	0.799236	0.000000	0.000000	0.000000	0.965292
18	occupancy	LogisticRegression	4	0.798600	0.000000	0.000000	0.000000	0.963483
19	occupancy	LogisticRegression	5	0.799555	0.000000	0.000000	0.000000	0.960492
20	occupancy	RandomForestClassifier	1	0.993955	0.985957	0.982327	0.989614	0.999790
21	occupancy	RandomForestClassifier	2	0.996182	0.991266	0.991266	0.991266	0.999668
22	occupancy	RandomForestClassifier	3	0.995546	0.989645	0.985272	0.994056	0.999668
23	occupancy	RandomForestClassifier	4	0.994273	0.986587	0.986587	0.986587	0.999092
24	occupancy	RandomForestClassifier	5	0.995546	0.989583	0.992537	0.986647	0.999914
25	occupancy	GaussianNB	1	0.976137	0.944155	0.898017	0.995290	0.993389
26	occupancy	GaussianNB	2	0.978683	0.950112	0.911429	0.992224	0.994013
27	occupancy	GaussianNB	3	0.978683	0.951625	0.915278	0.990977	0.995038
28	occupancy	GaussianNB	4	0.982183	0.957831	0.929825	0.987578	0.995010
29	occupancy	GaussianNB	5	0.974547	0.943583	0.896783	0.995536	0.992784
30	card	LogisticRegression	1	0.777000	0.000000	0.000000	0.000000	0.679706
31	card	LogisticRegression	2	0.779080	0.000000	0.000000	0.000000	0.680240
32	card	LogisticRegression	3	0.779160	0.000000	0.000000	0.000000	0.689312
33	card	LogisticRegression	4	0.778040	0.000000	0.000000	0.000000	0.682774
34	card	LogisticRegression	5	0.777320	0.000000	0.000000	0.000000	0.685451
35	card	RandomForestClassifier	1	0.815320	0.436883	0.678923	0.322064	0.775264
36	card	RandomForestClassifier	2	0.817720	0.441886	0.685932	0.325926	0.774085
37	card	RandomForestClassifier	3	0.813680	0.420503	0.672235	0.305938	0.770043
38	card	RandomForestClassifier	4	0.812800	0.448633	0.639570	0.345491	0.771375
39	card	RandomForestClassifier	5	0.820320	0.459836	0.677534	0.348016	0.773265
40	card	GaussianNB	1	0.785720	0.513221	0.514202	0.512244	0.731745
41	card	GaussianNB	2	0.780680	0.507589	0.504463	0.510754	0.729209
42	card	GaussianNB	3	0.786760	0.479953	0.526654	0.440860	0.727632
43	card	GaussianNB	4	0.779120	0.000362	1.000000	0.000181	0.692563
44	card	GaussianNB	5	0.789360	0.511321	0.526970	0.496575	0.734797
45	magic	LogisticRegression	1	0.649857	0.787774	0.649857	1.000000	0.811469
46	magic	LogisticRegression	2	0.647718	0.786200	0.647718	1.000000	0.812412
47	magic	LogisticRegression	3	0.644151	0.783567	0.644151	1.000000	0.816298
48	magic	LogisticRegression	4	0.646933	0.785621	0.646933	1.000000	0.807970
49	magic	LogisticRegression	5	0.651926	0.789292	0.651926	1.000000	0.818541
50	magic	RandomForestClassifier	1	0.870471	0.904047	0.868175	0.943011	0.924411
51	magic	RandomForestClassifier	2	0.865121	0.899591	0.866510	0.935299	0.924466
52	magic	RandomForestClassifier	3	0.871897	0.904600	0.878469	0.932333	0.926563
53	magic	RandomForestClassifier	4	0.868117	0.901967	0.870179	0.936166	0.925832
54	magic	RandomForestClassifier	5	0.871327	0.904063	0.870188	0.940682	0.931531
55	magic	GaussianNB	1	0.738088	0.822917	0.735074	0.934604	0.766424
56	magic	GaussianNB	2	0.726819	0.814402	0.726589	0.926359	0.753799
57	magic	GaussianNB	3	0.724679	0.812749	0.723777	0.926659	0.753567
58	magic	GaussianNB	4	0.731241	0.817707	0.730487	0.928579	0.757902
59	magic	GaussianNB	5	0.732168	0.818818	0.730207	0.931906	0.760655

Appendix 3. P-values of the Comparisons across algorithms



```
[Metric : accuracy]
LogisticRegression - RandomForestClassifier
p-value : 0.000
LogisticRegression - GaussianNB
p-value : 0.003
RandomForestClassifier - GaussianNB
p-value : 0.000
==================================================
[Metric : f1_score]
LogisticRegression - RandomForestClassifier
p-value : 0.000
LogisticRegression - GaussianNB
p-value : 0.000
RandomForestClassifier - GaussianNB
p-value : 0.048
==================================================
[Metric : precision]
LogisticRegression - RandomForestClassifier
p-value : 0.000
LogisticRegression - GaussianNB
p-value : 0.000
RandomForestClassifier - GaussianNB
p-value : 0.000
==================================================
[Metric : recall]
LogisticRegression - RandomForestClassifier
p-value : 0.000
LogisticRegression - GaussianNB
p-value : 0.000
RandomForestClassifier - GaussianNB
p-value : 0.032
==================================================
[Metric : AUC]
LogisticRegression - RandomForestClassifier
p-value : 0.000
LogisticRegression - GaussianNB
p-value : 0.514 Reject
RandomForestClassifier - GaussianNB
p-value : 0.000
==================================================
```

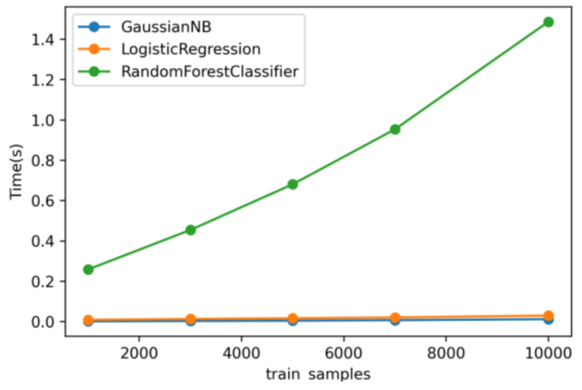


```
[Dataset : adult]
LogisticRegression - RandomForestClassifier
t-score : -5.075, p-value : 0.000
LogisticRegression - GaussianNB
t-score : -4.488, p-value : 0.000
RandomForestClassifier - GaussianNB
t-score : 1.134, p-value : 0.262
================================================
[Dataset : occupancy]
LogisticRegression - RandomForestClassifier
t-score : -7.171, p-value : 0.000
LogisticRegression - GaussianNB
t-score : -6.846, p-value : 0.000
RandomForestClassifier - GaussianNB
t-score : 4.097, p-value : 0.000
================================================
[Dataset : card]
LogisticRegression - RandomForestClassifier
t-score : -3.783, p-value : 0.000
LogisticRegression - GaussianNB
t-score : -3.374, p-value : 0.001
RandomForestClassifier - GaussianNB
t-score : 0.382, p-value : 0.704
================================================
[Dataset : magic]
LogisticRegression - RandomForestClassifier
t-score : -4.504, p-value : 0.000
LogisticRegression - GaussianNB
t-score : -0.451, p-value : 0.654
RandomForestClassifier - GaussianNB
t-score : 6.590, p-value : 0.000
================================================
```

5. Bonus

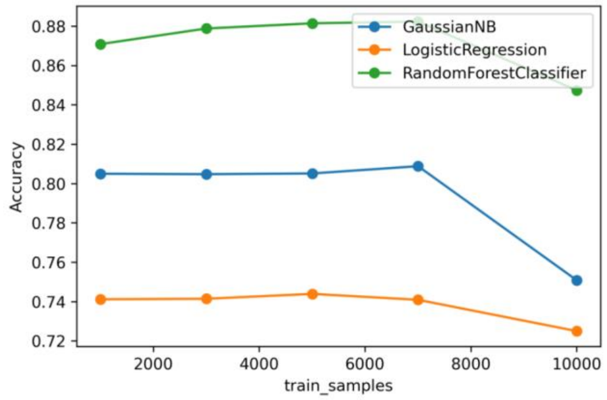
Appendix below is the time complexity curve of each algorithm varying the number of training samples. The Random Forest model has the longest training time and it increases the most drastically when the size of the training sample increases.

Appendix 4. Time Complexity Curve of Algorithms varying the Number of Training Samples



Appendix below is a learning curve that compares the test set performance for the best hyperparameter choice as varying the number of training samples. The performance for every three algorithms increases until the training sample size reaches ~7000, but significantly decreases after that.

Appendix 5. Learning Curve of Algorithm varying the Number of Training Samples



6. Reference

[1] Rich Caruana & Alexandru Niculescu-Mizil. (2006) An Empirical Comparison of Supervised Learning Algorithms. Ithaca, NY: Cornell University.
[2] UCI Machine Learning Data Repository. archive.ics.uci.edu/ml. Irvine, CA. University of California, Irvine.
[3] scikit-learn: Machine Learning in Python. scikit-learn.org.
[4] tensorflow:An open-source machine learning framework for everyone. tensorflow.org.