Proceedings of Machine Learning Research 1–4

Investigations of binary classification algorithms

Xinkai Wang x5wang@ucsd.edu

Abstract

There are many widely used supervised machine learning algorithms. In this article, I choose three of them: logistic regression, k-nearest neighbors, and random forest. Given four data sets and five trials on each data set , I compare the performance of the three algorithms and use paired t-test to check whether their performance are distinguishable.

Keywords: supervised machine learning, algorithm performance

1. Introduction

Different algorithms could have advantages in different areas of study and on different problems, so we choose to examine how well logistic regression, k-nearest neighbors, and random forest perform on four different classification scenarios with the evaluation criteria of three metrics: accuracy, ROC, and F1. Logistic regression is a binary classification algorithm using the logistic model, which uses a linear separating plane to do the probabilistic classification task. K-nearest neighbors is a non-parametric model that predicts a point based on the labels of the point's closest neighbors. Random forest is an ensemble method that builds a multitude of decision trees to make predictions. Since there are different combinations of algorithms, data sets, and metrics, it allows us to see the whole picture of the various combinations that produce significant or non-significant results. Before we train the entire training test and measure the accuracy on the test set, we would also explore a wide range of possible hyper-parameters to choose the best hyper-parameters that maximize the accuracy of the model.

2. Methodology

2.1. Learning Algorithms

As mentioned before, I choose three different supervised learning algorithms for binary classification: logistic regression, k-nearest neighbors, and random forest. The parameters for each algorithm are described below:

Logistic Regression I use both regularized $L2$ model and unregularized model. For the regularized model, we will vary the weight parameter C by a factor of 10 from 10^{-8} to 10^{4}.

K-Nearest Neighbors I use 26 values from 1 to 105, with a step size of 5. I do not use 1 to the size of the training set because the algorithm would not work efficiently using large k values. We also use both Euclidean Distance and weighted Euclidean Distance.

Random Forests I use 1024 as the number of trees in the forest. The maximum number of features for each tree is 1,2,4,6,8,12,16 or 20.

For logistic regression and k-nearest neighbors, I transform the data into a standard scale with mean of 0 and standard deviation of 1. For random forests, I do not need to scale the data. I train a sample of 5,000 for each model.

2.2. Performance Metrics

I use two types of metrics: threshold metrics and a rank metric. The threshold metrics are accuracy and F1 score. They predict in hard boundary or binary terms: whether a prediction is above or below the threshold. The rank metric is ROC, which predict upon the rank of performance of different predictions.

2.3. Data Sets

I use four data sets found on UCI Machine Learning Repository and Kaggle: Rain in Australia, Bank Marketing, Adult Income, and Occupancy Detection. For the first data set, based on the weather conditions in previous days and years, it predicts whether it will rain tomorrow. For the second data set, based on customer information, it predicts whether the customer will deposit money in the bank. For the third data set, based on an individual's work and family conditions, it predicts whether the individual has an annual income of 50K. For the last data set, based on the room conditions, it predicts whether there are people in the rooms. The columns and rows with too many null values are dropped, non-binary categorical variables are one-hot encoded, and binary categorical variables are encoded as 1 or 0. The links to detailed descriptions and access to the data sets are referenced in the code section. Some other basic information about the data sets are included in table 1 below:

Table 1: Description of problems

Problem	Attr	Train	Test	%Pos
rain	23/11	5000	130643	22
bank	17/49	5000	5162	47
adult	15/66	5000	41043	25
occu	7/6	5000	5808	25

3. Performance by problem and metric

First, I randomly pick 5000 samples from each data set, and then I use stratified 5 folds to do a 80/20 training and validation set split for 5 times. After that, by implementing a grid search from the parameter search spaces for each model, I find the best hyper-parameters measured on each metric for each model that lead to the highest validation score in the 5 number of times of validation. Next, I use the best parameters to train the model on the 5000 samples in the training set, and I can use the rest of the data apart from the training set as test set to fit into the model to measure the performance score of each combination of model, data set and metric, represented by each entry in table 2.

From table 2, the rows are the three models and there are three metrics in columns and they are repeated four times for four different data sets. Then, a mean along the rows are calculated. Boldfaced scores are the best score in the column, and scores with a * are the models whose performance are not significantly different from the performance of the best models with scores in boldface.

Looking at the results, we see that no matter which metric we use for a particular problem, we get the same best models and its associated not significantly different models. For the first three problems, logistic regression model gives the best scores; for the last problem, k-nearest neighbors gives the best scores. Therefore, logistic regression and k-nearest neighbors might perform better in different data sets depending on how the data is organized and how similar the algorithm predicts in a way that simulates real-world scenarios. Most scores are around 0.83, except the last data set has an average score of around 0.98, which might due to the strong

Table 2: Scores for each algorithm by problem and metric

	rain data set			bank data set			adult data set			occupancy data set			
Model	ACC	ROC	F1	ACC	ROC	F1	ACC	ROC	F1	ACC	ROC	F1	mean
LR	**.835**	**.835**	**.835**	**.824**	**.824**	**.824**	**.845**	**.845**	**.845**	.986	.985*	.986	**.872**
KNN	.829	.825*	.829	.748*	.735*	.75*	.825	.827*	.825	**.992**	**.99**	**.992**	.847
RF	.819*	.819*	.818*	.787	.785	.783	.805*	.804	.803	.976	.978	.978	.846*

correlations between the attributes and the labels.

Overall, logistic regression has the highest scores. Also, I notice that the logistic regression model takes the least amount of time to complete running the tests compared to the other two models. Given such results, I assume that logistic regression model is one of the most useful models for many classifications problems. However, we also see that after paired t-tests, one other model could perform differently just due to chance, not due to the nature of the difference in algorithms. Consequently, I qualify my previous statement that some other models could be nearly as predictive as the best performing model.

4. Performance by metric

Please refer to table 3 for this section. It is not surprising that when problems are averaged for each metric, Logistic Regression still have the best performance on every metric measured, and although random forest scores are lower than k-nearest neighbors on both accuracy and f1 measure, its performance not significantly distinguishable from logistic regression. K-nearest neighbors might be different because it predicts in a distinct way by referring to a point's neighbors, whereas logistic regression and random forest are trying to use lines to separate the points.

Table 3: Performance by metric

Model	ACC	ROC	F1
LR	**.872**	**.872**	**.872**
KNN	.848	.844	.849
RF	.847*	.847*	.846*

5. Conclusions

It is nice to see the many algorithms that are able to perform the same binary classification tasks nearly equally well. To some extent, there is not much difference or variability about the level of performance from the existing supervised machine learning algorithms, even though those algorithms are indeed implemented in completely different ways. On our experiments, if we really want to rank the algorithms, logistical regression model could be the best, but k-nearest neighbors and random forests are approximately at the same level.

There could be some limitations of the data, since I drop some null values and unique categorical variables that do not seem to relate to the prediction tasks, even though they probably do make a difference. Further research can be done on more algorithms and data sets if the conditions allow, so that variability of algorithms could be reduced, so that we will be more confident about our results.

Machine learning is a brand-new area at the intersections of statistics and computer science. In the previous year, we are more focused on how to process, store and output data, but now we are more close to making the data more meaningful so that computers learn from the data in complicated but precise ways, like neural networks. Hopefully, we will continue making progress toward more "intelligent" algorithms that get the most out of the existing data around the world.

6. Appendix

In general, k-nearest neighbors has the best training set performance, which is near perfect, which might because when we do k folds, we have understand data in most areas. Then, logistic regression model has slightly better training performance than random forests. Except k-nearest neighbors that has much higher training performance than test performance, other scores are pretty much similar to the test performance on each data set.

Table 4: Training Performance for algorithms and data sets

Model	rain	bank	adult	occupancy
LR	.837	.831	.855	.985
KNN	.979	.985	.933	1
RF	.82	.795	.806	.977

References

R. Caruana and A. Niculescu-Mizil. "An empirical comparison of supervised learning algorithms."In Proceedings of the 23rd international conference on Machine learning,161-168. 2006.

Table 5: Raw Scores for each algorithm by problem and metric with 5 trials

	rain data set			bank data set			adult data set			occupancy data set		
Model	ACC	ROC	F1	ACC	ROC	F1	ACC	ROC	F1	ACC	ROC	F1
LR_1	.835	.835	.835	.82	.82	.82	.847	.847	.847	.986	.985	.986
LR_2	.833	.834	.833	.821	.821	.821	.844	.844	.844	.987	.986	.987
LR_3	.836	.836	.836	.828	.828	.829	.843	.843	.843	.986	.985	.986
LR_4	.835	.835	.835	.817	.817	.817	.845	.845	.845	.984	.983	.984
LR_5	.834	.834	.834	.832	.832	.832	.845	.845	.845	.987	.985	.987
KNN_1	.827	.825	.827	.735	.728	.744	.824	.825	.824	.99	.99	.99
KNN_2	.83	.825	.83	.748	.728	.748	.827	.828	.827	.992	.99	.992
KNN_3	.83	.826	.83	.759	.754	.759	.826	.827	.826	.992	.99	.992
KNN_4	.831	.826	.831	.749	.727	.749	.822	.828	.822	.992	.989	.992
KNN_5	.83	.826	.83	.748	.736	.748	.827	.826	.827	.991	.99	.991
RF_1	.818	.822	.818	.791	.781	.786	.801	.805	.803	.981	.982	.98
RF_2	.82	.818	.815	.777	.78	.778	.801	.795	.794	.974	.976	.977
RF_3	.822	.822	.821	.788	.787	.779	.81	.808	.807	.974	.976	.978
RF_4	.817	.817	.819	.786	.789	.783	.804	.803	.8	.976	.976	.977
RF_5	.818	.818	.819	.795	.786	.791	.81	.811	.81	.976	.977	.98

Table 6: p-values between best algorithm and two others

	rain data set			bank data set			adult data set			occupancy data set		
Model	ACC	ROC	F1	ACC	ROC	F1	ACC	ROC	F1	ACC	ROC	F1
LR										.001	3.924	.001
KNN	.003	2.813	.003	3.047	2.164	1.237	.0	8.147	.0			
RF	7.725	7.765	2.628	.0	.0	.0	9.541	.0	.0	.001	.0	.0

Table 7: p-values between best algorithm and two others averaged

Model	ACC	ROC	F1
KNN	.004	.004	.004
RF	1.16	6.848	5.288