COGS 118A Coursework: An Empirical Supervised Machine Learning Algorithms' Comparison Test

Yuan Tang YUTANG@UCSD.EDU
Cognitive Science Class 118A
University of California San Diego
San Diego, CA 92093, USA

Abstract

This paper investigates a machine learning algorithms' performance comparison for binary classification problems, replicating a small-scale experiment carried out by Caruana and Niculescu-Mizil 2006 paper, which will be referred to as CNM06. Specifically, this paper will compare between three supervised machine learning methods, including Logistic Regression, Support Vector Machines, and K-nearest neighbor. The error metrics used to compare the algorithms are accuracy, the area under the receiver operating characteristic curve (ROC), and F1 score.

1 Introduction

Caruana and Niculescu-Mizil (2006) conducted a more comprehensive comparison test for supervised machine learning algorithms, following STATLOG (King et al., 1995), by adding comparisons between newly emerged learning methods, such as SVMS, random forests, and boosting. This experiment is a small-scale replication of the experiment conducted in CNM06 paper, aiming to validate the credibility of CNM06 in small numbers of datasets, and check if the result still stays true in 2021 as the algorithm calculation process may change along the development of computer science. In this project, models that are used to compare are logistic regression, SVMs, and KNN. The comparison measures are performance metrics, including Accuracy, ROC, and F1 score.

In this algorithm performance comparison test, four datasets in total are used. Three data sets used among these four are drawn from CNM06 paper and one more dataset, MAGIC, is unique to my experiment. The dataset, ADULT, is quite imbalanced with only 26% of positive and 74% of negative data points. A stratified random sampling for splitting training and testing sets are applied to all datasets.

The results of this experiment are quite similar to CNM06 (Caruana & Niculescu-Mizil, 2006). In my analysis, SVM's performance is generally better than KNN, and KNN is generally better than Logistic regression with one exception. These results are precisely consistent in CNM06 (Caruana & Niculescu-Mizil, 2006), even the exception is caught up in the same way. Specifically, in the ADULT dataset, logistic regression's average performance is like SVM, meaning that logistic regression outperforms KNN, which is also consistent with the result demonstrated in CNM06's Table 3 (Caruana & Niculescu-Mizil, 2006).

2 Method

2.1 Learning Algorithms

This paper explores 3 algorithms utilized within CNM06, in which SVMs are supposed to have more accurate overall performance over other two. The algorithms' hyperparameters' tuning is closely following the CNM06 paper with minor variation. And the performance metrics used in this paper are part of the choice from CNM06.

2.1.1 Logistic Regression

For logistic regression, I use different penalties, L1, L2 and none, going through every available solver choosing from “saga”, “lbfgs”, “sag”, “newton-cg” (Scikit-learn.org., 2021). More than this, the C parameters are ranging from 10^-8 to 10^4, which are the same C list as CNM06 used (Caruana & Niculescu-Mizil, 2006).

2.1.2 SVMs

SVMs hyperparameter tuning in this paper rigorously follows CNM06’s method (Caruana & Niculescu-Mizil, 2006). I train it with different kernel options, including linear, polynomial, and radial. For linear, I go through different Cs ranging from 10^-7 to 10^3. For a polynomial kernel, I set the polynomial degree to 2 and 3, and go through the same C list as the linear kernel does. For the radial basis function, besides searching through the same C list, I tune the algorithm with width, gamma values {0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 2}.

2.1.3 KNN

I use 26 different neighbor values for ranging from K = 1 to K = 101 with gap equals 4. Being different from paper, KNN in this paper uses uniform weight and weighted weights based on Euclidean distance, which is a simplified version compared to CNM06’s KKN manipulation.

2.2 Performance Metrics

Simplified from CNM06, this paper only examines 3 performance metrics: Accuracy, ROC-AUC, and F1. In this experiment, these performance metrics are used both in the selection of the best model in each 5-fold cross validation, and the training set and test set performance measures across 5 trials for the best models. All 3 metrics are numerically ranging from 0 to 1. Higher the numeric value, better the algorithms’ performance.

2.3 Data Sets

Four data sets are used in this experiment, including ADULT, LETTER.A, COVTYPE, and MAGIC, and all of them are from the UCI repository database (“UCI Machine Learning Repository,” 2021). When cleaning the data, I transform all data sets to binary classification problems even if they may be used to be multi-class problems. For the ADULT data set, one hot encoding method is used to transform nominal labels into 0 and 1. The LETTER.A data set regards A to M as positive class, and the rest as negative class, which is in the same way as CNM06 manipulated. The COVTYPE treats the largest class as positive and the rest as negative, which is also how CNM06 did (Caruana & Niculescu-Mizil, 2006). The MAGIC is a unique data set that CNM06 paper did not use, and it is originally a binary classification with all entries numerical. No further cleaning is required for the MAGIC. Table 1 shows the detailed information for each data set. Specifically, 14/108 stands for 14 original attributes for each data point and 108 attributes after One-Hot encoding.

Data set	Attributes	Train size	Test size	Positive rate
ADULT	14/108	5000	27561	24%
LETTER.A	16	5000	15000	50%
COVTYPE	54	5000	576012	49%
MAGIC	10	5000	14020	65%

Table 1. Detailed datasets information

3 Experiment & Results

This experiment has 3 algorithms, 4 data sets, and 3 performance metrics running through 5 trials in total. For each data set, a fixed number of 5000 data points are randomly selected and set to a training set, whereas the rest are treated as a testing set. Also, each randomized partition of the data set has a unique random state explicitly marked to ensure that difference processes within each trial are using the same set of training and testing data combination. After splitting the data into training and testing sets, 5-fold cross validation is used to select the best model parameters determined by each performance metrics. This process goes through each algorithm and data set combination in each trial, for every different performance metrics. After I have the best model parameters, these models with the best parameters refit into the training set respectively to get the training performance, and then, predict the testing set respectively to get the testing performance.

The best model winning the testing score will be marked in bold font. Paired sample t-test is utilized to compare the statistical difference between the best algorithm in each column and the rest (See table 2 and 3). Precisely, two different sets of t-test are done for Algorithm/Metric combination and Algorithm/Data set combination. P-value threshold is set to 0.05. If no significant difference is caught by the t-test, the algorithm score will be marked by *. Other entries with no bold-face and no * indicate that the algorithm performs significantly worse than the best algorithm in each column.

Table 2 shows the testing set scores average over data sets and 5 trials for each model and performance metric combination, meaning that each entry is the mean value averages over 5 trials and 4 data sets. The columns of the table list the different performance metrics, and the rows list the algorithms. With no exception, SVM is the best algorithm under all performance metrics' judgement. Table 7 suggests that both logistic regression and KNN are significantly worse than SVM across all performance metrics, given p=0.05. Even if I set p=0.01, the result will not change. The last column is the MEAN column, of which the value is mean over three metrics, demonstrating the overall performance of its corresponding algorithm. SVM has the highest performance score in general.

Model/Metrics	ACC	ROC_AUC	F1	MEAN
LOG	.780	.748	.747	.758
SVM	**.869**	**.838**	**.829**	**.845**
KNN	.849	.759	.801	.803

Table 2. Average Testing set performance over 5 trials with metrics as columns.

Table 3 shows the average testing set performance over 5 trials and all performance metrics, meaning that each entry contains the mean value averages over 5 trials and 3 performance metrics. The rows are different classifiers, and the columns register different datasets. It demonstrates quite the same result as Table 2 does. SVM is still the best algorithm for each model/data set combination. Table 8 indicates that most logistic regression and KNN show a significantly lower performance score than SVM. But one exception is that, under the ADULT data set column, logistic regression's performance is not significantly different from SVM, indicating that both may be the best model, and SVM as the best model may happen by chance. The MEAN column at last is the average value over all datasets representing the overall performance of each algorithm. In the MEAN column, SVM is still the best.

Model/Data	ADULT	MAGIC	LETTER.A	COVTYPE	MEAN
LOG	.755*	.795	.729	.753	.758
SVM	**.756**	**.865**	**.962**	**.798**	**.845**
KNN	.713	.826	.951	.780	.803

Table 3. Average Testing set performance over 5 trials with datasets as columns.

Table 6 shows the training set performance for each algorithm averages over 5 trials with columns listed as metrics, meaning that each entry is the mean value of training scores over 5 trials and all data sets for each model. When comparing Table 2 result and Table 6, the best performance model shifts to KNN in Table 6 rather than SVM in Table 2. Worth noting is that all training set performance for KNN is 1, meaning a perfect binary classification is conducted across all data sets. This is possible when the K is relatively small. The results generally hold true that training performance is better than testing performance with a few exceptions. Comparison between Table 4 and Table 5 indicates that, in the LETTER dataset, logistic regression's training performance is averagely worse than its testing performance. However, the difference may not be significant.

4 Discussion

The result is consistent with CNM06 (Caruana & Niculescu-Mizil, 2006) as SVMs are generally the best model, logistic regression is the worst, and KNN sits in the middle. And this is literally true for algorithms' average learning performance measured across datasets, or across performance metrics in this small-scale experiment, with few acceptable exceptions. This means that some algorithms are performing better in a domain-general sense. And in this study, SVM is no doubt the best domain-general classifier compared to the other two.

The exceptions found in my analysis worth noting. As aforementioned, the logistic regression performance is exceptionally well compared to SVM in the case of learning the ADULT dataset. The ADULT dataset is quite imbalanced featuring only 24% positive cases. Upsampling and downsampling techniques may apply to improve the results for all algorithms' performance, whereas in this study, none of them is applied. After these techniques are implemented, SVM may again be better than logistic regression in a statistically significant manner, but this point requires further investigation.

5 Appendix

MDMET	ACC				ROC_AUC				F1				MEAN
	ADULT	MAGIC	LETTER	COVTYPE	ADULT	MAGIC	LETTER	COVTYPE	ADULT	MAGIC	LETTER	COVTYPE	
LOG	.848	.791	.726	.754	.761	.747	.728	.754	.656	.848	.732	.752	.758
SVM	**.850**	**.865**	**.962**	**.798**	**.762**	**.829**	**.962**	**.800**	**.656**	**.902**	**.962**	**.797**	**.845**

KNN	.831	.830	.953	.781	.717	.767	.947	.780	.592	.880	.953	.779	.803

Table 4. Detailed Average Testing set performance over 5 trials.

MDMET	ACC				ROC_AUC				F1				MEAN
	ADULT	MAGIC	LETTER	COVTYPE	ADULT	MAGIC	LETTER	COVTYPE	ADULT	MAGIC	LETTER	COVTYPE	
LOG	.854	.793	.724	.762	.765	.748	.724	.762	.669	.850	.726	.760	.761
SVM	.855	.876	**1**	.871	.768	.854	**1**	.910	.667	.909	**1**	.871	.882
KNN	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

Table 5. Detailed Average Training set performance over 5 trials.

Model/Metrics	ACC	ROC_AUC	F1	MEAN
LOG	.783	.750	.751	.761
SVM	.901	.883	.862	.882
KNN	1	1	1	1

Table 6. Average Training set performance over 5 trials

Model/ Metrics	ACC	ROC_AUC	F1
LOG	0.000295	0.000299	0.000725
SVM	--	--	--
KNN	3e-7	0.000744	0.000016

Table 7. P-value for Testing set performance over 5 trials: Model/Metric.

Model/Data	ADULT	MAGIC	LETTER.A	COVTYPE
LOG	0.442678	3e-12	8e-28	2e-18
SVM	--	--	--	--

KNN	0.000001	4e-7	1e-8	2e-15

Table 8. P-value for Testing set performance over 5 trials: Model/Data.

Testing Set Raw Data in Trial 1												
	ACC				ROC_AUC				F1			
	ADULT	MAGIC	LETTR	COVTY	ADULT	MAGIC	LETTR	COVTY	ADULT	MAGIC	LETTR	COVTY
LOG	.848	.791	.727	.753	.763	.745	.728	.754	.657	.849	.733	.753
SVM	.849	.867	.961	.801	.762	.830	.961	.803	.655	.903	.960	.802
KNN	.831	.831	.953	.782	.711	.770	.948	.783	.587	.880	.952	.781
Testing Set Raw Data in Trial 2												
	ACC				ROC_AUC				F1			
	ADULT	MAGIC	LETTR	COVTY	ADULT	MAGIC	LETTR	COVTY	ADULT	MAGIC	LETTR	COVTY
LOG	.847	.793	.724	.753	.759	.749	.725	.754	.657	.849	.725	.751
SVM	.848	.864	.961	.799	.769	.829	.961	.799	.662	.901	.960	.798
KNN	.827	.831	.951	.780	.711	.770	.946	.779	.585	.880	.951	.779
Testing Set Raw Data in Trial 3												
	ACC				ROC_AUC				F1			
	ADULT	MAGIC	LETTR	COVTY	ADULT	MAGIC	LETTR	COVTY	ADULT	MAGIC	LETTR	COVTY
LOG	.850	.794	.724	.754	.760	.751	.726	.754	.655	.849	.728	.752
SVM	.852	.869	.961	.797	.757	.832	.961	.798	.653	.905	.961	.794
KNN	.832	.832	.952	.780	.719	.772	.944	.779	.588	.881	.951	.775
Testing Set Raw Data in Trial 4												
	ACC				ROC_AUC				F1			
	ADULT	MAGIC	LETTR	COVTY	ADULT	MAGIC	LETTR	COVTY	ADULT	MAGIC	LETTR	COVTY
LOG	.848	.793	.730	.755	.762	.748	.731	.755	.657	.849	.736	.754

SVM	.849	.866	.965	.799	.759	.830	.965	.802	.653	.903	.965	.796
KNN	.831	.829	.954	.784	.718	.763	.947	.782	.605	.879	.953	.780
Testing Set Raw Data in Trial 5												
	ACC				ROC_AUC				F1			
	ADULT	MAGIC	LETTR	COVTY	ADULT	MAGIC	LETTR	COVTY	ADULT	MAGIC	LETTR	COVTY
LOG	.848	.787	.728	.754	.761	.741	.730	.755	.655	.845	.736	.752
SVM	.850	.859	.964	.796	.762	.823	.964	.797	.658	.898	.964	.795
KNN	.830	.828	.957	.781	.723	.759	.949	.777	.597	.879	.957	.778

Table 8. Raw Data

References

Caruana, Rich, and Alexandru Niculescu-Mizil. "An empirical comparison of supervised learning algorithms." Proceedings of the 23rd international conference on Machine learning. 2006.

King, Ross D., Cao Feng, and Alistair Sutherland. "Statlog: comparison of classification algorithms on large real-world problems." Applied Artificial Intelligence an International Journal 9.3 (1995): 289-333.

Scikit-learn.org. 2021. Sklearn.Svm.SVC — Scikit-Learn 0.24.1 Documentation. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

Scikit-learn.org. 2021. sklearn.neighbors.KNeighborsClassifier — Scikit-Learn 0.24.21 Documentation. [online] Available at: https://scikitlearn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

Scikit-learn.org. 2021. sklearn.linear_model.LogisticRegression — Scikit-Learn 0.24.1 Documentation. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Scikit-learn.org. 2021. sklearn.model_selection.GridSearchCV — Scikit-Learn 0.24.1 Documentation. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

Archive.ics.uci.edu. 2021. UCI Machine Learning Repository. [online] Available at: https://archive.ics.uci.edu/ml/index.php [Accessed 18 March 2021].