A Comparison of Supervised Learning Classification Algorithms

Supratik Banerjee
University of California San Diego

Abstract

This research project explores various supervised learning algorithms. We do so by comparing the performance of different classifiers on various datasets obtained from the UCI repository. 5 classifiers were selected for the above-mentioned investigation namely - Logistic Regression, Support Vector Machines (SVM), Naive Bayes, Gradient Boosting, and Decision Tree. The study adheres to the methodology outlined in the Empirical Comparison of Supervised Learning Algorithms (Caruana & Niculescu-Mizil 2006). The datasets are subjected to cross-validation, with each classifier trained and tested across different partitions. The primary metric for evaluation is classification accuracy, aligning with the empirical study by Caruana and Niculescu-Mizil. The research not only seeks to compare the performance of different classifiers on each dataset but also investigates the impact of varying training data sizes on test accuracy.

1. Introduction

The sudden outburst in AI (Artificial Intelligence) is sweeping not only the technology industry of the United States, but it has also permeated every facet of our global landscape, transforming industries, institutions, and daily life across both developed and developing nations. From education and research to finance, agriculture, and transportation, AI has become a major yet unstoppable force shaping the trajectory of various sectors. Gone are the days where AI applications were exclusive to a select group of highly educated mathematicians who employed it for intricate tasks. The current landscape paints a radically different picture. AI has transitioned from being a rare tool to an integral and commonplace element of our interconnected world.

A well-trained artificial intelligence model can do many tasks better than humans can. This creates a push in society to use such tools to replace manpower. As AI continues to evolve and embed itself into the fabric of our society, an understanding of the underlying algorithms that drive such powerful tools becomes indispensable for harnessing the full potential of these technologies. A major class of such underlying algorithms that power generative AI are classifiers from supervised machine learning. In this paper, we evaluate 5 distinctly different supervised learning classifying algorithms. We use binary classification for the target. We do so by measuring the accuracy score as the primary comparison metric of each algorithm from 4 distinctly different multi-fielded datasets from the UCI repository. We ensure that the data is large, shuffled, scaled, properly partitioned and cross-validated to ensure minimization of random biasness. We further ensure validity of our accuracy score by hyperparameter tuning, high number of iterations, and averaging the results of multiple trials. Our final result is a comparison table of the 5 classifiers with their respective accuracy scores, cross-validation scores and, optimal hyperparameters for the 4 datasets.

2. Methodology

2.1 Data and Data preparation

Adult (Becker & Kohavi, 1996)

This dataset, taken from the UCI repository, contains census data of American adults with 14 features such as age, sex, education level, occupation, marital status and many more. We have used this data to predict whether yearly income exceeded 50,000 USD or it did not. Since the target has binary labels, it was used as is without any merges and we used income exceeding 50k as positive and not exceeding 50k as negative. All categorical data features were converted to numerical data using one-hot encoding. The categorical data converted involved the features with labels: “workclass”, “occupation”, “race”, “native-countries”. The features excluded in our study were with labels: “marital-status”, “education” (This is the categorical data for education level which was included), “sex”, “relationship”. All feature data was then normalized. The data was cleaned by discarding the rows with missing data values. After cleaning, all the datapoints were used. The Final input data had 32560 points and 10 features.

Cover-type (Blackard, 1998)

This dataset, taken from the UCI repository, contains cartographic data to predict forest cover types with 52 features such as elevation, aspect, slope and many more. Since the prediction target: cover type is multiclass with labels Spruce/Fir, Lodgepole Pine, Ponderosa Pine, Cottonwood/Willow, Aspen, Douglas-fir, Krummholz, we used the largest class as positive and the rest as negative. There was no categorical data in this dataset. All feature data was normalized. No features were excluded in this experiment. No data cleaning was needed as there were no missing values. Since there were over 500000 data points, we shuffled the dataset and chose 50000 points. Please do note that only 30000 points were chosen in the empirical study paper. The Final input data had 50000 points and 52 features.

Letter Recognition (Slate, 1991)

This dataset, taken from the UCI repository, contains statistical moments and edge counts extracted from images of handwritten capital letters of the 26 English alphabets. There are a total of 16 features such as high, width, x-bar, y-bar, and many more. Since the prediction target: the letter is multiclass with labels A to Z, we used the A-M as positive and the rest as negative. There was no categorical data in this dataset. All feature data was normalized. No features were excluded in this experiment. No data cleaning was needed as there were no missing values. All data points were used. The Final input data had 20000 points and 16 features.

Dry Bean (Dry Bean Dataset., 2020)

This dataset, taken from the UCI repository, contains numerical image data extracted from images of 13,611 grains of 7 different registered dry beans. There are a total of 16 features such as area, perimeter, Eccentricity, and many more. Since the prediction target, the letter is multiclass with labels corresponding to the names of the beans - Seker, Barbunya, Bombay, Cali, Dermosan, Horoz, and Sira, we used the largest class as positive and the rest as negative. There was no categorical data in this dataset. All feature data was normalized. No fields were excluded in this experiment. No data cleaning was needed as there were no missing values. All data points were used. The Final input data had 13611 points and 16 fields.

2.1 Classifiers

Support Vector Machine

We used sklearn.svm.LinearSVC from the sci-kit learn library (Pedregosa, et al., 2011). During experimentation we varied the hyperparameter by factors of 10 from 10^{-1} to 10^{2}. The max iterations were set to 10000. The other parameters were set to default. The classifier accuracy score was calculated as 1- error score.

Logistic Regression

We used sklearn.linear_model.LogisticRegression from the sci-kit learn library (Pedregosa, et al., 2011). During experimentation we varied the hyperparameter by factors of 10 from 10^{-1} to 10^{2}. The max iterations were set to 10000. The other parameters were set to default. The classifier accuracy score was calculated as 1- error score.

Naïve Bayes

We used gaussian naïve bayes algorithm with sklearn.naive_bayes.GaussianNB from the sci-kit learn library (Pedregosa, et al., 2011). The max iterations were set to 10000. The other parameters were set to default. The classifier accuracy score was calculated as 1- error score.

Decision Tree

We used sklearn.tree.DecisionTreeClassifier from the sci-kit learn library (Pedregosa, et al., 2011). The max iterations were set to 10000. Max_depth and splitter, and min_samples_split, min_samples_leaf and other parameters were left at default. The classifier accuracy score was calculated as 1- error score.

Gradient Boosting

Please do note that this classifier was not used in the empirical study paper amd was done as a bonus classifier. We used sklearn.ensemble.GradientBoostingClassifier from the sci-kit learn library (Pedregosa, et al., 2011). The max iterations were set to 10000. All parameters were left at default. The classifier accuracy score was calculated as 1- error score.

2.3 General experimentation details

After the data was finalized as noted in section 2.1 Data and Data preparation, each data set was split into 3 sets of partitions: 20% Training data - 80 % Test data, 50% Training data - 50 % Test data, 80% Training data - 20 % Test data. Each of these three sets were fed into the classifiers to get the classifier accuracy score.

Furthermore, we used sklearn.model_selection. cross_val_score (Pedregosa, et al., 2011) to calculate cross validation score. For each partition, 3 trials were run and their accuracy score, cross-validation score, cross validation standard deviation were averaged respectively to reduce random data selection error.

Additionally, each of these experiments were run 4 times for different hyperparameter guesses from 10^{-1} to 10^{2} (increased by a factor of 10). We then compared the accuracy scores to select the best hyperparameter which was used for in the final data table and graph.

3. Experimentation and results

Using the above-mentioned methodology (section 2.3), the experiment was carried out the following number of times. 5 (number of classifier) * (4 number of data sets) * 3 (number of partitions) * 3 (number of trials) * 4 (number of hyper parameters {0.1,1,10,100}). We carried out further experimentations with different max iteration values from 10^{2} to 10^{5} and found diminishing results to the improvement of accuracy score and increasing compute time hence those results were discarded, and all the recorded results were carried out with 10^{4} iterations. The experimentation was also carried out with unscaled data, however, due to low accuracy and low cross validation score, those records were discarded as well and only the results with normalized data was recorded.

3.1 Hyperparameter Tuning

We use table 1. to compare and the optimize hyper parameters for logistic regression and SVM. Please note that this is not the final result table hence cross validation score, and accuracy score for other classifiers have been omitted for readability.

		20/80	50/50	80/20	20/80	50/50	80/20
		SVM			LogReg		
	c = 0.1	0.768	0.764	0.764	0.767	0.762	0.762
	c=1	0.785	0.779	0.773	0.78	0.773	0.746
	c = 10	0.792	0.79	0.789	0.791	0.787	0.782
Adult	c = 100	0.793	0.791	0.793	0.791	0.79	0.789
	c = 0.1	0.559	0.559	0.557	0.571	0.571	0.56
	c=1	0.585	0.577	0.573	0.587	0.587	0.577
	c = 10	0.594	0.596	0.594	0.589	0.594	0.591
Cover Type	c = 100	0.596	0.595	0.594	0.591	0.593	0.592
	c = 0.1	0.743	0.74	0.739	0.743	0.74	0.739
	c=1	0.743	0.74	0.739	0.743	0.74	0.739
	c = 10	0.743	0.74	0.739	0.743	0.74	0.739
Dry Bean	c = 100	0.879	0.824	0.739	0.826	0.777	0.739
	c = 0.1	0.707	0.707	0.692	0.706	0.706	0.694
	c=1	0.724	0.724	0.72	0.719	0.719	0.71
	c = 10	0.727	0.73	0.728	0.723	0.723	0.724
Letters	c = 100	0.728	0.73	0.727	0.722	0.724	0.724

Table 1. Experiment output with C for SVM and LogReg.

Table 1. shows the classifier scores (accuracies) for SVM and Logistic Regression for the three partition sets for the 4 data sets. We used this table to compare accuracies. For each partition, we chose the hyperparameter C for the highest accuracy. For example, in the 20/80 partition from adult dataset, score = 0.768, c =0.1; score = 0.785, c =1; score = 0.792, c =10; score = 0.793, c =100. Since highest score is 0.793, the optimal hyperparameter here is 100 for adult data set's 20/80 partitioned data when SVM classifier was used. We repeated this process for all the data shown in table 1 to come up with optimal hyperparameters for each data set's partitions for each classifier used. The results are shown in the table 2. below

	20/80	50/50	80/20	20/80	50/50	80/20
	SVM			LogReg		
Adult	100	100	100	100	100	100
cover type	100	10	100	100	10	100
Dry Bean	100	100	100	100	100	100
letters	100	100	10	10	100	10

Table 2. Optimal Hyperparameters.

3.2 Final Results

After choosing the optimal hyperparameters, we discarded the data for the non-optimal hyperparameters and collated all the classifier accuracies, cross validation scores and cross validation standard deviations for the 4 datasets for each of the 3 partitions using the 5 classifiers onto tables 3.* as shown below. Please note that 20/80 on the top row means the data has been partitioned to 20 percent training 80 % testing.

		20/80	50/50	80/20
		SVM		
	Classifier Accuracy	0.793	0.791	0.793
	Hyper Parameter	100	100	100
	Cross Validation Score	0.793	0.795	0.791
Adult	Cross Validation s.d.	0.002	0.002	0.004
	Classifier Accuracy	0.596	0.594	0.594
	Hyper Parameter	100	10	100
	Cross Validation Score	0.596	0.597	0.592
Cover Type	Cross Validation s.d.	0.004	0.004	0.009
	Classifier Accuracy	0.879	0.824	0.739
	Hyper Parameter	100	100	100
	Cross Validation Score	0.857	0.789	0.742
Dry Bean	Cross Validation s.d.	0.004	0.004	0.004
	Classifier Accuracy	0.728	0.73	0.728
	Hyper Parameter	100	100	10
	Cross Validation Score	0.733	0.731	0.733
Letters	Cross Validation s.d.	0.007	0.009	0.009

Table 3.1 SVM Final Experiment Output

		20/80	50/50	80/20
		Naïve Bayes		
	Classifier Accuracy	0.788	0.785	0.787
	Hyper Parameter	/	/	/
	Cross Validation Score	0.786	0.785	0.781
Adult	Cross Validation s.d.	0.005	0.006	0.01
	Classifier Accuracy	0.645	0.648	0.646
	Hyper Parameter	/	/	/
	Cross Validation Score	0.645	0.646	0.643
Cover Type	Cross Validation s.d.	0.005	0.002	0.007
	Classifier Accuracy	0.898	0.892	0.893
	Hyper Parameter	/	/	/
	Cross Validation Score	0.893	0.896	0.901
Dry Bean	Cross Validation s.d.	0.005	0.008	0.006
	Classifier Accuracy	0.699	0.699	0.7
	Hyper Parameter	/	/	/
	Cross Validation Score	0.697	0.697	0.703
Letters	Cross Validation s.d.	0.006	0.01	0.018

Table 3.2 Naïve Bayes Final Experiment Output

		20/80	50/50	80/20
		Log Reg		
	Classifier Accuracy	0.791	0.79	0.789
	Hyper Parameter	100	100	100
	Cross Validation Score	0.791	0.793	0.788
Adult	Cross Validation s.d.	0.003	0.002	0.004
	Classifier Accuracy	0.591	0.589	0.592
	Hyper Parameter	100	10	100
	Cross Validation Score	0.593	0.594	0.591
Cover Type	Cross Validation s.d.	0.005	0.003	0.009
	Classifier Accuracy	0.826	0.777	0.739
	Hyper Parameter	100	100	100
	Cross Validation Score	0.806	0.759	0.742
Dry Bean	Cross Validation s.d.	0.004	0.004	0.004
	Classifier Accuracy	0.723	0.724	0.724
	Hyper Parameter	10	100	10
	Cross Validation Score	0.726	0.726	0.729
Letters	Cross Validation s.d.	0.007	0.008	0.013

Table 3.3 Logistic Regression Final Experiment Output

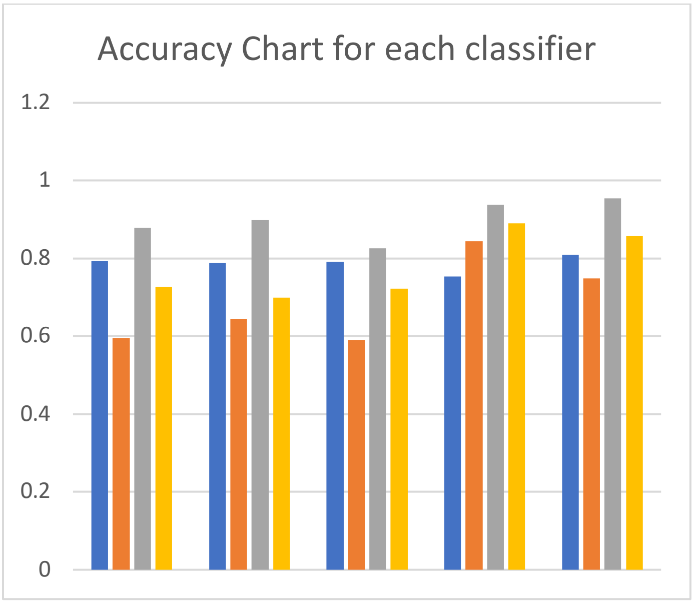
		20/80	50/50	80/20
		Decision Trees		
	Classifier Accuracy	0.754	0.749	0.747
	Hyper Parameter	/	/	/
	Cross Validation Score	0.753	0.752	0.739
Adult	Cross Validation s.d.	0.004	0.006	0.008
	Classifier Accuracy	0.844	0.845	0.77
	Hyper Parameter	/	/	/
	Cross Validation Score	0.829	0.833	0.758
Cover Type	Cross Validation s.d.	0.003	0.002	0.01
	Classifier Accuracy	0.938	0.94	0.938
	Hyper Parameter	/	/	/
	Cross Validation Score	0.94	0.936	0.939
Dry Bean	Cross Validation s.d.	0.003	0.005	0.009
	Classifier Accuracy	0.89	0.875	0.843
	Hyper Parameter	/	/	/
	Cross Validation Score	0.886	0.87	0.836
Letters	Cross Validation s.d.	0.008	0.01	0.013

Table 3.4 Decision Tree Final Experiment Output

		20/80	50/50	80/20
		GradBoost		
Adult	Classifier Accuracy	0.809	0.806	0.806
	Hyper Parameter	/	/	/
	Cross Validation Score	0.808	0.809	0.805
	Cross Validation s.d.	0.003	0.003	0.005
Cover Type	Classifier Accuracy	0.748	0.748	0.756
	Hyper Parameter	/	/	/
	Cross Validation Score	0.749	0.752	0.757
	Cross Validation s.d.	0.005	0.006	0.006
Dry Bean	Classifier Accuracy	0.954	0.953	0.951
	Hyper Parameter	/	/	/
	Cross Validation Score	0.954	0.954	0.953
	Cross Validation s.d.	0.004	0.005	0.011
Letters	Classifier Accuracy	0.858	0.854	0.849
	Hyper Parameter	/	/	/
	Cross Validation Score	0.856	0.855	0.847
	Cross Validation s.d.	0.004	0.01	0.009

Table 3.5 Gradient Booster Final Experiment Output

3.3 Interpretation of results



Graph 1. Visualization of results (20/80 partition)

To Interpret the results, we can take a look at 1 of the partitions such as 20/80 as shown in the histogram above(graph 1.). Each color represents a different dataset (in the order of Adult, cover type, dry beans, and letter) while each group of bars represents a classifier as SVM, NB, LogReg, DT and Grad Boost going from left to right. The Y axis represents the accuracy score. We can interpret a few things such as the orange dataset, that is the cover type data has been consistently performing the poorest among the rest regardless of any classifier used. Furthermore Decision tree and Gradient booster performed the best in the task of binary classifying the given data sets, while Logistic Regression performed the worst. It is also to be noted that the accuracy scores are consistently matching those done in the empirical comparison paper.

4. Conclusion

In conclusion, this research project delves into the comparative analysis of five distinct classifiers—Gradient Boosting, Naïve Bayes, Decision tree, Support Vector Machines (SVM), and Logistic Regression—across three datasets sourced from the UCI repository. The study rigorously follows the methodology articulated by Caruana and Niculescu-Mizil, with a focus on replicating consistent results.

The cross-validation process, conducted on different partitions (20/80, 50/50, 80/20), provides valuable insights into the performance of each classifier under varied training and testing conditions. The primary evaluation metric, classification accuracy, aligns with the empirical study by Caruana and Niculescu-Mizil, ensuring a direct comparison of results. Notably, the research extends beyond algorithmic performance comparisons by exploring the influence of training data sizes on test accuracy—a crucial aspect often overlooked in classifier assessments.

5. Bonus

This investigation delves into machine learning in much more depth than required by the assignment. Firstly, large datasets were used. For all the data sets all the possible data points were used (besides Covertype as there were over 500 thousand data points). Furthermore, I used 1 extra data set than the requirement. 2 extra classifiers were also used with 1 of them being a novel classifier in terms of this course as it was not taught in class. Nor was it used in the provided paper. Lastly, every effort has been made to explain each data input and out put through proper explanations, excessive experimentation(during the calculation of hyperparameters) and visual aids such as tables and graphs. Hence I believe that I am deserving of bonus points as I went above and beyond the required scope.

6. References

Becker,Barry and Kohavi,Ronny. (1996). Adult. *UCI Machine Learning Repository*. https://doi.org/10.24432/C5XW20.

Blackard,Jock. (1998). Covertype. *UCI Machine Learning Repository.* https://doi.org/10.24432/C50K5N.

Caruana, R., & Niculescu-Mizil, A. (2006). Predicting good probabilities with supervised learning. *Proc. 23nd International Conference on Machine Learning* (ICML'07).

Dry Bean Dataset. (2020). *UCI Machine Learning Repository*. https://doi.org/10.24432/C50S4B.

Pedregosa, Fabian., et al. (2011). *Scikit-learn: Machine Learning in Python*. https://scikit-learn.org.

Slate,David. (1991). Letter Recognition. *UCI Machine Learning Repository*. https://doi.org/10.24432/C5ZP40.