Binary Classification Model Selection

Alexa Hurtado amhurtad@ucsd.edu
University of California, San Diego

Abstract

This paper attempts to replicate the results of Niculescu-Mizil and Caruana's 2006 paper on Model Selection through the implementation of three supervised machine learning models on binary classification across four different datasets. The algorithms used across datasets were logistic regression, k-nearest neighbors and decision trees. After implementation of hyperparameter tuning through the use of grid search and k-fold cross validation, each algorithm's performance is assessed on three different metrics of accuracy, ROC Area and F-score. Findings seem to coincide with those of Caruana's and Niculescu-Mizil in that k-nearest neighbors tend to outperform both logistic regression and decision trees, and between decision trees and logistic regression, decision trees tend to exceed logistic regression performance.

1. Introduction

In their paper, Caruana and Niculescu-Mizil assess the performance of ten machine learning algorithms including SVMs, neural nets, logistic regression, naive bayes, memory-based learning, random forests, decision trees, bagged trees, boosted trees and boosted stumps. They analyzed their performance through implementation across eleven datasets and assessed them through metrics including but not limited to accuracy, precision and ROC area.

Their motivation comes from extending model selection research in machine learning. Previous studies such as had analyzed different models and assessed their performance but Caruana and Niculescu-Mizil's research further delve into the topic by this time around including new algorithms that have since come out.

By analyzing these algorithms through a variety of different metrics, it allows for a better understanding of each model's strengths and weaknesses and a more in depth insight on what model definitively outperforms the rest.

Caruana and Niculescu-Mizil found that out of all algorithms, bagged trees, random forests and neural nets performed the best when averaged across metrics and datasets. Among their results they also concluded that k-nearest neighbors outperform both decision trees and logistic regression, and decision trees outperform logistic regression.

I will attempt to replicate their findings when analyzing the performance of k-nearest neighbors, decision trees and logistic regression. By analyzing each model across four datasets and assessing with metrics of accuracy, ROC score and F-score, I will try to gain a better insight on the performances of each algorithm and compare them.

2. **Methodology**

2.1 Learning Algorithms

For each dataset and algorithm combination, training data was split into five trials where each trial consisted of 5000 different randomized samples from the dataset. These were then put through 5 fold cross validation and through use of grid search hyperparameter tuning I found the best models corresponding to each dataset. The grid search space for each algorithm was set up as such:

Logistic Regression: Hyperparamter tuning involved exploring three different models:
In the first model the solver was set to 'saga' with penalty either 'l1' or 'l2', and C values ranging from 10^{-8} to 10^{4} with a step size of 10x. In the second model we explore a solver value of 'lbfgs' with penalty 'l2' and C value remains the same as the first. In the third model the solver was set to either 'lbfgs' or 'saga' with penalty 'none' and no C values.
K-nearest Neighbors: Hyperparameter tuning involved only varying the n_neighbors argument ranging from 1 to 105 with step size of 4.
Decision Tree: Hyperparameter tuning involved varying criterion value between 'gini' and 'entropy', max_depth ranged from 1 to 100 with a step size of 3, and min_samples_leaf was also set with a range from 10 to 100 with a step size of 10.

2.2 Performance Metrics

All algorithms were assessed with metrics of accuracy score, ROC AUC score and F1 score. Within each trial, the best performing model parameters for each metric were extracted using grid search, resulting in 5 values per metric and 15 overall values per algorithm. These parameters were then refitted onto the entire 5000 samples of training data and tested on the testing data, the remaining samples not included in the 5000.

From this I obtained 15 performance scores on both the training and test data. Depending on the metric those models corresponded to, that metric was used to score the model on the training and test data. Ultimately 5 scores per metric were acquired, 15 scores in total per algorithm and 45 scores per dataset.

2.3 Data Sets

All Datasets used were obtained from the UCI Machine Learning Repository. The ADULT Dataset consisted of 14 total attributes but after limiting the 'Country' column to only include 'United States' values, the column was dropped and features were reduced to 13. After Hot encoding categorical variables and standardizing numerical variables with the StandardScaler function of sklearn,, features were increased to 53. The target variable of 'Salary' was changed to 1 if the person made over 50K and -1 if the person did not make over 50K. The BEAN Dataset consisted of 16 total numerical features which were all standardized, there were in total 7 classes of beans but in order to make this a binary problem, classes 'DERMASON' and 'SIRA' were changed to values of 1 and the rest of the classes to -1. OCCUPANCY Dataset consisted of 6 different numerical features including room temperature and humidity, with values of 1 in the target variable if the room was occupied and -1 if the room was not occupied. Lastly the EEG Dataset consisted of 14 different EEG features with target variables of 1 and -1, determining whether a person's eyes were open or closed.

Table 1. Problem Set Summary

Name	#ATTR	TRAIN SIZE	TEST SIZE	%POZ
ADULT	13/53	5000	15853	25%
BEAN	16	5000	8611	45%
OCCUPANCY	6	5000	15560	23%
EEG	14	5000	9480	45%

3. Experiment & Results

3.1 Performance Across Metrics

After performing cross validation and tuning on hyperparameters via gridsearch, all metric scores were calculated and resulted in a total 5 accuracy scores, 5 ROC AUC scores and 5 F1 scores reflecting the performance of the 5 best models from each algorithm on both the training and test set. Test set averages of metric scores across datasets are reflected in Table 2, where each algorithm's respective metric score is the average of all 20 scores of that metric across all datasets.

Table 2. Main Matter — Test Set Performance by Metric

Model	ACC	ROC AUC	F1	MEAN
LR	.855	.829	.776*	.820
kNN	**.902**	**.871**	**.835**	**.869**
DT	.887	.860	.817*	.855

After calculating averages, we see that k-nearest neighbors outperforms both decision trees and logistic regression when predicting on test set, and after performing ttest_rel to test for significance, k-nearest neighbors performance does reflect a significant difference in accuracy and ROC AUC scores but not in F1 scores, which in logistic regression resulted in a p value of .067 and in decision trees a p value of .06. This is reflected in Table 2 with insignificant values marked with * and bolded values representing the highest performing algorithm of that specific metric. Overall mean across metric for k-nearest neighbors is .869, compared to the values of .820 and .855 of logistic regression and decision trees respectively. Thus, decision trees performed second best with logistic regression in last place.

Training set averages were also obtained and are represented in Table 3. Training set values were obtained by extracting metric scores from implementing models on training sets of the 5000 samples as opposed to the test set. Training set scores also show that the highest performing algorithm was k-nearest neighbors, with an overall average of .898 across all metrics, followed by decision trees at .889 and logistic regression at .825. All average scores from training sets were higher than the average scores represented in Table 2, the test set scores.

Table 3. Main Matter — Training Set Performance per Metric

Model	ACC	ROC AUC	F1	MEAN
LR	.857	.834	.785	.825
kNN	.931	.888	.875	.898
DT	.917	.89	.861	.889

3.2 Performance Across Datasets

Next I analyzed performance across all datasets by averaging across all metrics and comparing on each different algorithm. With the exception of the ADULT Dataset, K-nearest neighbors once again resulted in the highest performing of the three algorithms. Logistic Regression and K-nearest neighbors performed the same on the BEAN dataset, both with mean .972, thus showing no significant difference there with p value .373. Other than that p value, all the rest did show that k-nearest neighbors performance was a significant difference between both logistic regression and decision trees. When averaging across all datasets, the mean of decision trees stands at .854, slightly larger than logistic regression's value of .82, and proving consistent with Table 2 results of decision tree's overall performance when comparing to logistic regression. Averaged scores across datasets can be seen in Table 4, bolded values represent the highest performing algorithm for that dataset and * indicates low significance.

Table 4. Main Matter — Test Set Performance per Dataset

Model	ADULT	BEAN	OCC	EEG	MEAN
LR	**.718**	**.972***	.986	.603	.82
kNN	.688	**.972**	**.987**	**.83**	**.869**
DT	.699	.963	.984	.771	.854

4. Conclusion & Discussion

Results indicate that k-nearest neighbors outperformed logistic regression and decision trees in both the training and test set performance. Training set performance for all algorithms was slightly higher than test set performance, which can be attributed to the fact that the model itself had been fit on the training data and would consequently have a higher performance across all metrics of accuracy score, ROC AUC and F1 score.

From these results I was able to conclude that of the three algorithms k-nearest neighbors clearly performed better than both logistic regression and decision trees across accuracy score, ROC AUC and F1 scores, as well as across all datasets with the exception of one. Furthermore I was able to replicate Caruana's and Niculescu-Mizil's results in their research on model selection. Their data found that k-nearest neighbors also outperformed logistic regression and decision trees, and that between the two, decision trees would come out the better algorithm when compared to logistic regression. For further research it would be insightful to research more algorithms and their performances but for this project I focused on only three algorithms, and the results did coincide with Caruana and Niculescu-Mizil's research.

Appendix

Code is embedded after references

Table 5: Table 2 Supplemental — p-values (p),

Models	p-LR	p-DT
KNN-ACC	.044	.043
KNN-ROC	.042	.01
KNN-F1	.067	.06

Table 6: Table 4 Supplemental — p-values(p)

Datasets (algo)	p-val
ADULT(LR)	9.27 e-7
ADULT(DT)	0.01
OCCUPANCY(LR)	.044
OCCUPANCY(DT)	.005
BEAN(LR)	.373
BEAN(DT)	6.154 e-11
EEG(LR)	1.864 e-11
EEG(DT)	5.086 e-7

Table 7: Raw Test Scores for Adult Dataset

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
LR -Accuracy	0.82	0.8184570744	0.8181416767	0.8191509493	0.8191509493
LR-ROC	0.7313480659	0.7303397857	0.7307176419	0.7303397857	0.7307176419
LR-F1	0.6062354072	0.6064004376	0.6044724928	0.6062354072	0.6062354072
KNN -Accuracy	0.7978300637	0.7950545638	0.789503564	0.7994070523	0.7970731092
KNN-ROC	0.7084236276	0.7045633649	0.7113196957	0.7145049513	0.7045633649
KNN-F1	0.5628154413	0.5546264565	0.5374913375	0.5766773163	0.561895683
DT-Accuracy	0.8137261086	0.8027502681	0.8106982905	0.8086797452	0.8109506087
DT-ROC	0.7192725942	0.7021183182	0.7286042374	0.7021183182	0.7089700742
DT-F1	0.562842339	0.5751935878	0.5974513749	0.5830927835	0.572284858

Table 8: Raw Test Scores for Bean Dataset

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
LR -Accuracy	0.9736383695	0.9736383695	0.9736383695	0.9736383695	0.9736383695
LR-ROC	0.9698803064	0.9693708634	0.9698176464	0.9743672093	0.9693708634
LR-F1	0.9713631157	0.9719887955	0.9719887955	0.9719887955	0.9701207883
KNN -Accuracy	0.9738706306	0.9744512832	0.9744512832	0.9744512832	0.9727093253
KNN-ROC	0.9730704476	0.9730704476	0.9730704476	0.9730704476	0.9730704476
KNN-F1	0.9701207883	0.9701207883	0.9701207883	0.9701207883	0.9701207883
DT-Accuracy	0.9649285797	0.9657414934	0.9657414934	0.9650447103	0.9667866682
DT-ROC	0.9643808884	0.9641504125	0.9625042224	0.9628446004	0.9578661201
DT-F1	0.9611525598	0.9621524202	0.9620187975	0.9611625514	0.962687524

Table 9: Raw Test Scores for EEG Dataset

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
LR -Accuracy	0.6366733467	0.6333667335	0.6366733467	0.6371743487	0.6366733467
LR-ROC	0.6231276137	0.6231276137	0.6231276137	0.6231276137	0.6231276137
LR-F1	0.5502356735	0.5433670286	0.5502356735	0.5505771379	0.5502356735
KNN -Accuracy	0.820741483	0.8652304609	0.8652304609	0.820741483	0.8652304609
KNN-ROC	0.8175869828	0.8175869828	0.8029711174	0.8029711174	0.8175869828
KNN-F1	0.7975557316	0.8503393791	0.8503393791	0.7975557316	0.8503393791
DT-Accuracy	0.7828657315	0.7907815631	0.7862725451	0.7805611222	0.7826653307
DT-ROC	0.7758624922	0.7688116125	0.7779214526	0.7713067593	0.7868790015
DT-F1	0.7528334287	0.7595454545	0.7554790591	0.7469740634	0.7494538347

Table 10: Raw Test Scores for Occupancy Dataset

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
LR -Accuracy	0.9900385604	0.9900385604	0.9900385604	0.9900385604	0.9900385604
LR-ROC	0.98715541	0.9914761211	0.98715541	0.9914761211	0.9914761211
LR-F1	0.9790907865	0.9790907865	0.9790907865	0.9790907865	0.9790907865
KNN -Accuracy	0.9906812339	0.9914524422	0.9906812339	0.9902313625	0.9906812339
KNN-ROC	0.9903330916	0.9903330916	0.9918264964	0.9906374606	0.9903330916
KNN-F1	0.9800137836	0.9814012026	0.9800137836	0.9791552386	0.9800137836
DT-Accuracy	0.9897172237	0.9865681234	0.9897172237	0.9857969152	0.9865681234

DT-ROC	0.9915714549	0.9915714549	0.9915714549	0.9816946093	0.9915714549
DT-F1	0.9784017279	0.9714442614	0.9784017279	0.9695046226	0.9713030345

References

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

R. Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830.

R. Dorpe, Steven Van. “Preprocessing with Sklearn: a Complete and Comprehensive Guide.” *Medium*, Towards Data Science, 13 Dec. 2018, towardsdatascience.com/preprocessing-with-sklearn-a-complete-and-comprehensive-guide-670cb98fcfb9.