No Free Lunch: Free at Last!

Ali Almashhadani[1], Neelang Parghi[2], Weihao Bi[2], and Raman Kannan[2](✉)

[1] Hunter College, City University of New York, New York, USA
[2] CSE, Tandon School of Engineering, New York University, Brooklyn, NY 10201, USA
rk1750@nyu.edu

Abstract. No Free Lunch (NFL) Theorem in M/L is rigid and inflexible, which states that "No particular classifier can outperform all the other classifiers for every dataset". In this paper we present a MISD machine that runs multiple classifiers in parallel against a given dataset, implementing a "Swiss Army Knife" to combine many different classifiers and review their performance, effortlessly. The service will be hosted as a public service over the internet for any Machine Learning practitioner to experiment with datasets.

1 Supervised Learners

Machine Learning seeks to learn from known data and apply it to never seen before data. Classification or Supervised Learning is one of the core Machine Learning tasks. In Supervised Learning, one learns to assign a label (class) given a vector of predictors. Interested readers may find summary introduction in [1] and deep introduction in [2] and there are several other classics on the subject matter [3–5], and for those interested in managing large scale machine intelligence projects [6] is an excellent source (Fig. 1).

There are many Classification algorithms as shown above and one is faced with a dilemma: Which algorithm should one use given a particular dataset?

1.1 Satisficing Solution[x]

The satisficing decision-making as discussed in [6] is a heuristic where people settle with a solution to a problem that is 'good enough' but may not be the optimal one. A "Satisficing Solution" can be considered as a vernacular description of Occam's Razor [7, 8]. The notion of Satisficing Solution does not run counter to the well known axiom "No Free Lunch Theorem" [10] in Machine Learning. In combination with the razor, a satisficing solution is good enough.

1.2 No Free Lunch (NFL) Theorem

There is considerable debate [9, 10] about NFL [11] as to its meaning and interpretation and there is even an organization dedicated to NFL [12].

D. D. Schmorrow and C. M. Fidopiastis (Eds.): HCII 2020, LNAI 12196, pp. 217–225, 2020.
https://doi.org/10.1007/978-3-030-50353-6_16

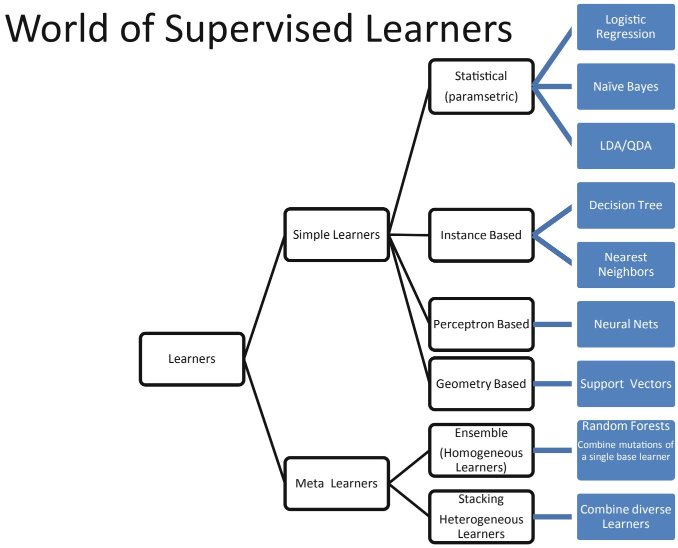


Fig. 1. Machine learning classifier tree

1.3 What Is Cost?

If it cannot be free, what is the cost? As outlined in [6] and [13], misclassification error is not the only cost. There are other costs including:

a) demand on memory,
b) processing time and
c) interpretability.

1.4 Need for Automated Algorithm Selector

In our opinion, as M/L is adopted more and more, the most impactful consideration for practitioners is that there is no single classifier can outperform in all domains. Consequently and it is imperative for practitions to ask the fundamental question posed in [14] "Among all the available classification algorithms, and in considering a specific type of data and cost, which is the best algorithm for my problem?" before settling on a particular algorithm. As the number of practitioners increase, ability to run a model will cease to be an advantage. The need for automating the algorithm selection process will become all too important and immediate. There have been several experiments comparing classifier performance [15–18], but none is available as a service to practitioners.

In this paper we will present our efforts, the Swiss Army Knife for No Free Lunch (NFL-SAK) to make lunch free for anyone with a dataset. Consistent with Occam's Razor, we allow users to submit a dataset, provide some hints to the structure of data and run several established classification algorithms of different types (parametric, instance based, logic based, ensemble and stacked-generalization). The NFL-SAK presents a useful tabulation of performance metrics. In its current form we present Area Under the Curve (AUC) [20] and Accuracy. There are several other performance metrics, see chapter 7 in Practical Data Mining [6] for a detailed overview and we plan to incorporate them in later revisions.

2 Implementation

Given a dataset, a model Formula, and a set of algorithms, NFL-SAK platform, performs a classification over the given set of algorithms. System uses readily available packages in R [21] including:

1. library (DMwR)
2. library (caret)
3. library (e1071)
4. library (pROC)
5. library (randomForest)
6. library (rattle)
7. library (rpart)

The process is intuitive as shown below (Fig. 2):

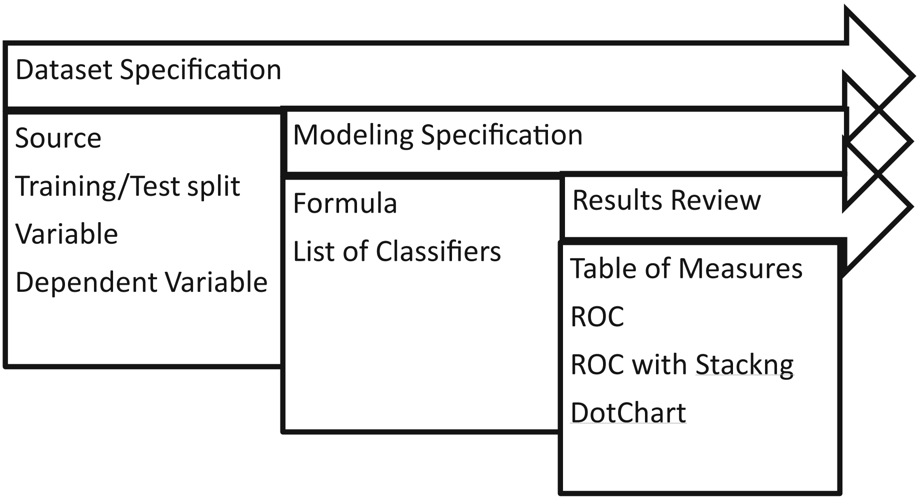


Fig. 2. NFL-SAK process and user interaction frameworks

The Shiny UI implements a "Classify By Example" model where the practitioner can specify One or more classifiers, the independent variable and the dependent variables. Consistent with Ockham's Razor, each selected classifier will be run with the simplest default model without parameter tuning and the results displayed for review (Fig. 3).

2.1 Dataset Specification

Here we have loaded the Hepatitis [29–31] dataset. We want to use 70% for training and the rest for testing.

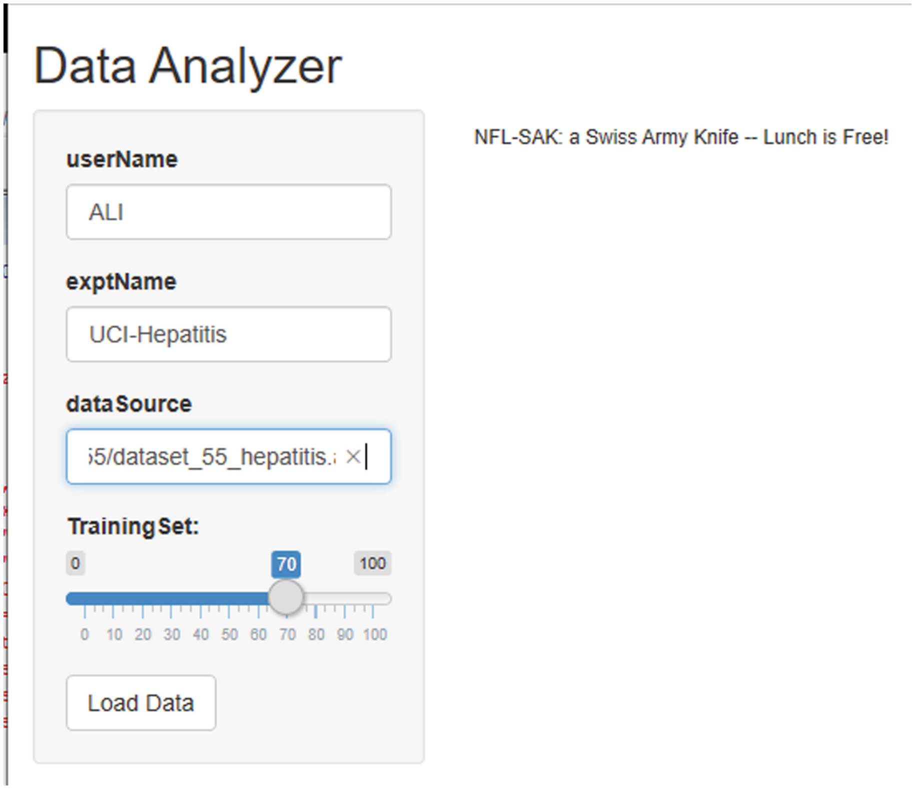


Fig. 3. Users first interaction with NFL-SAK, users name the experiment, specify the dataset.

2.2 Modeling Specification

Users can specify the model Formula and train one or more of the learners. Here we have identified the class variable and the list of learners we want to evaluate. Note that one parametric classification (Logistic Regression), one instance based classifier (kNN), one logic based classifier (Decision Tree) and a Support Vector Machine alongside RandomForest (an Ensemble classifier is given. Stacking with voting is also run by default) (Fig. 4).

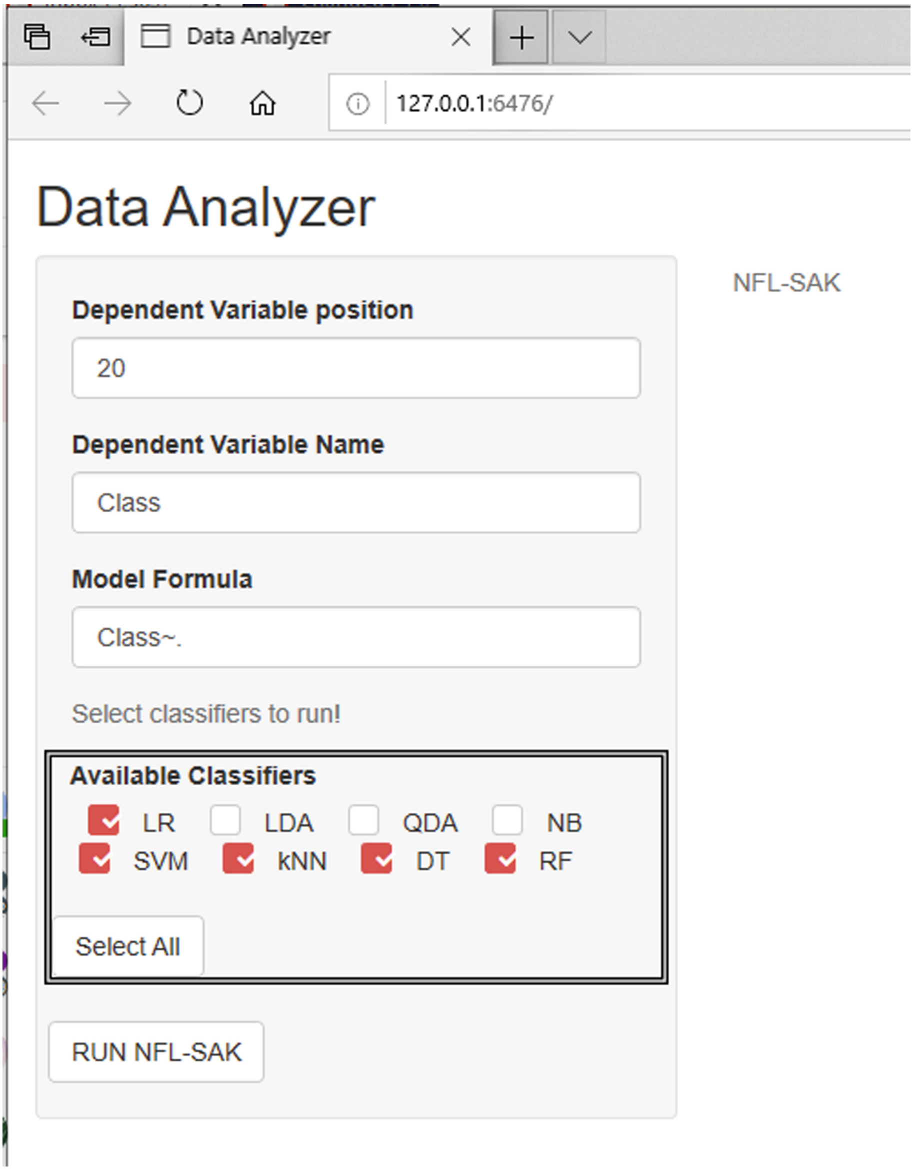


Fig. 4. Experimental design specification

Now we will run the model and review the results.

2.3 Model Output

First numeric performance measures including Accuracies and AUC are presented (Figs. 5 and 6).

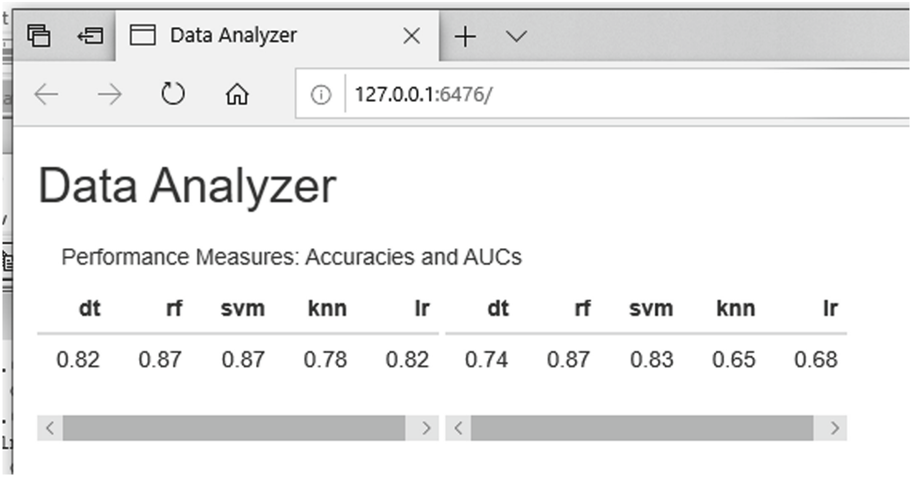


Fig. 5. Table of numeric performance metrics.

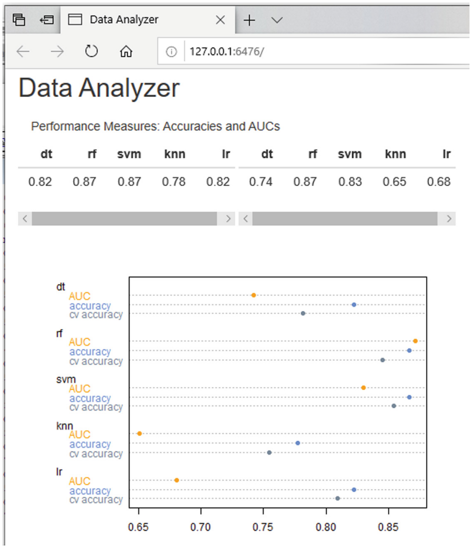
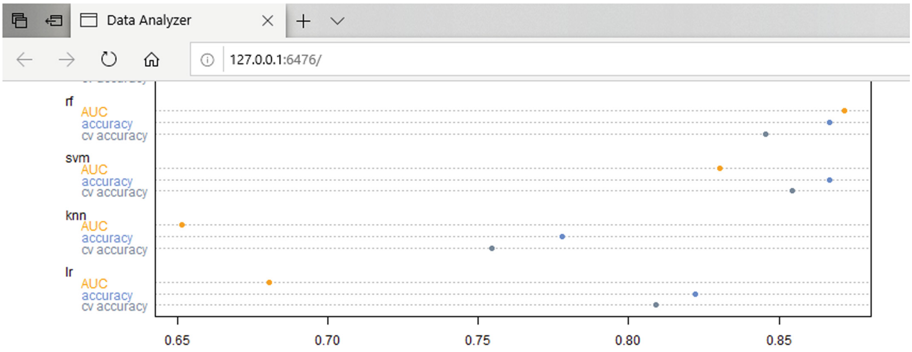


Fig. 6. Table of classifier performance accuracy and AUC.

Modest visualizations are presented allowing one to compare the relative measures. We used shiny [22] and shinyWidgets [23] for generating these visualizations and without the swarm wisdom available from netizens [32] none of this is possible, given that we are unfunded, staffed by 1 TA,1 Volunteer and 1 undergraduate student.

Results of stacked generalization is presented below (Fig. 7).



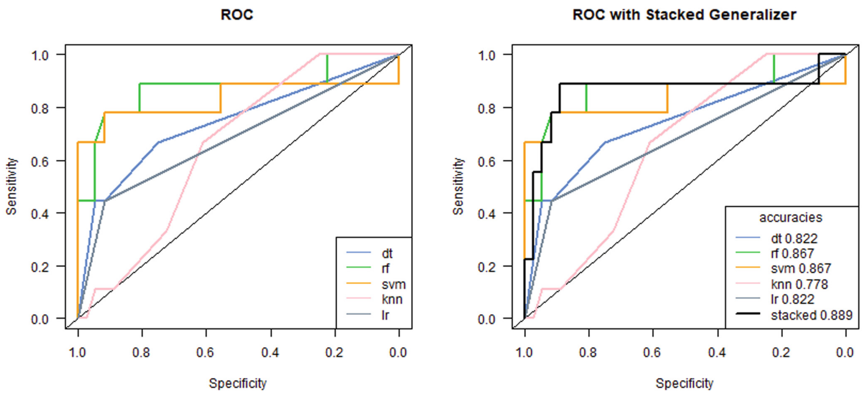


Fig. 7. ROC curves with and without stacked generalization

For the Hepatitis dataset, the stacked-generalizer using LogisticRegression, DecisionTree, Nearest Neighbor, Support Vector Machines, randomForest and the RandomForest are shown as specified. The Stacked Generalizer results in the highest performance of 0.899 combining all the above classifiers including the Ensemble classifier.

3 Conclusion

In this paper we summarized the import of No Free Lunch theorem, efficacy of Occam's Razor in searching for the best performing classifier for any given dataset. Guided by Occam's Razor, weak learners are trained at default configuration. User is allowed to pick and choose algorithms, specify a training set proportion. The system then runs the stacked-generalizer using voting mechanism. Comparative performance measures are displayed with Accuracy and AUC. ROC curves are generated for the specified algorithms. Users can perform multiple experiments and save them for further analysis.

Acknowledgements. Generous support from IBM PowerSystems Academic Initiatives for all of Raman's course is acknowledged.

References

1. Kotsiantis, S.B.: Supervised machine learning: a review of classification techniques. Informatica **31**, 249–268 (2007)
2. Alpaydin, E.: Introduction to Machine Learning. MIT Press
3. Duda, R.O., Stork, D.G., Hart, P.E.: Pattern Classification. Wiley
4. Mitchell, T.: Machine learning. McGraw Hill
5. Murphy, K.P.: Machine Learning, A Probabilistic Perspective. MIT Press
6. Hancock Jr., M.F.: Practical Data Mining. CRC Press
7. Saisficing Solution. https://www.kbmanage.com/concept/satisficing
8. Domingo, P.: The role of occam's razor in knowledge discovery. Data Min. Knowl. Discov. **3**, 409–425 (1999)
9. No Free Lunch Theorem. https://medium.com/@LeonFedden/the-no-free-lunch-theorem-62ae2c3ed10c
10. https://peekaboo-vision.blogspot.com/2019/07/dont-cite-no-free-lunch-theorem.html
11. Wolpert, D.: The lack of a priori distinctions between learning algorithms. Neural Comput. **8**(7), 1341–1390 (1996)
12. http://no-free-lunch.org/
13. Turney, P.: Types of Cost in Inductive Concept Learning. https://arxiv.org/ftp/cs/papers/0212/0212034.pdf
14. Shilbayeh, S.A.: Cost Sensitive meta-learning. http://usir.salford.ac.uk/id/eprint/36278/1/Cost%20sensitive%20meta%20learning_2015.pdf
15. Salzberg, S.L.: On comparing classifiers: pitfalls to avoid and a recommended approach. Data Min. Knowl. Disc. **1**, 317–328 (1997). http://people.sabanciuniv.edu/~berrin/cs512/reading/salzberg-comparing-pitfalls.pdf
16. Dietterich, T.G.: Approximate statistical tests for comparing supervised classification learning algorithms. http://web.cs.iastate.edu/~honavar/dietterich98approximate.pdf
17. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006), http://jmlr.org/papers/volume7/demsar06a/demsar06a.pdf
18. Caruana, R., et al.: An Empirical Evaluation of Supervised Learning in HighDimensions. http://lowrank.net/nikos/pubs/empirical.pdf
19. https://www.wired.co.uk/article/master-algorithm-pedro-domingos
20. Fawcett, T.: An introduction to ROC analysis. Pattern Recogn. Lett. **27**, 861–874 (2006)
21. R Core Team: R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2018). https://www.R-project.org/
22. Chang, W., Cheng, J., Allaire, J.J., Xie, Y., McPherson, J.: Shiny: Web Application Framework for R. R package version 1.2.0 (2018). https://CRAN.R-project.org/package=shiny
23. Perrier, V., Meyer, F., Granjon, D.: shinyWidgets: Custom Inputs Widgets for Shiny. R package version 0.5.0 (2019). https://CRAN.R-project.org/package=shinyWidgets
24. Robin, X., et al.: pROC: an open-source package for R and S + to analyze and compare ROC curves. BMC Bioinform. **12**, 77 (2011). https://doi.org/10.1186/1471-2105-12-77 http://www.biomedcentral.com/1471-2105/12/77/
25. Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F. (2019). e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.7-0.1. https://CRAN.R-project.org/package=e1071
26. Kuhn, M., et al.: caret: Classification and Regression Training. R package version 6.0-84 (2019). https://CRAN.R-project.org/package=caret
27. Therneau, T., Atkinson, B.: rpart: recursive Partitioning and Regression Trees. R package version 4.1-13 (2018). https://CRAN.R-project.org/package=rpart

28. Liaw, A., Wiener, M.: Classification and regression by randomForest. R News **2**(3), 18–22 (2002)
29. https://www.openml.org/data/get_csv/55/dataset_55_hepatitis.arff
30. https://archive.ics.uci.edu/ml/datasets/Hepatitis
31. https://github.com/datasets/hepatitis
32. https://stackoverflow.com/questions/34384907/how-can-put-multiple-plots-side-by-side-in-shiny-r