José Tristani | jtristani3

Charles Isbell

CS-7641: Machine Learning

September 22, 2021

Supervised Learning Analysis

This analysis will contrast different methods used for Supervised Learning by using two specially selected datasets where some methods exceed and some struggle to achieve good performance. The algorithms that will be contrasted are Decision Trees, Neural Networks, Random Forests, K-Nearest Neighbors and Support Vector Machines. The datasets that will be used for contrasting these algorithms are the Poker dataset and the MADELON dataset.

Why are these datasets interesting? They are interesting because these highlight the different algorithms strengths and weaknesses. The Poker dataset (Cattral and Franz) is composed of 25,000 training examples and 1,000,000 testing examples. Each record is an example of a hand consisting of five playing cards drawn from a standard deck. Each card is described using two attributes (suit & rank), for a total of 10 attributes per example. There is 1 class attribute that describes the Poker Hand. It is a very interesting dataset because of the class distributions, for example, there are 4 instances of a Royal Flush Hand, but there are 480 combinations of hands for those 4 instances. This makes the poker dataset challenging for different algorithms. The second is the MADELON dataset (Guyon, Gunn and Ben-Hur), this dataset is an artificial dataset containing data points grouped into 32 clusters placed on the vertices of a five-dimensional hypercube and randomly labeled 1 or -1. The five dimensions compose the 5 informative features, included are 15 linear combinations of those features for a combined 20 informative features. The remaining 400+ features have no predictive power. The dataset was part of the NIPS 2003 feature selection challenge. The difficulty is that the problem is multivariate and highly non-linear.

Starting with one of the more broadly used methods of machine learning, Deep Neural Networks. The results on the two datasets are very varied with DNNs. After using multiple hyperparameter settings for both datasets, the poker dataset performs excellently with DNNs, achieving a 99.99% accuracy with the training data and 98.82% on the test set comprising 1 million examples. These are very impressive results, the reason being that the poker dataset is in oner words curated and designed one might say, this is because there are certain poker hands that are oversampled, making the training dataset very good for training the network, with very little to no noise in the training data. It took very little effort to train DNNs on the poker dataset, considering overfitting is not much of a concern since hands are fixed. Swapping over to the MADELON dataset yields somewhat disappointing results, the more than 400 non-predictive features make the DNNs learning very hard and the network struggles to discern the patterns from the real features, even after changing multiple hyperparameters for learning rate the network only managed to achieve a 56.96%, not a very good accuracy considering there are only 2 classes in the dataset. Other methods have much overall performance for the MADELON dataset, without the added training time and tuning.

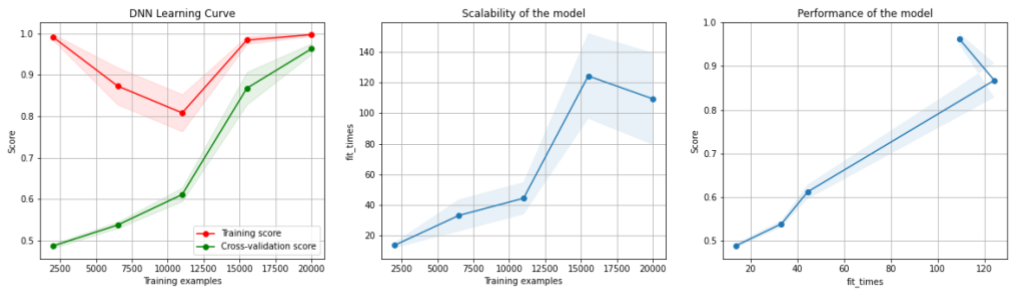

Figure 1: DNN Learning Curve for poker dataset

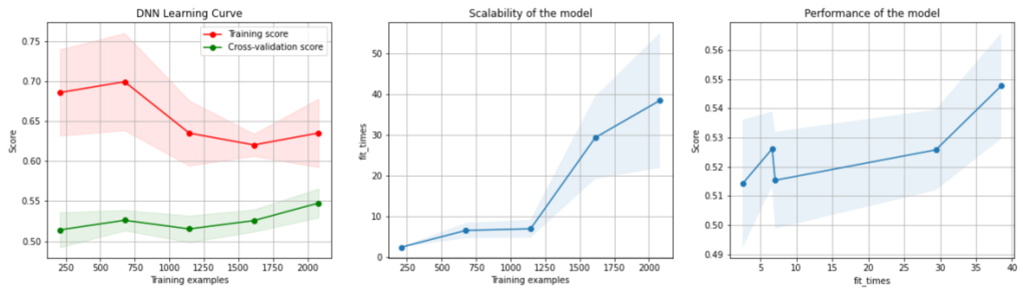

Figure 2: DNN Learning Curve for MADELON dataset

In contrast with neural networks, the KNN results are surprising, but expected. The poker dataset struggles to get good performance indicating the data is sparse and doesn’t cluster well with this method, multiple ***k*** parameters were used with little improvements, using Euclidian distance to compute neighbors yields better training set results, but not much improvement for test set data. Test set results yielded a mere 57.88%, with 10 classes in the poker dataset, the results are definitely tangible, but not close to the performance of the DNN method. Switching over to the MADELON dataset, performance much improved over the DNN method, much more consistent and kNN wall clock time was consistent and fast at just a few seconds. Performance for the MADELON dataset sits at 76.25% for the test set, a great result compared to the training time for other methods like DNN, SVMs. The performance for kNN is explained by diving into the datasets specifics, it consists of 32 clusters, kNN is very good at identifying these clusters and with consistent accuracy with small subsets of the data.

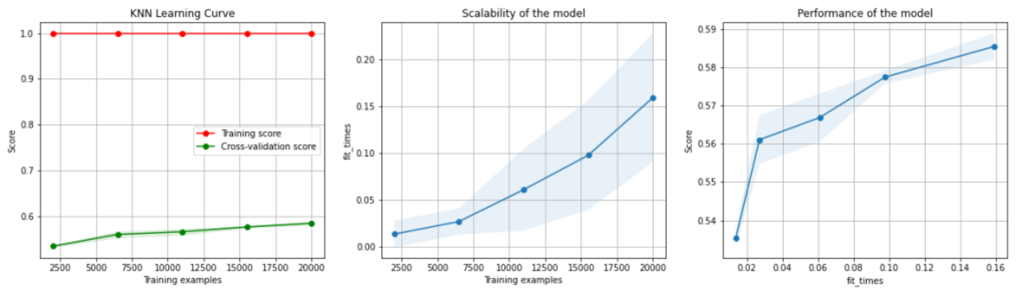

Figure 3: KNN Learning Curve for poker dataset

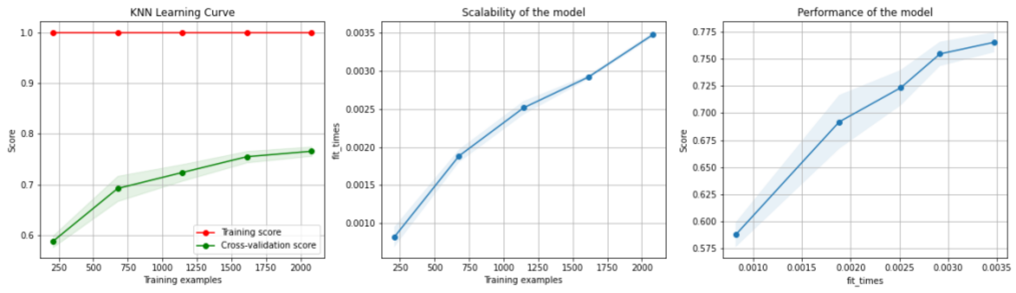

Figure 4: KNN Learning Curve for MADELON dataset

Decision trees achieves comparable results to kNN, ~55% accuracy with the poker dataset and ~75% with fast training times. After much tuning, smaller leaf sizes worked better for both datasets, settling on a *minimum leaf size* of 5 on both yielded the best results. There is no pruning by default on *scikit* Decision trees, but using pruning decreased accuracy. Diving into the poker dataset, Decision trees also failed to accurately generalize hands into accurate leaves of the tree, it may be primarily explained by the fact that the training set is not representative of real-world distribution of the data, the training sets data is true, representing permutations for all the unique hands, for instance, it contains 5 examples for a Royal Flush, but there are 480 combinations.

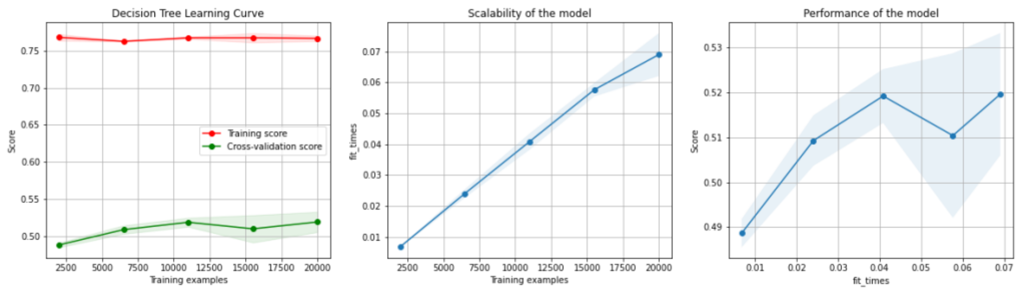

Figure 5: Decision Tree Learning curve for poker dataset

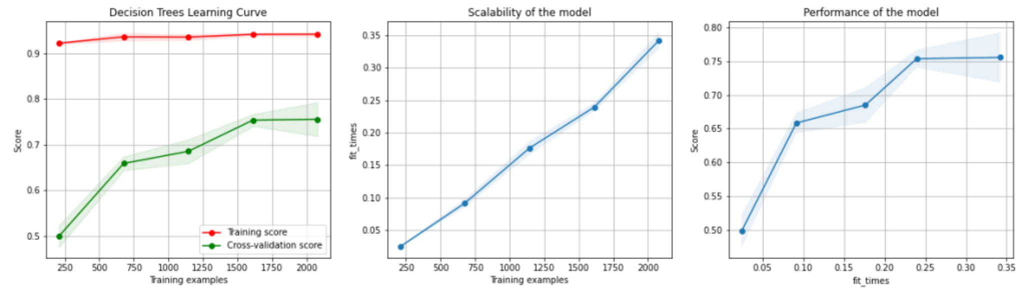

Figure 6: Decision Tree Learning Curve for MADELON dataset

For a boosted version of a decision tree, Random Forests of 100 trees were chosen, this method achieved better accuracy than just a single Decision tree, but improvements fell of quick, tweaking hyperparameters to higher *min_samples_leaf* from 5 to 12 in both datasets, in the poker dataset, accuracy improved from 55% to 60%, not much but some improvement. This is not a surprising result, the poker dataset requires learning a pattern that card order doesn't matter when it comes to class, and since those examples are not in the training data, just combinations, decision trees or forests will not reach the level of accuracy that DNNs do. In a surprising result, when using the MADELON dataset, Random Forests's performance takes a notable hit, going from 75% in Decision trees to 70% in forests, this anomaly was checked and even after changing multiple hyperparameters, the single decision tree still performed better, indicating that the bagging features in forests are detrimental for learning the patterns in the MADELON dataset.

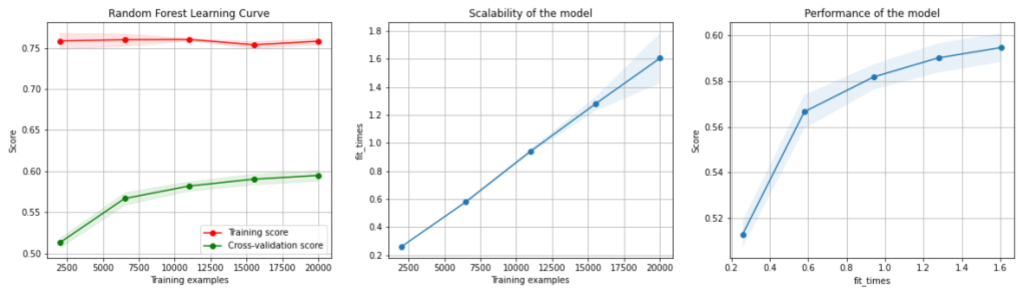

Figure 7: Random Forest Learning Curve for poker dataset

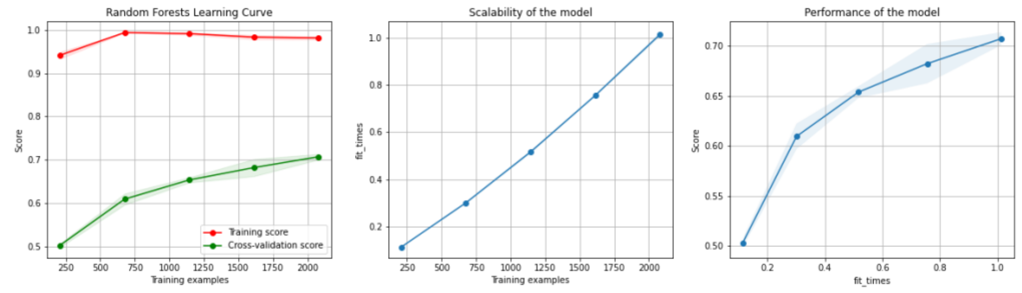

Figure 8: Random Forest Learning Curve for MADELON dataset

SVMs performed more or less the same as previous methods, for the poker dataset, the DNN is still the best performing method, able to infer the patterns in the training data, other methods don't have the same capability to do this on the curated poker dataset. This method was among the slowest to train and test, the 1 million test examples for the poker dataset took ~15 minutes to process. SVMs, although among the slowest to train as well as DNNs, performed among better that DNNs for the MADELON dataset, reaching a ~66% accuracy.

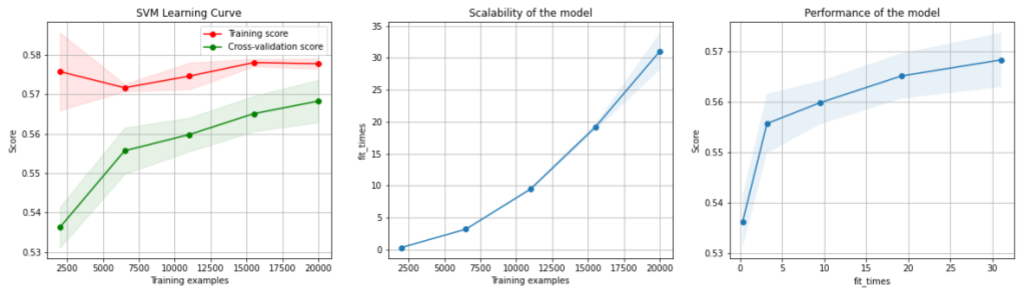

Figure 9: SVM Learning Curve for poker dataset

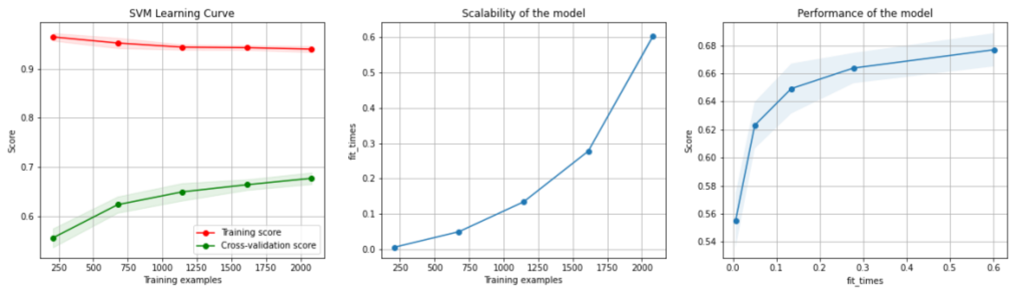

Figure 2: SVM Learning Curve for MADELON dataset

In conclusion, both datasets highlight the strengths and weaknesses in the 5 methods tested throughout this analysis. The best performing method for the poker dataset was DNNs, exclusively due to the ability of neural networks to find patterns, in this instance the pattern most important in this is dataset is learning how to interpret poker hands independently of card positions, this is what gave DNNs such an advantage with the limited training examples. On the other hand, the MADELON datasets traits make it incredibly hard for a DNN to extract pattern with that many non-predictive variables, some method of feature selection would be preferable before training the network. All the other methods performed better for the MADELON dataset, with the best being kNN. Adjusting the learning rate for DNN on MADELON had little effect. These results although not entirely surprising, are a great showcase of how a dataset's structure and feature sampling affect different machine learning methods. The training data for the poker dataset greatly impacted the results, whereas sampling from all the real-world poker hands might have had better results for other the methods. For MADELON, the fact that the dataset is composed of 32 clusters makes kNN or even k-means one of the best methods for this dataset, that is without some sort of feature selection that might help the other methods.

Works Cited

Cattral, Robert and Oppacher Franz. "Poker Hand Data Set ." 1 January 2007. *UCI Machine Learning Repository [https://archive.ics.uci.edu/ml/datasets/Poker+Hand].* 20 September 2021.

developers, Scikit-learn. "Plotting Learning Curves." 2021. *Scikit Learn.* 16 September 2021. <https://scikit-learn.org/stable/auto_examples/model_selection/plot_learning_curve.html>.

Dua, D and C Graff. *UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]*. 2019. 15 September 2021.

Guyon, Isabelle, et al. *UCI: Machine Learning Repository [https://archive.ics.uci.edu/ml/datasets/madelon]*. 2004. 18 September 2021.