Prediction of electromagnetic shower particles

Noa Miller

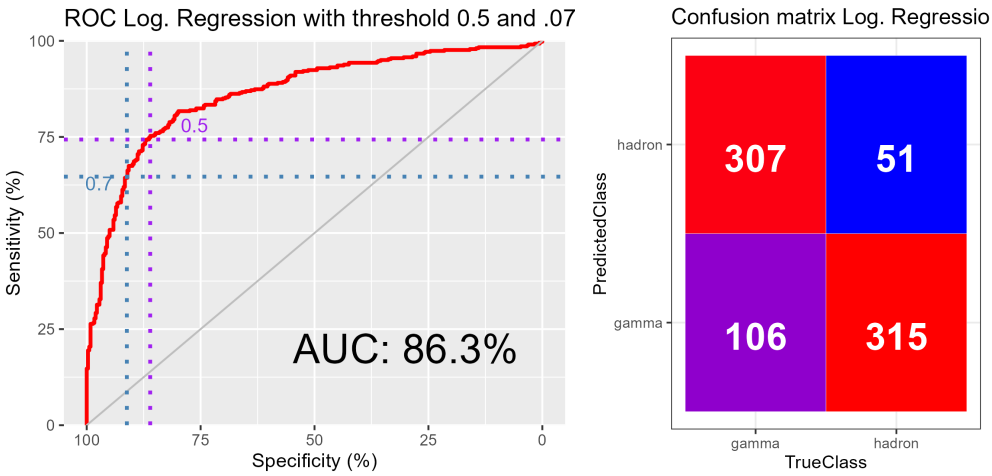
January 18 2023

Introduction

The gamma data is extracted from MAGIC Gamma Telescope (https://archive.ics.uci.edu/ml/datasets/magic+gamma+telescope) and contains simulated data with ten variables of electromagnetic variation, as well as a categorical "class" variable. It indicates whether an observation belongs to gamma (g) or hadron (h) particles. The objective of this analysis is to distinguish the class variable as successfully as possible for unseen data, predict the test accuracy, and explain why that specific method works best for the gamma data. First of all an exploratory analysis was done, where the magnitude of the independent variables and their covariance matrices were taken into account. The variables are on a very different scale (their unit measures are different too), e.g. distance or length are very dominant in value. Since some models (e.g. Neural Networks) are particularly sensitive to heterogeneity in terms of magnitude, the data got standardized. After that, an analysis was done to see whether classes are linearly separable. As part of this exploration missing data was excluded, some features were plotted in 2D, and the covariance matrices by class were analysed. In terms of the variance matrices by class, there are quite some differences, however the covariance of many variables,i.e. Size vs Conc, or Size vs Distance are very similar. We expect the decision boundaries to be quadratic, hence methods beyond linearity to perform better on the data.

Logistic regression

Logistic regression returns an accuracy of 0.7984596 at a threshold of 0.5 and the area under the curve (AUC) is 86.3% when plotting a ROC curve (specificity against sensibility). Although we could optimize this benchmark to fine-tune our prediction on either gamma or hadrons. However, the data is well balanced and this would eventually lead to a trade off: increasing prediction accuracy on one class will ultimately reduce accuracy on the other. As expected, the Logistic Regression model does not perform very well on the gamma data. By 10-fold cross-validation on the training set we get an accuracy of 0.779, by using the whole data for 10-fold CV 0.7757. The prediction accuracy on the test set is 0.7581. Even if it fares much better than a random guess (which would mean an AUC 50%),it is still a modest result.



Linear (LDA) and Quadratic (QDA) Discriminant Analysis

LDA assumes that the features are normally distributed and have similar covariance matrices, then we can expect linear boundary, otherwise a quadratic method will work better on the data. But the variance matrices are not similar, and a formal multivariate normality test (mvn test "royston") rejects normality for all variables. If the distribution of the predictors were approximately normal in each class, and the sample size relatively small, then LDA would work better than Logistic Regression. Despite the below QQ-Plots we assume normality as per the CLT (Central Limit Theorem), and for this sample size LDA indeed seems to have a somewhat better prediction accuracy than Logistic Regression with 0.78819.

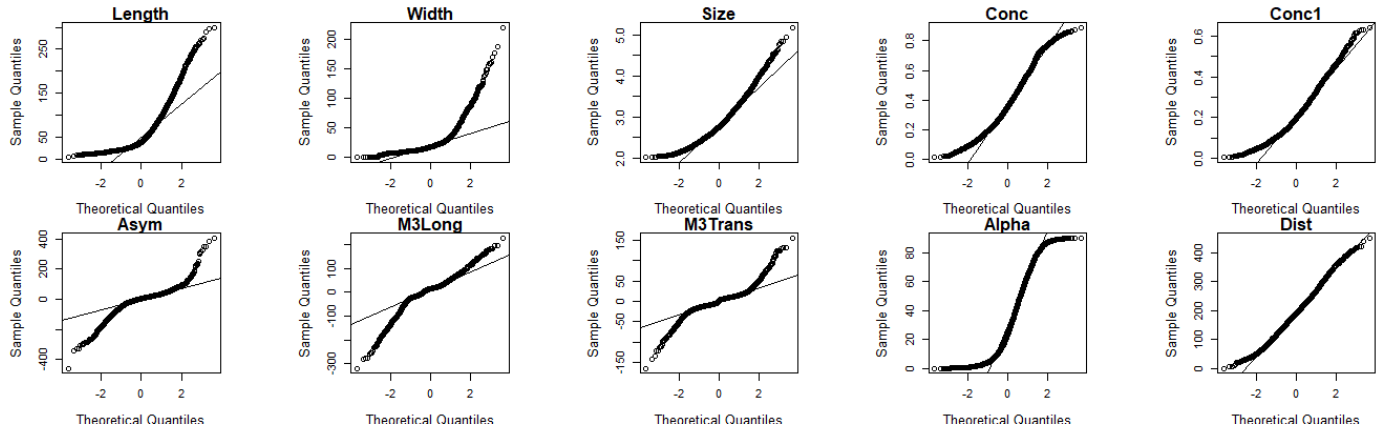
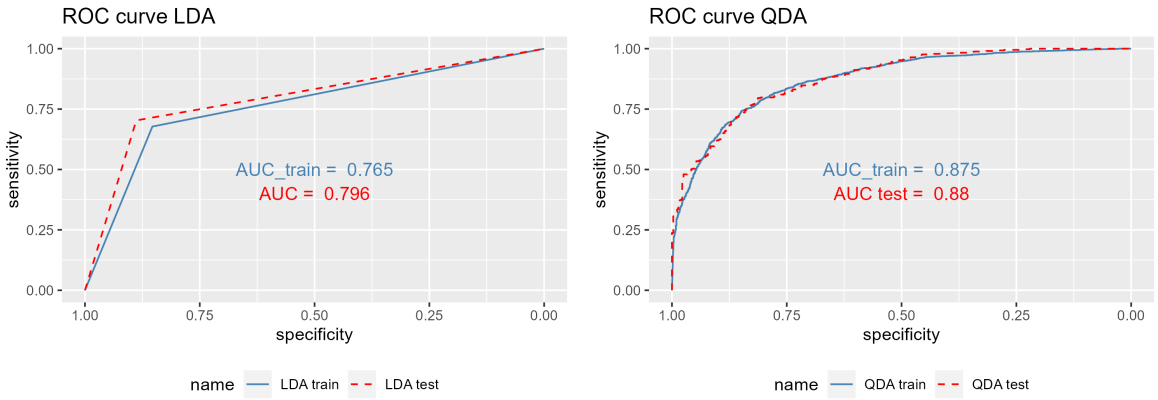


Figure 1: Normality check

When plotting the first three principal components we find that despite dimensionality reduction, the two classes are not well separated - there is no main variables splitting the data.

And even though the covariances are somewhat different, QDA is not performing very well either. The sample size is still relatively small (only about 2,000 observations each), and the 10-fold cross validated mean accuracy is 0.7447205 with a test accuracy of 0.7265 (the lowest up to this point).

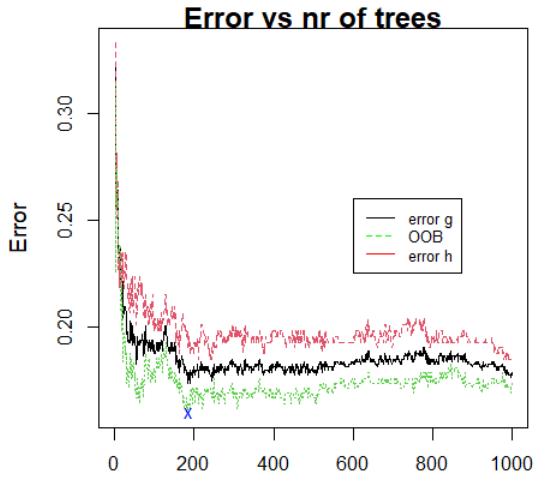


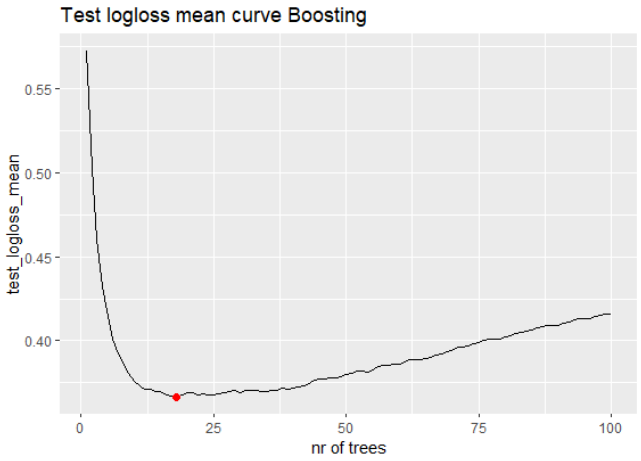
Tree based models

A linear regression will likely work well if the relationship between the features and response can be well approximated by a linear model. However, if a highly non-linear and complex relationship exists, a decision tree might outperform linearity. This proves to be the case for the gamma data.

When fitting a tree based model, a complexity parameter plot indicates that a cp = 0.011 or lower will be optimal for pruning. A pruned tree predicts the classes on the test set with an accuracy of 0.7946085, which is the best performance up to this point. To exploit the potential of "wisdom of crowds" and increase the accuracy further,

several trees got bagged into a Random Forest (classifying by majority vote, and many classifiers will reduce variability). We see on the error plot, that the OOB error stabilizes around 300 trees, and reaches its minimum at 185 trees. With this Random Forest we get a best test accuracy of 0.8536585 up to this point.

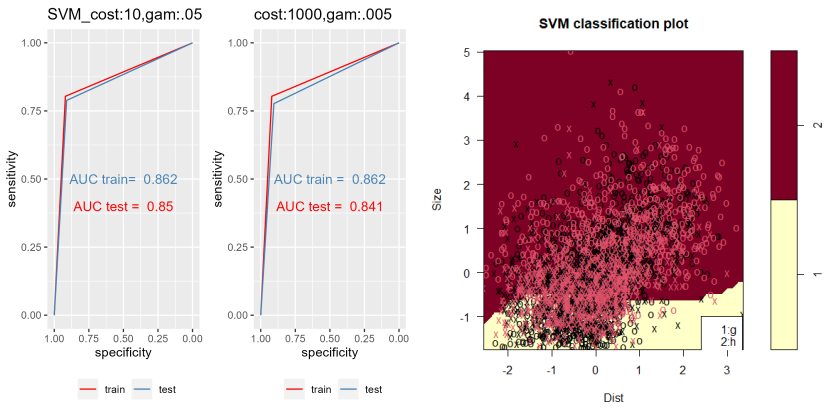




Boosting is a very performant method, which often improves ensemble methods even further. It uses many trees which are not independent from each other, but rather formed sequentially and learn from each other. The test maximum AUC value is reached at 42 trees, and the test logloss mean value reaches its minimum at 18 trees, which got plugged in as parameter "nrounds" into the xboost model. By exploiting this machine learning algorithm, we manage to increase the test prediction rate to 0.8408216 with an AUC of 0.912. By reducing the learning rate parameter "eta" from 0.3 to 0.01, we further increase the accuracy to 0.8523748 and AUC 0.931. Thus far Random Forest and the Xboost method with a slow-learning rate have the best performance with about 0.852-0.854.

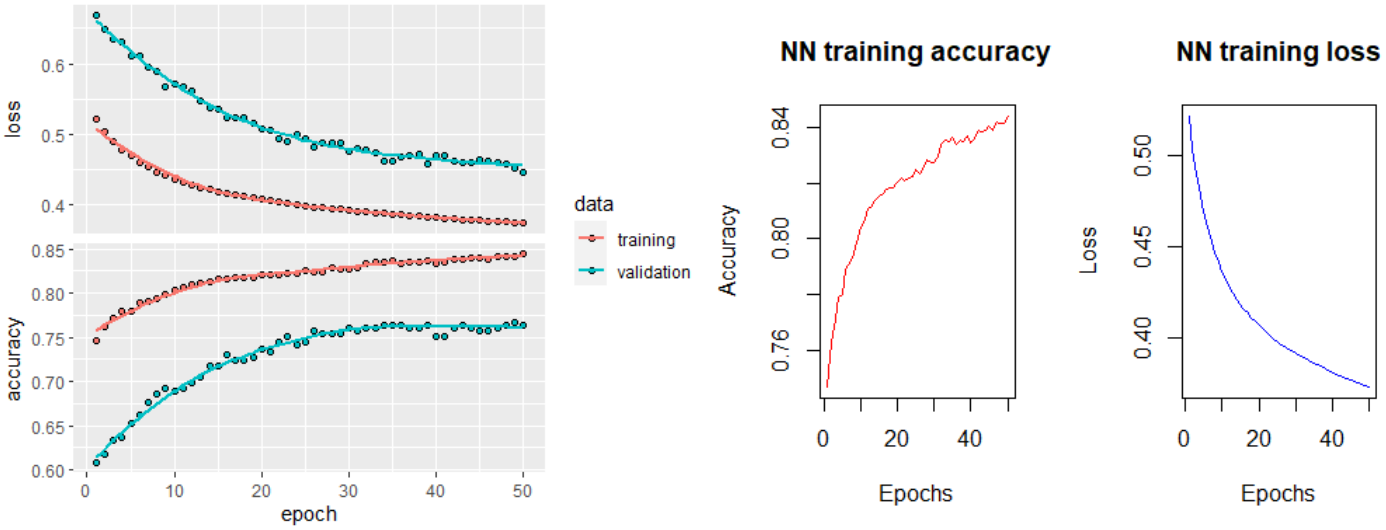
Support vector machines (SVM)

SVM accommodate non-linear boundaries and find hyperplanes that separate the data well. A non-linear "radial" kernel got instantiated, which acts locally on observation level (like each observation Gaussian), and measures the interaction between two data points. After inserting a list of parameter ranges, the optimum cost, as well as gamma values were calculated. We compare two models (one with the best parameters, and another with random values) with radial kernel and get a best model accuracy of 0.8446727, with an AUC 0.85. By plotting some of the decision boundaries in a two-dimensional space, we get a good insight about the complexity of the task, since the two classes virtually overlap in several aspects. This explains the relatively high cost parameter needed to accommodate the data, and allow for boundary violations, while controlling over-fitting through other parameters (gamma). This explains that a cost of 100.000 results in a even higher accuracy of 0.8498074.



Neural Networks (NN)

When fitting a network with no hidden layer, "rmsprop" optimizer, softmax activation and 10, 30 and 50 epochs, the best test accuracy is 0.843389, which is the second largest after Random Forest and Xboost. Even by adding an additional hidden layer or changing the optimizer to adam, the performance is not increasing.



Conclusion

Andre Ye describes in his post (https://towardsdatascience.com/when-and-why-tree-based-models-often-outperform-neural-networks-ceba9ecd0fd8) that tree based models are basically simplified neural networks, modelling by a deterministic approach (following certain yes/no-rules, just like human thinking) instead of a probabilistic technique. Neural networks work well on data with massive scale, such as text or images. But for our binary classification at hand, a downscaled tree based method will outperform a such complex network. Logistic Regression and LDA slightly outperformed QDA, since the differences in the covariance matrices were not significant enough to justify more flexibility by QDA. However, when looking at classification plots and the improving results of tree based models or SVM, we can convince ourselves that quadratic methods ware needed for this data set.

In summary, we conclude that either a Random Forest or a slow-learner Xboost model will perform best on unseen gamma data, with an accuracy of about 0.852 - 0.855.

References

https://cran.r-project.org/web/packages/caret/vignettes/caret.html https://www.geeksforgeeks.org/cross-validation-in-r-programming/ An Introduction to Statistical Learning, Gareth James,Daniela Witten,Trevor Hastie,Robert Tibshirani https://stackoverflow.com/questions/37438461/ggplot2-use-of-scale-x-reverse-on-roc-plot http://r-statistics.co/Information-Value-With-R.html https://towardsdatascience.com/linear-discriminant-analysis-lda-101-using-r-6a97217a55a6 http://r-statistics.co/Information-Value-With-R.html https://mlr-org.com/posts/2015-07-28-visualisation-of-predictions/ https://www.statology.org/roc-curve-ggplot2/ https://medium.datadriveninvestor.com/understanding-the-log-loss-function-of-xgboost-8842e99d975d https://towardsdatascience.com/when-and-why-tree-based-models-often-outperform-neural-networks-ceba9ecd0fd8 https://stackoverflow.com/questions/37897252/plot-confusion-matrix-in-r-using-ggplot