MA8701 Advanced methods in statistical inference and learning

Mette Langaas IMF/NTNU

04 April, 2021

Part 4: Explainable AI

Why a part on XAI?

In **Part 1** we worked with *interpretable* methods:

- linear regression (LS/MLE, ridge and lasso)
- logistic regression (MLE, ridge and lasso)

By studying the estimated regression coefficients we could (to some extent) explain what our fitted model could tell us about the data we had analysed.

In **Part 2** we started by studying a classification and regression tree, which is also an interpretable method.

See Chapter 4.1, 4.2 and 4.4 of Molnar (2019) (Chapther 4_ Interpretable Models, 4.1 Linear regression, 4.2 Logistic regression and 4.4 Decision tree) on a discussion around what to report or plot from interpretable methods.

In Part 2 we then moved on to different versions of ensemble methods (bagging, random forest, xgboost, superlearner) - which are not interpretable. In **Part 3** we studied artificial neural networks (deep nets, recurrent nets, Bayesian nets) - again not interpretable methods.

L12: Introduction slide set

L12: LIME slide set

L13: Shapley values slide set

L13: Counterfactuals slide set

The two methods can be studied in detail here:

-
- Dandl et al. (2020)

We train a random forest on the Boston dataset: data(“Boston,” package = “MASS”) library(“randomForest”) rf <- randomForest(medv ~ ., data = Boston, ntree = 50) mod <- Predictor$new(rf, data = Boston)

Compute the accumulated local effects for the first feature

```
eff <-
FeatureEffect$new(mod, feature = "rm", grid.size = 30)$effplot()
```

Again, but this time with a partial dependence plot and ice curves

eff <- FeatureEffect$new(mod, feature = “rm,” method = “pdp+ice,” grid.size = 30) plot(eff)

Since the result is a ggplot object, you can extend it:

library(“ggplot2”) plot(eff) + # Adds a title ggtitle(“Partial dependence”) + # Adds original predictions geom_point(data = Boston, aes(y = mod$predict(Boston)[[1]], x = rm), color = “pink,” size = 0.5)

If you want to do your own thing, just extract the data:

```
eff.dat <- eff$results head(eff.dat)
```