Principles of supervised learning

B.D. in Business Administration and Economics
Course in Quantitative Methods III

Rosario Barone
University of Rome “Tor Vergata”

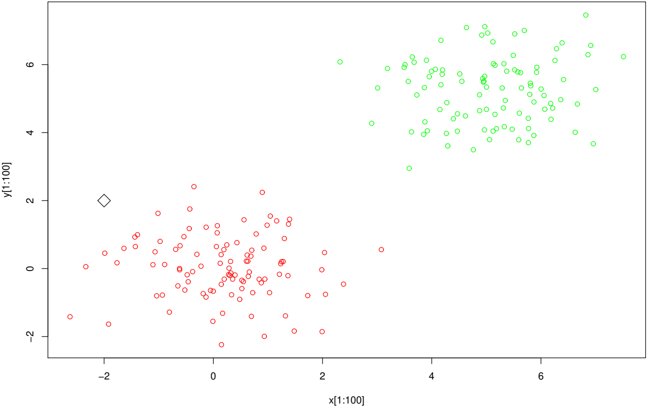
rosario.barone@uniroma2.it

credits to Prof. Alessio Farcomeni

Focus on the prediction

- So far we have mostly restricted our attention to understanding interrelationships among predictors and outcome
- Certain tasks are more focused on prediction of the outcome in future settings (when it has not been measured, only the predictors)
- A sample of examples/training data is available where an outcome has been measured together with (possibly, cheap) predictors
- Will a firm go into default? Has there been a fraud with this transaction? Etc.
- Logistic regression can be used, but it is not the only option. Different predictors make a different prediction, anyway.

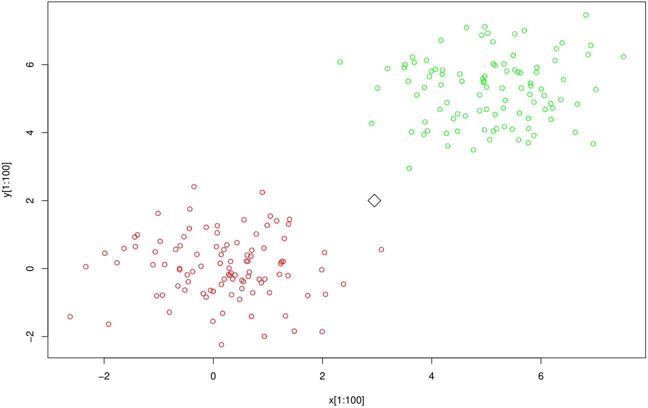
Basic Idea: which class for the diamond?



Why?

- There are a huge number of methods for classification. They will all agree.
- Idea is always that of dividing the feature space in segments, one for each class.
- Main differences: complexity, interpretability
- If your only target is prediction: only aim is performance (on new data)

A more challenging case



Take home messages

- Groups might not be separated, so that the classifier will make errors also on the training set
- Non-linear classifiers tend to make less errors on the available data (less biased) than linear/simpler classifiers
- Non-linear classifiers tend to make more errors on the new data (more variable) than linear/simpler classifiers.
- Simple and interpretable method: logistic regression.
- Complex and not interpretable method: random forests.

Performance on new data

- The estimated performance on available data is a bad predictor of performance on new data
- Better predictor: performance on available data set aside (test set)
- The rest is a training set used for model estimation.
- Split data at random (`sample`), maybe stratify if there are small categories
- What performance measure? The total number of misclassified objects.

Procedure in practice

1. Select (a) few methods, by varying prediction method, number of predictors, tuning parameters.
2. Split the data in a training set (50-80% of your sample) and a test set (the remaining).
3. Train selected methods on the training set, predict test set. For each method, estimate the prediction error (or the prediction accuracy)
4. Best method is the one with lowest prediction error (or the highest accuracy) on the test set.

Can you improve on the best performance? How? Compare training and test error rates.

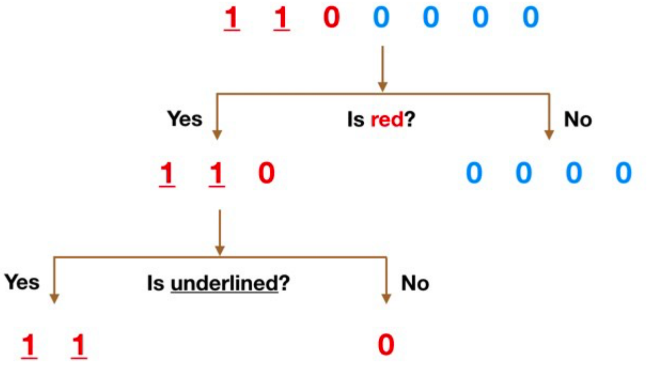
Comparing performance on test and training

- Training error almost always smaller than test error
- Very small (or zero) error on the training set, large on test: overfitting. Simplify method.
- Large error on the training set: get new and better variables, increase sample size
- Large or small depends on comparison. Always get a benchmark (no predictors, human performance, etc.).
- Balance complexity and fit: the simplest model that fits best is the best

What if I do not have enough predictors?

- If performance can be improved by removing predictors, life is easy.
- What if more are needed? (And no new measurements can be performed or the experts have no idea what to do)
- Augmentation: do polynomial regression by creating new variables (squares, products, etc.)

Classification tree



Prediction through classification tree

- Prediction is done by following the estimated tree to the leaf. E.g., a family with unknown status with 52$ per day, two members, will belong to a leaf with 0.5% deprived.
- Trees are also easy to visualize.

For instance

- Y: indicator of deprived household
- X: income, number of members, etc.
- First level of the tree: for daily income$>$ 2\$, there are only 14% poors while 86% for lower income
- Further level: for income $>$ 2\$, deprived are found only with number of members $>$ 2; no further splits for low incomes.

How to read it? If *condition 1* and *condition 2* and *condition 3*, then outcome.

Generalization of classification trees

- We have seen multinomial outcome logistic regression;
- Classification trees are naturally extended to nominal outcomes. Prediction is assigned to the relatively more prevalent category.

Classification trees in R

- These are estimated using function `rpart` in `library(rpart)`. Usual syntax and methods (including `predict`).
- Visualize through `prp` function in `library(rpart.plot)`

Random forests

A forest made of (classification) trees

- You have a target Y and a set of predictors $X_1, \ldots, X_p$
- Select the variable which separates the most presence and absences.
- Within each subgroup (node), split further.
- Iterate till stopping rule satisfied (e.g., less than 50 units in the leaf, three levels of split, etc.)

Note: heuristic approach!

Random Forests

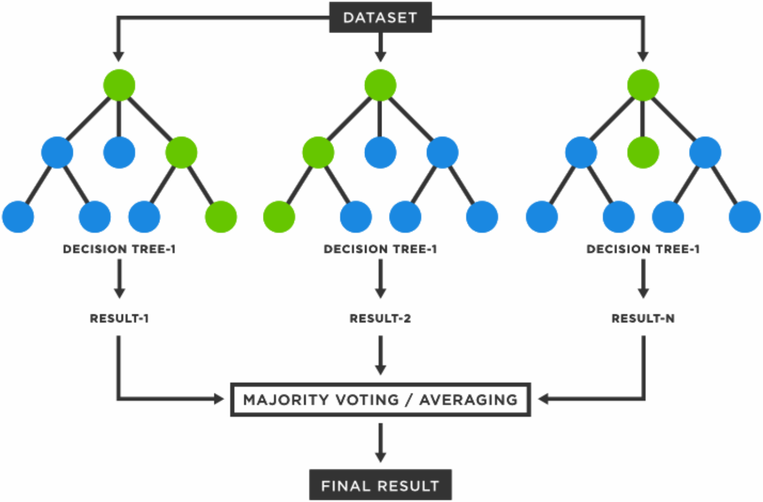
In other words, the entire group is a forest where each tree has a different independent random sample that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction.

In the case of the random forest algorithm, many trees can make the algorithm too slow and inefficient for real-time prediction. In contrast, the results are generated based on randomly selected observations and features built on different decision trees in the random forest algorithm.

The fundamental concept behind random forest is a simple but powerful. The reason that the random forest model works so well is:

A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models.

Random Forest



Random Forests

- The predictive performance of trees is good, but it is incredibly better when you put many together.
- Random Forests: select at random a large number (usually, 500) subsets of the data (both samples and predictors), grow 500 trees separately.
- Each tree makes a prediction as we just discussed.
- Final prediction is the majority vote. If 480 out of 500 trees vote that the household is deprived, you predict deprivation.

Forests work

- They predict well *off-the-shelf* (usually no tuning is needed)
- They seldom overfit: they adapt to the right amount of complexity automatically.
- Problem: you need a large training set. With $n < 100$ do not even attempt
- Problem: interpretation is impossible.
- Problem: they can adapt to a good level of complexity but only so much. Artificial intelligence applications (when features are text, images, etc.) use even more complex procedures.

Random forests in R

- Use function `randomForest` in `library(randomForest)`. Usual synthax and methods. The outcome must be a factor (or you get regression).
- Useful options for tuning: `ntree`, `mtry` (number of variables in each subset), etc. (see `help`)

RF vs logistic regression

- RF: usually better predictions (especially in complex cases, big data)
- Logistic: interpretable (mandatory in certain circumstances)
- Final recommendation: try both (maybe with different tuning for RFs), evaluate on a test set, pick the minimum error.