Basic Machine Learning Methods: Part 2

Christian Seberino, Ph.D.
cs@automaticprogramming.info
DAAML

Outline

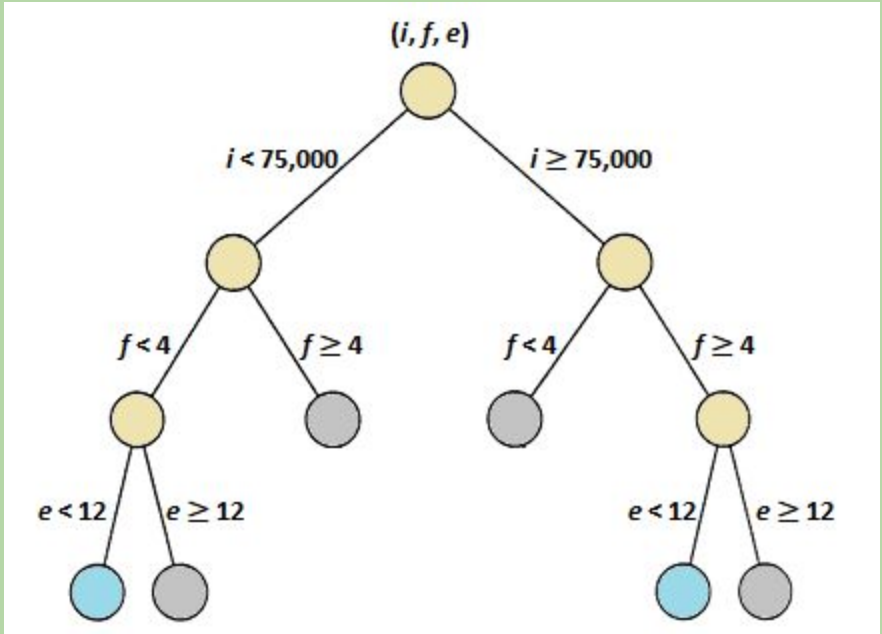
- Decision Trees
- Ensemble Learning
- Random Forests
- Gradient Boosting

Decision Trees

- flowcharts of inequalities
- Each test involves a single feature.
- Can be used for classification and regression!

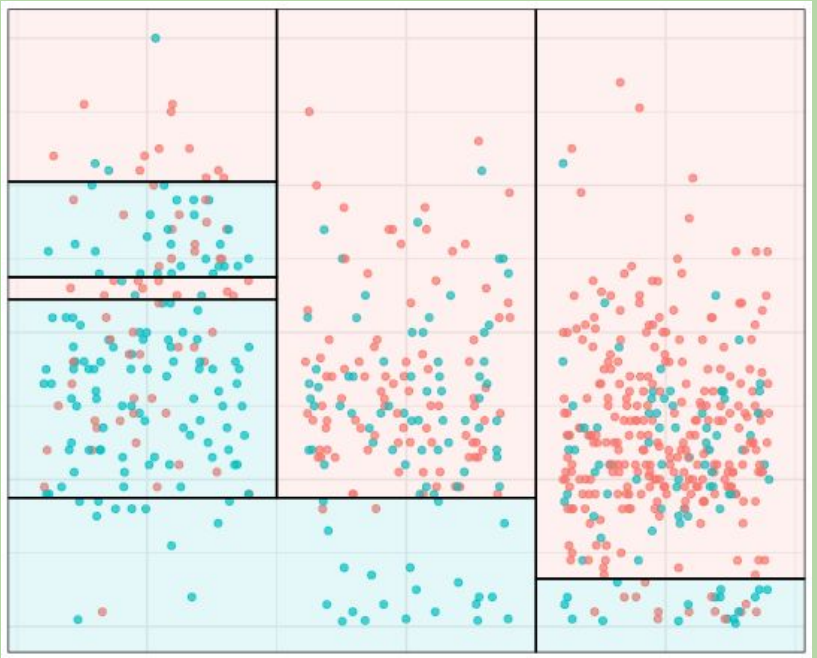
Decision Trees

- **Purchases depend on income, family size and education:**



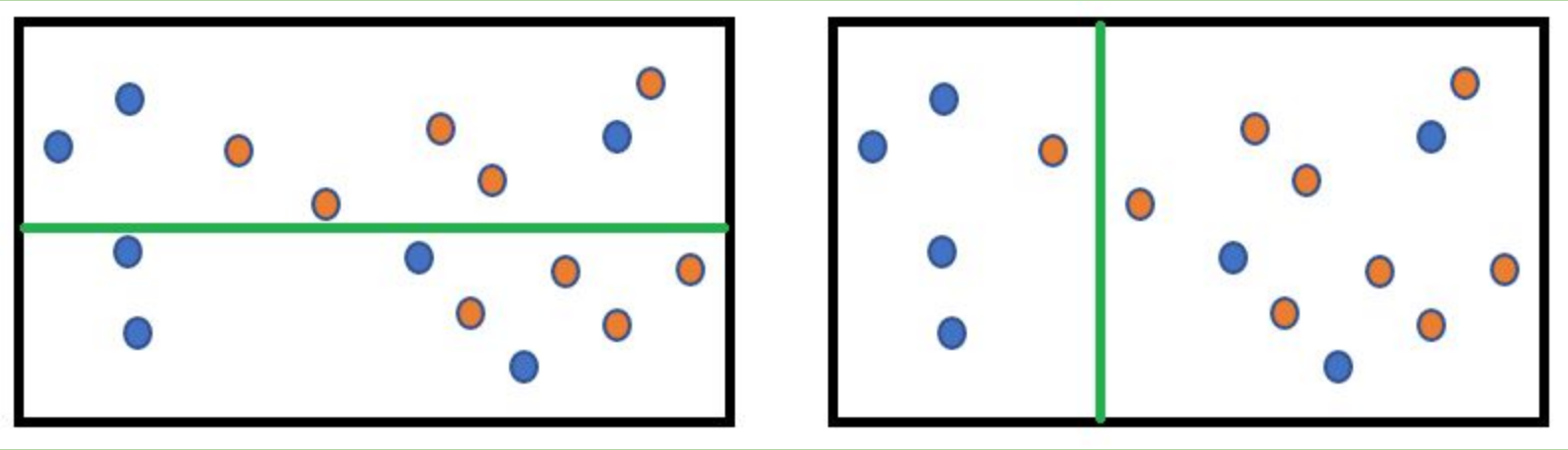
Decision Trees

- Can also equivalently interpret decision trees geometrically:



Decision Trees

- At each stage in building a decision tree, pick the best feature for the inequality.

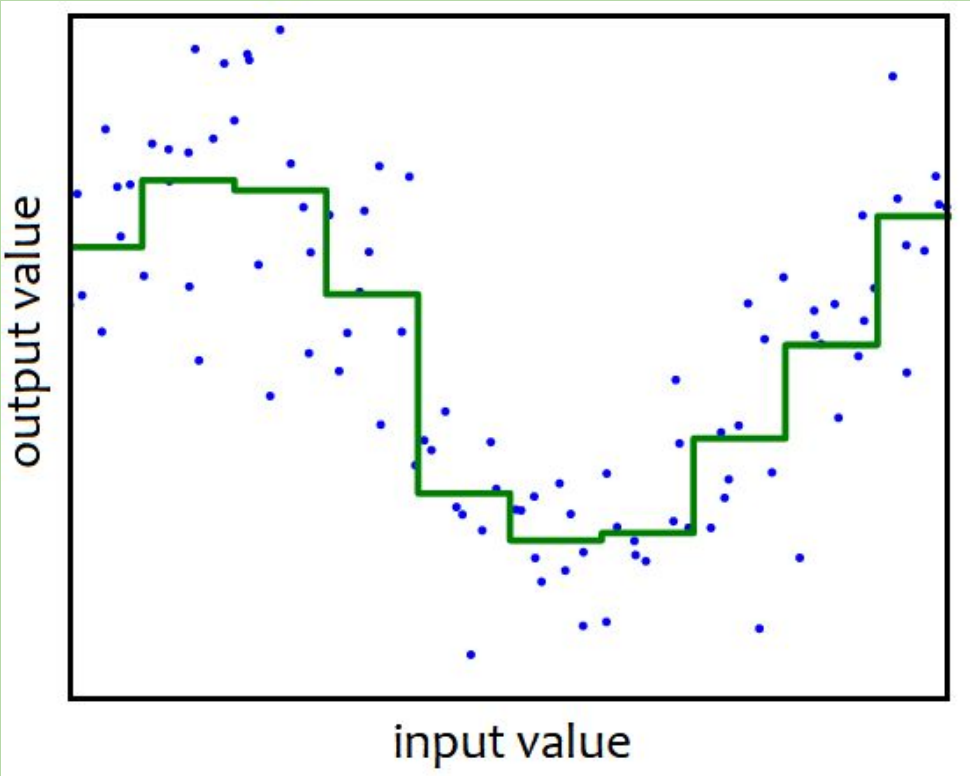


Decision Trees

- Decision trees can also be used for regression!

- Results are the averages of output values in regions.

Decision Trees



Decision Trees

- Decision trees do not require normalization and are easy to understand.
- Decision trees are very susceptible to overfitting.
- One way to try to avoid overfitting is to limit the number of levels.

Ensemble Learning

- Uses multiple models and methods!
- No reason in general must use one model!
- Consider advisors helping solve a problem.
- Can pick one result or combine results.

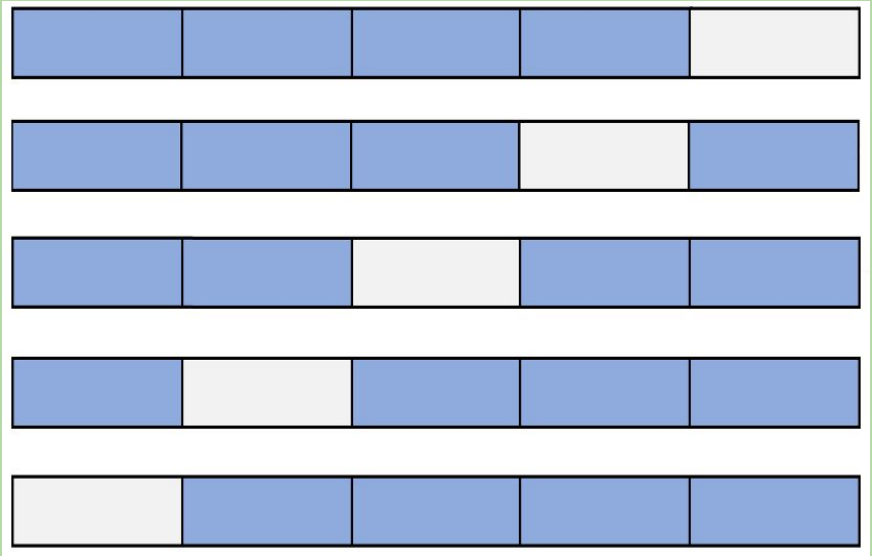
Ensemble Learning

- Also no reason in general must use datasets in one way!

- Can use several different derivations for testing and training.

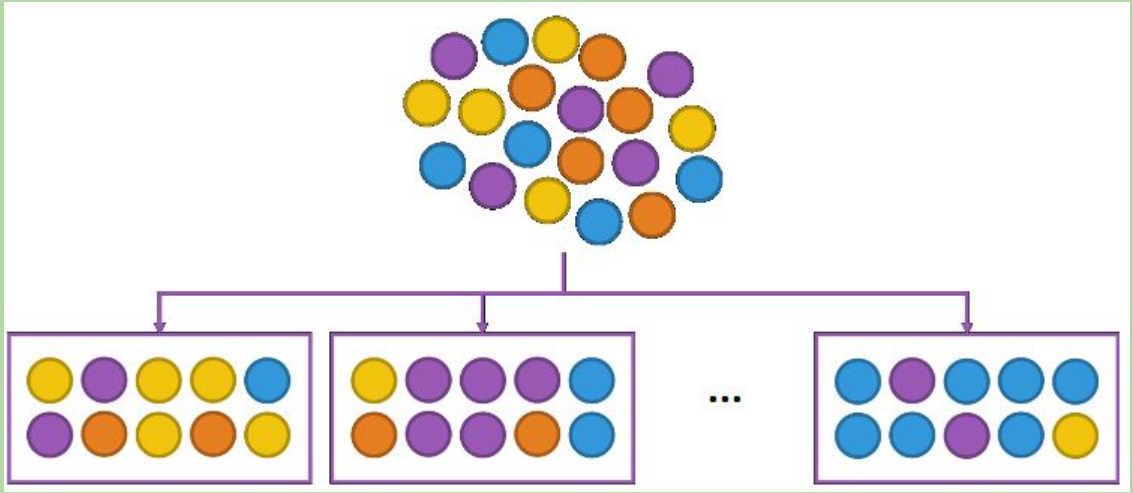
Ensemble Learning

- K fold cross validation has independent testing subsets but limited in quantity.



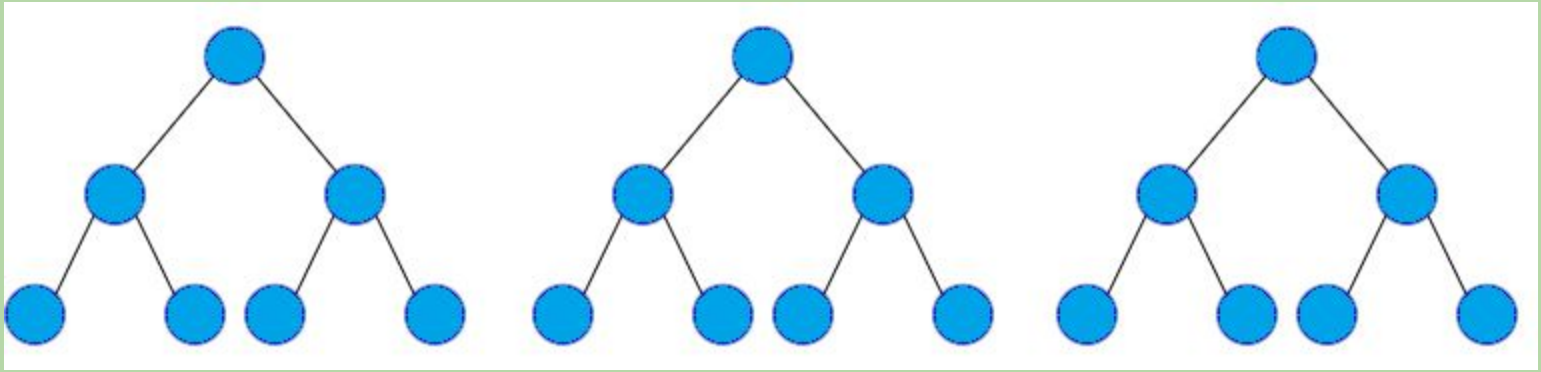
Ensemble Learning

- **Bootstrapping creates nearly unlimited training subsets but has redundancy.**



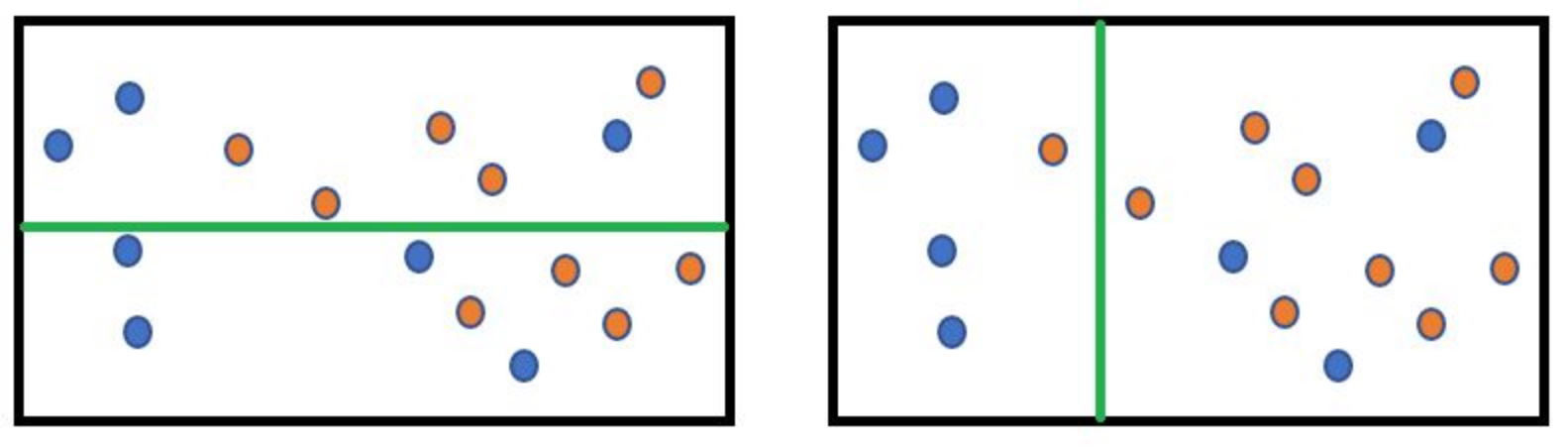
Random Forests

- **Use ensembles of decision trees with voting and averaging to avoid overfitting!**



Random Forests

- If bootstrapping, and feature selections limited to random subsets, then ensembles are referred to as random forests.



Random Forests

- Does not require normalization.
- Only a few parameters need to be tuned.
- Training requires more resources than decision trees but is parallelizable!

Random Forests

Journal of Machine Learning Research 15 (2014) 3133-3181 Submitted 11/13; Revised 4/14; Published 10/14

Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?

Manuel Fernández-Delgado MANUEL.FERNANDEZ.DELGADO@USC.ES
Eva Cernadas EVA.CERNADAS@USC.ES
Senén Barro SENEN.BARRO@USC.ES
CITIUS: Centro de Investigación en Tecnoloxías da Información da USC
University of Santiago de Compostela
Campus Vida, 15872, Santiago de Compostela, Spain

Dinani Amorim DINANIAMORIM@GMAIL.COM
Departamento de Tecnologia e Ciências Sociais- DTCS
Universidade do Estado da Bahia
Av. Edgard Chastinet S/N - São Geraldo - Juazeiro-BA, CEP: 48.305-680, Brasil

Random Forests

We evaluate **179 classifiers** arising from **17 families** (discriminant analysis, Bayesian, neural networks, support vector machines, decision trees, rule-based classifiers, boosting, bagging, stacking, random forests and other ensembles, generalized linear models, nearest-neighbors, partial least squares and principal component regression, logistic and multinomial regression, multiple adaptive regression splines and other methods), implemented in Weka, R (with and without the caret package), C and Matlab, including all the relevant classifiers available today. We use **121 data sets**, which represent **the whole UCI** data base (excluding the large-scale problems) and other own real problems, in order to achieve significant conclusions about the classifier behavior, not dependent on the data set collection. **The classifiers most likely to be the bests are the random forest** (RF) versions, the best of which (implemented in R and accessed via caret) achieves 94.1% of the maximum accuracy overcoming 90% in the 84.3% of the data sets. However, the dif-

Gradient Boosting

- Consider instead building decision trees sequentially while decreasing errors.

- Additional decision trees are instead trained to offset previous errors.

Gradient Boosting

training inputs	training outputs	decision tree model	testing residuals
	O	T_0	R_0
I	$-R_0$	T_1'	R_1
I	$-R_1$	T_2'	R_2
I	$-R_2$	T_3'	R_3
I	…	…	…
I	$-R_{N-1}$	T_N'	R_N

$$T_0 = \text{average of O} \;\&\; T_k' = T_{k-1} + \mathit{1}T_k$$

Gradient Boosting

- Often more accurate than random forests!
- XGBoost has won several competitions.

Gradient Boosting

- Much harder to tune, and often more computationally expensive, due to a learning rate parameter.

- Not parallelizable.

Gradient Boosting

- Much more prone to overfitting.

- Overfitting often makes unsuitable for very high dimensional datasets.