Trees & Forests

02/18/19

Andreas C. Müller

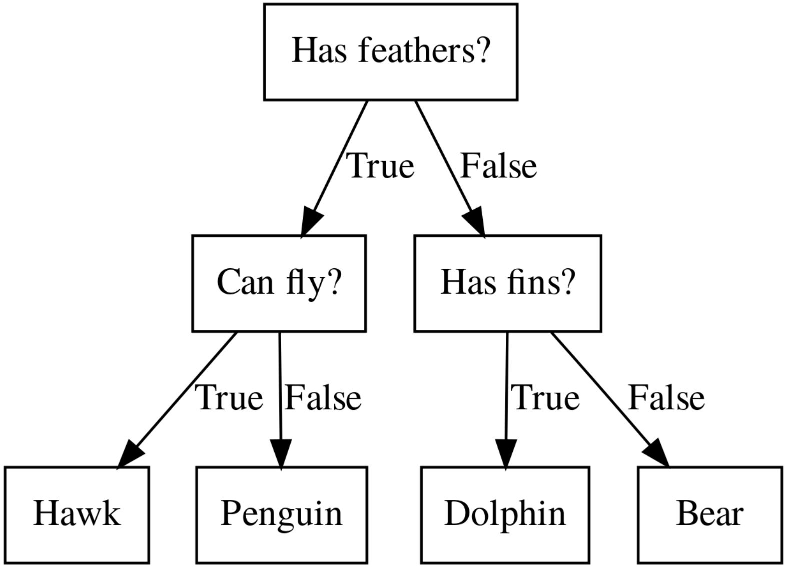
(Adapted and modified for CC 6021236 @ PCC/Ciencias/UCV by Eugenio Scalise, September 2019)

Why Trees?

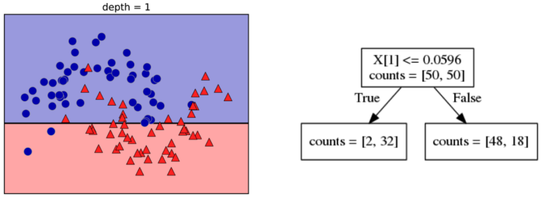
- Very powerful modeling method – non-linear!
- Doesn’t care about scaling of distribution of data!
- “Interpretable”
- Basis of very powerful models!

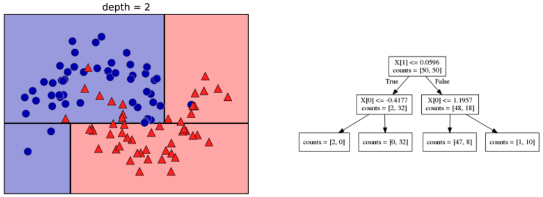
Decision Trees for Classification

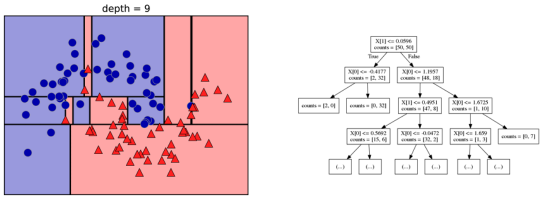
Idea: series of binary questions



Building Trees



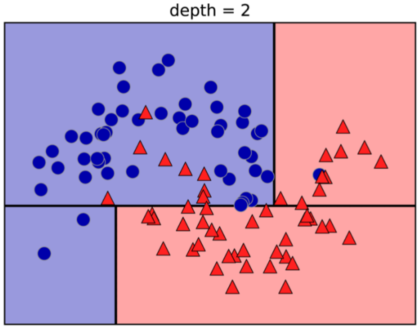


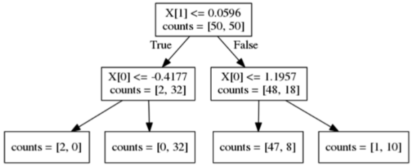


Continuous features:

- “questions” are thresholds on single features.
- Minimize impurity

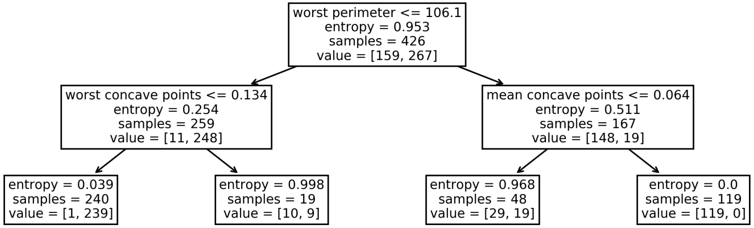
Prediction


depth = 2



Visualizing trees with sklearn

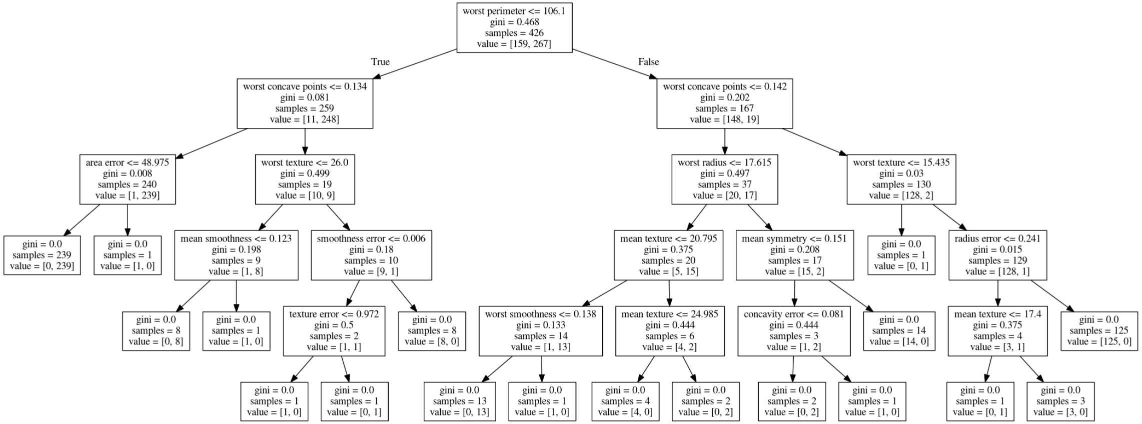
Visualizing trees with sklearn



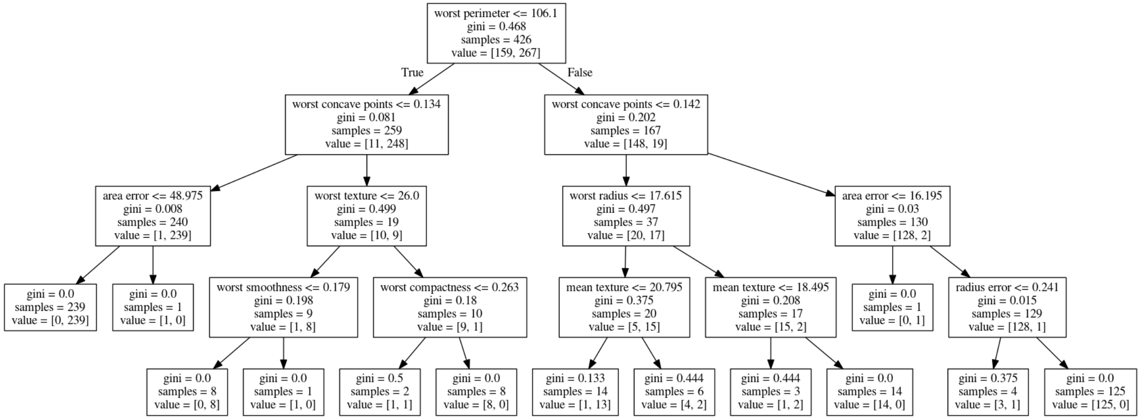
Parameter Tuning

- Limit tree size (pick one, maybe two):
  - max_depth
  - max_leaf_nodes
  - min_samples_split
  - min_impurity_decrease
  - ...

No pruning


worst perimeter <= 106.1 gini = 0.468 samples = 426 value = [159, 267]
True
False
worst concave points <= 0.134 gini = 0.081 samples = 259 value = [11, 248]
worst concave points <= 0.142 gini = 0.202 samples = 167 value = [148, 19]
area error <= 48.975 gini = 0.008 samples = 240 value = [1, 239]
worst texture <= 26.0 gini = 0.499 samples = 19 value = [10, 9]
worst radius <= 17.615 gini = 0.497 samples = 37 value = [20, 17]
worst texture <= 15.435 gini = 0.03 samples = 130 value = [128, 2]
gini = 0.0 samples = 239 value = [0, 239]
gini = 0.0 samples = 1 value = [1, 0]
mean smoothness <= 0.123 gini = 0.198 samples = 9 value = [1, 8]
smoothness error <= 0.006 gini = 0.18 samples = 10 value = [9, 1]
mean texture <= 20.795 gini = 0.375 samples = 20 value = [5, 15]
mean symmetry <= 0.151 gini = 0.208 samples = 17 value = [15, 2]
gini = 0.0 samples = 1 value = [0, 1]
radius error <= 0.241 gini = 0.015 samples = 129 value = [128, 1]
gini = 0.0 samples = 8 value = [0, 8]
gini = 0.0 samples = 1 value = [1, 0]
texture error <= 0.972 gini = 0.5 samples = 2 value = [1, 1]
gini = 0.0 samples = 8 value = [8, 0]
worst smoothness <= 0.138 gini = 0.133 samples = 14 value = [1, 13]
mean texture <= 24.985 gini = 0.444 samples = 6 value = [4, 2]
concavity error <= 0.081 gini = 0.444 samples = 3 value = [1, 2]
gini = 0.0 samples = 14 value = [14, 0]
mean texture <= 17.4 gini = 0.375 samples = 4 value = [3, 1]
gini = 0.0 samples = 125 value = [125, 0]
gini = 0.0 samples = 1 value = [1, 0]
gini = 0.0 samples = 1 value = [0, 1]
gini = 0.0 samples = 13 value = [0, 13]
gini = 0.0 samples = 1 value = [1, 0]
gini = 0.0 samples = 4 value = [4, 0]
gini = 0.0 samples = 2 value = [0, 2]
gini = 0.0 samples = 2 value = [0, 2]
gini = 0.0 samples = 1 value = [1, 0]
gini = 0.0 samples = 1 value = [0, 1]
gini = 0.0 samples = 3 value = [3, 0]

max_depth = 4



worst perimeter <= 106.1
gini = 0.468
samples = 426
value = [159, 267]

True

False

worst concave points <= 0.134
gini = 0.081
samples = 259
value = [11, 248]

worst concave points <= 0.142
gini = 0.202
samples = 167
value = [148, 19]

area error <= 48.975
gini = 0.008
samples = 240
value = [1, 239]

worst texture <= 26.0
gini = 0.499
samples = 19
value = [10, 9]

worst radius <= 17.615
gini = 0.497
samples = 37
value = [20, 17]

area error <= 16.195
gini = 0.03
samples = 130
value = [128, 2]

gini = 0.0
samples = 239
value = [0, 239]

gini = 0.0
samples = 1
value = [1, 0]

worst smoothness <= 0.179
gini = 0.198
samples = 9
value = [1, 8]

worst compactness <= 0.263
gini = 0.18
samples = 10
value = [9, 1]

mean texture <= 20.795
gini = 0.375
samples = 20
value = [5, 15]

mean texture <= 18.495
gini = 0.208
samples = 17
value = [15, 2]

gini = 0.0
samples = 1
value = [0, 1]

radius error <= 0.241
gini = 0.015
samples = 129
value = [128, 1]

gini = 0.0
samples = 8
value = [0, 8]

gini = 0.0
samples = 1
value = [1, 0]

gini = 0.5
samples = 2
value = [1, 1]

gini = 0.0
samples = 8
value = [8, 0]

gini = 0.133
samples = 14
value = [1, 13]

gini = 0.444
samples = 6
value = [4, 2]

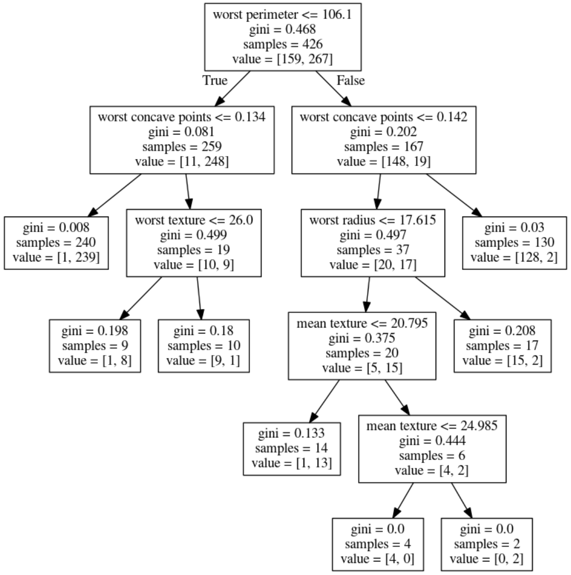
gini = 0.444
samples = 3
value = [1, 2]

gini = 0.0
samples = 14
value = [14, 0]

gini = 0.375
samples = 4
value = [3, 1]

gini = 0.0
samples = 125
value = [125, 0]

max_leaf_nodes = 8



worst perimeter <= 106.1
gini = 0.468
samples = 426
value = [159, 267]

True

False

worst concave points <= 0.134
gini = 0.081
samples = 259
value = [11, 248]

worst concave points <= 0.142
gini = 0.202
samples = 167
value = [148, 19]

gini = 0.008
samples = 240
value = [1, 239]

worst texture <= 26.0
gini = 0.499
samples = 19
value = [10, 9]

worst radius <= 17.615
gini = 0.497
samples = 37
value = [20, 17]

gini = 0.03
samples = 130
value = [128, 2]

gini = 0.198
samples = 9
value = [1, 8]

gini = 0.18
samples = 10
value = [9, 1]

mean texture <= 20.795
gini = 0.375
samples = 20
value = [5, 15]

gini = 0.208
samples = 17
value = [15, 2]

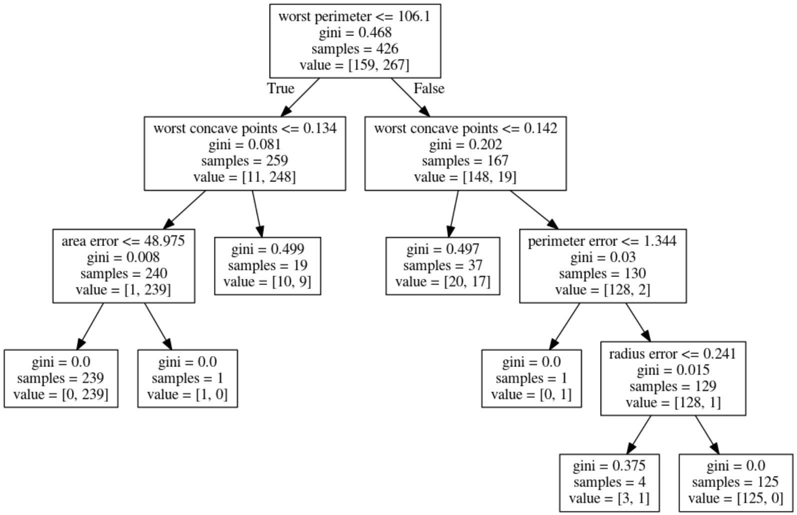
gini = 0.133
samples = 14
value = [1, 13]

mean texture <= 24.985
gini = 0.444
samples = 6
value = [4, 2]

gini = 0.0
samples = 4
value = [4, 0]

gini = 0.0
samples = 2
value = [0, 2]

min_samples_split = 50



worst perimeter <= 106.1
gini = 0.468
samples = 426
value = [159, 267]

True

False

worst concave points <= 0.134
gini = 0.081
samples = 259
value = [11, 248]

worst concave points <= 0.142
gini = 0.202
samples = 167
value = [148, 19]

area error <= 48.975
gini = 0.008
samples = 240
value = [1, 239]

gini = 0.499
samples = 19
value = [10, 9]

gini = 0.497
samples = 37
value = [20, 17]

perimeter error <= 1.344
gini = 0.03
samples = 130
value = [128, 2]

gini = 0.0
samples = 239
value = [0, 239]

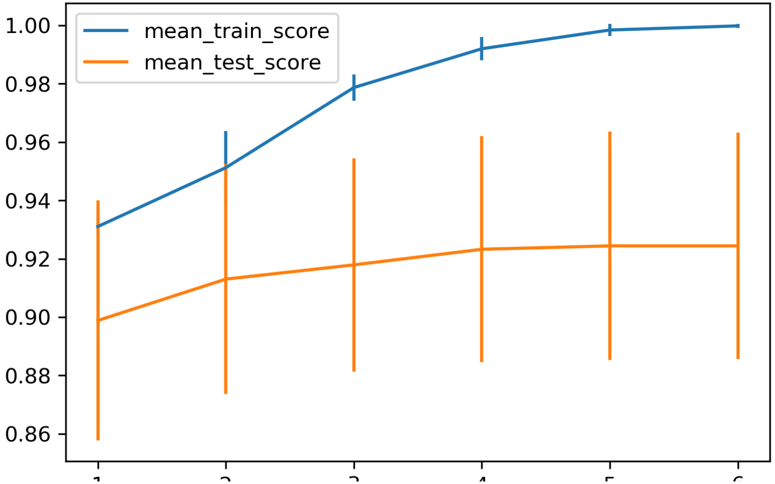
gini = 0.0
samples = 1
value = [1, 0]

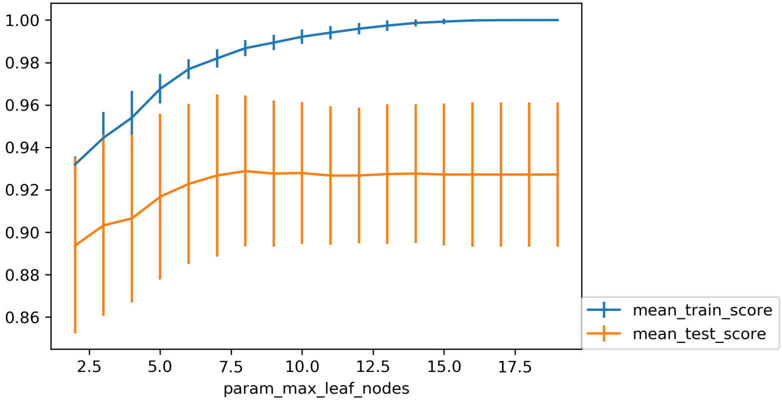
gini = 0.0
samples = 1
value = [0, 1]

radius error <= 0.241
gini = 0.015
samples = 129
value = [128, 1]

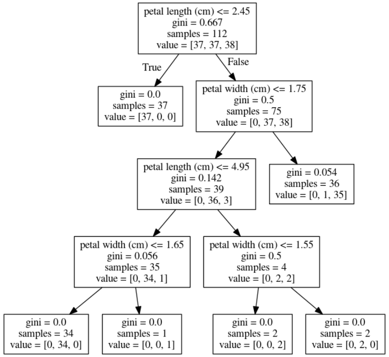
gini = 0.375
samples = 4
value = [3, 1]

gini = 0.0
samples = 125
value = [125, 0]


mean_train_score
mean_test_score
1.00
0.98
0.96
0.94
0.92
0.90
0.88
0.86
1
2
3
4
5
6


1.00
0.98
0.96
0.94
0.92
0.90
0.88
0.86
2.5
5.0
7.5
10.0
12.5
15.0
17.5
param_max_leaf_nodes
mean_train_score
mean_test_score

Instability



petal length (cm) <= 2.45
gini = 0.667
samples = 112
value = [37, 37, 38]

True

False

gini = 0.0
samples = 37
value = [37, 0, 0]

petal width (cm) <= 1.75
gini = 0.5
samples = 75
value = [0, 37, 38]

petal length (cm) <= 4.95
gini = 0.142
samples = 39
value = [0, 36, 3]

gini = 0.054
samples = 36
value = [0, 1, 35]

petal width (cm) <= 1.65
gini = 0.056
samples = 35
value = [0, 34, 1]

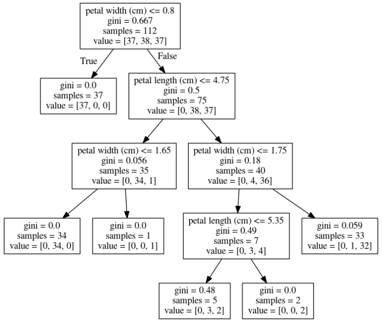
petal width (cm) <= 1.55
gini = 0.5
samples = 4
value = [0, 2, 2]

gini = 0.0
samples = 34
value = [0, 34, 0]

gini = 0.0
samples = 1
value = [0, 0, 1]

gini = 0.0
samples = 2
value = [0, 0, 2]

gini = 0.0
samples = 2
value = [0, 2, 0]



petal width (cm) <= 0.8
gini = 0.667
samples = 112
value = [37, 38, 37]

True

False

gini = 0.0
samples = 37
value = [37, 0, 0]

petal length (cm) <= 4.75
gini = 0.5
samples = 75
value = [0, 38, 37]

petal width (cm) <= 1.65
gini = 0.056
samples = 35
value = [0, 34, 1]

petal width (cm) <= 1.75
gini = 0.18
samples = 40
value = [0, 4, 36]

gini = 0.0
samples = 34
value = [0, 34, 0]

gini = 0.0
samples = 1
value = [0, 0, 1]

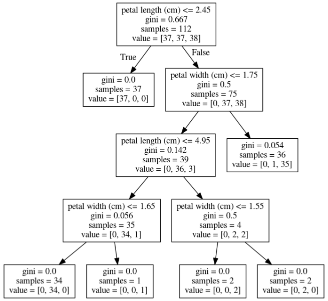
petal length (cm) <= 5.35
gini = 0.49
samples = 7
value = [0, 3, 4]

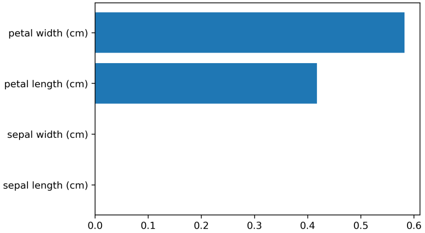
gini = 0.059
samples = 33
value = [0, 1, 32]

gini = 0.48
samples = 5
value = [0, 3, 2]

gini = 0.0
samples = 2
value = [0, 0, 2]

Feature importance





Ensemble Models
(Random Forests)

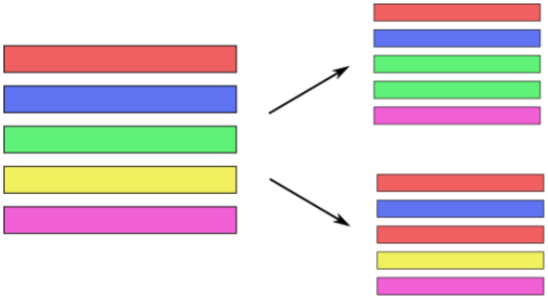
Poor man's ensembles

- Build different models
- Average the result
- More models are better – if they are not correlated.
- Also works with neural networks
- You can average any models as long as they provide calibrated ("good") probabilities.
- Scikit-learn: VotingClassifier

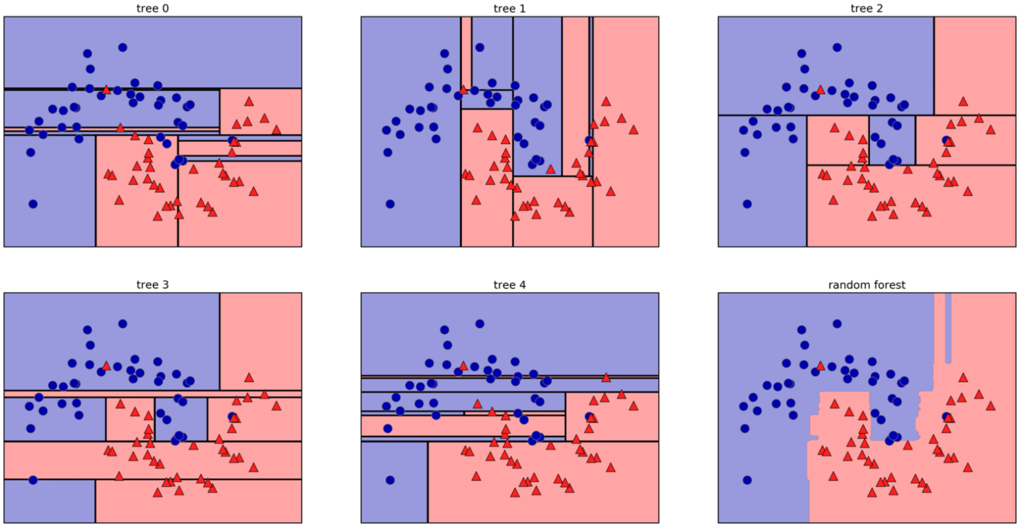
Bagging (Bootstrap AGGregation)

- Generic way to build “slightly different” models

- BaggingClassifier, BaggingRegressor

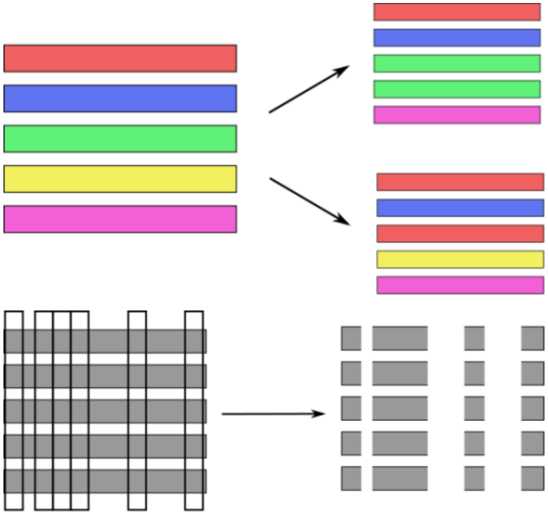


Random Forests



Randomize in two ways

- For each tree:
  - Pick bootstrap sample of data
- For each split:
  - Pick random sample of features
- More trees are always better



Tuning Random Forests

- Main parameter: max_features
  - around sqrt(n_features) for classification
  - Around n_features for regression
- n_estimators > 100
- max_depth, max_leaf_nodes, min_samples_split again