Trees and Forests

PSY-GS 8875 Behavioral Data Science

Overview

Overview: Week 6

Readings

- ESL Chapters: 8.7, 9.2, 10.1-10.9, and 15
- HML Chapters: 9-12
- Fife and D'Onofrio - 2023

Optional

- Strobl et al. - 2009
- Goretzko and Bühner - 2020
- Breiman - 2001 - random forest

- Classification and Regression Trees (CART)
- Bagging
- Random Forests
- Boosting
- Activity: predicting functioning and schizophrenia symptoms with schizotypy

Schizotypy

Schizotypy

Rather than jumping into the models, we'll started by getting acquainted with our data...

schizotypy: susceptibility or vulnerability to develop schizophrenia-spectrum disorders

> *Schizotypy offers explanatory power for understanding the development and expression of schizophrenic psychopathology, and it encompasses a broad sprectrum of conditions including schizophrenia and related disorders, personality disorders, the prodrome, and subclinical expressions. Schizotypy, and by extension schizophrenia, are heterogeneous in etiology, symptoms, and treatment response.*

– Gross et al. (p. 397, 2014)

Schizotypy

Wisconsion Schizotypy Scales – Short Forms (60 items; Gross et al., 2015)

negative: characteristics/symptoms that are **absent** that are present in typical-developing populations

Schizotypy

Wisconsion Schizotypy Scales – Short Forms (60 items; Gross et al., 2015)

negative: characteristics/symptoms that are **absent** that are present in typical-developing populations

- social anhedonia (15 items): asociality and indifference to others
  - `SA07`: *I don't really feel close to my friends*
  - `SA04`: *Just being with friends can make me feel good* (reversed)

Schizotypy

Wisconsion Schizotypy Scales – Short Forms (60 items; Gross et al., 2015)

negative: characteristics/symptoms that are **absent** that are present in typical-developing populations

- social anhedonia (15 items): asociality and indifference to others
  - `SA07`: *I don't really feel close to my friends*
  - `SA04`: *Just being with friends can make me feel good* (reversed)
- physical anhedonia (15 items): deficits in sensory and aesthetic pleasure
  - `PY05`: *The beauty of sunsets is greatly overrated*
  - `PY07`: *It has often felt good to massage my muscles when they are tired or sore* (reversed)

Schizotypy

positive: characteristics/symptoms that are **present** that are absent in typical-developing populations

Schizotypy

positive: characteristics/symptoms that are **present** that are absent in typical-developing populations

- magical ideation (15 items): belief in implausible or invalid causality
  - `MI13`: *I think I could learn to read others' minds if I wanted to*
  - `MI11`: *Numbers like 13 and 7 have no special powers* (reversed)

Schizotypy

positive: characteristics/symptoms that are **present** that are absent in typical-developing populations

- magical ideation (15 items): belief in implausible or invalid causality
  - `MI13`: *I think I could learn to read others' minds if I wanted to*
  - `MI11`: *Numbers like 13 and 7 have no special powers* (reversed)
- perceptual aberration (15 items): schizophrenic-like perceptual and bodily distortions
  - `PB07`: *Sometimes I have had a passing thought that some part of my body was rotting away*
  - `PB10`: *I sometimes have to touch myself to make ure I'm still there*

Schizotypy

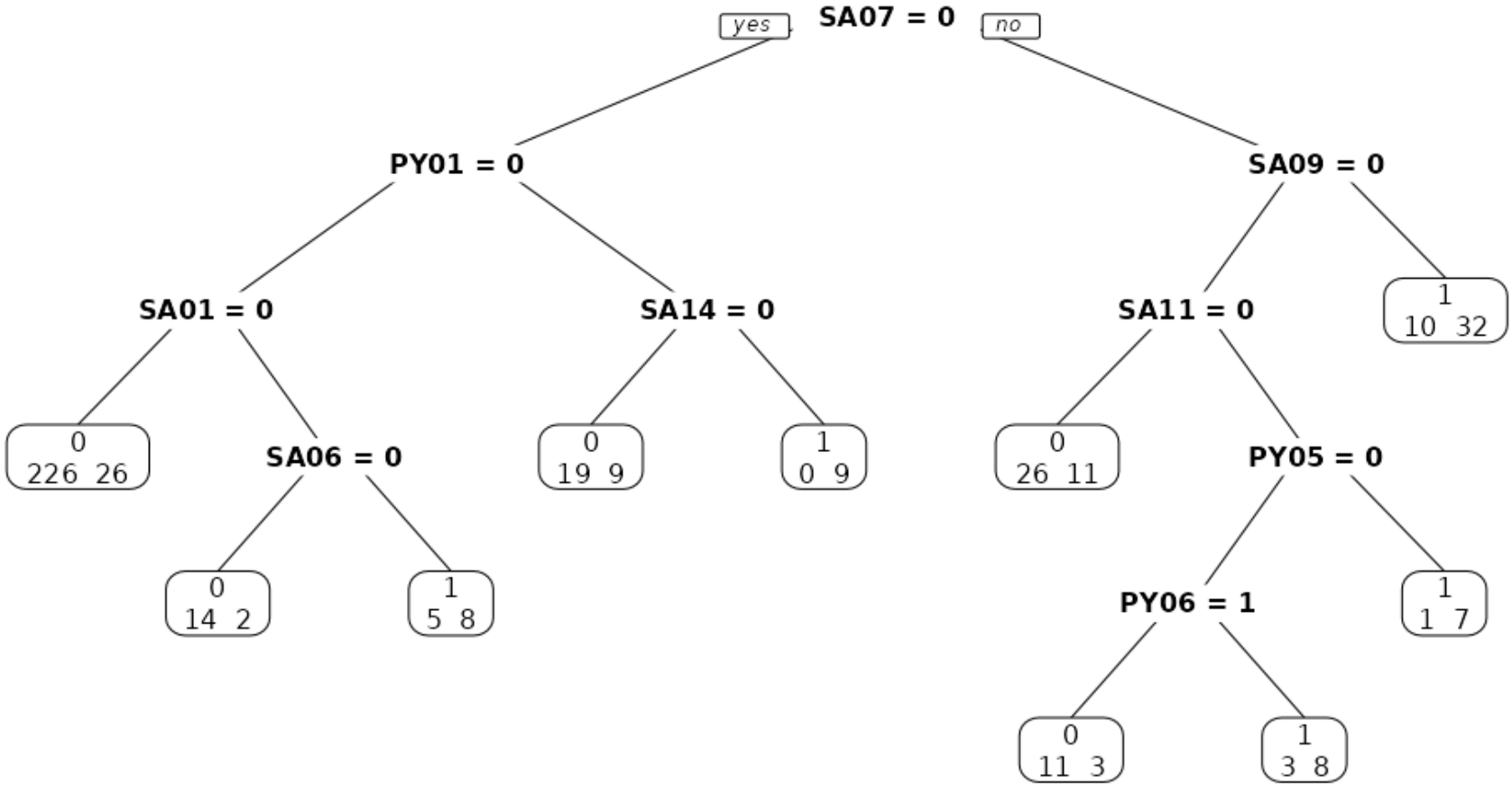
- $N = 430$ people with an oversample of folks higher on overall schizotypy
- People were administered structured interviews (based on DSM-IV) with clinicians who rated the likely presence of schizophrenia-spectrum symptoms
  - `negative` symptoms: alogia (lacking speech), flattened affect, anhedonia, social indifference, avolition/anergia (lacking motivation), impairment in attention
  - psychotic-like (`positive`) symtpoms: transmission of thoughts, passivity experiences, thought withdrawal, voice and auditory hallucinations, abberant beliefs, visual hallucinations, and olfactory hallucinations
  - functioning (`gas`): global assessment scale ranging from 0-100 with marked psychopathology on the low end and "superior" functioning on the high end

For more details, see Kwapil et al., 2008

Classification and Regression Trees

Classification and Regression Trees

Let's start with an example of a decision tree classifying negative symptoms



Classification and Regression Trees

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 296  51
         1  19  64

               Accuracy : 0.8372
                 95% CI : (0.7989, 0.8708)
    No Information Rate : 0.7326
    P-Value [Acc > NIR] : 0.0000001785

                  Kappa : 0.5443

 Mcnemar's Test P-Value : 0.0002112

            Sensitivity : 0.5565
            Specificity : 0.9397
         Pos Pred Value : 0.7711
         Neg Pred Value : 0.8530
             Prevalence : 0.2674
         Detection Rate : 0.1488
   Detection Prevalence : 0.1930
      Balanced Accuracy : 0.7481

       'Positive' Class : 1
```

Compare with logistic regression

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 291  51
         1  24  64

               Accuracy : 0.8256
                 95% CI : (0.7863, 0.8603)
    No Information Rate : 0.7326
    P-Value [Acc > NIR] : 0.000003618

                  Kappa : 0.519

 Mcnemar's Test P-Value : 0.00268

            Sensitivity : 0.5565
            Specificity : 0.9238
         Pos Pred Value : 0.7273
         Neg Pred Value : 0.8509
             Prevalence : 0.2674
         Detection Rate : 0.1488
   Detection Prevalence : 0.2047
      Balanced Accuracy : 0.7402

       'Positive' Class : 1
```

Under the Hood

$$\hat{f}(X) = \sum_{m=1}^{t} c_m I(X_1, X_2) \in R_m$$

- X = one variable or feature
- c_m = split-point for variable X
- R_m = region of data points

For $\in R_m$, some data points will be in a different split, so only the values in the current split are considered

Terms

- **nodes**: points in the tree where the data split
- **root node**: dataset before the first split
- **branch**: segments of the node split (flow of decision)
- **terminal**: end point of the branches (no further splitting)

How to Split?

Regression: sum/mean squared error

$$MSE = \frac{\sum_{i \in R_1}(y_i - \bar{y}_1)^2 + \sum_{i \in R_2}(y_i - \bar{y}_2)^2}{n_{R_1} + n_{R_2}}$$

where R_1 and R_2 represent the regions based on the split of X

How to Split?

Classification: Gini impurity

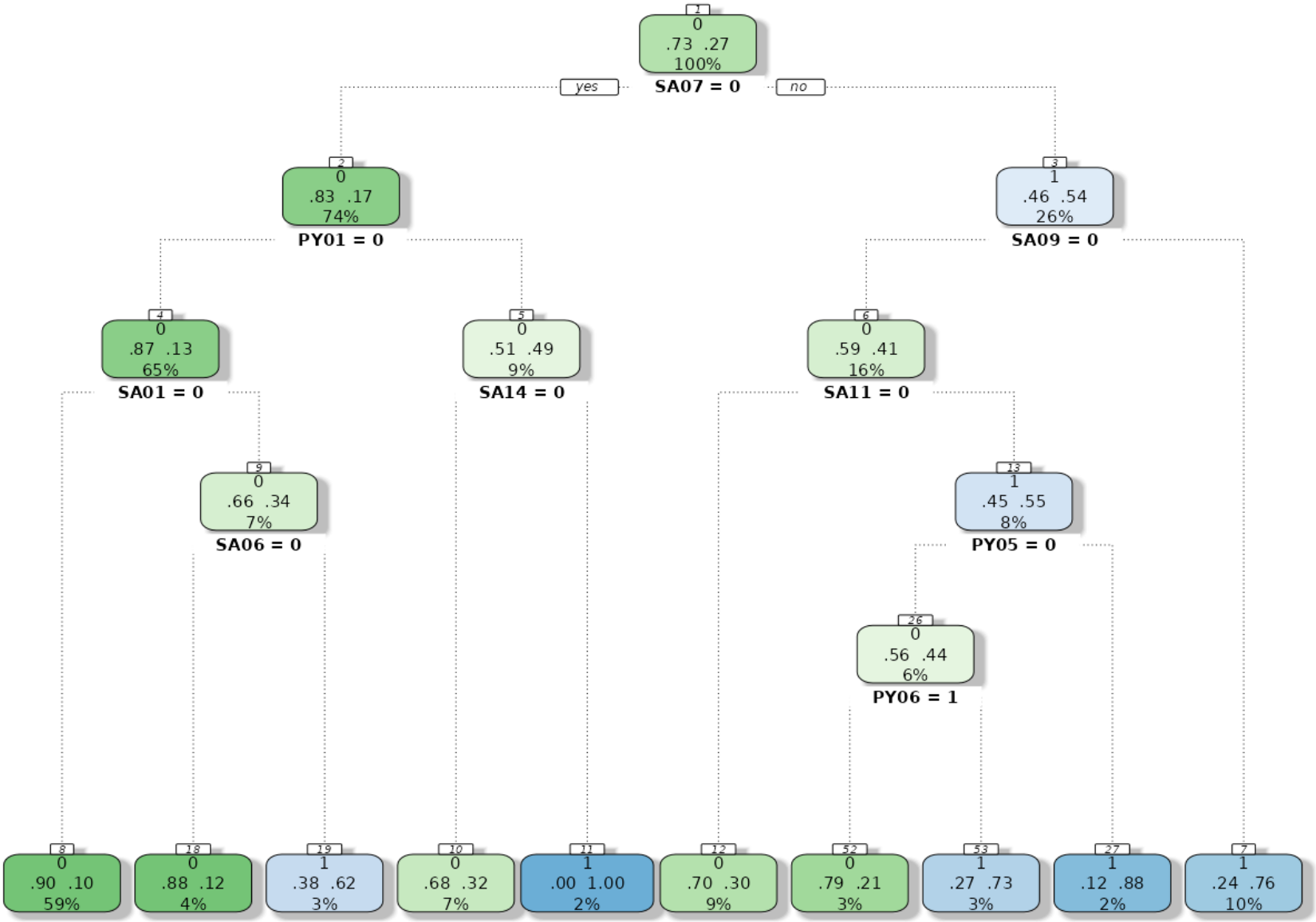
$$Gini = 1 - (p_A^2 - p_B^2)$$

where p_A and p_B are the probabilities of class A and B, respectively

When to Stop?

- Minimum number of samples (need at least N cases to split)
- Maximum depth (only T branches deep)
- Non-substantial improvements in MSE/Gini
- **Pruning**, based on evaluation metric, can be used to remove "less important" branches

Classification and Regression Trees



Limitations?

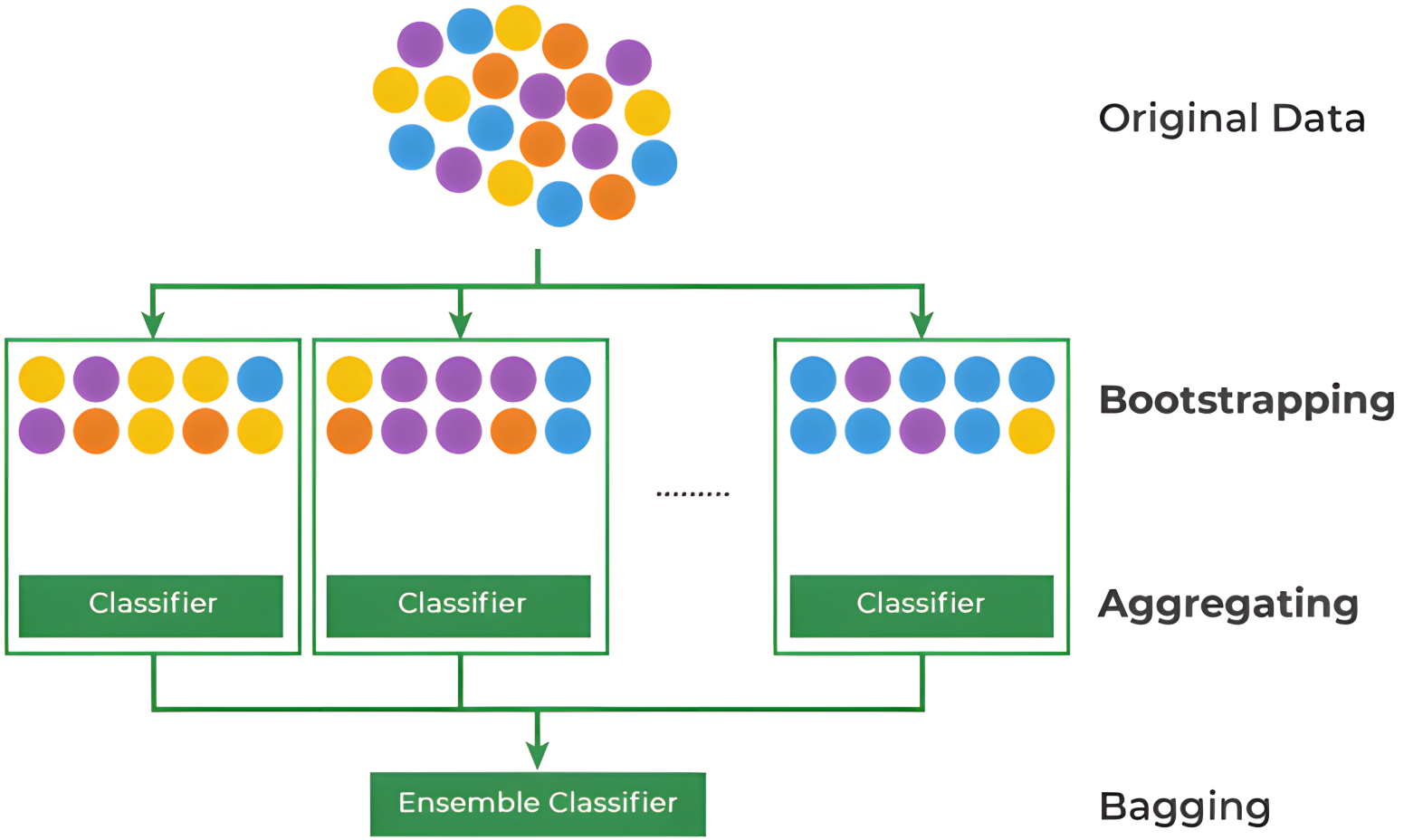
- Single tree – only 9 out of 60 variables were used
- Overfitting – lack of generalizability

Bootstrap **Agg**regation

Bootstrap Aggregation: "bagging"

- A bootstrap strategy to overcome issues related to using a single tree
- Bootstraps with resampling to develop different trees
- Aggregates across bootstraps to arrive at a better model

- regression: average predicted values from across the bootstraps are used
- classification: majority "voting" is used (e.g., 60/40 yes/no = yes)
- Often referred to as an **ensemble** model or many models combined together to make a single prediction/classification


Original Data
Bootstrapping
Classifier
Classifier
Classifier
Aggregating
Ensemble Classifier
Bagging

Why does it work?

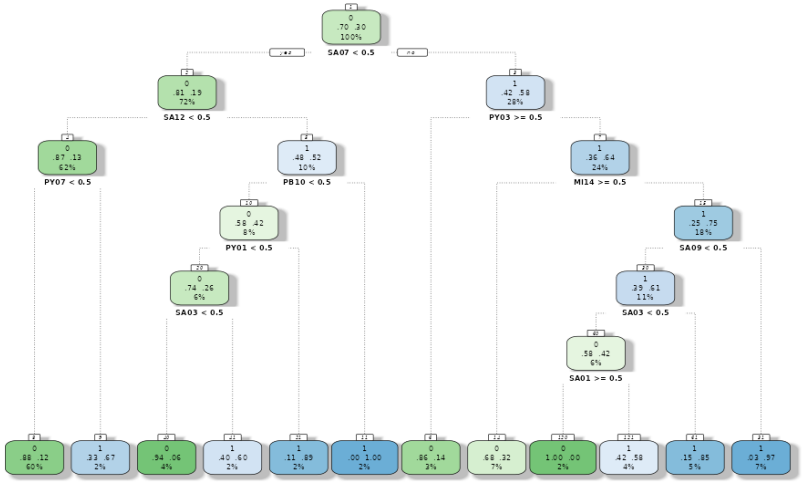
- Trees tend to overfit to their sample (high variance, low bias)
- Although trees will overfit to their respective bootstrap samples, the aggregation across the samples leads to results that are more generalizable
- For regression, there is a "regression-to-the-mean" effect
- For classification, there is a "wisdom-in-the-crowd" effect

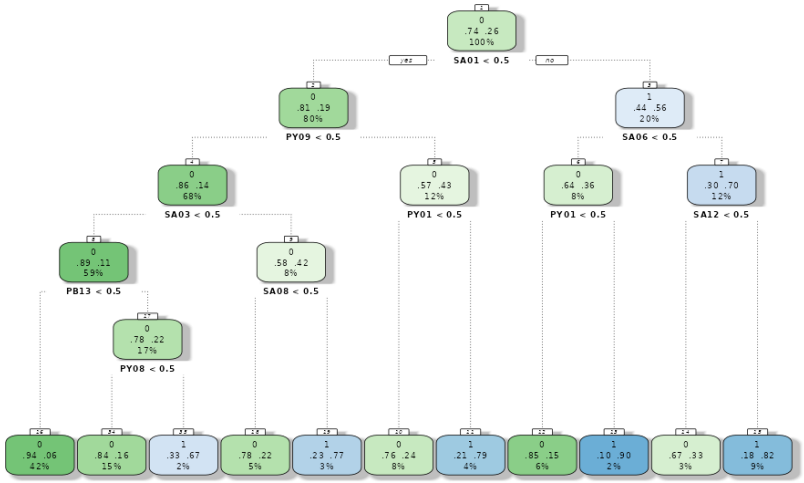
Random Forest

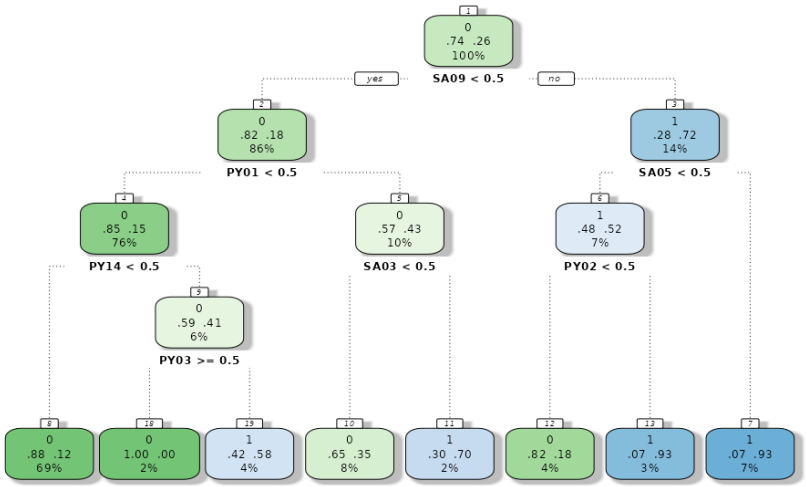
Random Forest

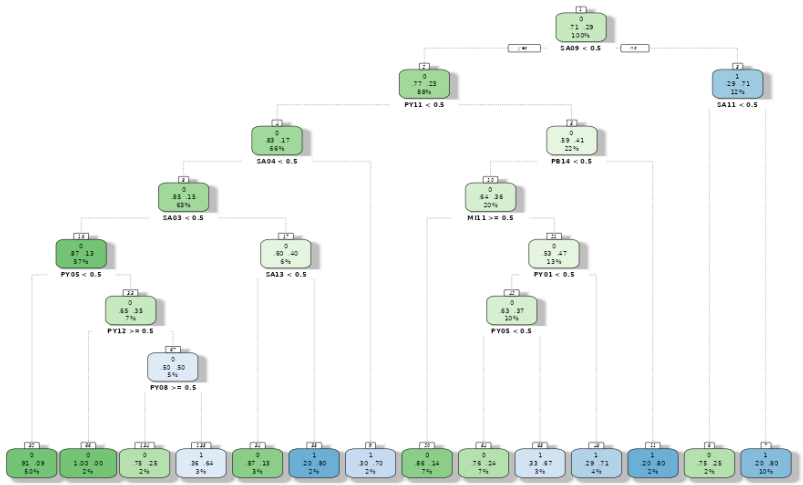
- Uses bagging trees
- An extra layer of randomness is added by only considering a random subset of variables for each tree
- Effect: more diverse trees that are less correlated (usually $\sqrt{p}$)
- Result: better generalizability

Random Forest









Internal Mechanics

- "Out-of-Bag" error estimation: prediction error on data left out of bootstraps
- Strength: correlation between each tree's predicted values and actual values (strong trees = more accurate)
- Correlation between trees: lower correlations = better diversity and (usually) more accurate predictions

Random Forest

Advantages

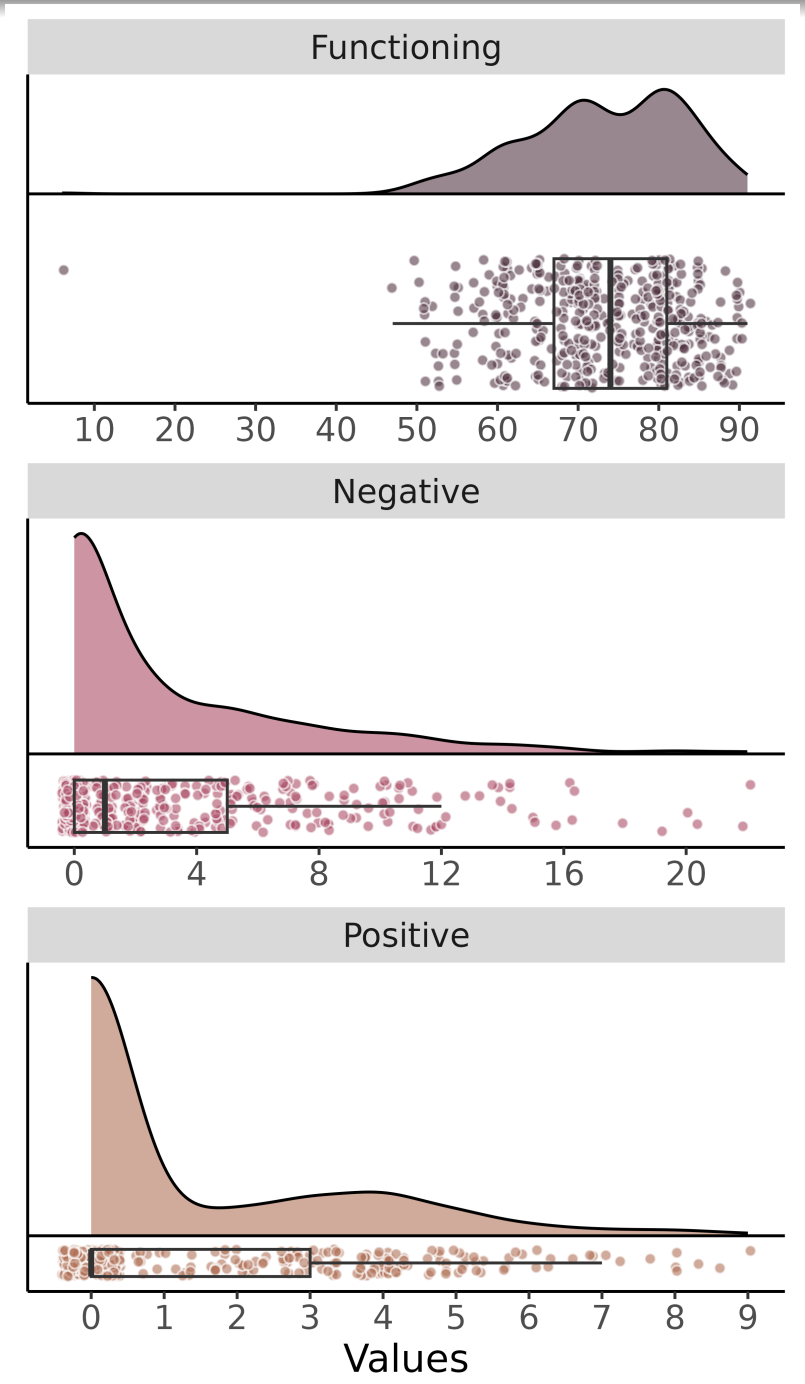
- More robust to outliers: smooths over variance in single trees due to influential variables or cases
- Relative importance: allows importance across variables to be understood (single or bagged trees may over emphasize a few variables)

Limitations

- Less interpretable: many trees means less direct interpretations can be made (though relative importance helps)
- Computationally intensive: time of one tree multiplied by 10, 50, 100, 250, 500, etc.

Random Forest in R

```r
# Load packages
library(ranger); library(caret)

# Load data
schizotypy <- read.csv("../../data/schizotypy/share_430n_interview.csv")

# Fill in `NA` with 0
schizotypy[,4:63][is.na(schizotypy[,4:63])] <- 0

# Set up group variables
## Negative symptoms
schizotypy$negative <- factor(
  as.numeric(schizotypy$nsm_tot > 4), levels = c(0, 1)
)
## Positive symptoms
schizotypy$positive <- factor(
  as.numeric(schizotypy$psx_high > 2), levels = c(0, 1)
)
```


Functioning
10
20
30
40
50
60
70
80
90
Negative
0
4
8
12
16
20
Positive
0
1
2
3
4
5
6
7
8
9
Values

Workflow

1. Perform grid search with k-folds cross-validation for hyperparameters
2. Estimate model with best performing parameters (RMSE/Accuracy/Kappa)
3. Compute variable importance
4. Check out some of the trees (optional)

Random Forest with {ranger}

```r
# Get {ranger} model
ranger_model <- ranger(
  formula = negative ~ ., # `lm` style formula
  data = schizotypy[,c(4:63, 74)], # data
  mtry = 43, # number of variables to split at each node
  # (defaults to `sqrt(p)`)
  splitrule = "gini", # how splits should be made
  min.node.size = 9, # minimum cases before split stop
  num.trees = 250, # number of trees in the forest
  importance = "impurity", # relative importance metric
  seed = 42 # for reproducibility
)
```

Hyperparameters

- `mtry` = number of variables to possibly split at each node (bagging)
  - Defaults to `sqrt(p)`; search along `1` and `p` (variables)
- `min.node.size` = minimum cases to perform split (less than this value will stop splitting)
  - Defaults to `1` (classification) and `5` (regression); search along 1-10 is usually good
- `num.trees` = number of decision trees in the forest (more *neq* better)
  - Defaults to `500`; search along distributed range of low (around 25) and high (around 1500)

Use *k*-folds cross-validation to determine "optimal" values for these parameters

```r
# Set seed for reproducibility
set.seed(42)

# Random forest cross-validation for parameters
store_caret <- train(
  x = schizotypy[,4:63], y = schizotypy$negative,
  method = "ranger", # use {ranger}
  metric = "Kappa", # better for imbalanced datasets
  trControl = trainControl(
    method = "cv", number = 5 #, sampling = "smote"
    # For class imbalances:
    # https://topepo.github.io/caret/subsampling-for-class-imbalances.html
  ),
  tuneGrid = expand.grid(
    mtry = seq(1, 60, 1), # 1:nrow(data)
    min.node.size = seq(1, 10, 1), # 1-10 is usually good
    splitrule = "gini" # classification
  ),
  num.trees = 500, # keep at 500 for the initial search
  importance = "impurity" # set up for later
); store_caret
# This process will take a long time
# On my laptop (16 cores): 83.822 sec elapsed
```

```
Tuning parameter 'splitrule' was held constant at a value of gini
Kappa was used to select the optimal model using the largest value
The final values used for the model were mtry = 43,
splitrule = gini and min.node.size = 9
```

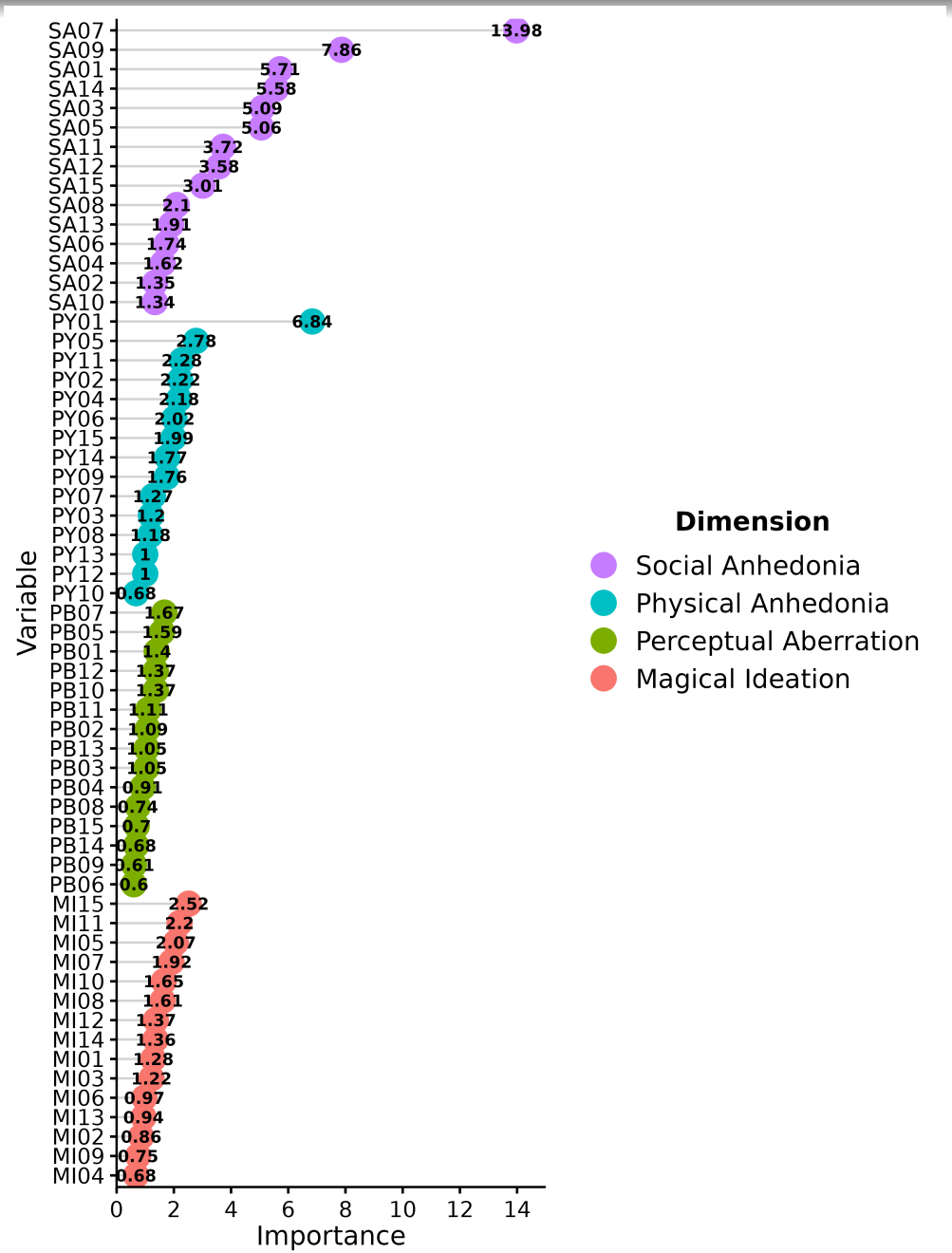
```r
# With the `mtry` and `min.node.size` parameters, search over `num.trees`
trees <- c(10, 50, 100, 250, 500, 1000, 1500)

# Store results
results <- vector("list", length(trees))

# Perform cross-validation (will be much faster than before)
for(i in seq_along(trees)){

  results[[i]] <- train(
    x = schizotypy[,4:63], y = schizotypy$negative,
    method = "ranger", metric = "Kappa",
    trControl = trainControl(method = "cv", number = 5),
    tuneGrid = data.frame(
      mtry = 43, splitrule = "gini",
      min.node.size = 9
    ), num.trees = trees[i],
    importance = "impurity"
  )$results

}

# Combine results
combined <- do.call(rbind.data.frame, results)
combined$num.trees <- trees
combined
```

num.trees	Accuracy	Kappa	AccuracySD	KappaSD
10	0.730	0.273	0.048	0.142
50	0.756	0.319	0.027	0.061
100	0.740	0.278	0.024	0.067
250	0.772	0.384	0.049	0.125
500	0.751	0.318	0.033	0.079
1000	0.753	0.318	0.045	0.084
1500	0.753	0.308	0.061	0.197

Final Model

```r
# Get final model
ranger_model <- ranger(
  formula = negative ~ .,
  data = schizotypy[,c(4:63, 74)],
  mtry = 43, splitrule = "gini",
  min.node.size = 9, num.trees = 250,
  importance = "impurity",
  seed = 42 # for reproducibility
)

# Get predictions
ranger_predictions <- predict(ranger_model, data = schizotypy)$predictions
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 313  18
         1   2  97

               Accuracy : 0.9535
                 95% CI : (0.9291, 0.9714)
    No Information Rate : 0.7326
    P-Value [Acc > NIR] : < 0.00000000000000022

                  Kappa : 0.8758

 Mcnemar's Test P-Value : 0.0007962

            Sensitivity : 0.8435
            Specificity : 0.9937
         Pos Pred Value : 0.9798
         Neg Pred Value : 0.9456
             Prevalence : 0.2674
         Detection Rate : 0.2256
   Detection Prevalence : 0.2302
      Balanced Accuracy : 0.9186

       'Positive' Class : 1
```


SA07 13.98
SA09 7.86
SA01 5.71
SA14 5.58
SA03 5.09
SA05 5.06
SA11 3.72
SA12 3.58
SA15 3.01
SA08 2.1
SA13 1.91
SA06 1.74
SA04 1.62
SA02 1.35
SA10 1.34
PY01 6.84
PY05 2.78
PY11 2.28
PY02 2.22
PY04 2.18
PY06 2.02
PY15 1.99
PY14 1.77
PY09 1.76
PY07 1.27
PY03 1.2
PY08 1.18
PY13 1
PY12 1
PY10 0.68
PB07 1.67
PB05 1.59
PB01 1.4
PB12 1.37
PB10 1.37
PB11 1.11
PB02 1.09
PB13 1.05
PB03 1.05
PB04 0.91
PB08 0.74
PB15 0.7
PB14 0.68
PB09 0.61
PB06 0.6
MI15 2.52
MI11 2.2
MI05 2.07
MI07 1.92
MI10 1.65
MI08 1.61
MI12 1.37
MI14 1.36
MI01 1.28
MI03 1.22
MI06 0.97
MI13 0.94
MI02 0.86
MI09 0.75
MI04 0.68
Variable
0 2 4 6 8 10 12 14
Importance
Dimension
Social Anhedonia
Physical Anhedonia
Perceptual Aberration
Magical Ideation

In-class Activity

In-class Activity

- Use the `share_430n_interview.csv` data to predict positive symptoms (`positive`) with all schizotypy items (columns 4-63)
    - The data wrangling is already handled in the `trees-and-forests.R` script
- Perform random forest classification following the workflow in the `trees-and-forests.R` script
- Compare accuracy and kappa of random forest with logistic regression
- Compare significant (logistic regression) and important (random forest) variables

At Home Activity

At Home Activity

- Use the `share_430n_interview.csv` data to predict general functioning (`gas`) with all schizotypy items (columns 4-63)
- This task is **regression** and *differs* from the in-class example and activity
- Perform linear regression and evaluation with RMSE and R^2
- Perform random forest regression (use cross-validation and grid search to select hyperparameters) and evaluate with RMSE and R^2
- Compare models on:
  - Significant (linear regresion) and important (random forest) variables
  - RMSE and R^2: which model performed better?

Code you'll need

```r
# R-squared and RMSE
continuous_accuracy <- function(prediction, observed)
{

  # Compute square error
  square_error <- (prediction - observed)^2

  # Return metrics
  return(
      c(
        R2 = 1 - (
          sum(square_error, na.rm = TRUE) /
          sum((observed - mean(observed, na.rm = TRUE))^2, na.rm = TRUE)
        ),
        RMSE = sqrt(mean(square_error, na.rm = TRUE))
      )
    )

}

# Usage
continuous_accuracy(prediction, actual)
```

Readings for Next Week

Readings

- ESL Chapters: 14.3
- HML Chapters: 20 and 21

Optional

- Steinley - 2006
- Forbes et al. - 2023

Boosting

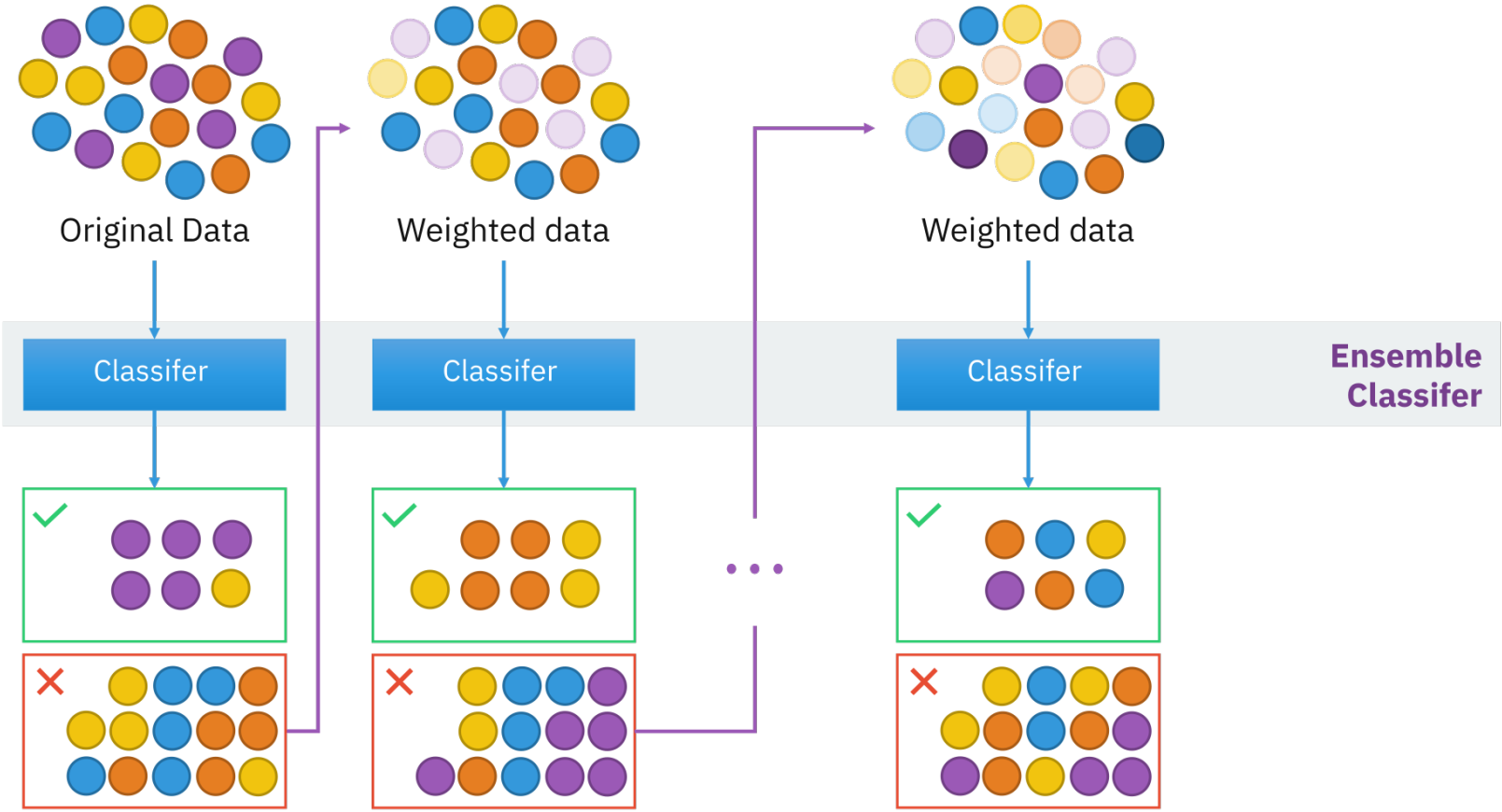
Boosting

Premise

- Large errors (regression) or misclassifications (classification) receive greater weight in the next iteration of model prediction
- Weight good predictions less (the model is already capturing them) and weight bad predictions more
- Different styles
  - Adaptive boosting (adaboost): focuses on errors directly
  - Gradient boosting (gbm): focuses on the gradient (think gradient descent)
  - Extreme gradient boosting (xgboost): trees are built in parallel and focuses across the gradient

Often referred to as the best "out-of-the-box" prediction models

Boosting


Original Data
Weighted data
Weighted data
Classifer
Classifer
Classifer
Ensemble Classifer

R Packages

- Adaptive boosting: {adabag}
- Gradient boosting: {gbm}
- Extreme boosting: {xgboost}