INSEAD

The Business School for the World®

Prof. Anton Ovchinnikov

Prof. Spyros Zoumpoulis

DSB Sessions 7-8, February 7, 2020

Advanced Classification; Overfitting and regularization; From .R to Notebooks

Structure of the course

INSEAD
The Business School for the World®

- SESSIONS 1-2 (AO): Data analytics process; from Excel to R
  - Tutorial 1: Getting comfortable with R
- SESSIONS 3-4 (AO): Time Series Models
- SESSIONS 5-6 (AO): Introduction to classification
- Tutorial 2: Midterm R help / classification
- SESSIONS 7-8 (SZ): Advanced Classification; Overfitting and Regularization; From .R to Notebooks
- Tutorial 3: Setup with GitHub and knitting notebooks
- SESSIONS 9-10 (SZ): Dimensionality Reduction; Clustering and Segmentation
- SESSIONS 11-12 (SZ): AI in Business; The Data Science Process; Guest speaker
  - Hands-on help with projects
- SESSIONS 13-14 (AO+SZ): Project presentations

Plan for the day
Learning objectives

INSEAD
The Business School for the World®

- Assignment 2
- Advanced Classification: more metrics and methods
- Overfitting & Regularization
- Feature Engineering
- From .R scripts to Notebooks
  - New way/process for doing and communicating analytics with reproducible, publication-quality output

Assignment 2…

Overfitting…

INSEAD
The Business School for the World®

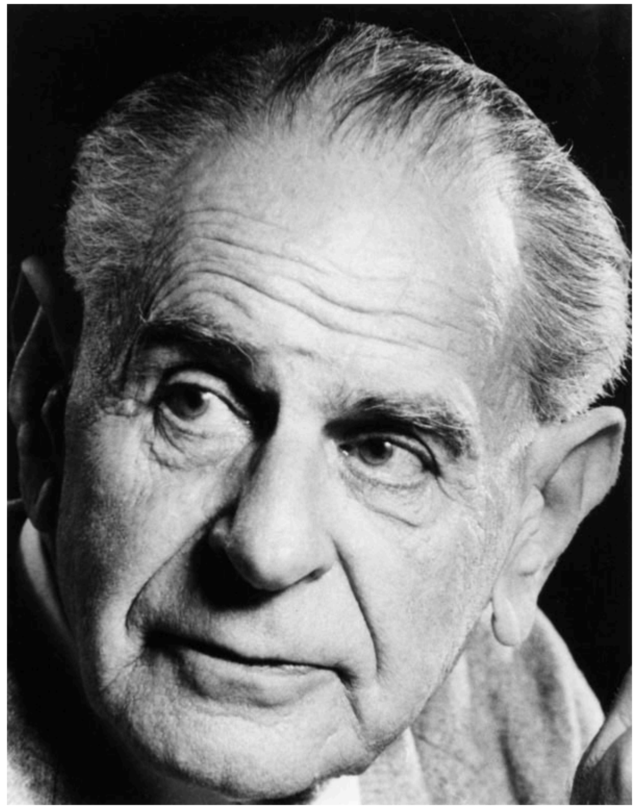
- What happened when in Assignment 2, you made a `rpart` CART tree with very small cp?
- Fundamental tradeoff of learning with data
  - Models that are too simple: are not accurate on the training set, nor are they accurate on the test set
  - Models that are too complex: are very accurate on the training set, but don't generalize well on the test set…
    - …exactly because they too closely capture the nuances of the training set, which may not be present in testing.

Overfitting...

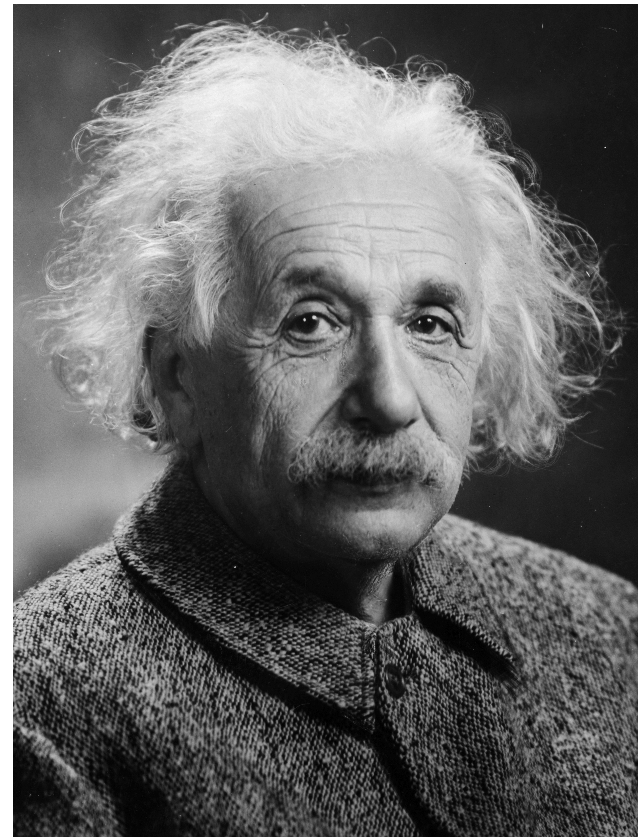
INSEAD

The Business School for the World®

Immanuel Kant

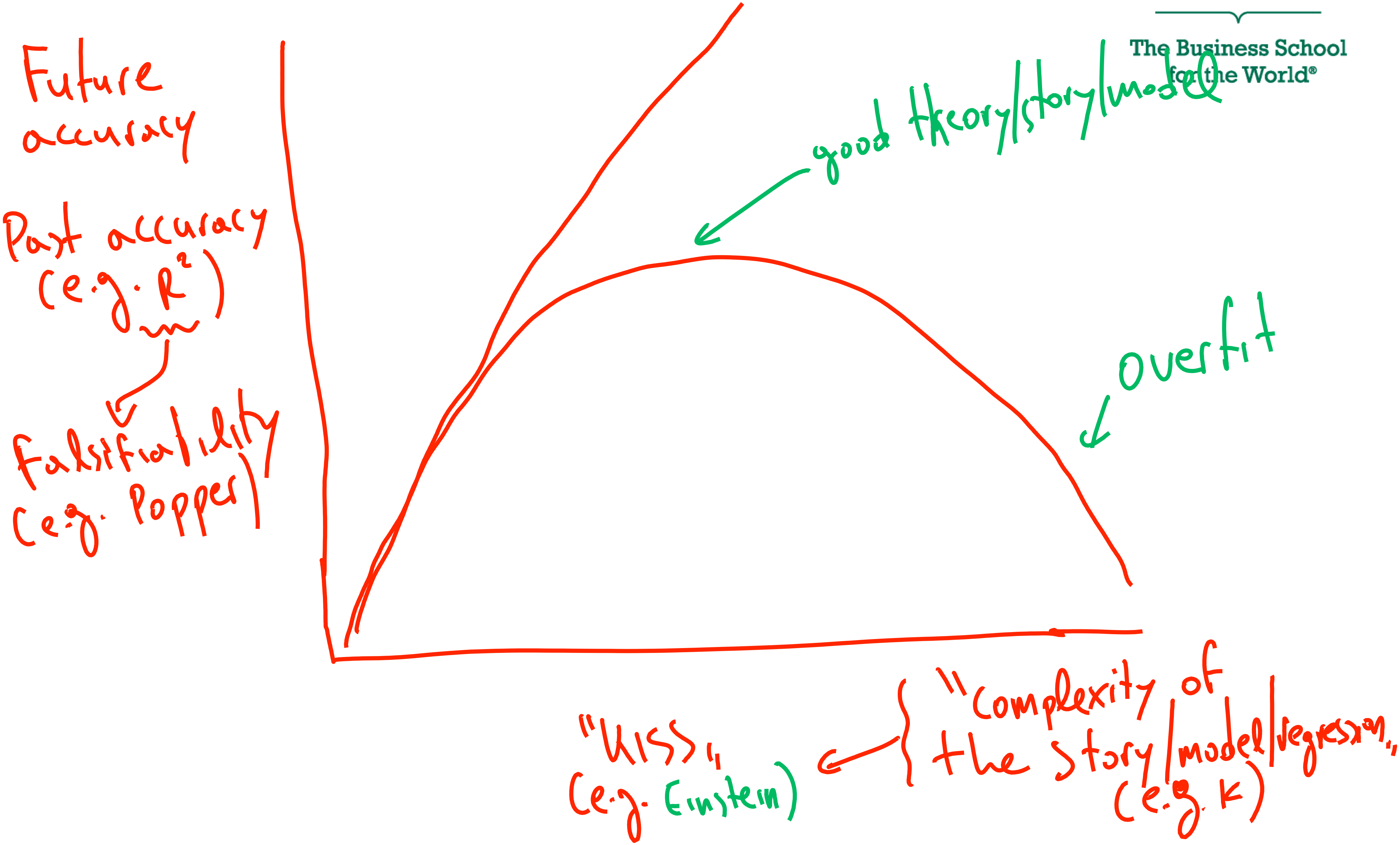


Karl Popper



Albert Einstein

GOOD THEORIES



Cross-validation

INSEAD
The Business School for the World®

- Need to fine-tune the model so that is strikes a good balance between accuracy and simplicity
- Cross-validation does this fine-tuning
  - Break the data into training data, validation data, test data
  - Train model using training data
  - Test on validation data to fine-tune parameters, and iterate
  - "When happy," test (once) on test data to simulate how model would do in the real world

Regularization

INSEAD
The Business School for the World®

- Regularization: set of techniques to reduce overfitting
  - For logistic regression (β are the coefficients):

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left\{ -\log likelihood(\beta, data) + \lambda \left(\frac{1-\alpha}{2} \sum_i \beta_i^2 + \alpha \sum_i |\beta_i| \right) \right\}$$

measures **fit**

controls trade off between maximizing fit and minimizing complexity

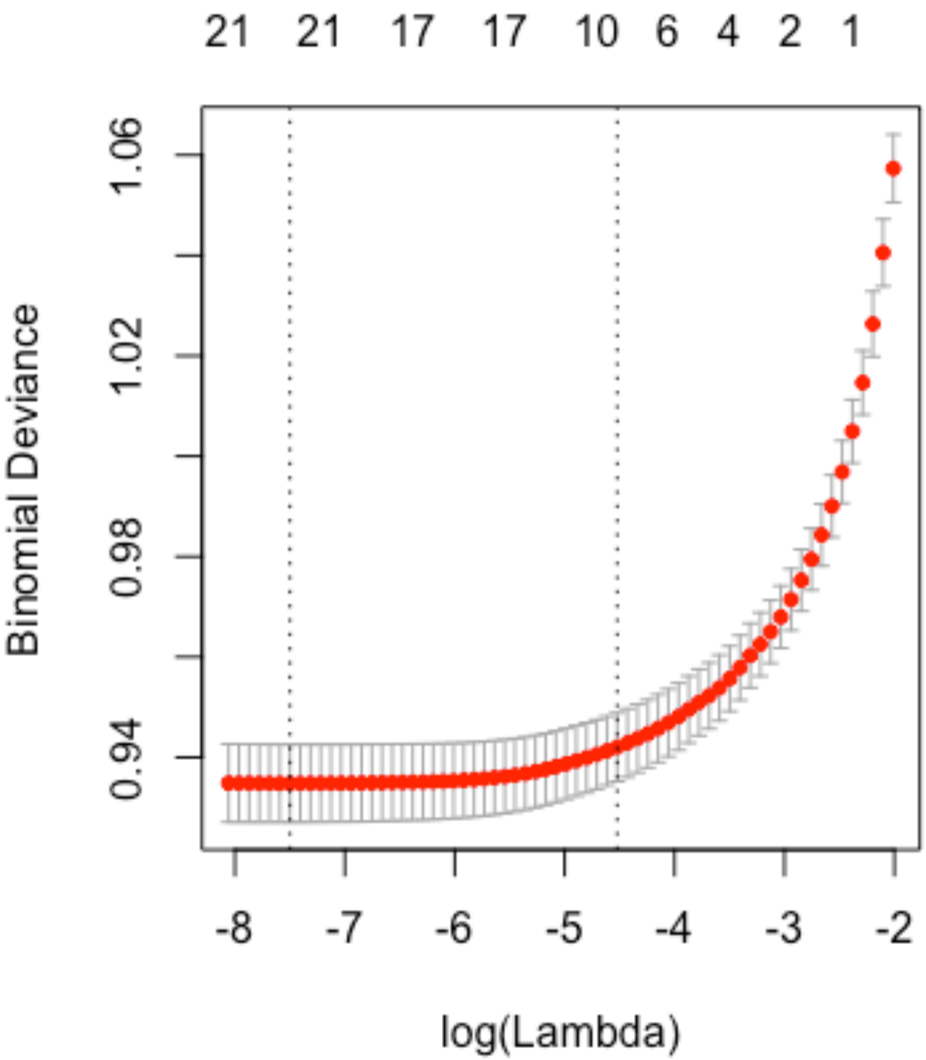
measures **complexity**

  - α= 1: penalize sum of absolute values of coefficients. Lasso regression
  - α=0: penalize sum of squares of coefficients. Ridge regression

```
Package: glmnet
cv.out <- cv.glmnet(as.matrix(estimation_data[,independent_variables]),estimation_data[,dependent_variable],alpha=1,
                    family="binomial" )
#family= "binomial" => logistic regression
#alpha=1: Lasso
lambda <- cv.out$lambda.1se #choose value of λ
log_reg_coefficients <- as.matrix(coef(cv.out,s=lambda)) #extract the estimated coefficients
```

Overfitting & Regularization

> plot(cv.out)



- λ that minimizes mean cross-validated error:

> log(cv.out$lambda.min)

[1] -7.498859

- Largest λ s.t. error is within 1 standard error of the minimum:

> log(cv.out$lambda.1se)

[1] -4.52178

Emphasizes simplicity (even) more

Back to Assignment 2…
Time to make decisions

INSEAD
The Business School for the World®

Important classification metric: Profit Curve

INSEAD
The Business School for the World®

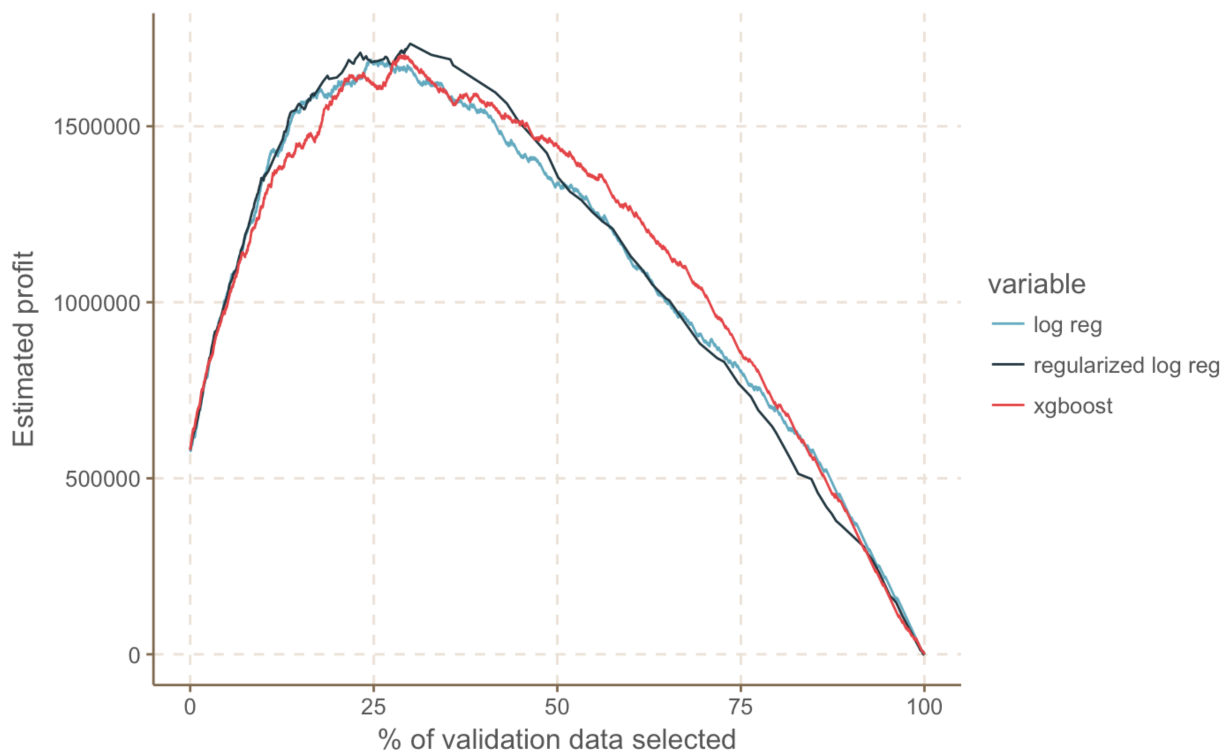
- Measure business profit if we only select the top cases in terms of the probability of "response"
- For this, we need to define values and costs of correct classifications and misclassifications

	Actual: default	Actual: no default
Predicted: default	\$0	\$0
Predicted: no default	-\$5000	\$1500

Profit = # of 1's correctly predicted * value of capturing a 1
+# of 0's correctly predicted * value of capturing a 0
+# of 1's incorrectly predicted as 0 * cost of missing a 1
+# of 0's incorrectly predicted as 1 * cost of missing a 0

Important classification metric: Profit Curve

- Given a classifier, rank instances in the test data from highest predicted probability of belonging to class 1 (= default) to lowest
- Can put the cutoff for giving vs. not giving credit at any rank
- As I move the cutoff, calculate the corresponding profit…



Back to Assignment 2…
Feature engineering?

INSEAD

The Business School for the World®

Feature Engineering

INSEAD
The Business School for the World®

Your data may have more information than what is contained in your existing variables

- Spend lots of time thinking of ways to combine your variables into new ones!
- “Engineering” good features may be more important than using a better method
- Requires contextual knowledge of the business
  - Can not be outsourced

Feature Engineering

INSEAD
The Business School for the World®

Example for credit card default case

(Code on Github repo: INSEADAnalytics/CourseSessions/ClassificationProcessCreditCardMoreMethods.Rmd):

```
tmpx = t(apply(ProjectData[,7:12], 1,
                    function(r) matrix(c(sum(r==-2), sum(r==-1), sum(r==0),sum(r > 0)), nrow=1)))
                    #apply: apply the function to an array of values
                    # argument “1”: apply the function over rows
                    # Summarize the PAY variables for each customer with a vector of how
                    many -2s, -1’s, 0’s, >0’s
ProjectData = cbind(ProjectData[,2:5], #cbind: combine a set of columns
                      tmpx,
                      apply(ProjectData[,13:18], 1, function(r) median(r[!is.na(r)])),
                    # Replace the BILL_AMT variables for each customer with their median
                      apply(ProjectData[,19:24]/ProjectData[,13:18], 1, function(r)
                             ifelse(sum(!is.na(r) & !is.infinite(r)), mean(r[!is.na(r) & !is.infinite(r)]),0)),
                    # Replace the PAY_AMT variables for each customer with the  mean of the ratio of
                    PAY_AMT/BILL_AMT (paid over consumed)
                       ProjectData[,25])


dependent_variable = 11
independent_variables = c(1:10) # use all the new attributes
```

Back to Assignment 2…

Sensitivity and Specificity

INSEAD
The Business School for the World®

		True condition	
	Total population	Condition positive	Condition negative
Predicted condition	Predicted condition positive	**True positive**, Power	**False positive**, Type I error
	Predicted condition negative	**False negative**, Type II error	**True negative**
		True positive rate (TPR), Recall, Sensitivity, probability of detection $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$
		False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$

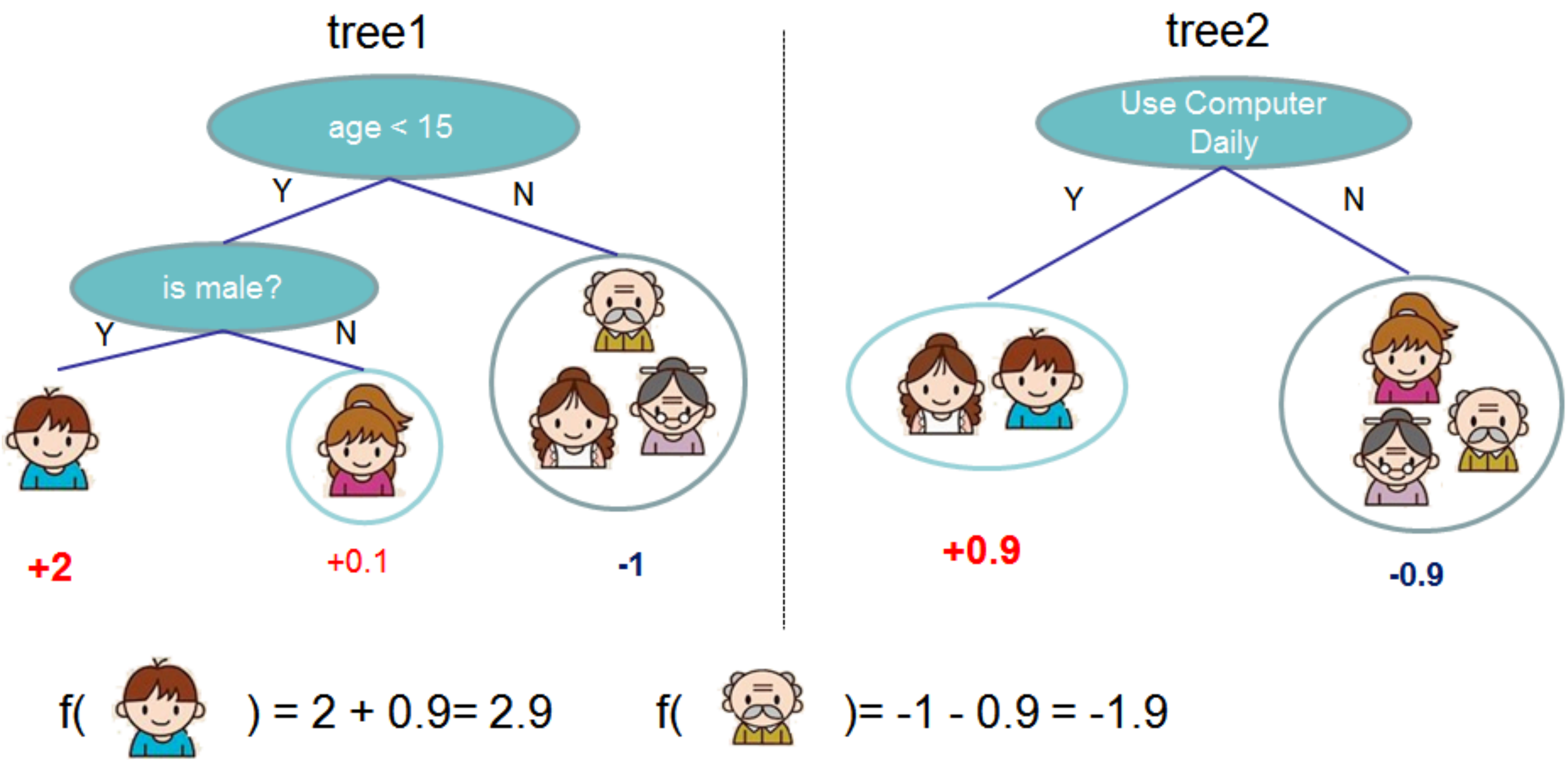
Source: Wikipedia

Tree Ensemble Methods

INSEAD
The Business School for the World®

- Main idea: put a set of CARTs together, output a combination (e.g., mode, mean) of the respective outputs the CARTs

Does someone like computer games?



Source: http://xgboost.readthedocs.io/en/latest/model.html#

Tree Ensemble Methods

INSEAD
The Business School for the World®

Both **random forests** and **boosted trees** generate multiple random samples from the training set (with replacement), and train a different CART for each sample of the data. This is called bagging.

- Random Forests
  - The samples are completely random. No adaptiveness.
  - Use fully grown CARTs (each with low bias, high variance). Reduce variance by bagging together many uncorrelated trees.
  - Final prediction is the simple average

- Boosted trees
  - Based on small trees: weak learners with high bias, low variance
  - But adaptive: instances modeled poorly by the overall system before, have larger probability of being picked now → higher weight
  - Final prediction is a weighted average

Tree Ensemble Methods

INSEAD
The Business School for the World®

- Random Forests

Package: `randomForest`

```
model_forest <- randomForest(x=estimation_data[,independent_variables],
                             y=estimation_data[,dependent_variable],
                             importance=TRUE, proximity=TRUE, type="classification”)
```

- Boosted trees

Package: `xgboost`

```
model_xgboost <- xgboost(data = as.matrix(estimation_data[,independent_variables]),
                         label = estimation_data[,dependent_variable],
                         eta = 0.3, max_depth = 10, nrounds=10, objective = "binary:logistic",
verbose = 0)
#objective= "binary:logistic" => logistic regression for classification
#eta: step size of each boosting step. max.depth: maximum depth of tree.
#nrounds: the max number of iterations
```

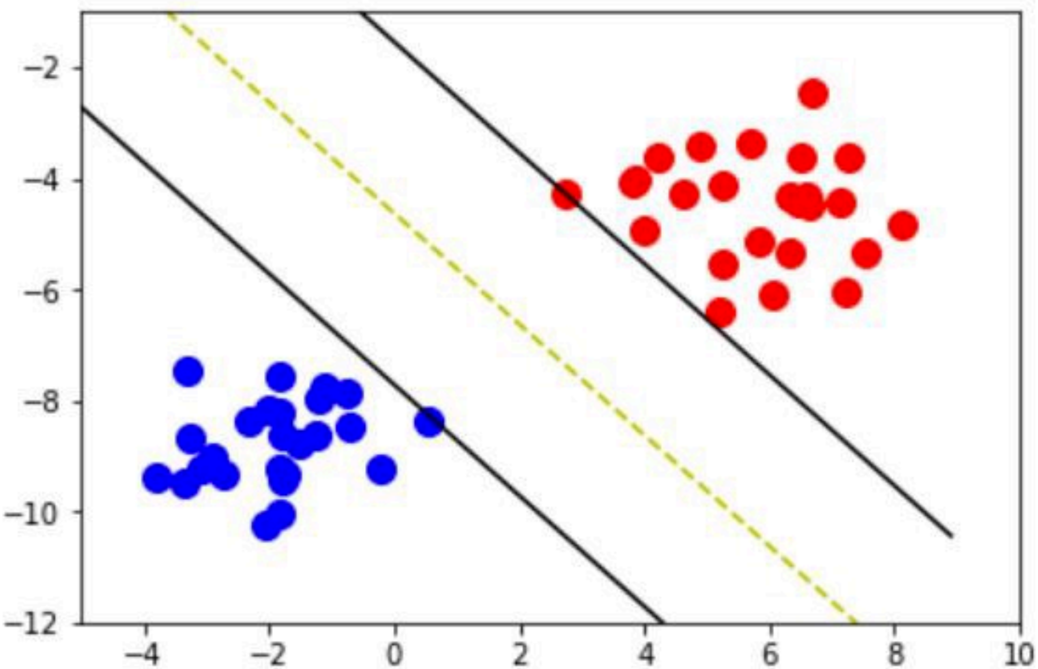
How to then retrieve predicted probabilities (and therefore also classes)?

```
validation_Probability_class1<-
        predict(model,newdata=as.matrix(validation_data[,independent_variables]),
                                  type=  "prob" )
```

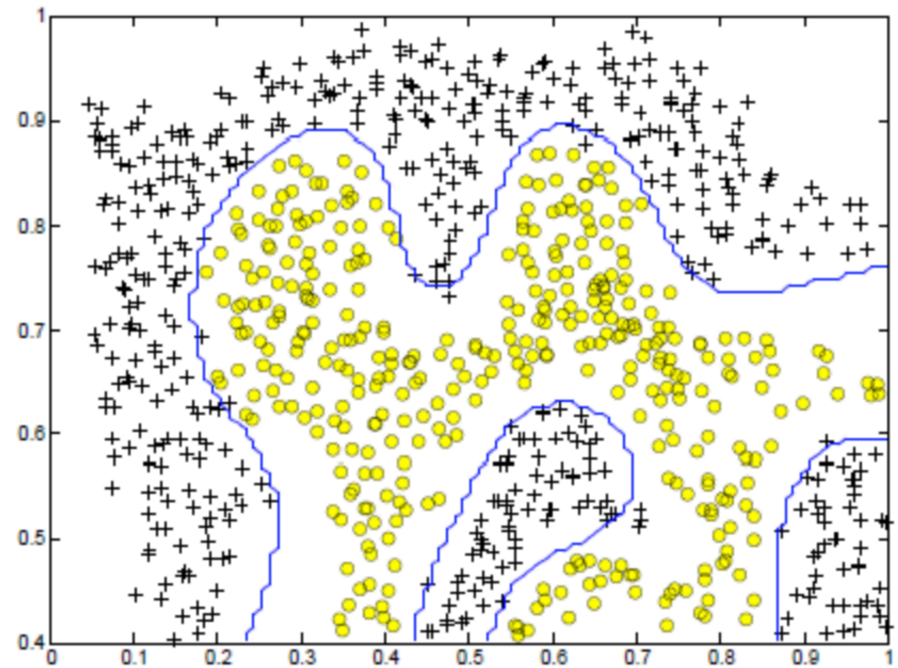
Support Vector Machines

INSEAD
The Business School for the World®

- Main idea
  - Training: Divide parameter space in two regions using maximum-margin hyperplanes, based on training set.
  - Decision: read the label of the region where the new instance falls



Linear kernel



Radial basis (Gaussian) kernel

Package: `e1071`

Model_svm <- svm(Retained.in.2012.~., data=training)

#Can choose the kernel, and parameters such as the kernel parameter, the cost of constraint violations, etc. Default is radial kernel.

(A) Process for Classification

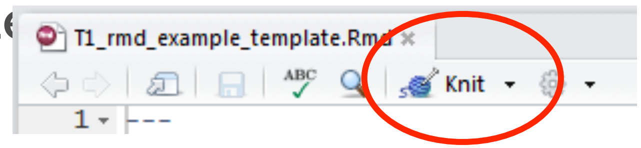
INSEAD
The Business School for the World®

1. Split the data
2. Set up the dependent variable
3. Simple Analysis
4. Classification and Interpretation
5. Validation accuracy
   - Use various classification metrics you know
6. Test accuracy

From R to Notebooks

- You traditional approach for "using" analytics has been two-step:
  - "do" analytics (e.g., plot a graph in Excel)
  - "communicate" analytics (e.g., copy-paste the graph into a PowerPoint presentation / Word file report, etc.)

- With coding (and R) there is a better way: "notebooks"
  - "knit" the R markdown (*.Rmd) file



  - This will create a *.html report (a webpage) with the analysis outputs, graphs, text. Can also create a PDF report
  - Main advantage of this approach: ALL IN ONE PLACE
    - When the new data is available (e.g., next quarter's sales numbers come in), creating an updated report will take you… 1 click
  - Along with sharing tools (GitHub): reusable, replicable, easy to share, all-in-one-place way of doing and communicating analytics with publication-quality output

The course on GitHub

- The course's GitHub repo:

  github.com/InseadDataAnalytics/INSEADAnalytics

  - For next time, you get set up with GitHub and copy the repo on your machine
  - You find there code – really, templates for business solutions – for
    - classification material covered today
    - dimensionality reduction and clustering, covered next time

- Course website on GitHub (parallel to Canvas)

  inseaddataanalytics.github.io/INSEADAnalytics/home.html

- Issues page:

  github.com/InseadDataAnalytics/INSEADAnalytics/issues/

Summary of Sessions 7-8

INSEAD
The Business School for the World®

- Advanced classification:
  - Profit curve, more methods (regularized regression, XGBoost, SVM), a process for classification
- Overfitting and regularization
- Feature engineering
- From R scripts to Notebooks
  - New way/process for doing and communicating analytics with reproducible, publication-quality output

Next…

INSEAD
The Business School for the World®

- Tutorial 3: [Tonight Fri, Feb 7]
  - Set up with GitHub repo and knitting

- Sessions 9-10: [Tue, Feb 11]
  - Dimensionality Reduction/Cluster Analysis and Segmentation
  - Please come to class having set up and knitted MarketSegmentationProcessInClass.Rmd
  - BOR – work on the market segmentation process for the Boats (A) case

- Assignment 3 (due Feb 14):
  - Complete the market segmentation process for the Boats (A) case

- Proposal for Final Project (due Feb 14)

Final Project (due before last class)

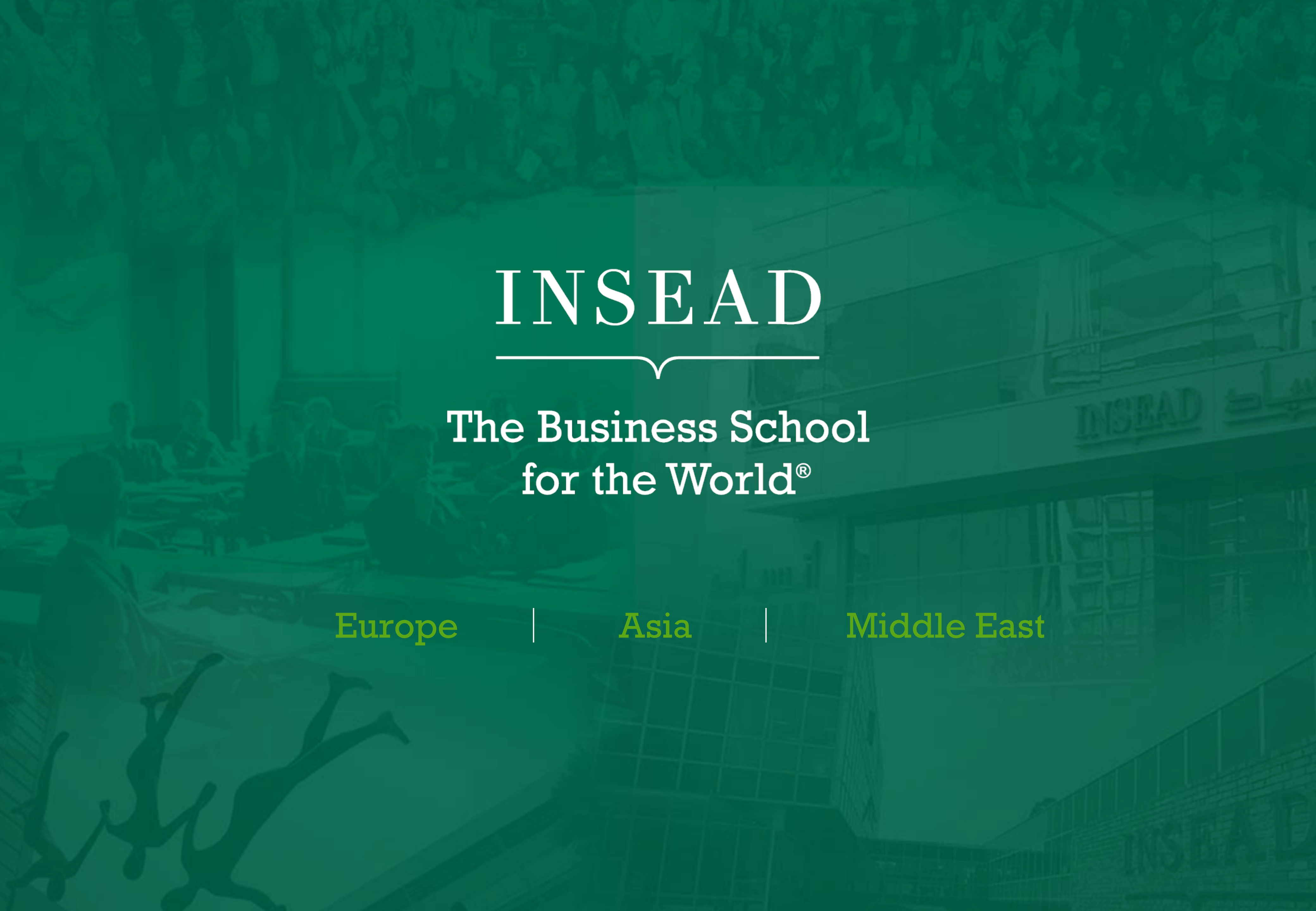
INSEAD
The Business School for the World®

- Develop a data analytics solution to a business problem
  - Relevant business problem, ideally from your past or future workplace
  - Develop a process for how to solve the problem with steps codified in a notebook
  - Show application on a dataset
  - Draw relevant and actionable business insights

- You are expected to share the data you use

- Examples of past projects on GitHub course website

- You will present in class



INSEAD

The Business School for the World®

Europe | Asia | Middle East