Applied Machine Learning for Educational Data Scientists

EDLD 654

Joe Nese

Daniel Anderson

An Introduction to the Course

Week 1, Class 2

Agenda

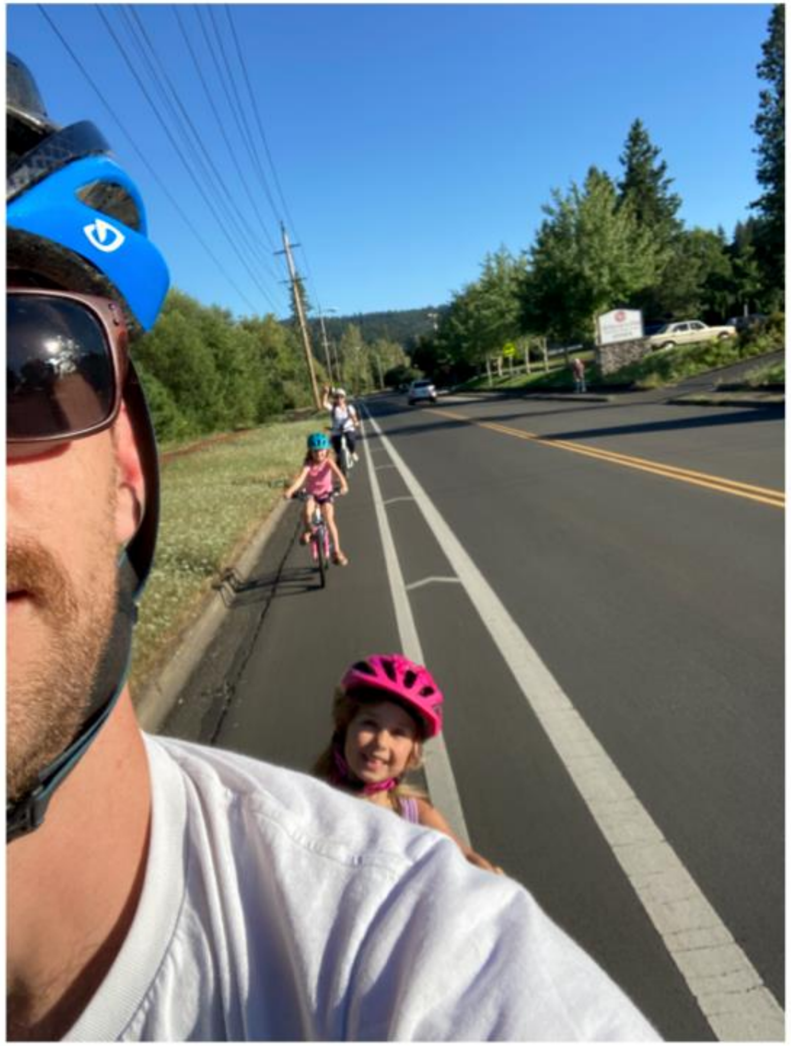
- Introductions
- About the course
- Syllabus
- Kaggle

About Me

- BA: UC Santa Barbara
- PhD, School Psychology: University of Maryland
- Behavioral Research & Teaching (BRT) since 2009
- Research Associate Professor
    - Research
        - Applied statistical methods to measure and monitor student growth
        - Inform the applied research methodologies used by researchers
        - Developing and improving systems that support data-based decision making using advanced technologies to influence teachers' instructional practices and increase student achievement
    - CORE and CORE II
    - Teaching
        - EDLD 651 – Introduction to Data Science with R
        - EDLD 654 – this one!

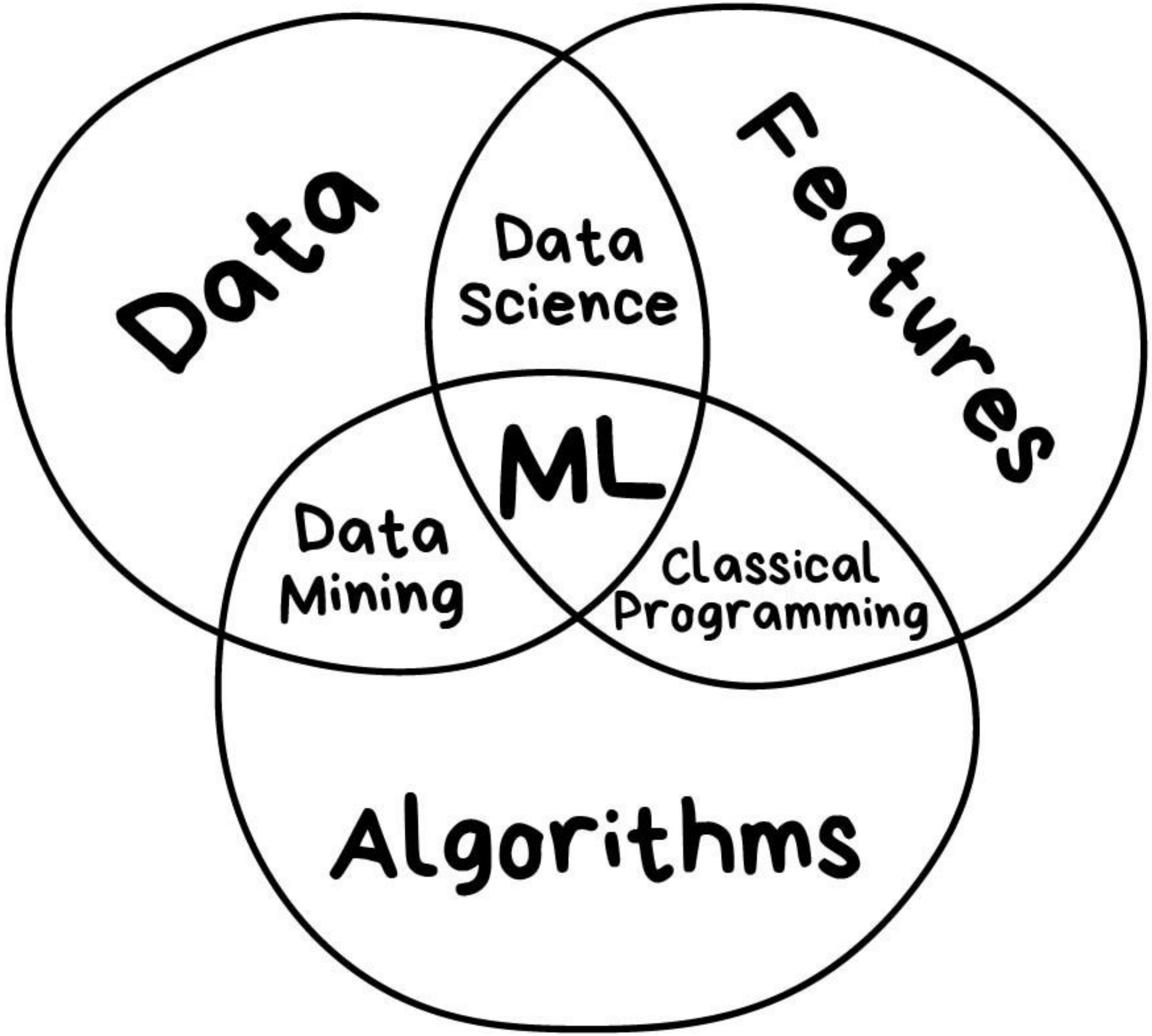
whoami

- Research Assistant Professor: Behavioral Research and Teaching
- Dad (two daughters: 8 and 6)
- Pronouns: he/him/his
- Primary areas of interest
    - 💗💗R💗💗, computational research, and open science
    - Achievement gaps, systemic inequities, and variance between educational institutions



About You

- Please introduce yourself
  - Name and program/year of study
  - How are you doing?
    - Tell me whatever you’d like the class to know

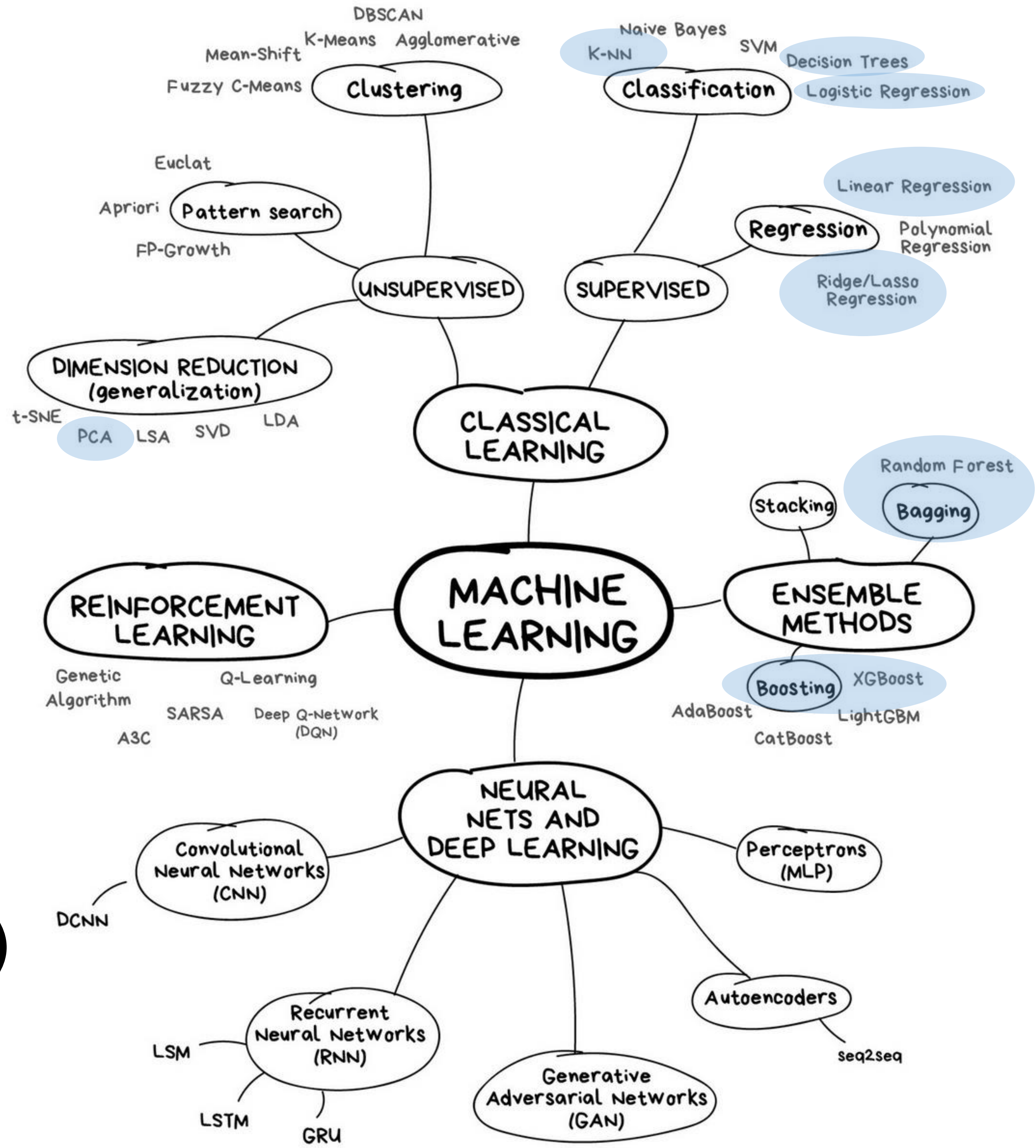


https://vas3k.com/blog/machine_learning/

Inference vs. Prediction

Variable Selection

Bias-Variance Tradeoff

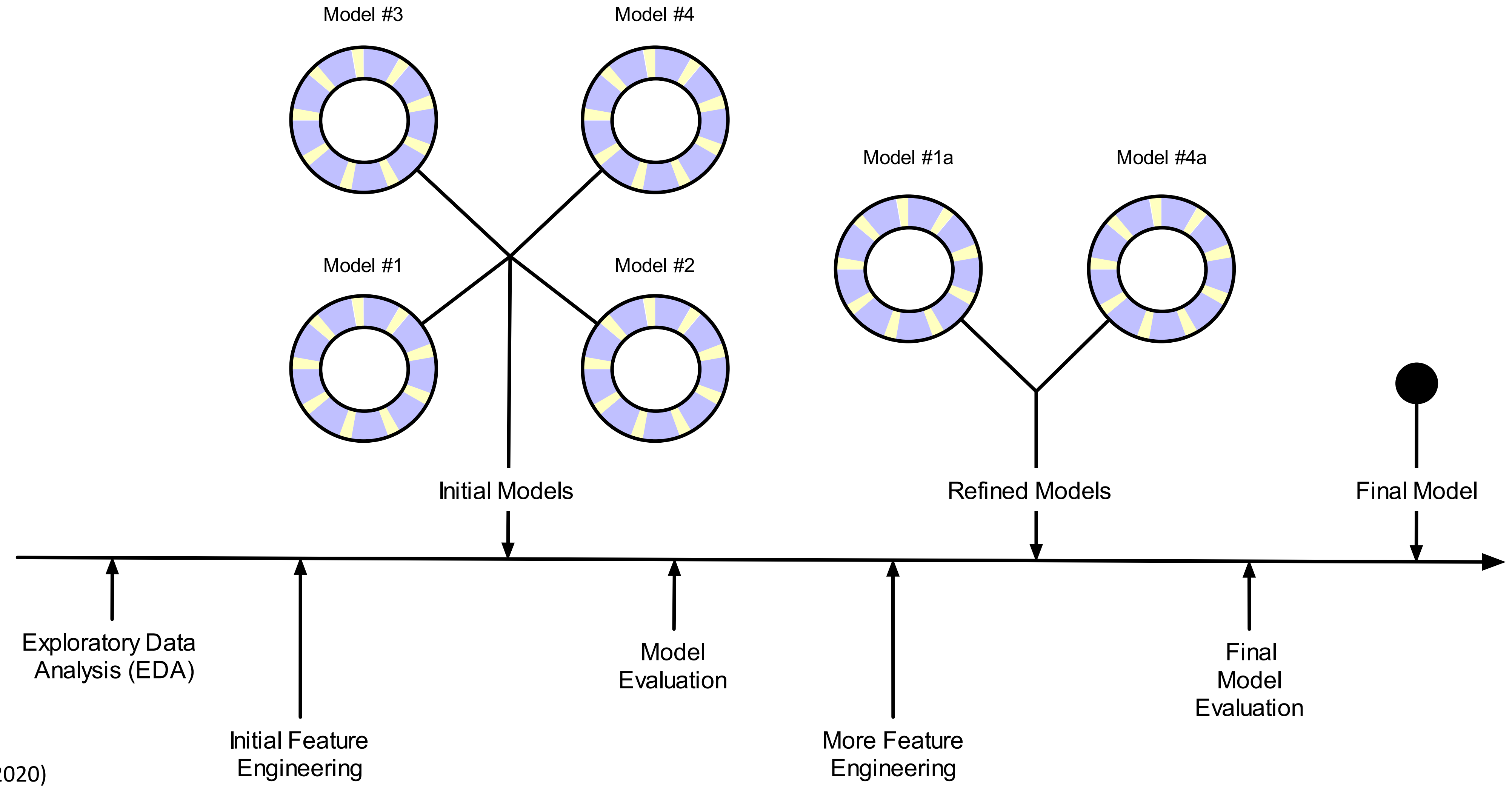


ML Ethics

Feature Engineering

https://vas3k.com/blog/machine_learning/

ML Modeling Process



Kuhn & Silge (2020)

Housekeeping

- Course website: https://uo-datasci-specialization.github.io/c4-ml-fall-2020/index.html
  - Syllabus: https://uo-datasci-specialization.github.io/c4-ml-fall-2020/site-syllabus.html
  - Schedule: https://uo-datasci-specialization.github.io/c4-ml-fall-2020/schedule.html
  - Assignments: https://uo-datasci-specialization.github.io/c4-ml-fall-2020/assignments.html

- Course github: https://github.com/uo-datasci-specialization/c4-ml-fall-2020

- Course Kaggle: https://www.kaggle.com/c/edld-654-fall-2020/overview

How the pandemic affects this course

Remote instruction

- In-person format
  - Present new content
  - Lab applying learned content
- Remote format
  - Present new content in a Zoom meeting
    - two instructors to help field questions
    - Zoom will be recorded, and posted to Canvas to view at a later time if need be
  - Labs held via zoom
    - two instructors available to help and answer questions
    - screen shares
    - breakout rooms to facilitate individual help as necessary
    - attendance is encouraged

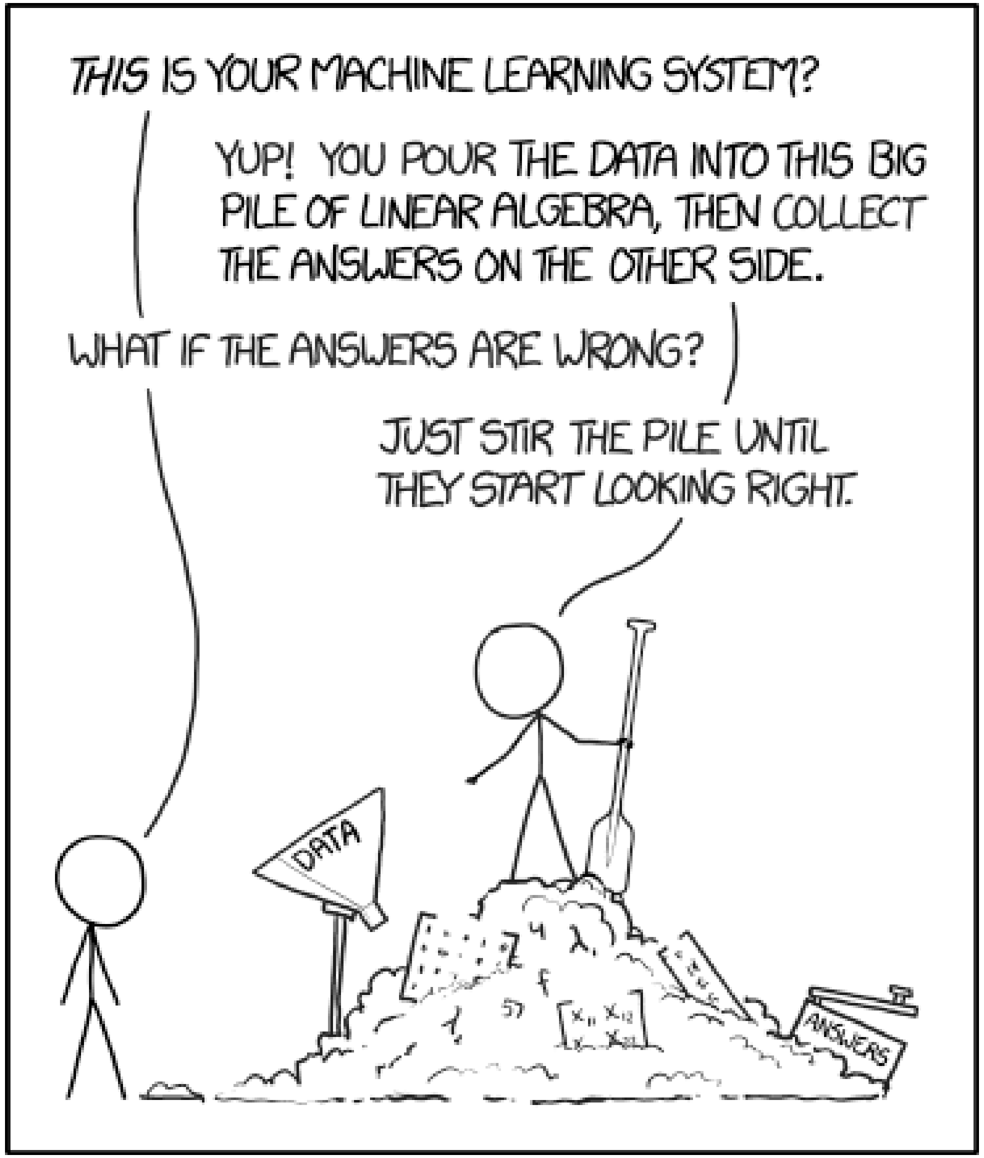
Practical changes

- Expectation is that you still engage with each week's materials
- Communication
  - Please be open with your communication regarding unexpected interruptions or uncertainties so that we can help
    - illness
    - caring for family
    - technology
- Accessibility
  - students with disabilities or medical conditions that encounter barriers with remote instruction should contact the Accessible Education Center as soon as possible so that appropriate accommodations can be determined

How we can help each other

- Patience
- Grace
- Empathy
- Understanding
- Open communication

About the course


THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

This class

- This class is
  - an introduction to applied machine learning techniques
  - experimental
  - utilizing R
  - delivered remotely

- This class is not
  - all encompassing
  - perfectly complete (the field and software change)
  - an online course

Why R?

- R has cutting edge ML models
  - Some ML developers use R as their primary computing environment and their work often results in R packages
- R and R packages are built by people who do data analysis
- It is easy to link to other applications
  - You can implement python, C, C++, tensorflow, keras, stan, or Weka without leaving R
- You already know R!

Why not R?

- R is a data analysis language, and is not C or Java
  - If a high-performance deployment is required, R can be a prototyping language
- R is mostly memory-bound
  - But there are plenty of exceptions to this
- The interfacing functions have been inconsistent
  - There are two methods for specifying what a model
    - formula (`y ~ x`)
    - `x = x , y = y`
  - Nearly all model functions auto-generate dummy variables
  - Some packages have an argument for resampling

Syntax for computing predicted class probabilities

Function	**Package**	**Code**
`lda`	`MASS`	`predict(obj)`
`glm`	`stats`	`predict(obj, type = "response")`
`gbm`	`gbm`	`predict(obj, type = "response", n.trees)`
`mda`	`mda`	`predict(obj, type = "posterior")`
`rpart`	`rpart`	`predict(obj, type = "prob")`
`Weka`	`RWeka`	`predict(obj, type = "probability")`
`logitboost`	`LogitBoost`	`predict(obj, type = "raw", nIter)`
`pamr.train`	`pamr`	`pamr.predict(obj, type = "posterior", threshold)`

Kuhn (2020)

Syntax for computing predicted class probabilities

Function	Package	Code
`lda`	`MASS`	`predict(obj)`
`glm`	`stats`	`predict(obj, type = "response")`
`gbm`	`gbm`	`predict(obj, type = "response", n.trees)`
`mda`	`mda`	`predict(obj, type = "posterior")`
`rpart`	`rpart`	`predict(obj, type = "prob")`
`Weka`	`RWeka`	`predict(obj, type = "probability")`
`logitboost`	`LogitBoost`	`predict(obj, type = "raw", nIter)`
`pamr.train`	`pamr`	`pamr.predict(obj, type = "posterior", threshold)`

Syntax for computing predicted class probabilities

Function	Package	Code
`lda`	`MASS`	`predict(obj)`
`glm`	`stats`	`predict(obj, type = "response")`
`gbm`	`gbm`	`predict(obj, type = "response", n.trees)`
`mda`	`mda`	`predict(obj, type = "posterior")`
`rpart`	`rpart`	`predict(obj, type = "prob")`
`Weka`	`RWeka`	`predict(obj, type = "probability")`
`logitboost`	`LogitBoost`	`predict(obj, type = "raw", nIter)`
`pamr.train`	`pamr`	`pamr.predict(obj, type = "posterior", threshold)`

Syntax for computing predicted class probabilities

Function	**Package**	**Code**
lda	MASS	predict(obj)
glm	stats	predict(obj, type = "response")
gbm	gbm	predict(obj, type = "response", n.trees)
mda	mda	predict(obj, type = "posterior")
rpart	rpart	predict(obj, type = "prob")
Weka	RWeka	predict(obj, type = "probability")
logitboost	LogitBoost	predict(obj, type = "raw", nIter)
pamr.train	pamr	pamr.predict(obj, type = "posterior", threshold)

Syntax for computing predicted class probabilities

Function	Package	Code
`lda`	`MASS`	`predict(obj)`
`glm`	`stats`	`predict(obj, type = "response")`
`gbm`	`gbm`	`predict(obj, type = "response", n.trees)`
`mda`	`mda`	`predict(obj, type = "posterior")`
`rpart`	`rpart`	`predict(obj, type = "prob")`
`Weka`	`RWeka`	`predict(obj, type = "probability")`
`logitboost`	`LogitBoost`	`predict(obj, type = "raw", nIter)`
`pamr.train`	`pamr`	`pamr.predict(obj, type = "posterior", threshold)`

Why `tidymodels`?

- Consistent interface functions and language
  - Irrespective of modeling package
  - This makes your workflow consistent across models and projects
- RStudio generally makes good products
  - You are already familiar with the `tidyverse`
- `tidymodels` has the support of a respected development team
- Cutting edge!
  - taking the place of the well-known and widely-used `carat` package

Why not `tidymodels`?

- Cutting edge!
  - currently in development
  - stable/durable code?
  - documentation ~~non-existent~~ is growing!
    - Tidymodels website (launched spring 2020)
    - Tidy Modeling with R (released on 2020-9-22)
  - makes it challenging both to learn and to teach
- Not the `tidyverse`
  - but it is tackling something much larger and more complex
  - all the packages

Why not `tidymodels`?

- Cutting edge!
  - currently in development
  - stable/durable code?
  - documentation ~~non-existent~~ is growing!
    - Tidymodels website (launched spring 2020)
    - Tidy Modeli with R (released on 2020-9-22)
  - makes it both learn an to teach
- No he `ti`


4GIFs
.com

tidymodels

- This is a lot of packages
  - Some of these packages work in the background
  - Others perform specific tasks, small, important tasks in the modeling process
- Can be overwhelming, but
  - we will not be calling all of these directly
  - you will learn how each of the major packages fit into your ML workflow

Some Resources

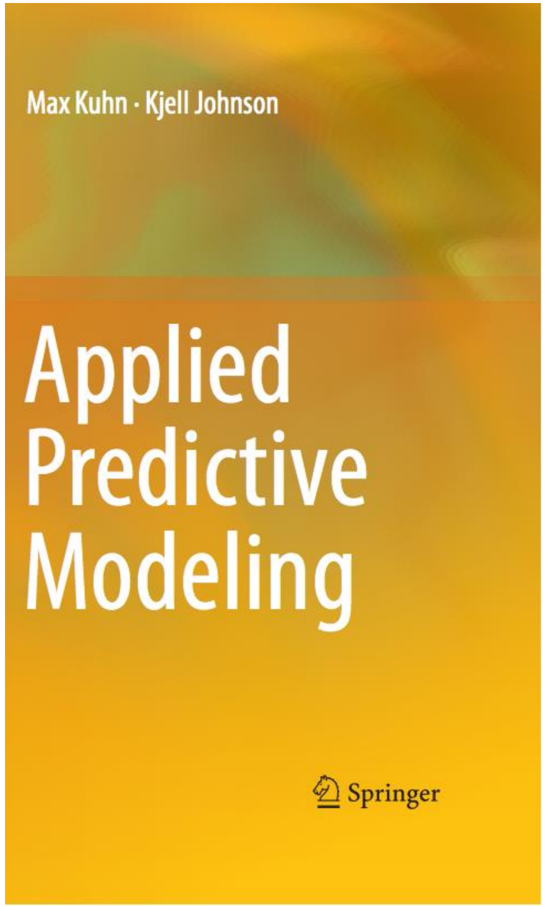
- Tidy Modeling with R (just released!)
- Tidymodels website (recently released!)
- Learning to Teach Machines to Learn (Allison Hill)
- ML Learning Resources (Bradley Boehmke)
- Intro to Tidy ML materials (Allison Hill)
- Applied ML materials (Max Kuhn)
- Julia Silge blog
  - screencasts and blogs using `{tidymodels}` and `#TidyTuesday` data
- Your peers
- Your instructors
- RStudio Community

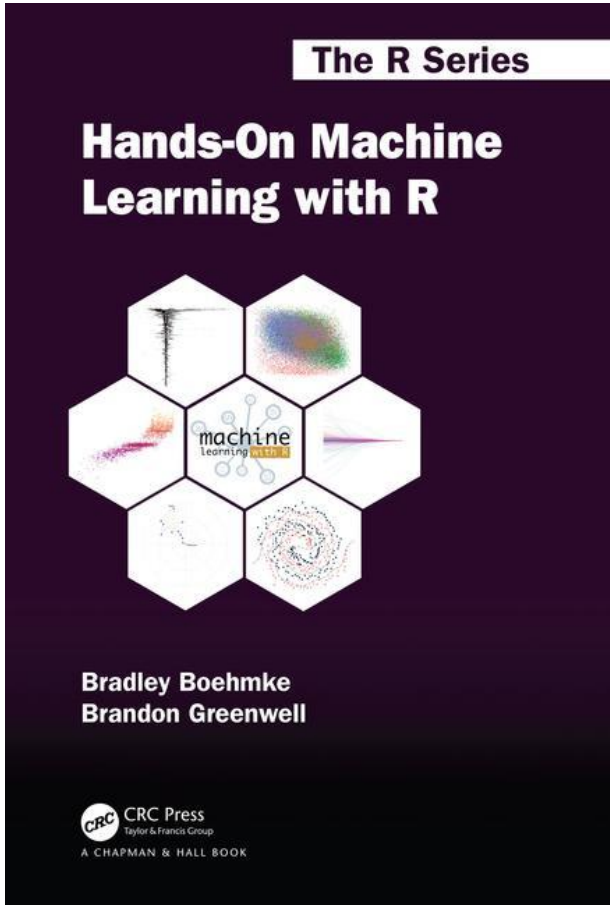
Syllabus

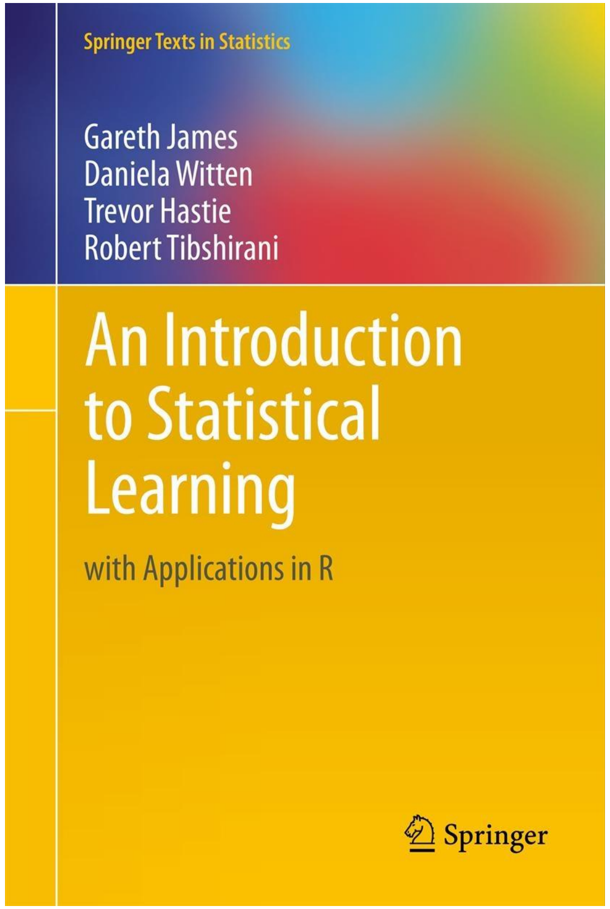
Course learning objectives

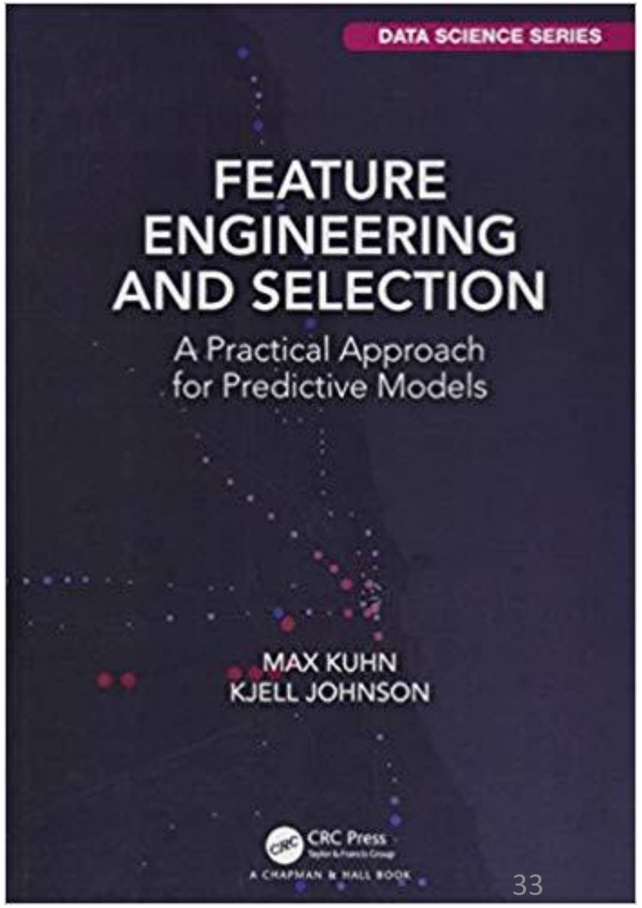
- Describe the framework of machine learning (i.e. supervised vs. unsupervised learning) and how it differs from standard inferential statistics
- Discuss the bias-variance tradeoff in supervised learning and apply the concept in making decisions about model selection
- Construct various supervised learning models, including linear regression (for prediction rather than inference), penalized regression (ridge/lasso), various decision tree models (including bagged and boosted trees, and random forests), and *k*-nearest neighbor models
- Measure and contrast the performance of various models
- Construct models for both classification- and regression-based problems
- Conduct feature engineering, including dimension reduction, to increase model performance (and quantify the degree to which model performance changed)

Required Texts (free)

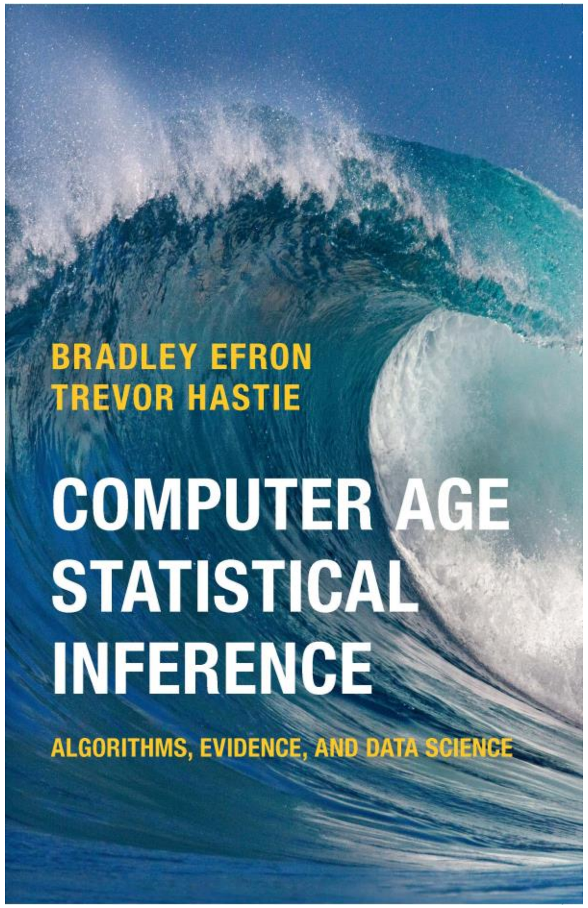


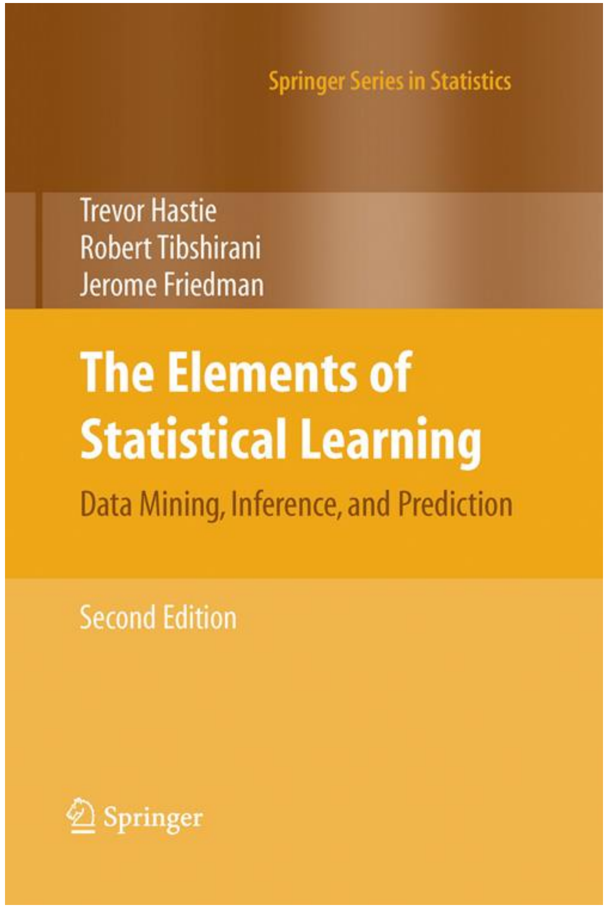


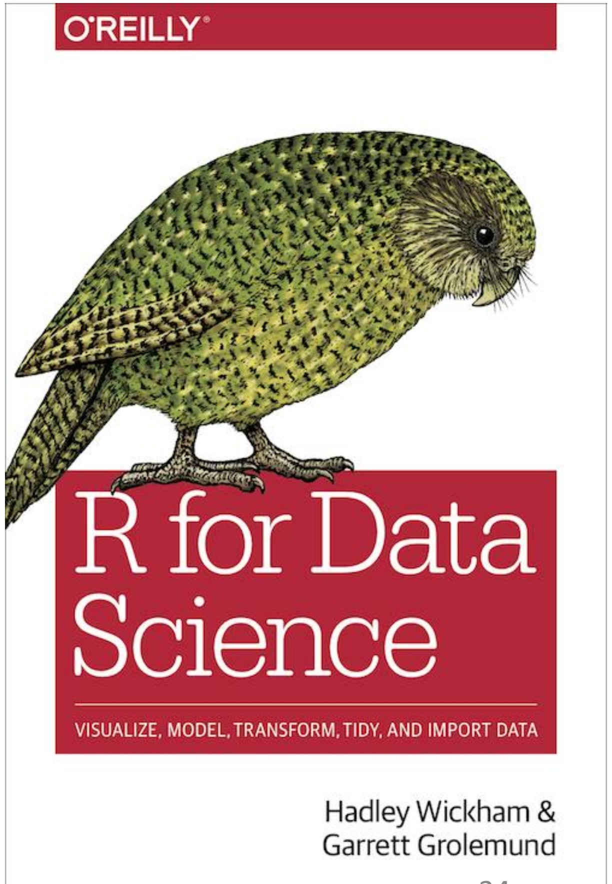




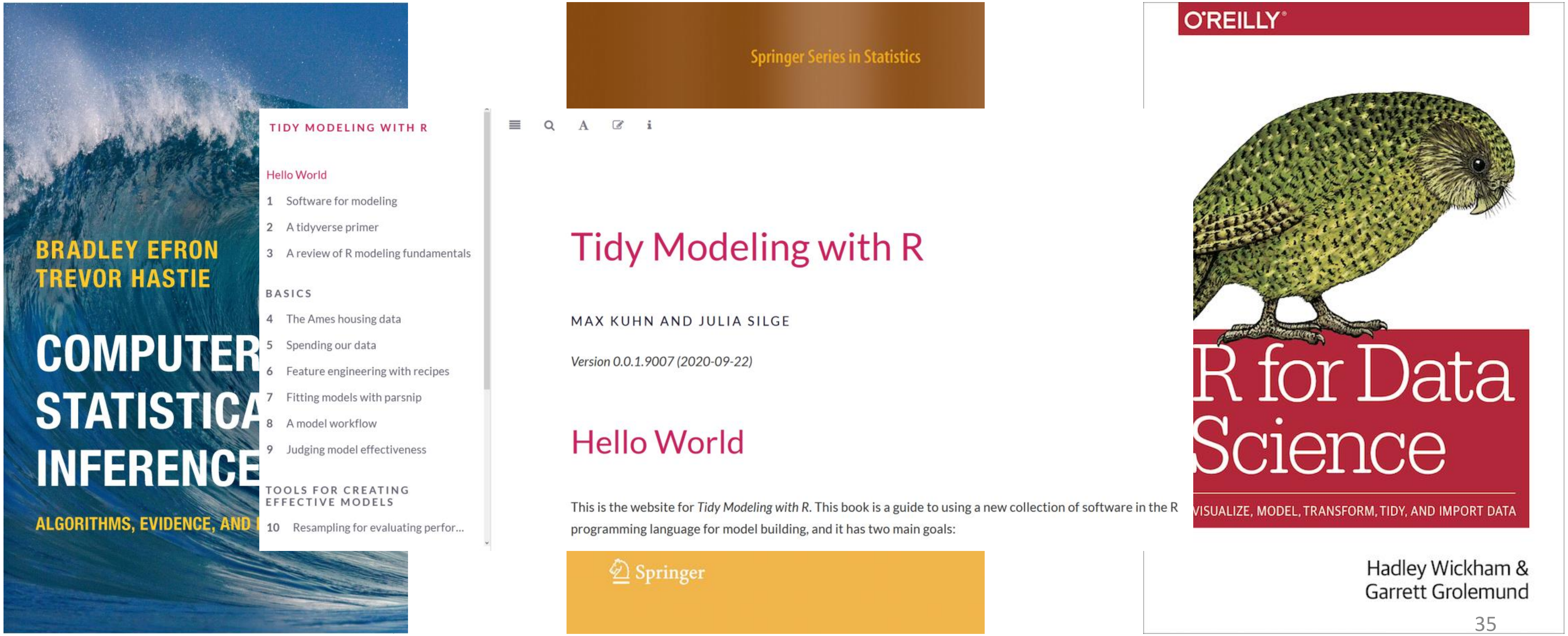
Books not required (but possibly helpful)







Books not required (but possibly helpful)



Weekly Schedule

- Class time M & W – 12:15-1:45
- Readings – do before class
- New content presented via Zoom
  - attend to ask questions
  - recordings will be posted to Canvas
- Labs conducted via Zoom
  - to provide you with practice and time with the instructors to help you puzzle through your problems
  - what you do not complete in Zoom labs must be completed on your own time

Assignments (200 points total)

- Data Quiz (5 points)
- Labs (100 points)
  1) Resampling (20 points)
  2) Penalized Regression (20 points)
  3) Feature Engineering (20 points)
  4) *K* Nearest Neighbors (20 points)
  5) Ensemble Methods (20 points)
- Final Project (95 points)
  - Preliminary fit 1 (10 points)
  - Preliminary fit 2 (10 points)
  - Blog post (75 points)
    - Data description (20 points)
    - Model fit description (25 points)
    - Model fits (20 points)
    - Data visualization (5 points)
    - Reproducibility (5 points)

Labs

- Scored on a “best honest effort” basis
  - generally, zero or full credit
- If you find yourself stuck and unable to proceed, **please contact the instructors for help rather than submitting incomplete work**
  - Contacting the instructor is part of the “best honest effort” and can result in full credit for an assignment even if the work is not fully complete.
- **If the assignment is not complete, and the student has not contacted the instructor for help, it is likely to result is a partial credit score or a zero**.
- Labs submitted late will be docked by 30% (6 points)
  - Labs are due a week after they are assigned, before class starts

Final Project

- We will be using the kaggle platform to host a local competition
- Work as a team to build, tune, evaluate a predictive model based on the training data
- Make predictions on the test data that has all the same features (variables) except the outcome
- Upload predictions to kaggle, which will provide you with an estimate of your model performance on a portion of the test data (but not the full test data)
- At the end of the course, each team's most performant model will be evaluated against the full test data
  - this final test regularly leads to changes in the leaderboard ranking for real kaggle competitions
  - the team with the best model will be awarded five points extra credit.
- Link to outside data to help increase the performance of your model (e.g., NCES)

Final Project – Preliminary fits

- At Week 6 and Week 8 each team will be required to submit preliminary predictions to kaggle
- You may submit predictions at any time, but you must submit your first predictions by Week 6, and predictions from a new model by Week 8
- A quantitative indicator of prediction accuracy will automatically be provided
- Submissions will be scored on an "all or nothing" basis
  - If your group provides a set of predictions, you will all receive credit, regardless of the performance of the model

Final Project

Group project, 3 to 4 people

Blog post (or a series of blog posts)

- Data description
  - Describe core features of the data, any additional data you joined in and why, basic descriptives, feature engineering, and data splitting
- Model fit description
  - At least three models must be fit to the data. Describe each model, hyperparameters, assumptions, and a description of what the model is doing and why it is appropriate
- Model fits
  - Describe model fitting procedure(s) and the results of your model evaluation. Compare and contrast the different fits, including a discussion of model performance
- Data visualization
  - Include at least two plots (you may include more) to help communicate your findings
- Reproducibility
  - All code should be housed in a GitHub repository and be fully reproducible

Final Project – Dates

- **Week 6a (11/2)**: Preliminary Fit 1
- **Week 8b (11/18)**: Preliminary Fit 2
- **Week 11 (12/7)**: Final Fits, Blog post (final product) due

Final Project – Scoring Rubric

Criteria	Points possible
Preliminary Fit 1	10
Preliminary Fit 2	10
Blog Post(s)	
Description of the data	20
Description of the model fits	25
Model fits	20
Data visualization	5
Reproducibility	5
Total	95

Grading

Lower percent	Lower point range	Grade	Upper point range	Upper percent
97	(194 pts)	A+		
93	(186 pts)	A	(194 pts)	97
90	(180 pts)	A-	(186 pts)	93
87	(174 pts)	B+	(180 pts)	90
83	(166 pts)	B	(174 pts)	87
80	(160 pts)	B-	(166 pts)	83
77	(154 pts)	C+	(160 pts)	80
73	(146 pts)	C	(154 pts)	77
70	(140 pts)	C-	(146 pts)	73
		F	(140 pts)	70

Kaggle