DECISION PATTERNS

Machine Learning Fundamentals

Practical Machine Learning (with R)

UC Berkeley

Fall 2015

Topics

- Administrativa
  - *Applied Predicative Modeling,* Max Kuhn
  - *The Art of R Programming,* Norm Matloff
  - *Elements of Statistical Learning,* Hastie, Friedman, Tibshirani
- Review Q&A
  - Assignments and Grading

REVIEW

EXPECTATIONS

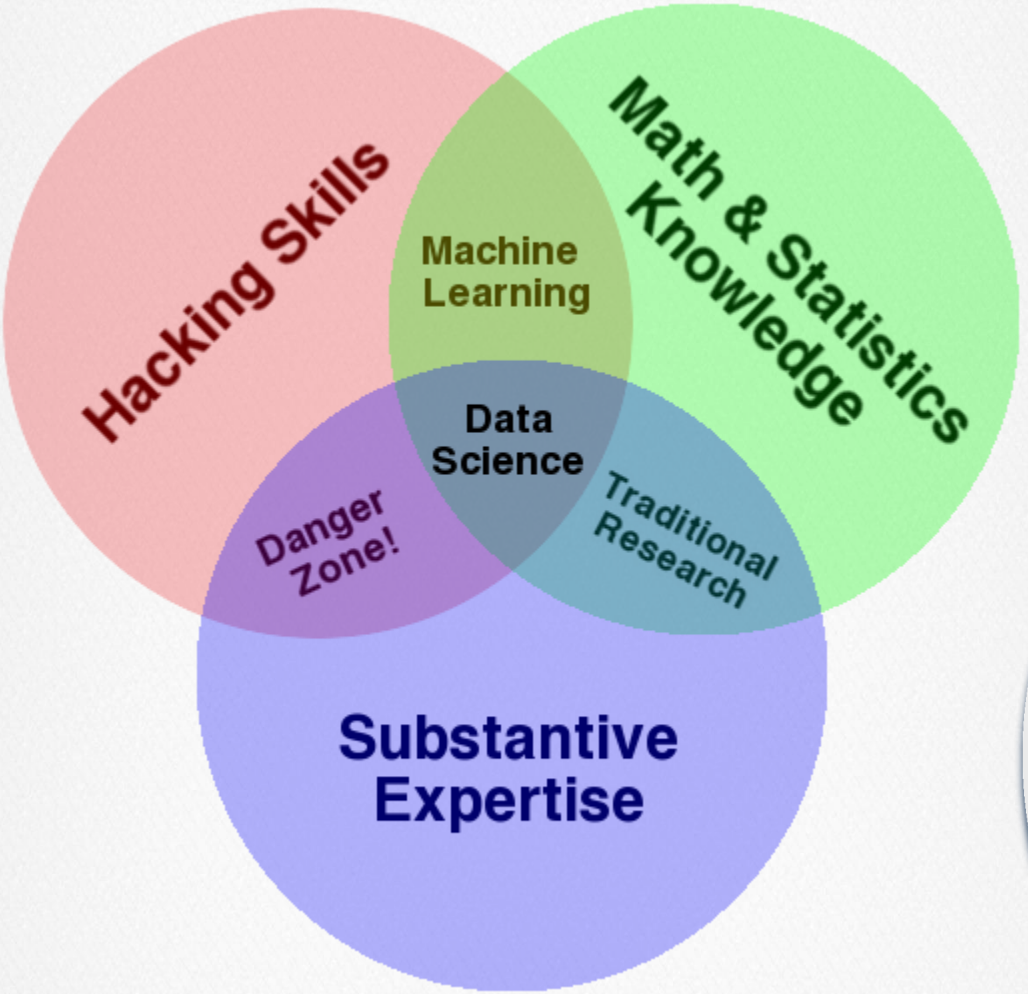
- You have installed **R** and **Rstudio**
- If you are new to **R**, you will have checked out one of the resources and have started becoming familiar with syntax and functions.

- If you are not familiar with source control, you will have investigated **git/github** and/or **sourcetree**

Review

- Class Objective → Practice of ML
- Advantages of R
  (popularity, community, extensibility)
- Elite Coding
  - Follow Established Design Patterns
  - Adopt Standards
  - Use Version Control → git(hub) / sourcetree
- Set-up R / Rstudio

Data Science Venn Diagram



Ref. http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram

USEFUL R PACKAGES

- ***ML Framework***: *caret* (Classification and Regression Training)
- **Pipe operators**: *magrittr, pipeR, backpipe* (shiny)
- **Tables**: *data.tables, dplyr*
- **Visualization**: *ggvis, ggplot2*
- **Reporting**: *knitr, rmarkdown*

R Resources

ONLINE

- (META)CRAN
  - Packages
  - Task Views
- Stackoverflow.com
- r-bloggers.com
- Advanced R Programming
- Github

Offline

The Art of R Programming

Norm Matloff

ISBN-13: 858-2592222227

ISBN-10: 1593273843

R in Action: Data Analysis and Graphics with R

Robert Kabacoff

ISBN-13: 978-1617291388

ISBN-10: 1617291382

Course Website

- https://github.com/CSX460

Git / GitHub / Source Tree

- **Workflow**
  - clone
  - branch
  - (work)
  - add
  - commit (early and often)
    - tag
  - push
  - Also checkout, status, log

ADMINISTRATIVE

GRADING

Grades

- **Exams and Quizzes (20%)**
- **Class Participation and Exercises(30%)**
- **Project (50%)**
  - Identify problem you want to tackle
  - Frame the problem
  - Build Features
  - Review linear model and cart
  - Build Model
  - Deploy
- **Attendance is Mandatory**

High Dimensional Spaces

Example of ML algorithm(s)

- Spam Filter
- handwriting recognition (svm)
- Traffic engineering (lights)
- Weather prediction
- Sentiment analysis (social media)
- Netflix Recommender
- Fraud detection (Visa)
- Imaging processing
- Intrution detection
- Self-driving cars

QUESTION 1

What is machine learning?

A formal ***process*** for building a ***model***

QUESTION 2

What is a model?

a ***function*** that ***estimates*** a ***response*** associated with (a set of) known ***predictors***

$$\hat{y} = f(\vec{x})$$

QUESTION 1

What is machine learning?

A formal ***process*** of building a model

How do we find f?

What are the Properties of f

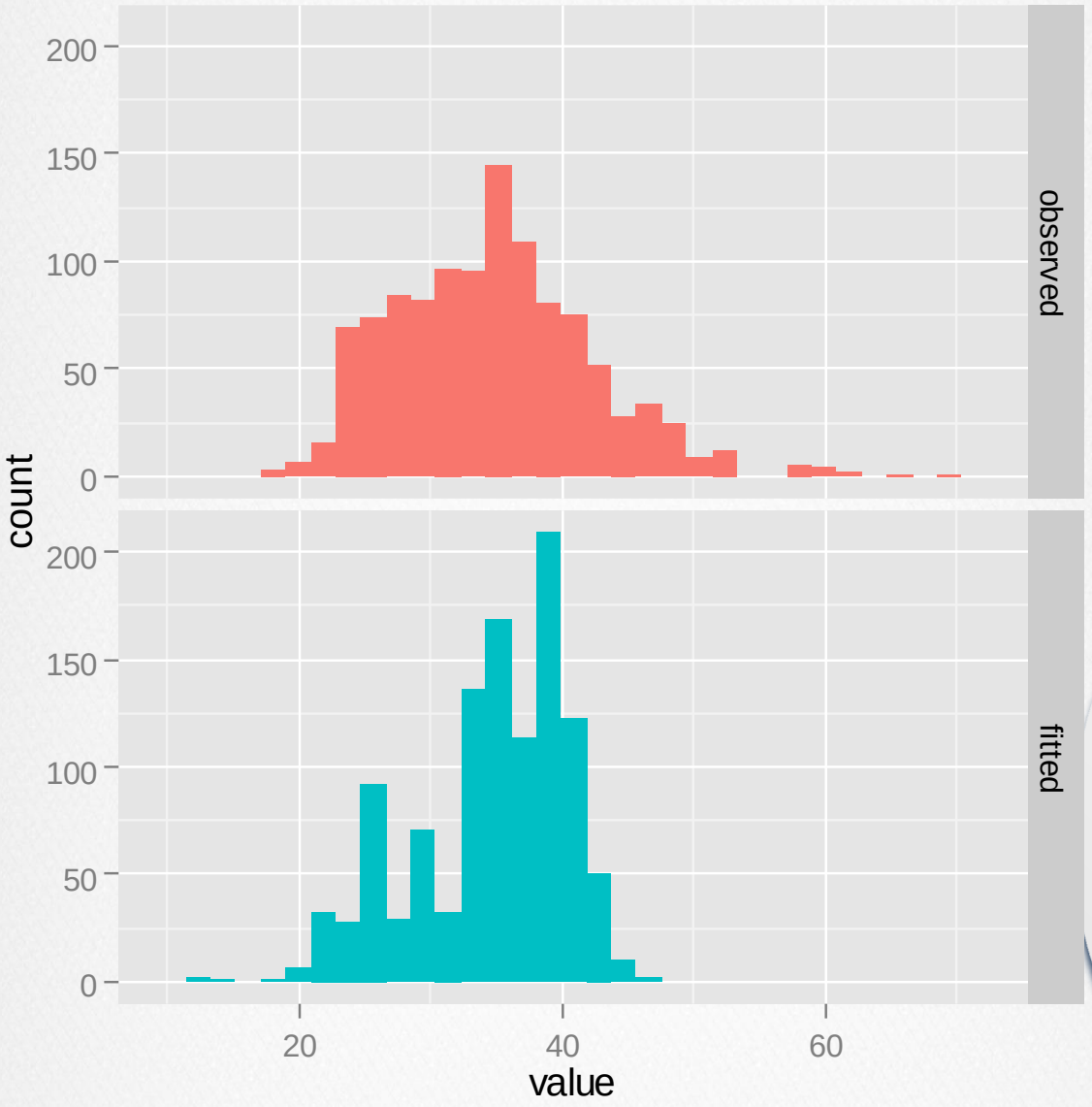
- Should be easy to evaluate
- Takes a one or more values of inputs
- Yields a single output value for each input

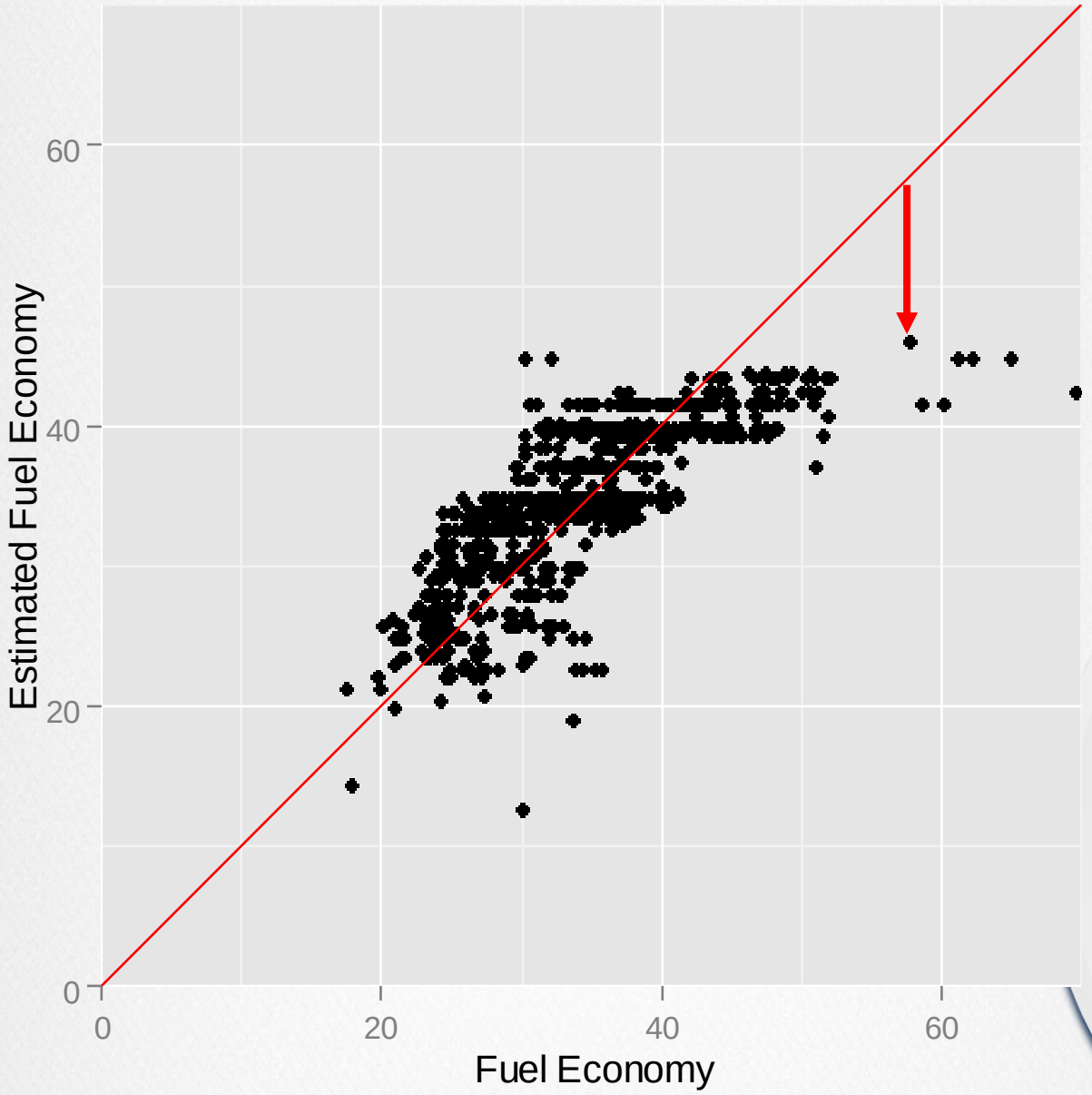
- Output, $\hat{\boldsymbol{y}}$, should be "close" to observed values, $\mathbf{y}$:

$$\hat{\boldsymbol{y}} \sim \boldsymbol{y}$$

QUESTION 3

How good is the model?


200
150
100
50
0
count
observed
fitted
20
40
60
value


Estimated Fuel Economy
60
40
20
0
0
20
40
60
Fuel Economy

Measurements for Error For Model

What are the Properties of f

- Should be easy to evaluate
- Takes a one or more values of inputs
- Yields a single output value for each input
- Can measure the error
- ~~Output, $\hat{y}$, should be "close" to observed values, y: $\hat{y} \sim y$~~

What else ….

- The number of functions available?

3 Requirement for Algorithm

- A method for evaluating how well the algorithm performs (**ERRORS**)
- A restricted class of function (**MODEL**)
- A process for proceeding through the restricted class of functions to identify the functions (**SEARCH/OPTIMIZATION**)

Linear Regression

- Errors : Minimize Squared Error
- Model:

$$\hat{y} = \beta_0 + \sum_{i=1}^{p} \beta_i x_i$$

$$\hat{y} = \beta_0 + \boldsymbol{x}^T \boldsymbol{\beta}$$

Optimization Techniques

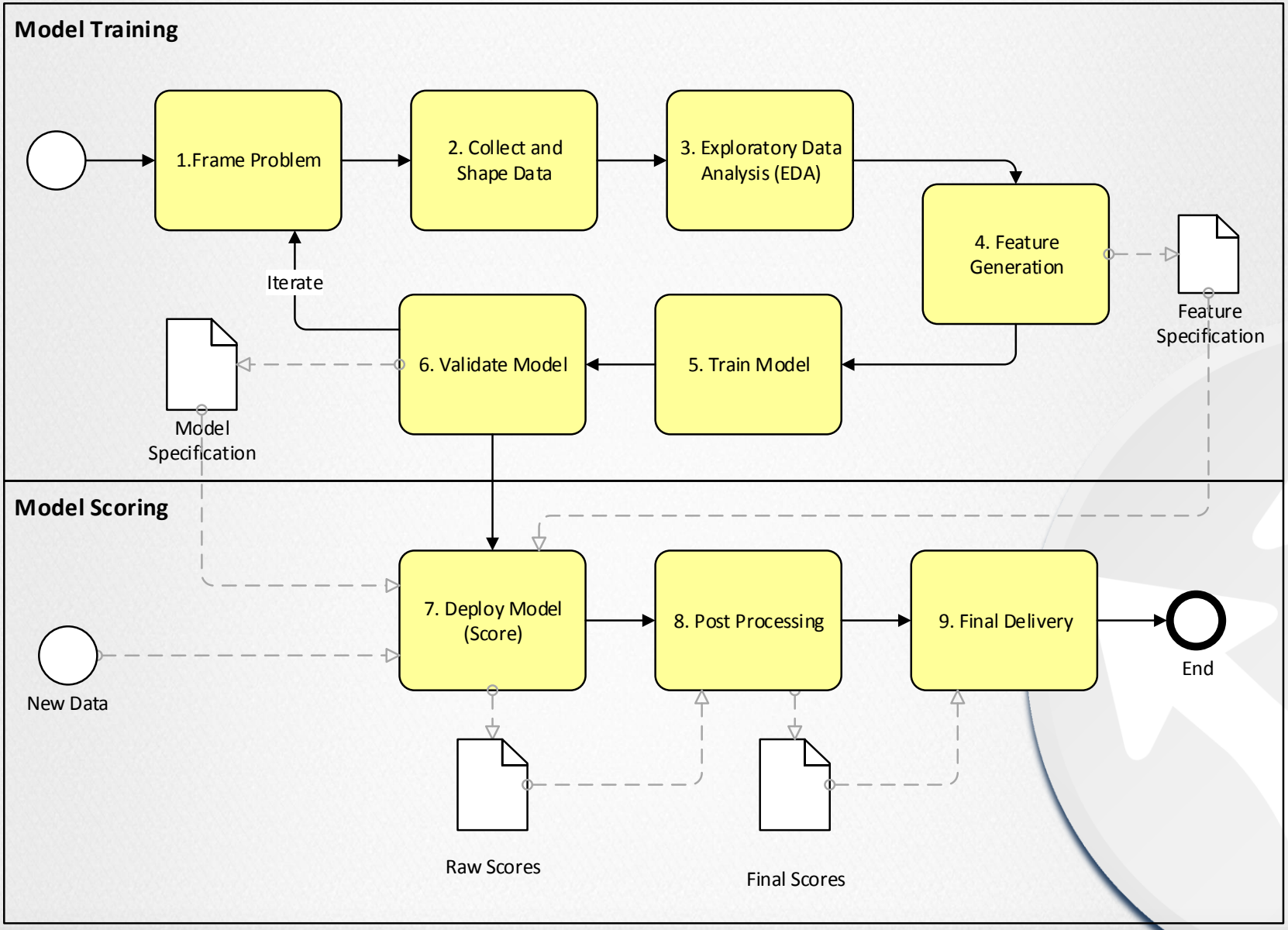
- Direct Solution
- Recursive Goal Seeking

More Than Learning

- Making a practical model entails more than learning…

What are the other requirements?

Comprehensive ML Process



Problem

Write a function to calculate the RMSE,

Write a function to calculate the MAE

Machine Learning Example

- AppliedPredictiveModelling::FuelEconomy

APPENDIX

Problem

Given a vector of numbers (x), write a function (f) that returns a vector of numbers containing the *product* of every other number excluding the current index.

Example:

```
> x <- c( 1, 5, 2, 8 )
> f(x)
[1] 80 16 40 10

# 5*2*8, 1*2*8, 1*5*8, 1*2*5
```

Solution

Given a vector of numbers (x) write a function (f) that returns a vector of numbers containing the *product* of every other integer excluding the current index.

Example:

```
> x <- c( 1, 5, 2, 8 )
> f(x)
[1] 80 16 40 10

# 5*2*8, 1*2*8, 1*5*8, 1*2*5
```

Solution:

```
f <- function(x) prod(x) / x
```

Class Overview : 1

- Introduction to R, setting up the ML developers environment
  - Installing R
  - Installing R Studio
  - Installing packages from CRAN, Bioconductor and Github
  - Exercises

Class Overview : 2

- Fundamentals of Machine Learning
  - Machine learning overview
  - Regression and classification
  - Supervised, unsupervised, and semi-supervised
  - Algorithm types and requirements
  - Exercises

Class Overview : 3

- Linear Regression
  - OLS Regression
  - Data partitioning
  - Model evaluation and tuning
  - Exercises

Class Overview : 4

- Logistic Regression
  - Logistic Regression
  - Exercises

CLASS OVERVIEW : 5

- Advanced Techniques: Partitioning Methods
  - CART/Regression Trees
  - Clustering
  - K Nearest Neighbors
  - Exercises

Class Overview : 6

- Advanced Techniques: Partitioning Methods
  - CART/Regression Trees
  - Clustering
  - K Nearest Neighbors
  - Exercises

Class Overview : 7

- Advanced Techniques
  - Bagging
  - Bagged Trees / Random Forests
  - Exercises

Class Overview : 8

- Advanced Techniques: Boosting
  - Boosting
  - Neural Networks
  - Support Vector Machines
  - Exercises

Class Overview : 8

- Deployment
  - Diving into the data lake
  - Optimization
  - Delivery and Production

Class Overview : 9

- Final Lecture
  - Exercises
  - Exam