Machine Learning

INFO 201

Today’s Objectives

Discuss differences/similarities between **statistics** and **machine learning**

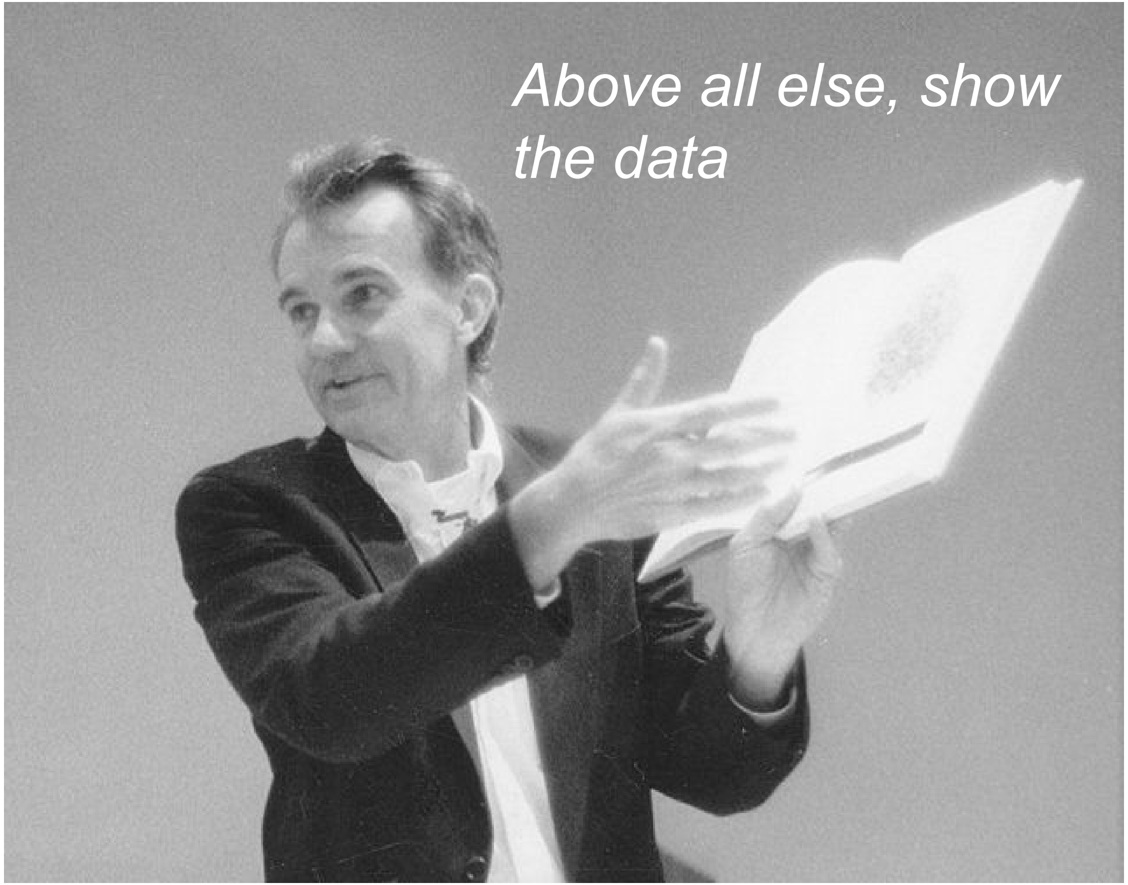
Describe the task of **classification**

Introduce **decision tree** approach to classification

Practice building a Shiny App for machine learning

Machine Learning

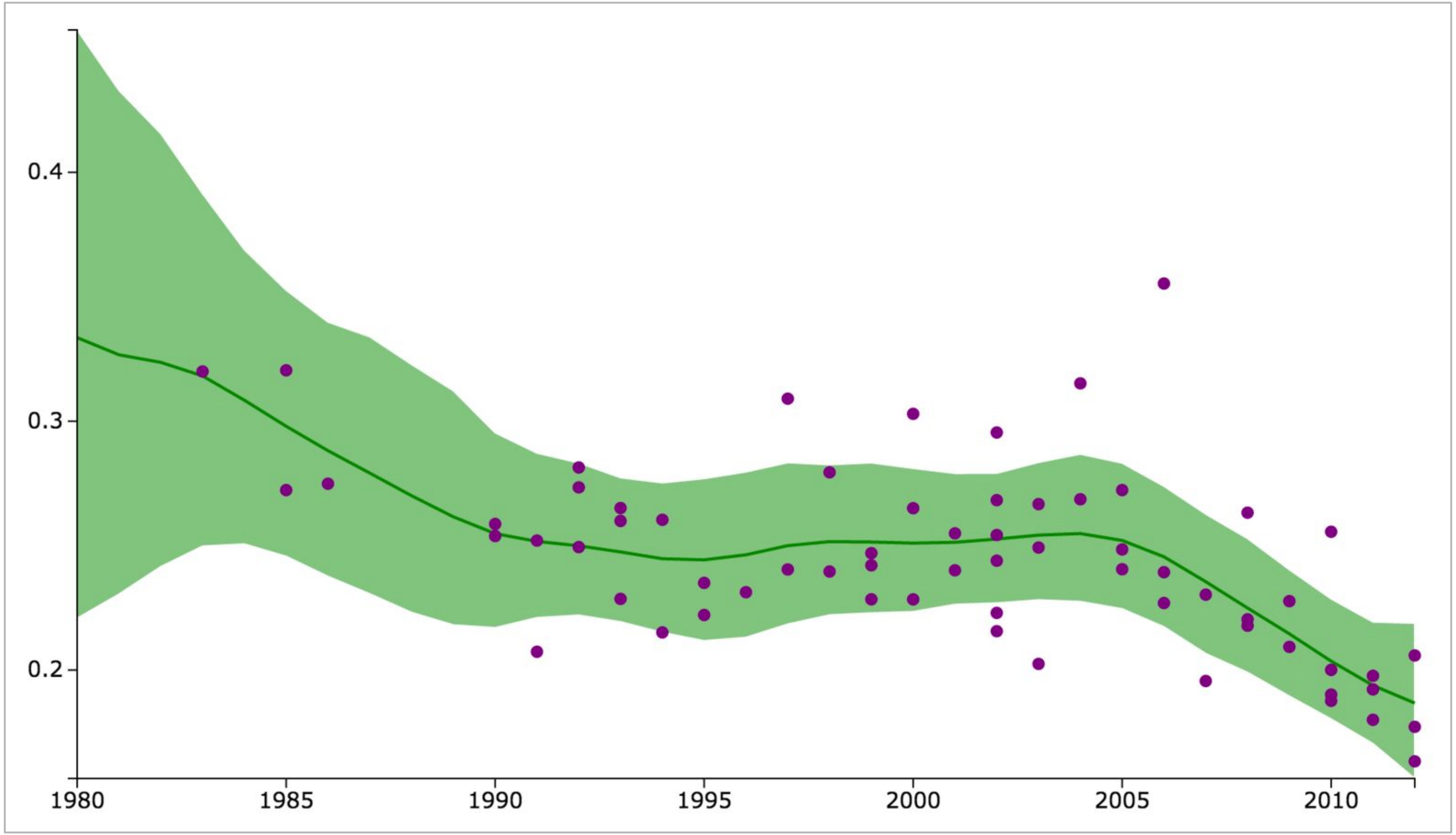
What do we do to data?



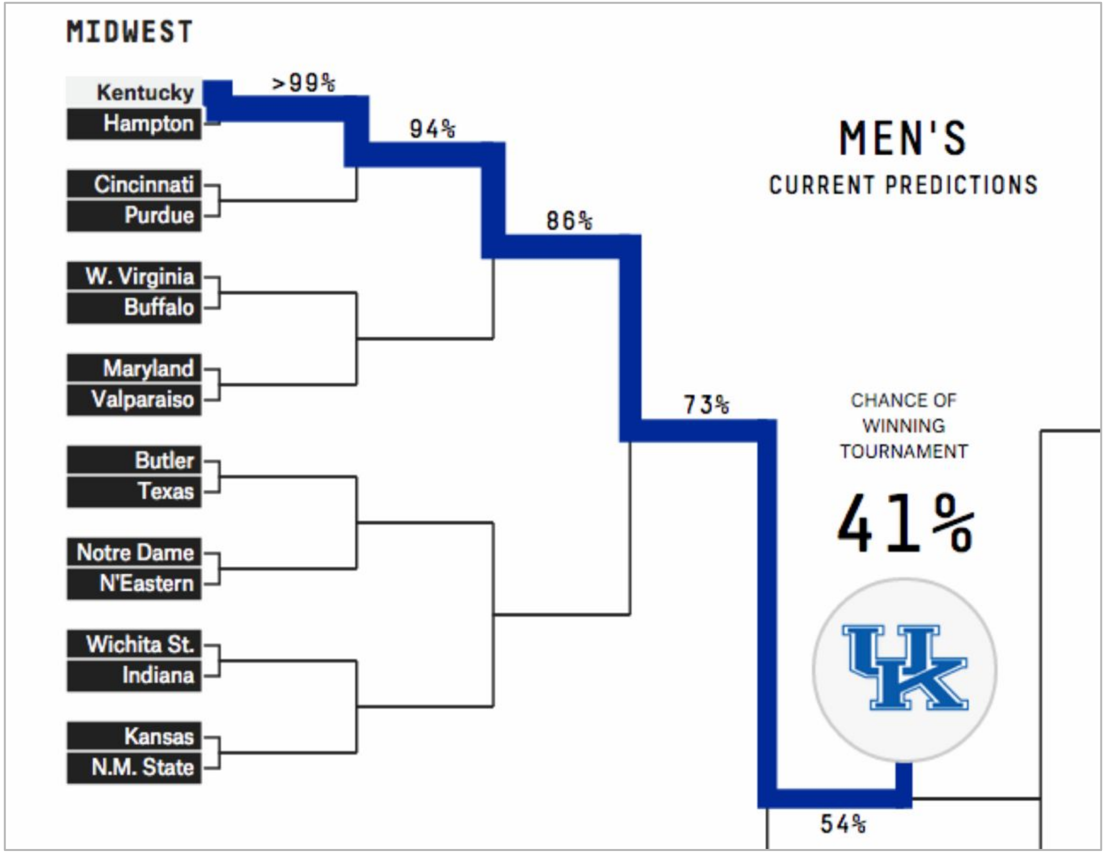
Above all else, show the data

Nothing

Clean



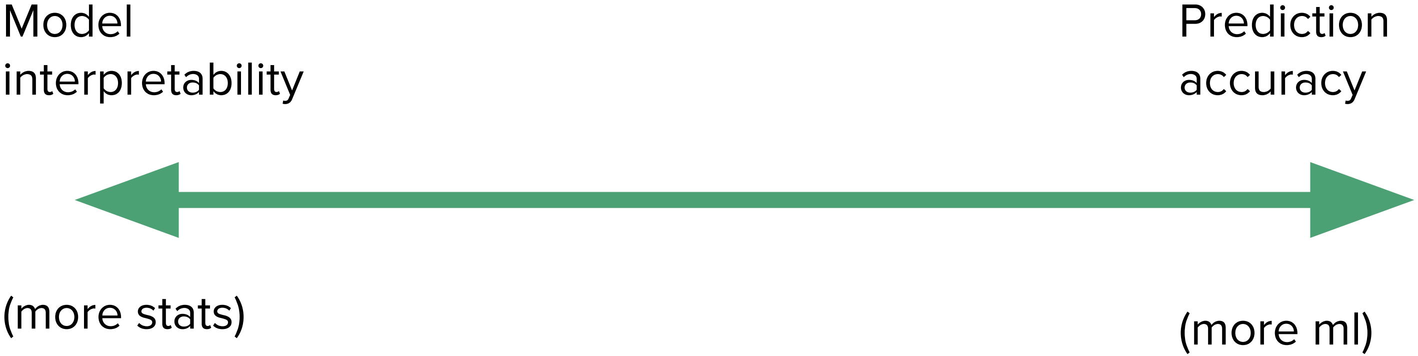
Model



MIDWEST
Kentucky
Hampton
Cincinnati
Purdue
W. Virginia
Buffalo
Maryland
Valparaiso
Butler
Texas
Notre Dame
N'Eastern
Wichita St.
Indiana
Kansas
N.M. State
>99%
94%
86%
73%
54%
MEN'S
CURRENT PREDICTIONS
CHANCE OF
WINNING
TOURNAMENT
41%

Predict

Machine learning and **statistics** are a set of tools used to **ask questions about data**. They leverage *mathematical concepts* and *computational abilities* to **make inferences** about relationships, or **make predictions** about unobserved contexts.



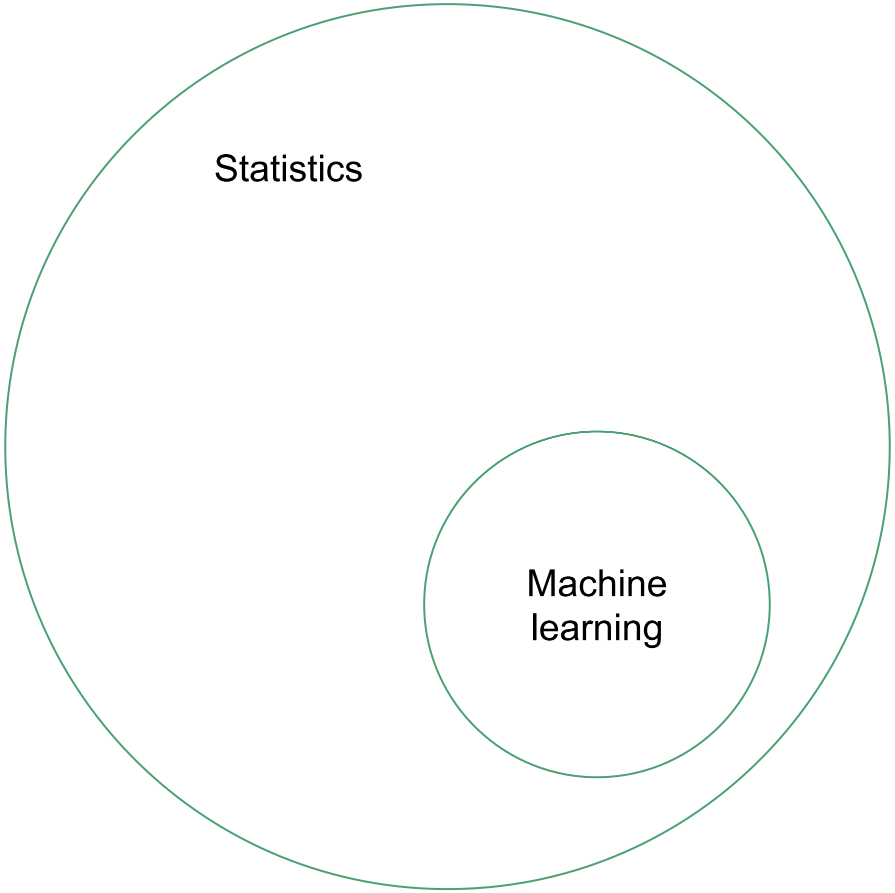
Model interpretability

Prediction accuracy

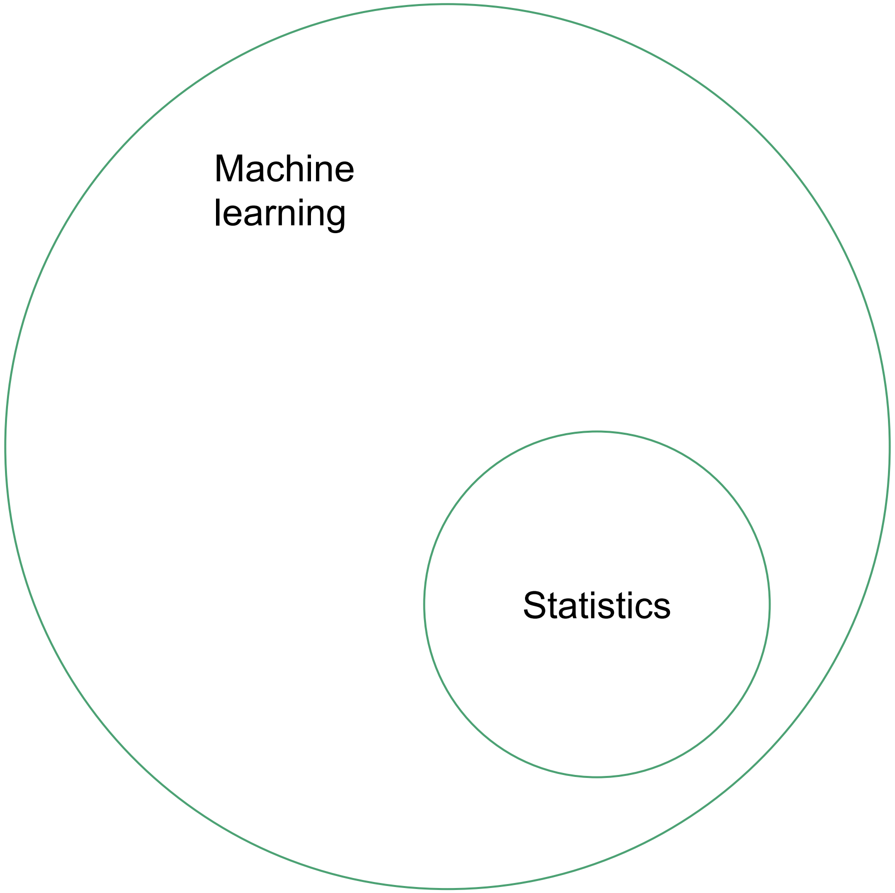
(more stats)

(more ml)

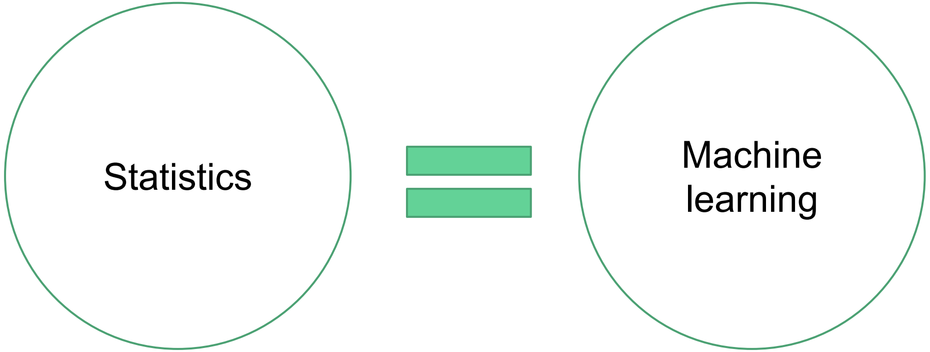
Tradeoffs



Many people argue this....

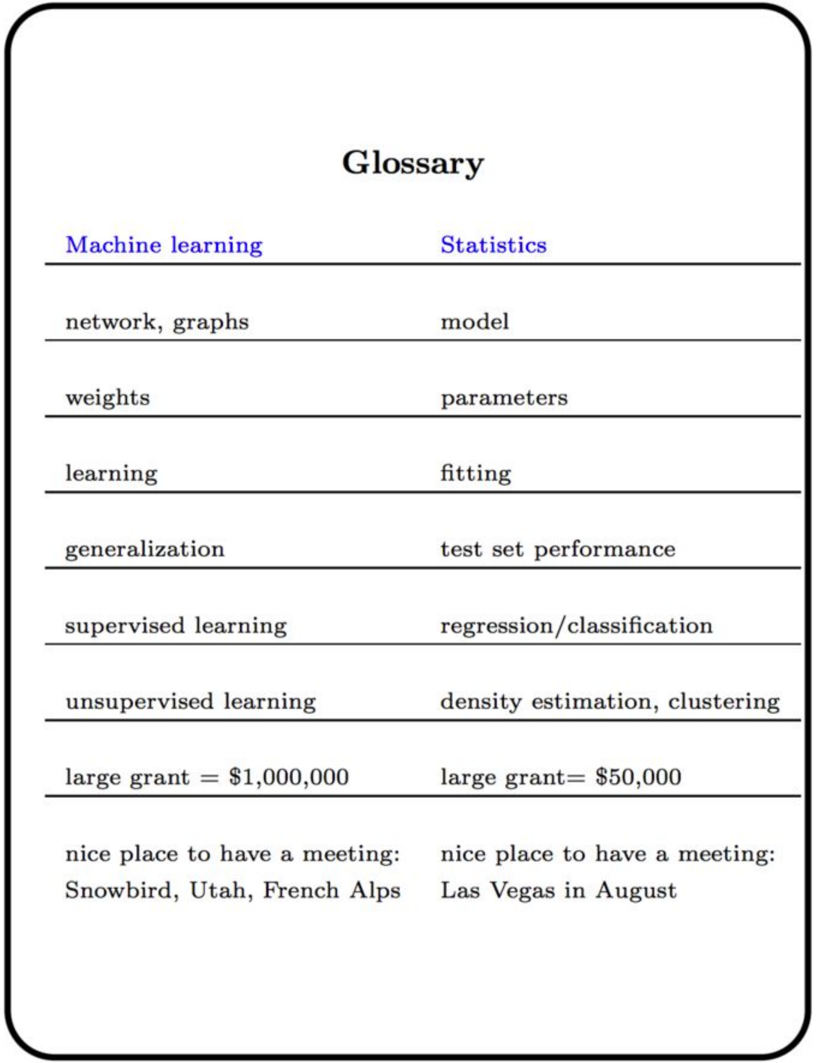


While many others argue this...



Statistics

Machine learning



Glossary

Machine learning	Statistics
network, graphs	model
weights	parameters
learning	fitting
generalization	test set performance
supervised learning	regression/classification
unsupervised learning	density estimation, clustering
large grant = $1,000,000	large grant= $50,000
nice place to have a meeting: Snowbird, Utah, French Alps	nice place to have a meeting: Las Vegas in August

And others think this...

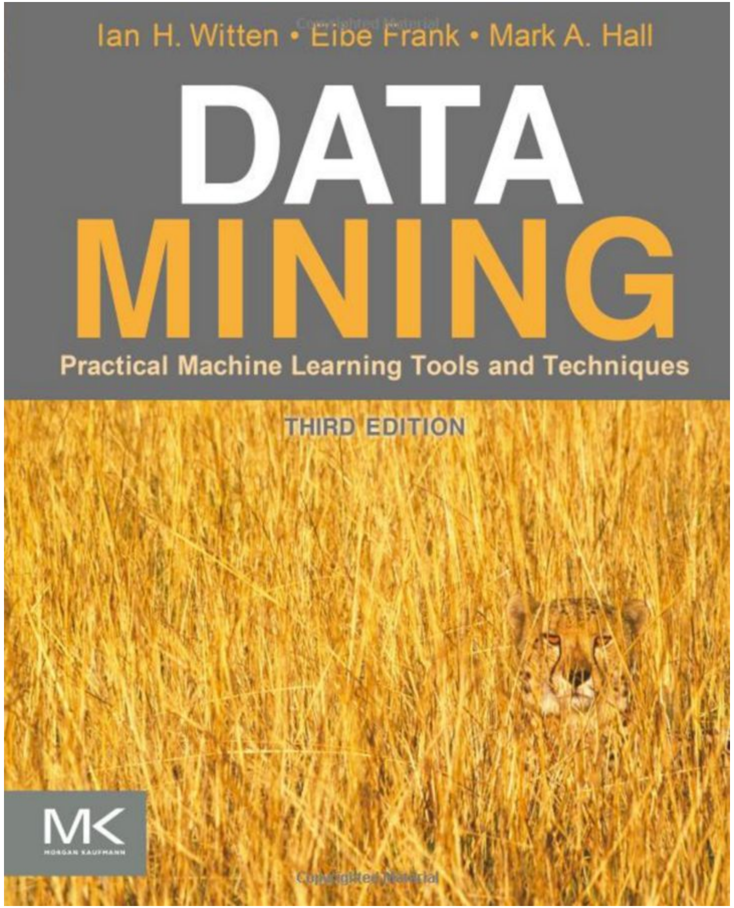
Other thoughts on ml v.s. stats

Machine learning is statistics on a mac

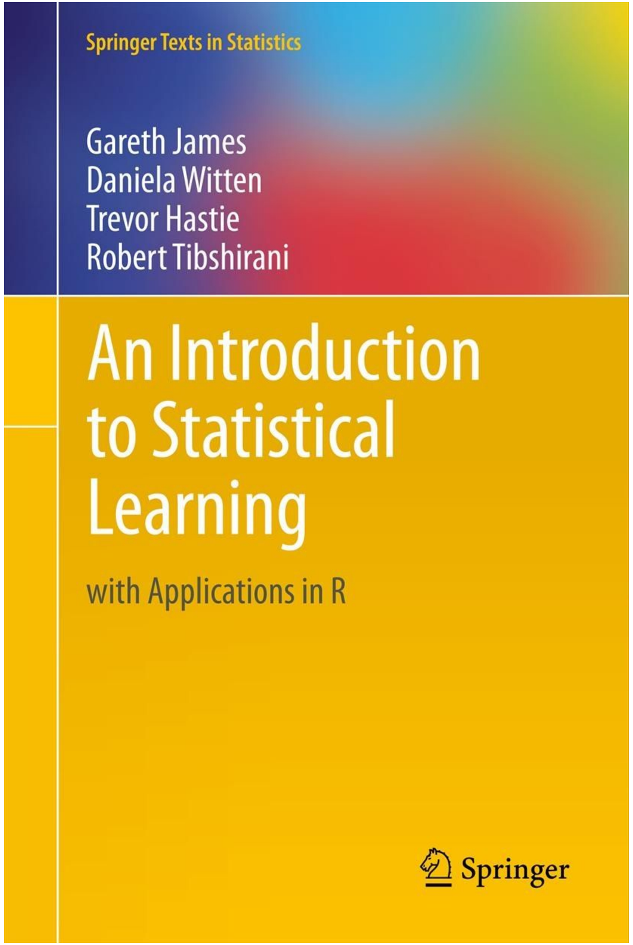
machine learning is statistics minus any checking of models and assumptions (source)

All valid tools to choose from, but you must select the right tool for the task

Simple to use, difficult to use well


Ian H. Witten • Eibe Frank • Mark A. Hall
DATA
MINING
Practical Machine Learning Tools and Techniques
THIRD EDITION
MK
MORGAN KAUFMANN

A good resource (source of some examples today)



Springer Texts in Statistics

Gareth James
Daniela Witten
Trevor Hastie
Robert Tibshirani

An Introduction to Statistical Learning

with Applications in R

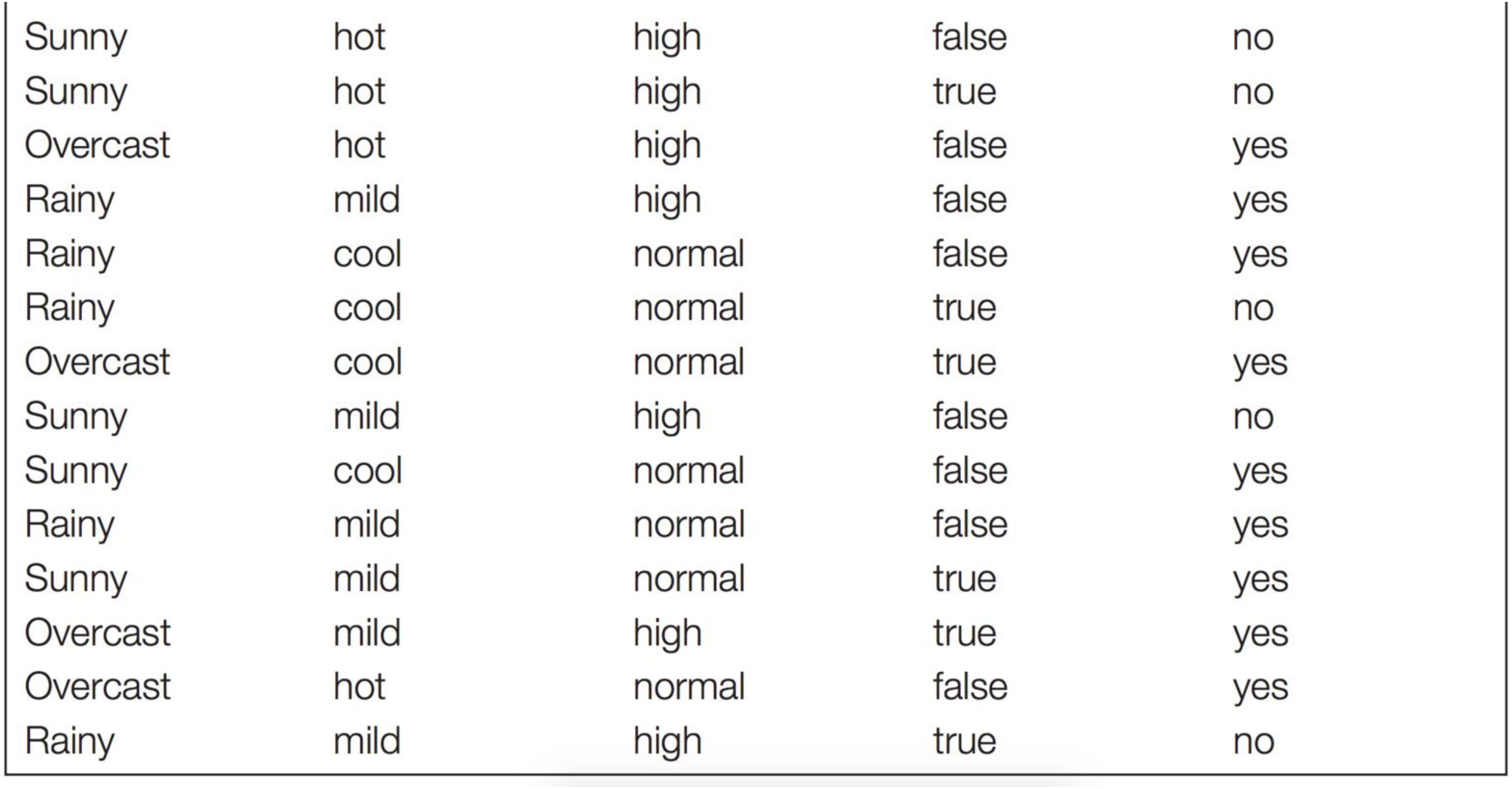
Springer

A great (free) resource

Classification

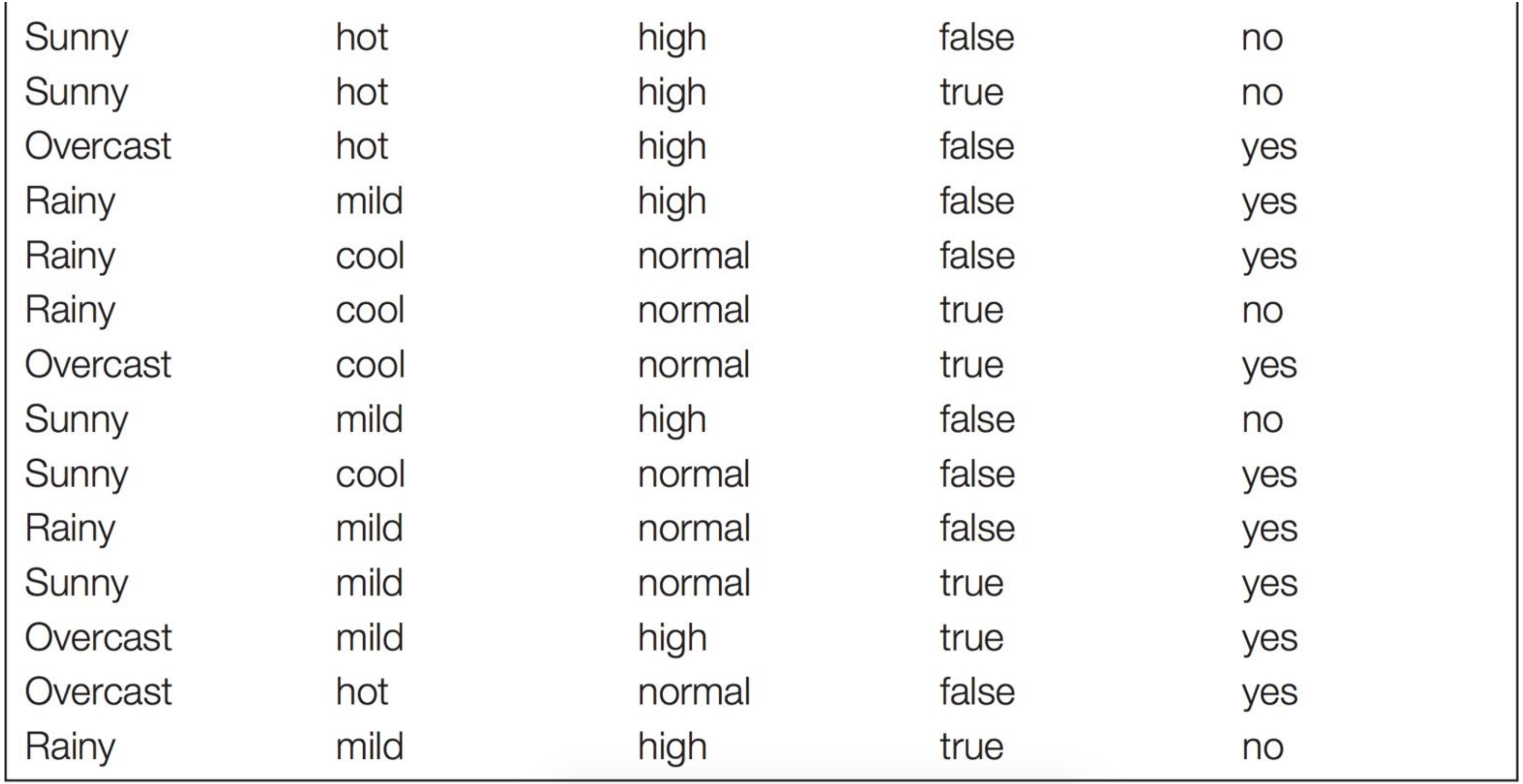
Classification is an attempt to determine if an **instance** (observation) is a member of a particular **class**.

In other words, classification predicting a **categorical variable.**



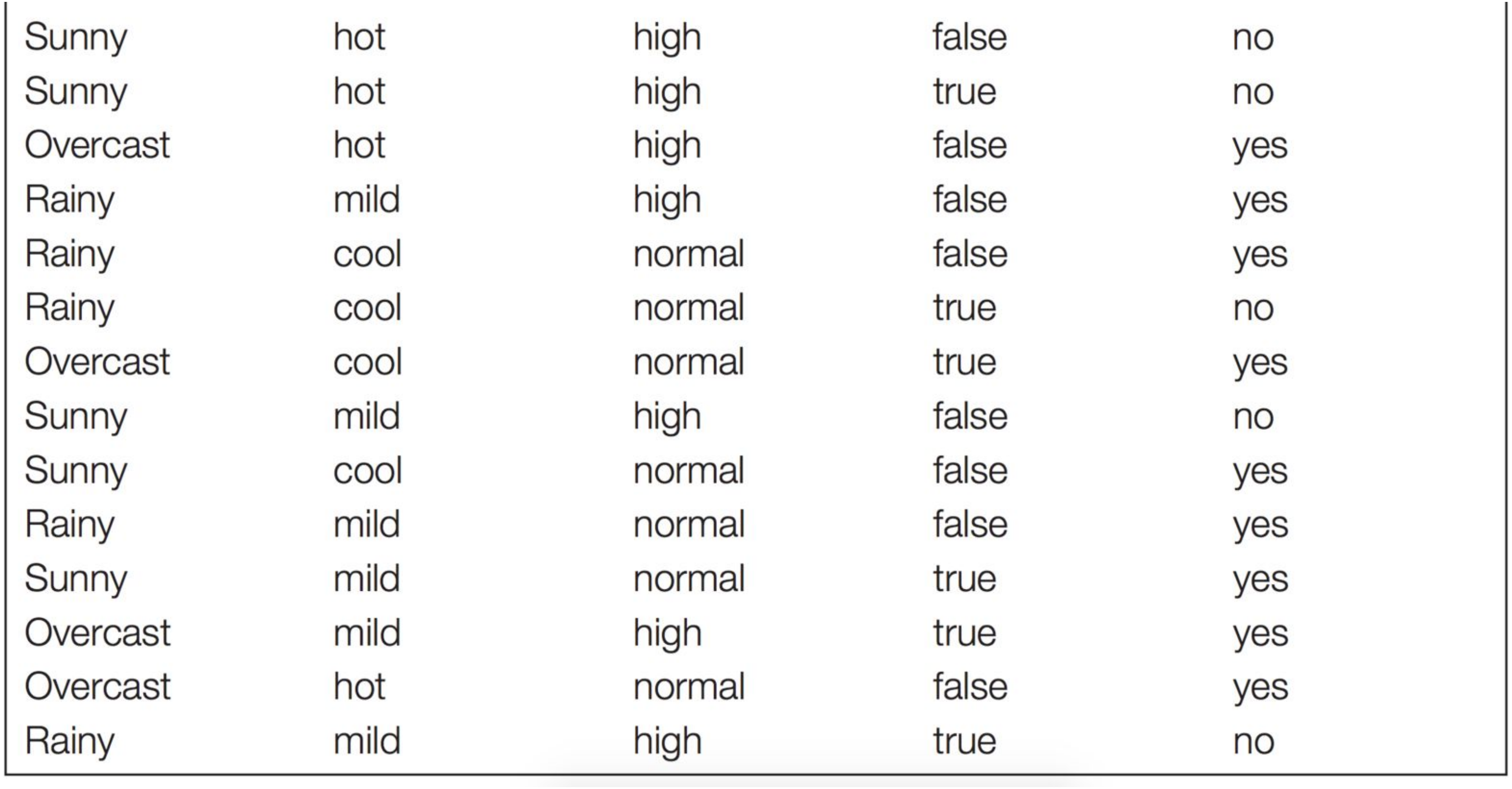
outlook	temp	work load	Likes R	Skips class
Sunny	hot	high	false	no
Sunny	hot	high	true	no
Overcast	hot	high	false	yes
Rainy	mild	high	false	yes
Rainy	cool	normal	false	yes
Rainy	cool	normal	true	no
Overcast	cool	normal	true	yes
Sunny	mild	high	false	no
Sunny	cool	normal	false	yes
Rainy	mild	normal	false	yes
Sunny	mild	normal	true	yes
Overcast	mild	high	true	yes
Overcast	hot	normal	false	yes
Rainy	mild	high	true	no

Let's imagine I'm trying to predict if a student will come to class



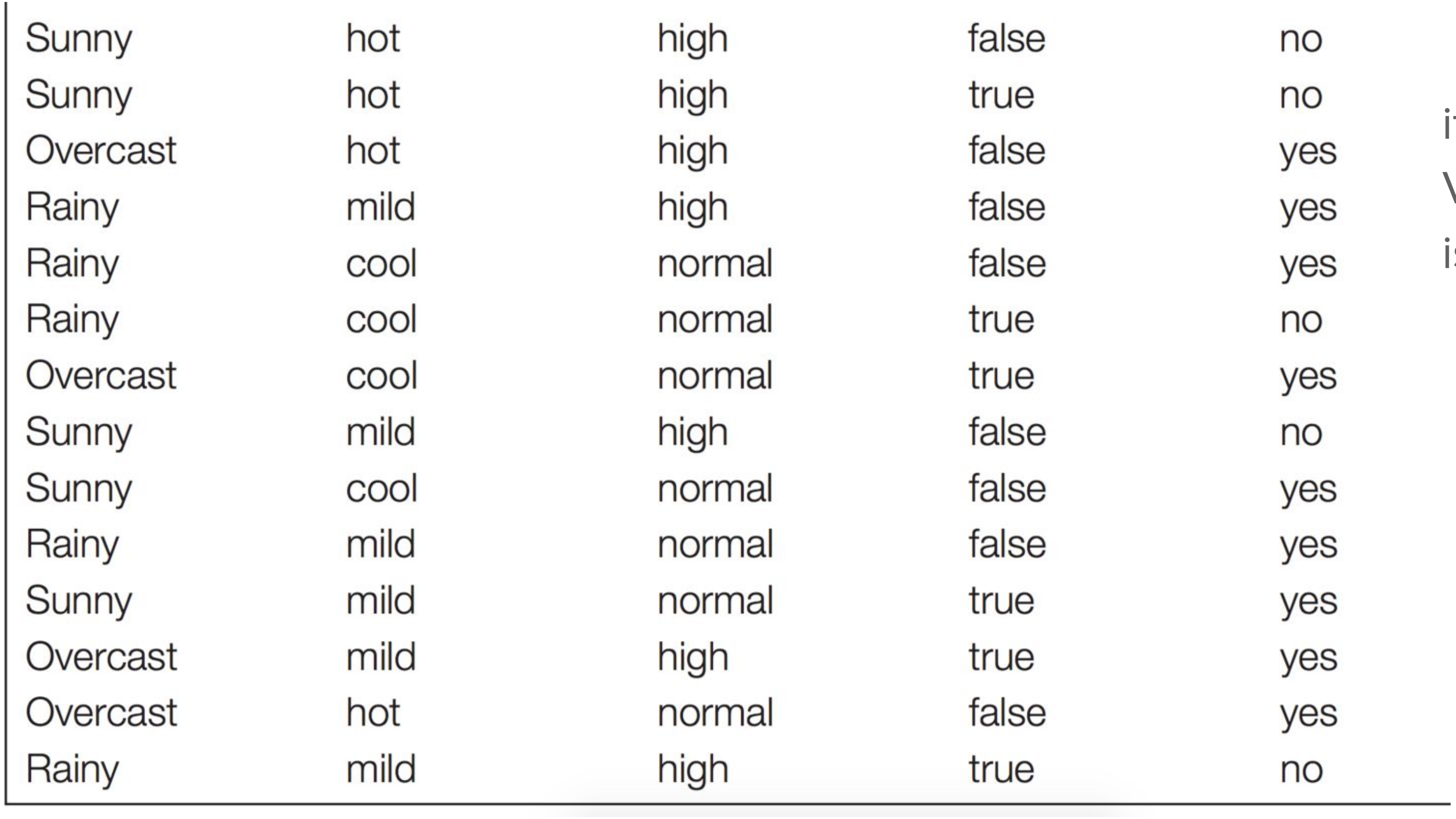
				outcome
outlook	**temp**	**work load**	**Likes R**	**Skips class**
Sunny	hot	high	false	no
Sunny	hot	high	true	no
Overcast	hot	high	false	yes
Rainy	mild	high	false	yes
Rainy	cool	normal	false	yes
Rainy	cool	normal	true	no
Overcast	cool	normal	true	yes
Sunny	mild	high	false	no
Sunny	cool	normal	false	yes
Rainy	mild	normal	false	yes
Sunny	mild	normal	true	yes
Overcast	mild	high	true	yes
Overcast	hot	normal	false	yes
Rainy	mild	high	true	no

Let's imagine I'm trying to predict if a student will come to class



Attributes (features)				outcome
outlook	**temp**	**work load**	**Likes R**	**Skips class**
Sunny	hot	high	false	no
Sunny	hot	high	true	no
Overcast	hot	high	false	yes
Rainy	mild	high	false	yes
Rainy	cool	normal	false	yes
Rainy	cool	normal	true	no
Overcast	cool	normal	true	yes
Sunny	mild	high	false	no
Sunny	cool	normal	false	yes
Rainy	mild	normal	false	yes
Sunny	mild	normal	true	yes
Overcast	mild	high	true	yes
Overcast	hot	normal	false	yes
Rainy	mild	high	true	no

Let's imagine I'm trying to predict if a student will come to class



Attributes (features)				outcome
outlook	**temp**	**work load**	**Likes R**	**Skips class**
Sunny	hot	high	false	no
Sunny	hot	high	true	no
Overcast	hot	high	false	yes
Rainy	mild	high	false	yes
Rainy	cool	normal	false	yes
Rainy	cool	normal	true	no
Overcast	cool	normal	true	yes
Sunny	mild	high	false	no
Sunny	cool	normal	false	yes
Rainy	mild	normal	false	yes
Sunny	mild	normal	true	yes
Overcast	mild	high	true	yes
Overcast	hot	normal	false	yes
Rainy	mild	high	true	no

if FEATURE(s) is VALUE, OUTCOME is VALUE

Write 3 rules to classify observations as skipping/attending class

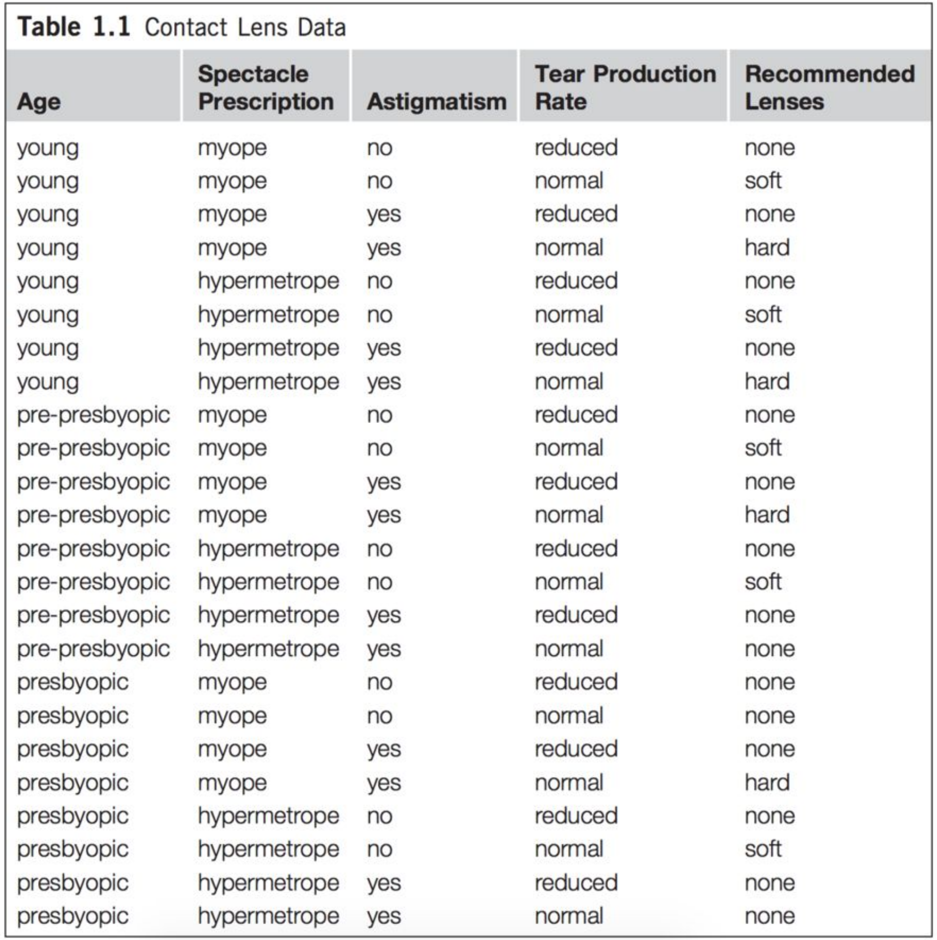
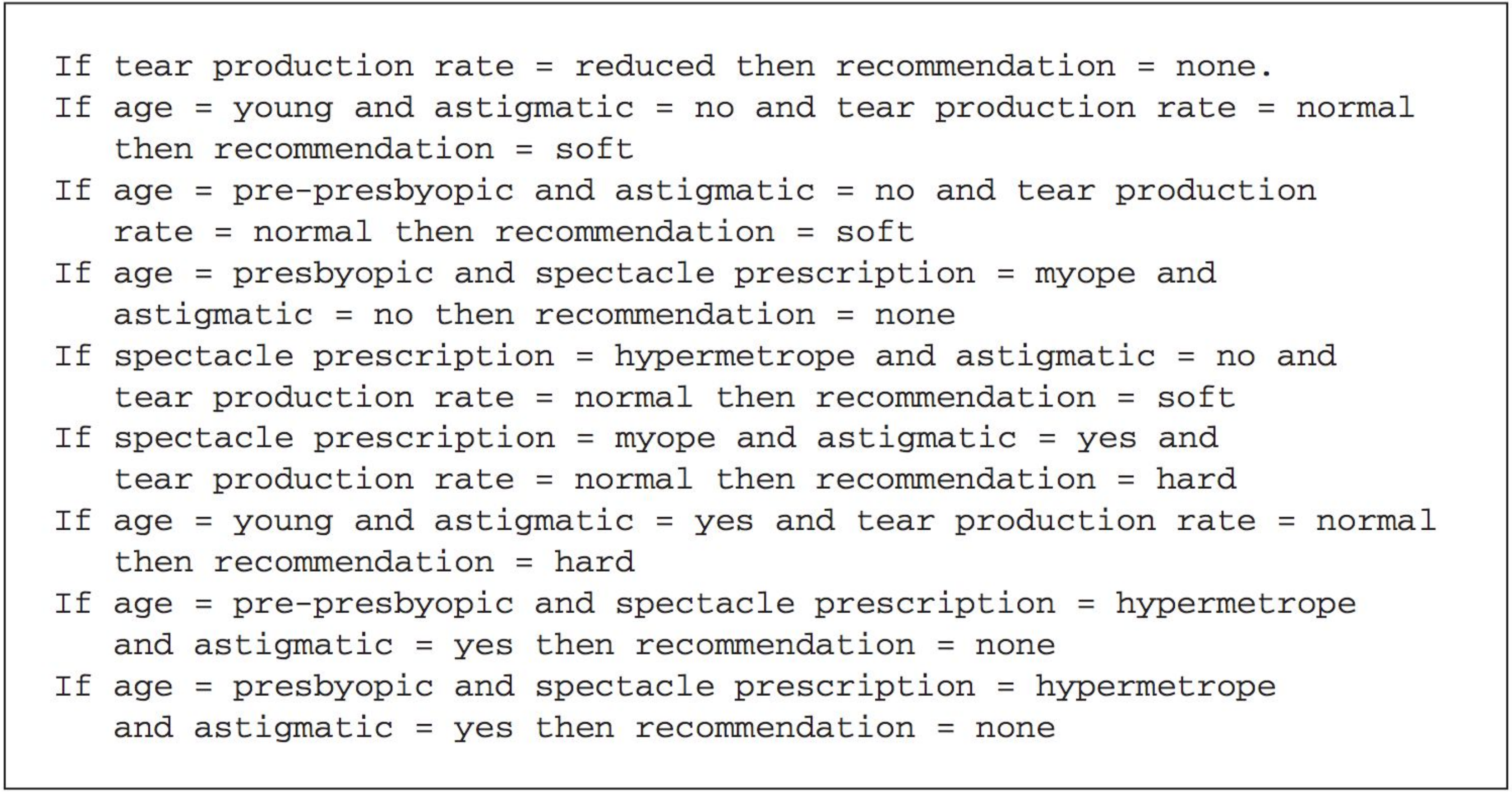


Table 1.1 Contact Lens Data

Age	Spectacle Prescription	Astigmatism	Tear Production Rate	Recommended Lenses
young	myope	no	reduced	none
young	myope	no	normal	soft
young	myope	yes	reduced	none
young	myope	yes	normal	hard
young	hypermetrope	no	reduced	none
young	hypermetrope	no	normal	soft
young	hypermetrope	yes	reduced	none
young	hypermetrope	yes	normal	hard
pre-presbyopic	myope	no	reduced	none
pre-presbyopic	myope	no	normal	soft
pre-presbyopic	myope	yes	reduced	none
pre-presbyopic	myope	yes	normal	hard
pre-presbyopic	hypermetrope	no	reduced	none
pre-presbyopic	hypermetrope	no	normal	soft
pre-presbyopic	hypermetrope	yes	reduced	none
pre-presbyopic	hypermetrope	yes	normal	none
presbyopic	myope	no	reduced	none
presbyopic	myope	no	normal	none
presbyopic	myope	yes	reduced	none
presbyopic	myope	yes	normal	hard
presbyopic	hypermetrope	no	reduced	none
presbyopic	hypermetrope	no	normal	soft
presbyopic	hypermetrope	yes	reduced	none
presbyopic	hypermetrope	yes	normal	none

What about when the data scales...?



```
If tear production rate = reduced then recommendation = none.
If age = young and astigmatic = no and tear production rate = normal
   then recommendation = soft
If age = pre-presbyopic and astigmatic = no and tear production
   rate = normal then recommendation = soft
If age = presbyopic and spectacle prescription = myope and
   astigmatic = no then recommendation = none
If spectacle prescription = hypermetrope and astigmatic = no and
   tear production rate = normal then recommendation = soft
If spectacle prescription = myope and astigmatic = yes and
   tear production rate = normal then recommendation = hard
If age = young and astigmatic = yes and tear production rate = normal
   then recommendation = hard
If age = pre-presbyopic and spectacle prescription = hypermetrope
   and astigmatic = yes then recommendation = none
If age = presbyopic and spectacle prescription = hypermetrope
   and astigmatic = yes then recommendation = none
```

This becomes more cumbersome

Decision Trees

Each **node** tests an **attribute** (feature)

Terminal **nodes (leafs)** assign a **classification**

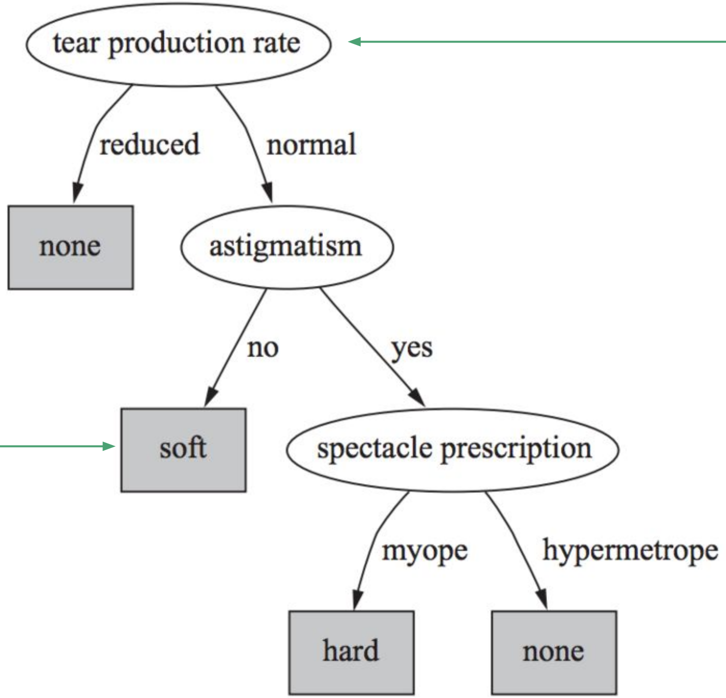


FIGURE 1.2

Decision tree for the contact lens data.

Translate rules into trees

Translate rules into trees: how to

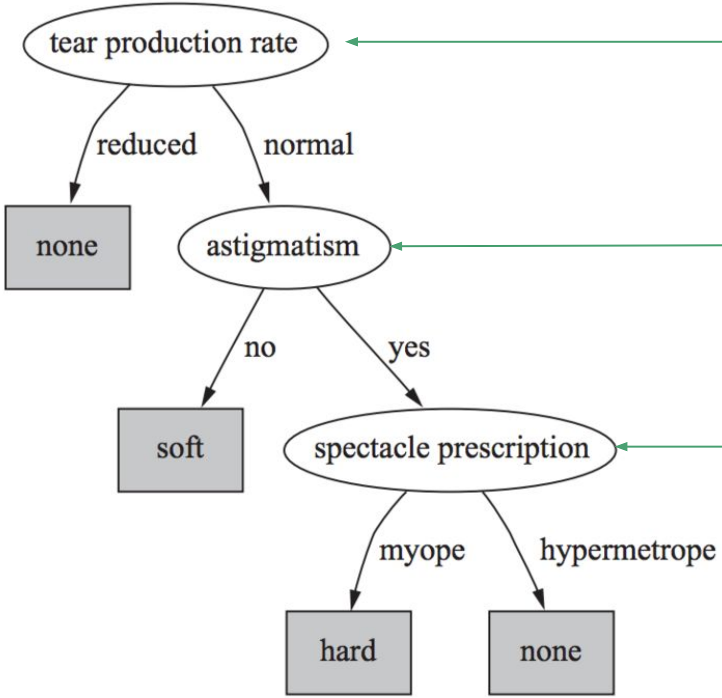


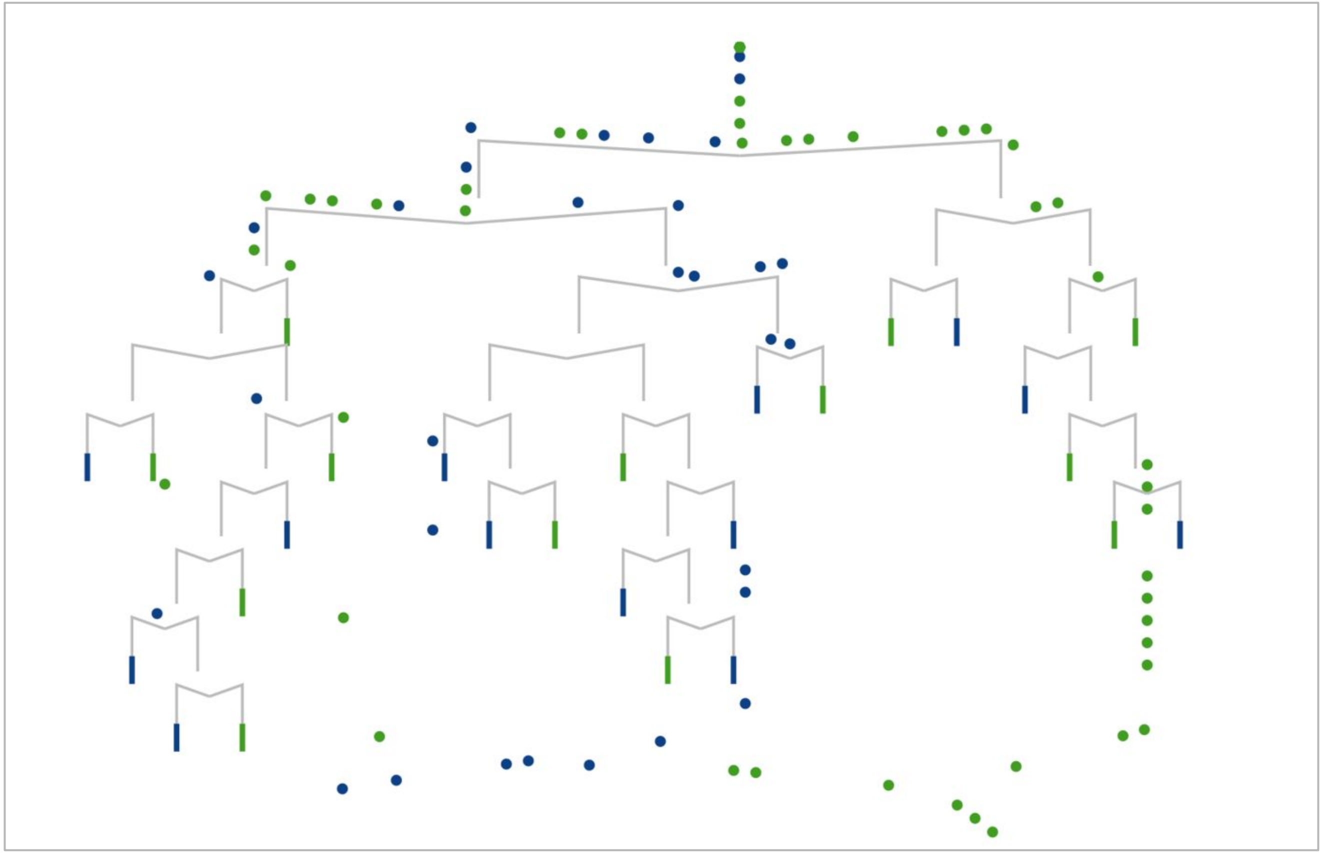
FIGURE 1.2

Decision tree for the contact lens data.

Pick attributes that produce the most "pure" branches

repeat

repeat



Explained visually

Classification in R

Classification in R

Pick one (of many) appropriate libraries

Load data into models

Visualize results

```r
# One of many libraries for classification / ML
library(rpart)

# Read in data
homes <- read.csv('part_1_data.csv')

# Use rpart to fit a model: predict `in_sf` using all variables
basic_fit <- rpart(in_sf ~ ., data = homes, method="class")

# How well did the model perform?
predicted <- predict(basic_fit, homes, type='class')
accuracy <- length(which(data[,'in_sf'] == predicted)) / length(predicted) * 100
```

Using the rpart library

Training/Testing Data

Right now, we're testing the model on the data used to train it (uh oh...)

To test prediction, we need to set aside data from the model

Many ways to separate data into train/test sets:

```r
train.indicies <- sample(seq_len(nrow(homes)), size = 100)
training.data <- homes[train.indicies,]
test.data <- homes[-train.indicies,]
```

Repetition

You often want to repeat your process

Helps avoid errors due to randomness

Allows you to create confidence intervals

May vary based on approach you're using

Module 15 Exercise-2

Integration with Shiny

How could a Shiny app help the machine learning process?

Reactive Expressions

Don't repeat time intensive tasks in Shiny

> *"Reactive expressions are a bit smarter than regular R functions. They **cache their values** and know when their values have become outdated. What does this mean? The first time that you run a reactive expression, the expression will **save its result** in your computer's memory. The next time you call the reactive expression, **it can return this saved result** without doing any computation (which will make your app faster)." -* source

```r
# Use a reactive expression so that you only run the code once
 getResults <- reactive ({
   return(simple_tree(input$features))
 })
 output$plot <- renderPlot({
   results <- getResults()
   return(results$plot)
 })
```

Reactive expression example

```
sidebarPanel(
     checkboxGroupInput("features", label = h3("Features to Use"),
          choices = colnames(homes)[2:ncol(homes)],
          selected = colnames(homes)[2:ncol(homes)])
   ),
```

Hint: using data to create options

Module 15 Demo-1

Upcoming...

Keep working on your final projects!