Machine Learning, Part I

INFO 370

Learning Objectives

Discuss final projects

Understand the use-cases for **machine learning**

Distinguish between **supervised** and **unsupervised** tasks

Understand the algorithm behind **decision trees**

Understand the importance of and syntax for creating **training and testing** data

Understand the algorithm behind **K Nearest Neighbors**

Be able to create and use a **validation** data set

Search for the best parameters for you models using **grid search**

Articulate the importance of (and process for) **normalizing** (scaling) your data

Process

Discuss machine learning at a high(er) level, then:

- Discuss a concept
- Implement **together** in this week's notebook (nb-6)
- Answer any questions
- Repeat

Final Projects

Machine Learning Overview

So much machine learning

So much machine learning

So much machine learning

Inference
(interpretability)

Prediction
(accuracy)

(more stats)

(more ml)

Tradeoffs

Statistical Approach

Estimate relationship between features (sq. feet, # bedrooms) and price.

Strength is **inference**, prediction is *possible*.

$Price = B_0 + B_1 * SqFeet + B_2 * Bedrooms$

One Machine Learning Approach

Find the price of the most similar house(s) based on features.

Strength is **prediction**, inference is difficult.

Price = Price of most similar house(s)

Doing data science on housing prices

Some definitions // thoughts

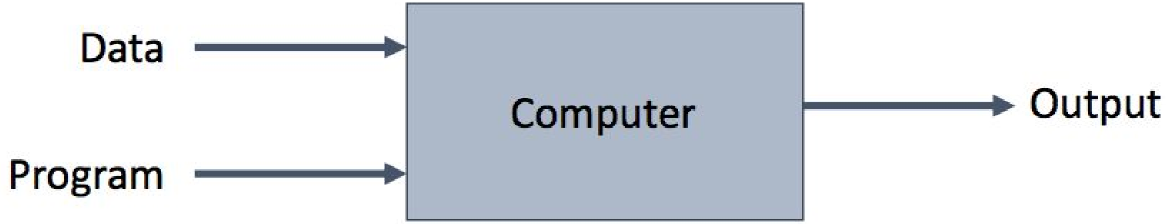
Machine learning is the science of getting computers to act *without being explicitly programmed*

Machine learning is a scientific discipline that explores the construction and study of **algorithms** that can **learn from** and **make predictions** using data

Machine learning is a natural outgrowth of the intersection of **Computer Science** and **Statistics**

What is ML?

Traditional Programming



Machine Learning

?

Diagram from Lavi Aulck

What is ML?

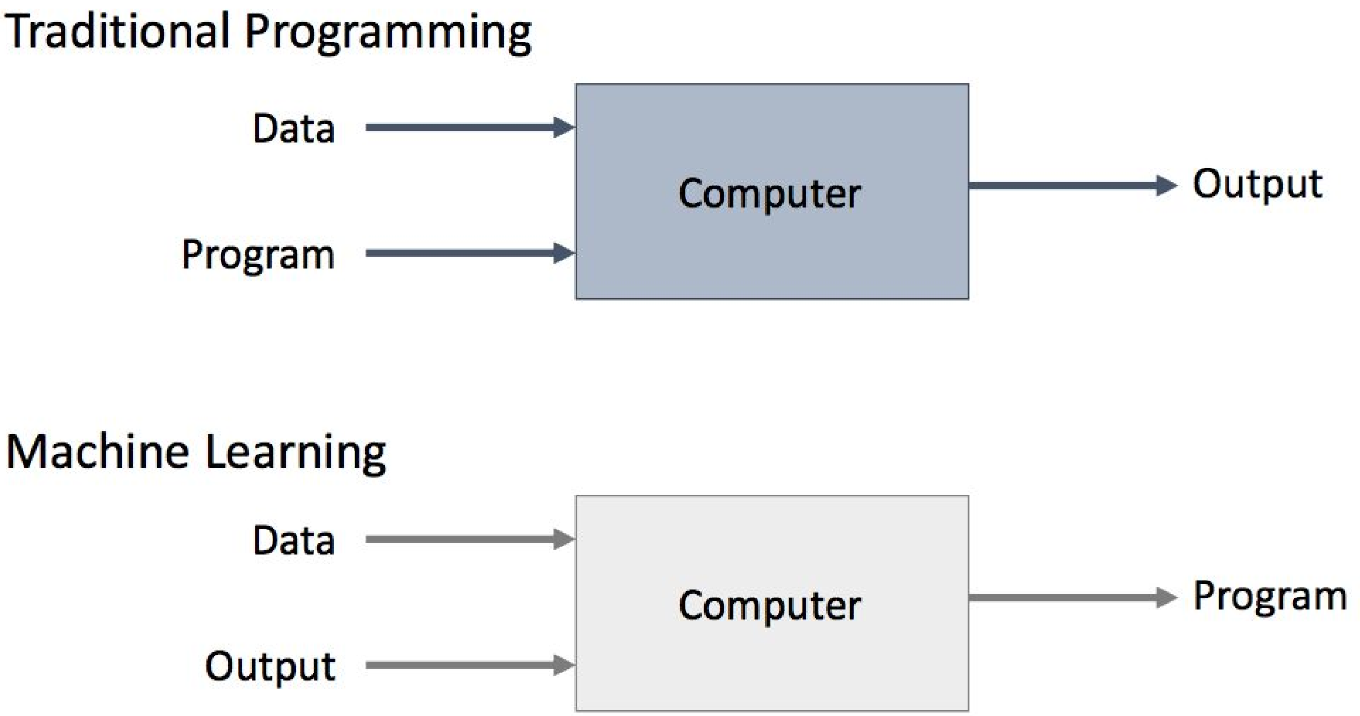


Diagram from Lavi Aulck

Types of Machine Learning

Generally two types of machine learning: **supervised** and **unsupervised**

Key distinction is whether you know the correct answer (**outcome**)

Supervised Learning

Develop a model that can accurately predict the outcome based on data features

You know the outcome (in your sample)

Examples

- Predict if students will graduate from UW
- Predict health outcome for patients
- Classify handwritten numbers

Methods

- Linear regression
- Decision trees
- K Nearest Neighbors

Unsupervised Learning

Develop a model to create groupings from unlabeled observations

You **don't know** the outcome (in your sample)

Examples

- Grouping similar books together
- Cluster customers by behavior (no label)

Methods

- K-means clustering
- Principal component analysis
- Topic modeling

In this course, we'll be using **supervised learning** approaches to **make predictions** in unobserved contexts.

Decision Trees

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

Challenge: predict outcome (lenses) based on features

```
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
```

Write rules for our dataset

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

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

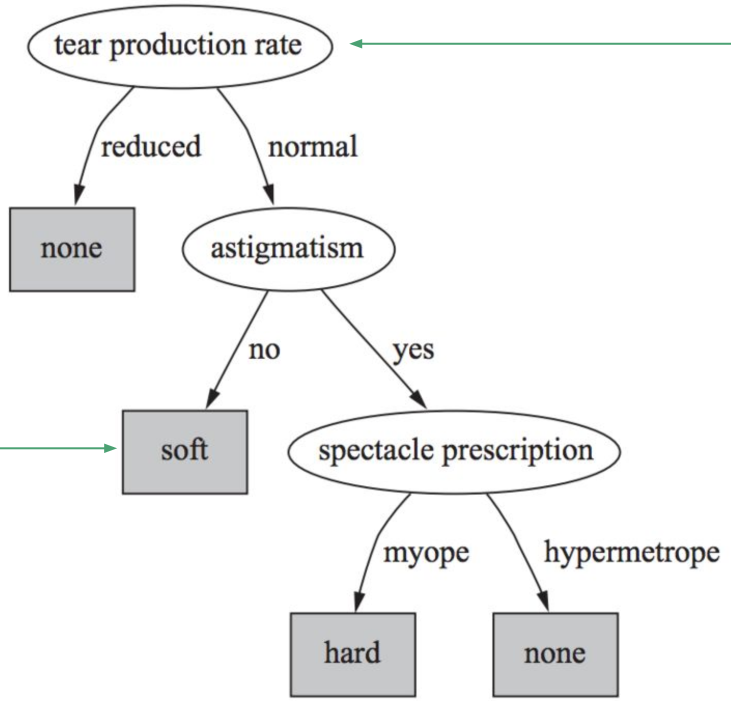


FIGURE 1.2

Decision tree for the contact lens data.

Translate rules into trees

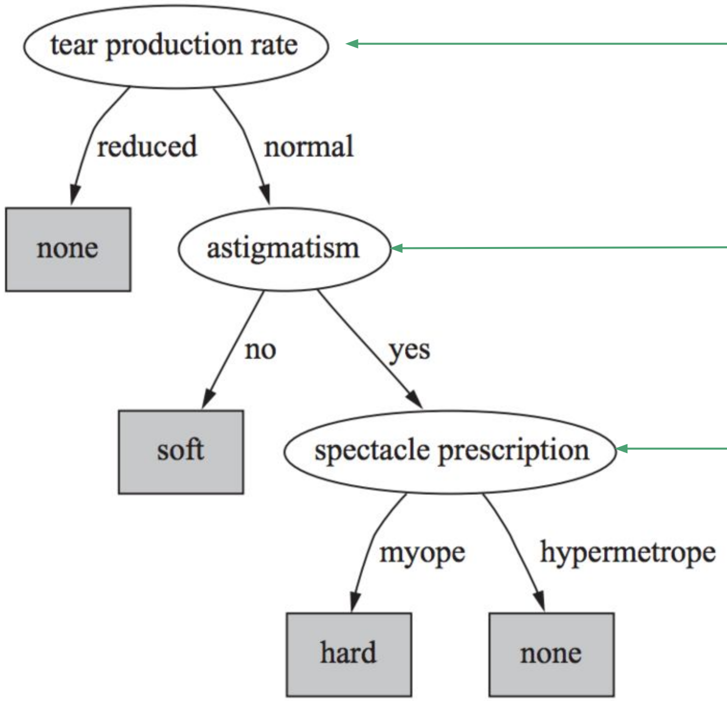


FIGURE 1.2

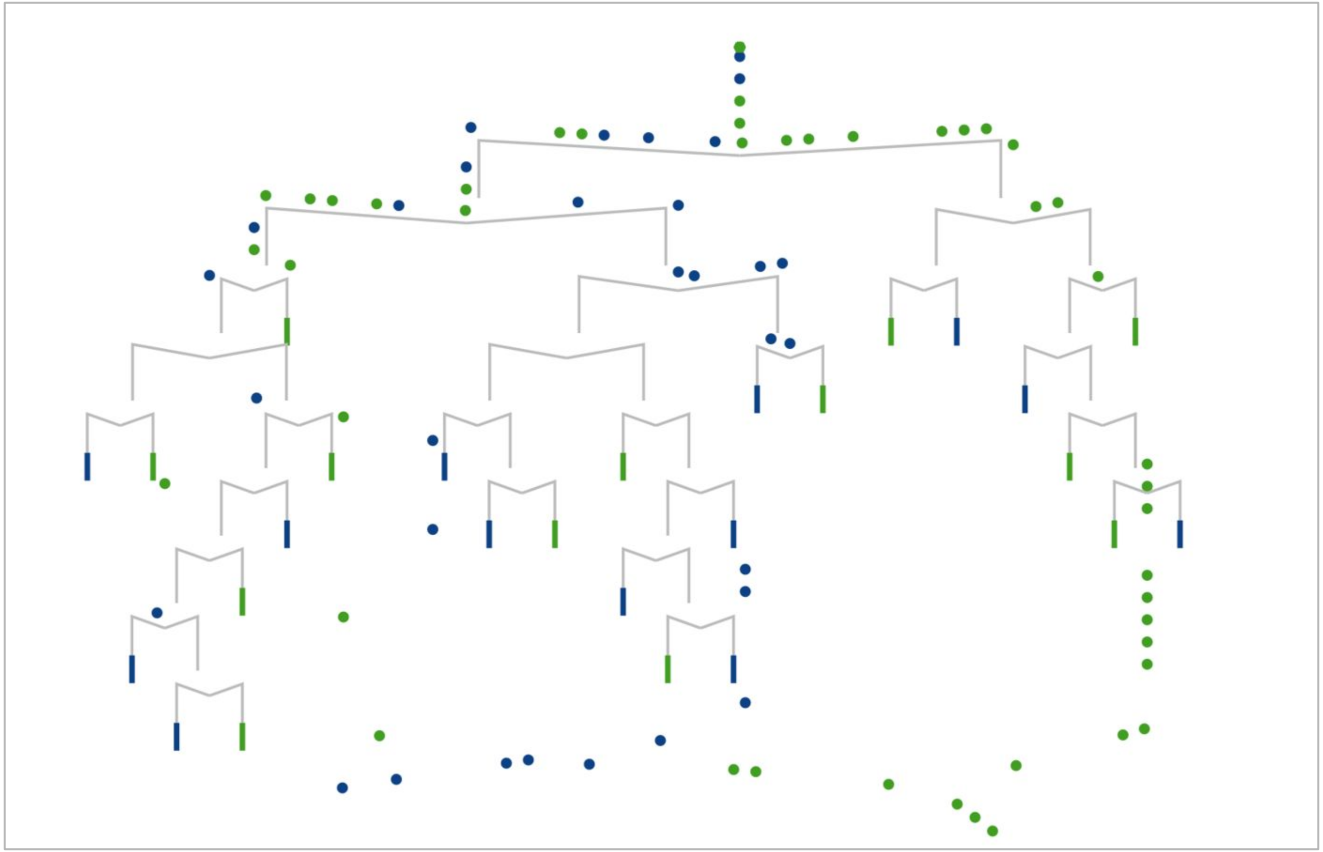
Decision tree for the contact lens data.

Pick attributes that produce the most "pure" branches

repeat

repeat

Translate rules into trees: how to



Explained visually

nb-set-6

Training // Testing Data

Training and Testing Data

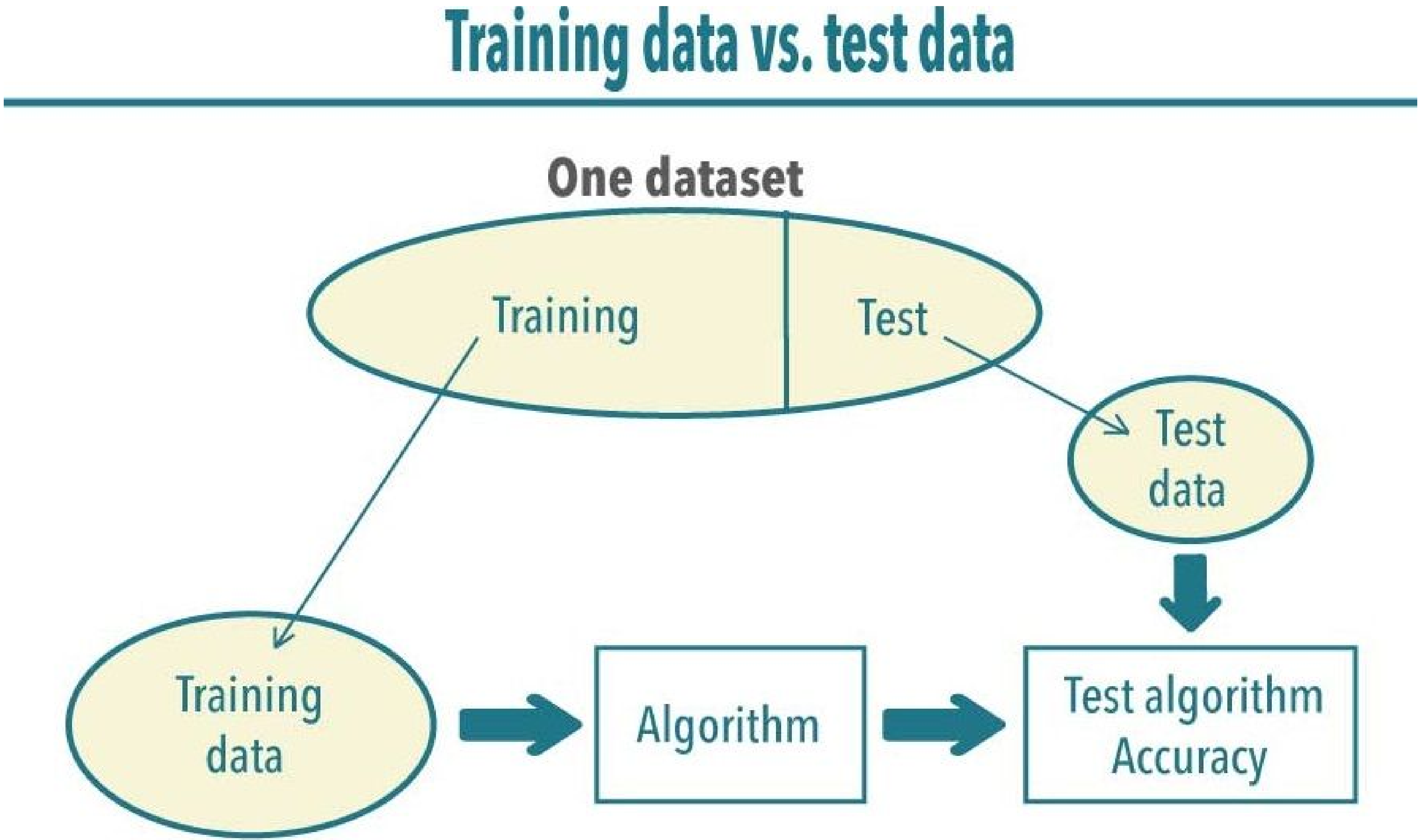
We need to both **create** and **assess** our algorithm using our data

We can't use the same data to do both

Split our data into:

- **Training data:** to build our model
- **Testing data**: to assess our model

Optimize performance on the **test data**



Training and Testing data (source)

K Nearest Neighbors

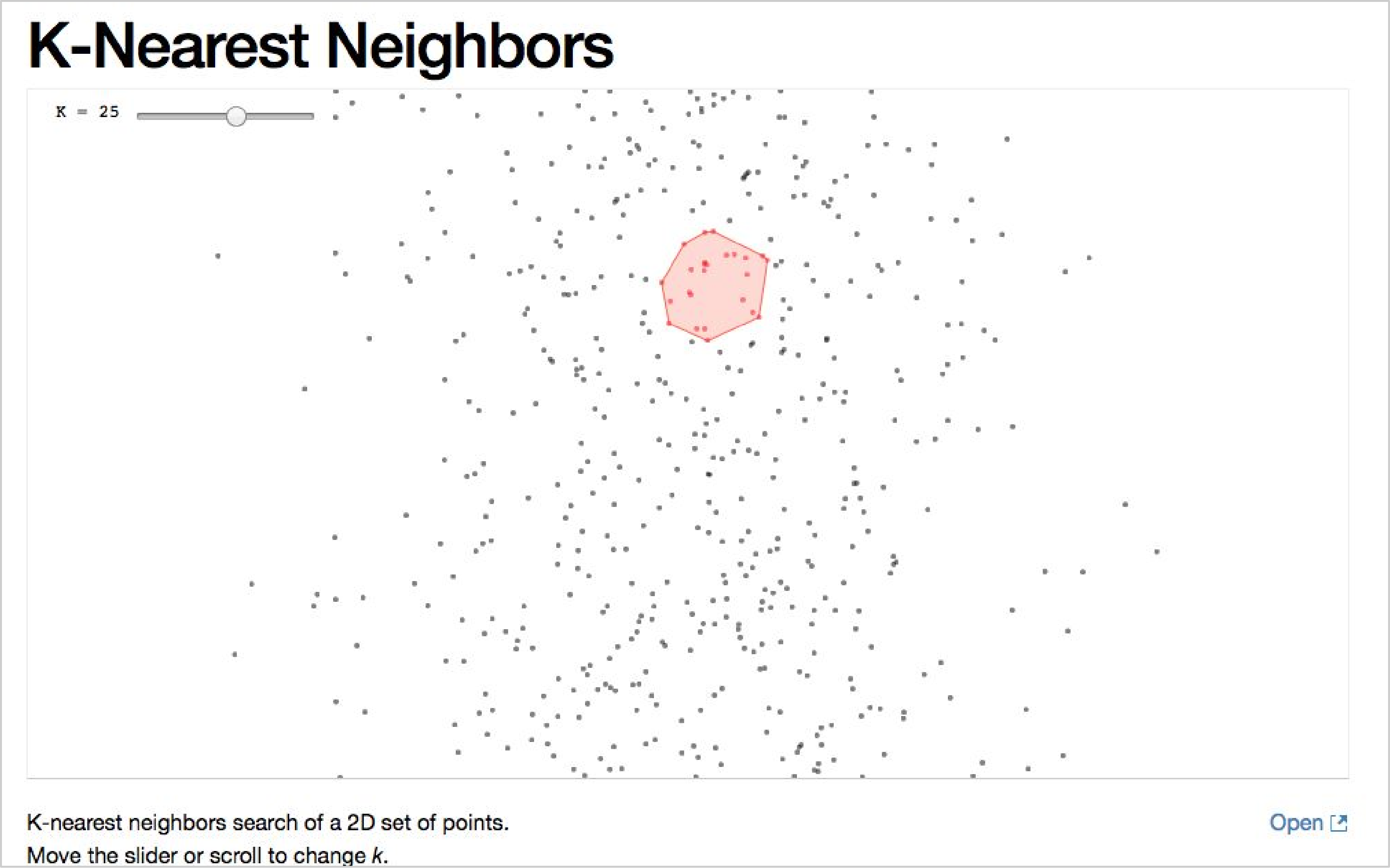
K Nearest Neighbors

Predict outcome based on most similar **K** neighbors in *feature space*

Can be used for **classification** (predict categories) or **regression** (predict values)

Uses instance-based or lazy learning to compute value at run-time

Uses **majority class** or **average value** to predict for each observation


K-Nearest Neighbors
K = 25
K-nearest neighbors search of a 2D set of points.
Move the slider or scroll to change k.
Open

KNN (link)

Validation Data

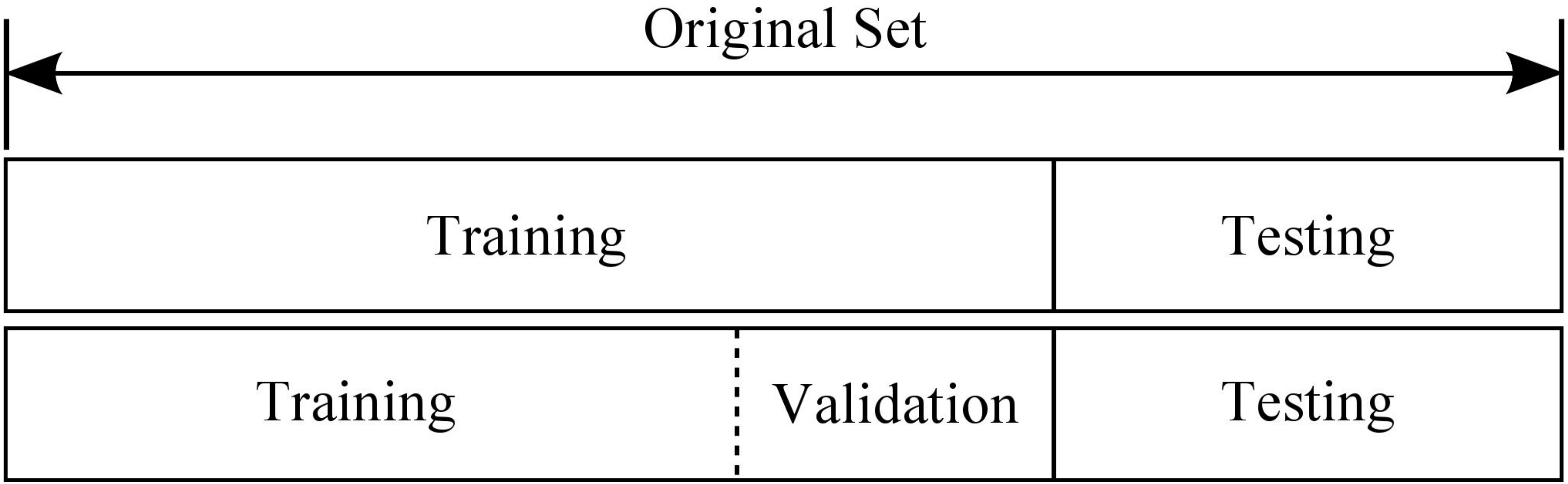
Validating results

We **never look at the test data** until our model is complete

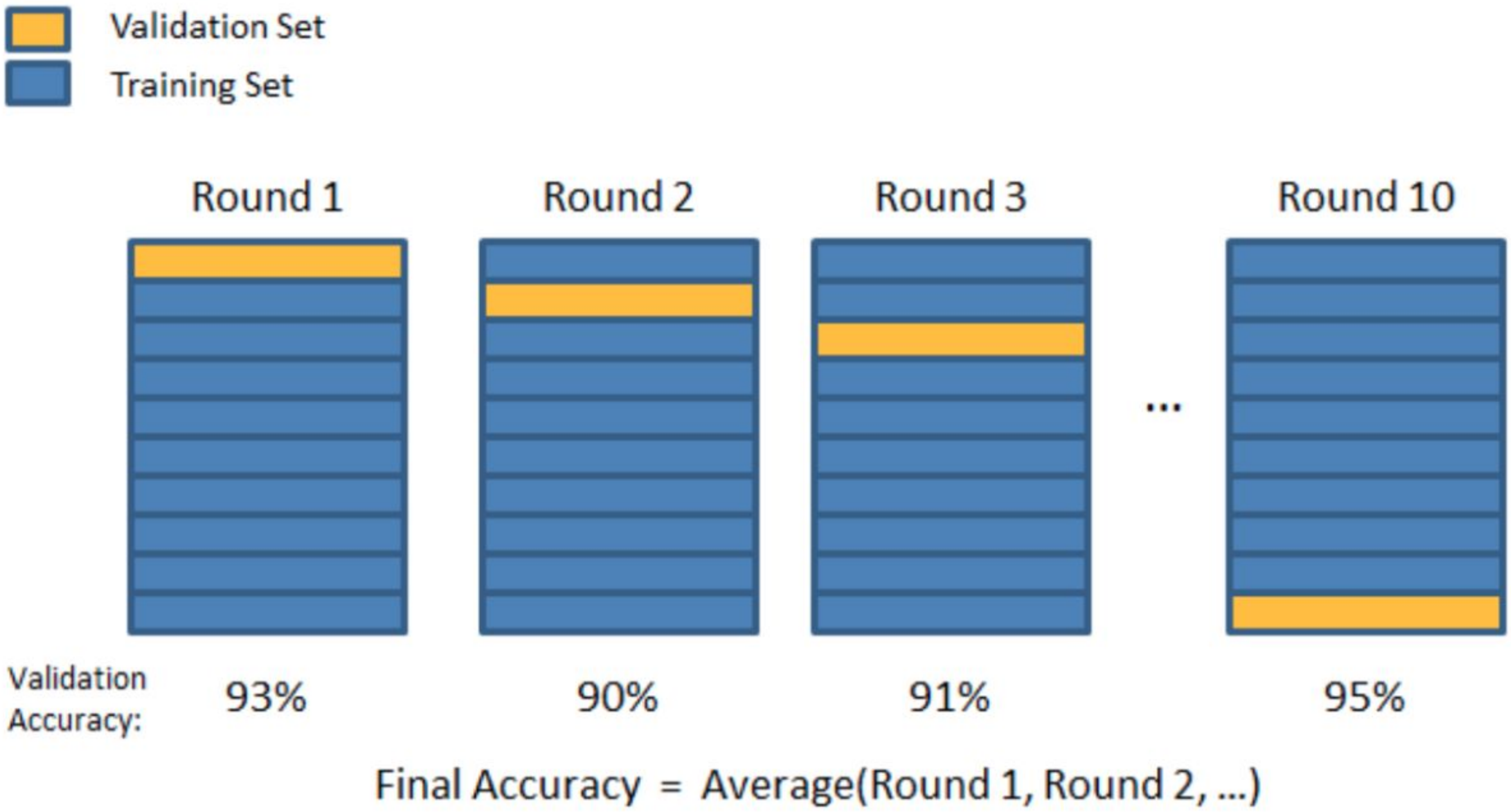
We may want to compare multiple models to one another

We can use some of the training data to assess (**validate**) our models

This is something we may want to repeat to avoid errors due to randomness



Creating validation data (source)



KFold Cross Validation (source)

Grid Search

Searching for parameters

So far:

- Create testing and training data
- Use cross validation to assess model performance
- Predict on our dataset

We'll want to find the **optimal set** of parameters for creating our models

This is call **tuning** your model (or, to be really fancy, *hyperparameter tuning*)

To tune our model, we'll perform the above steps separately for each parameter set

Normalizing // Scaling Data

Normalizing // Scaling Data

Many algorithms are distance based (KNN)

You'll need to normalize (scale) your data to weight features consistently

Various ways to normalize your data

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

$$x_{new} = \frac{x - \mu}{\sigma}$$

Upcoming...

r4-ethics due ***next Tuesday***

Project proposals due ***next Tuesday***

Notebook 6 due **Friday night**

<u>This week</u>

- Machine Learning