Machine Learning Intro

Fall R ’23

Dominic Bordelon, Research Data Librarian, ULS

Agenda

1. What is machine learning?
2. Supervised learning
    - Regression, K-nearest neighbors, decision trees…
3. Model assessment
4. Unsupervised learning
    - Principal component analysis, K-means clustering
5. Reinforcement learning: learning with rewards

About the trainer

Dominic Bordelon, Research Data Librarian
University Library System, University of Pittsburgh
dbordelon@pitt.edu

Services for the Pitt community:

- Consultations
- Training (on-request and via public workshops)
- Talks (on-request and publicly)
- Research collaboration

Support areas and interests:

- Computer programming fundamentals, esp. for data processing and analysis
- Open Science and Data Sharing
- Data stewardship/curation
- Research methods; science and technology studies

Fall R Series

#	Date	Title
1	8/29	Getting Started with Tabular Data
2	9/5	Working with Data Frames
3	9/12	Data Visualization
4	9/19	Inference and Modeling Intro
5	**9/26**	**Machine Learning Intro**

R and RStudio Drop-In Hour

Bring your R questions!
If we don't know the answer,
we'll help you look for it.
Students, post-docs, faculty, and staff are all welcome.

For R users in the Pitt community

Tuesdays, 4:30–5:30 pm
Fall semester 2023

Hillman Library, Rm. 255

https://bit.ly/pitt-r-office-23

Service provided by:
University Library System,
Digital Scholarship Services

University of Pittsburgh | Library System

Scan QR code for more info

DSS
DIGITAL SCHOLARSHIP SERVICES
@PITT

Today's packages

`tidymodels`, particularly `parsnip`: standardized modeling interface

```
1  install.packages(c("tidymodels", "e1071"))
```

💡 In terms of writing code, there are a variety of approaches to modeling in R, even for fitting the same type of model (e.g., when implemented by different package developers). We will favor the tidymodels approach.

…and using penguins examples

```
1 library(palmerpenguins)
2 data(penguins)
3 names(penguins)
```

What is machine learning?

What is machine learning?

- Using the computer to learn from data (yes, it is that simple)
- Applies **statistics** (finding information in data) and **computer science** (algorithmic design and implementation)
- Not quite equivalent to AI; AI is built on ML
  - Historically speaking, AI dates back to 1950s and aims to emulate human cognition and behavior, while ML originates in finding insights in data
  - The two have converged because ML are the instruments used to approach AI
  - Ex. an AI which “sees” subjects in an image, is in fact deploying ML methods that parse the image into regions/forms of interest, then classify the form into some category, based on what has been seen and categorized before. Roughly speaking, each component of this process corresponds to an ML algorithm.

ML (and statistics) terminology

Often describes statistical concepts with different language, due to separate disciplinary traditions.

Some terminology encountered in ML and statistics. Source: Adapted from Zachary Kurtz, "Translating Between Statistics and Machine Learning" (2018)

Statistics term	**ML / Computer Science term**
observation, case	example, instance
response variable, dependent variable	label, output
predictor, independent variable	**feature**, input
regression	regression, supervised learner, machine
estimation	learning
outlier	anomaly

Terminology caveats

⚠ Terms/concepts to be careful with in ML, coming from stats:

- hypothesis (sometimes an output of a classifier model)
- bias (broader meaning)
- causality (sometimes less rigorous than stats)

Supervised learning

Term review: regression and classification

Regression

- Understand a **numeric / continuous variable**'s relationship to one or more predictors
- or, Predict some numeric / continuous value from observations
- "How tall (y) will my plant be if I give it x amount of water?"

Classification

- Understand a **categorical variable**'s relationship to one or more predictors
- or, Predict some numeric / continuous value from observations
- "What kind of seeds (y) did I plant if the current height is x?"

Supervised learning

Supervised learning has a predictive output or target (regression or classification of a variable). A model is fit which predicts (or retrodicts) some y from one or more x.

- Regression (esp. linear and logistic)
- K-Nearest Neighbors classifier
- Decision trees
- Support vector machines (SVM)
- Naive Bayes classifier
- Linear discriminant analysis (LDA)
- Neural networks (incl. deep learning)

Regression

- Plot a scatterplot of x and y, then fit a line through it.
- We want to solve the equation $y = mx + b$, where m is the slope of the line of best fit, and b is the y-intercept. x is an independent or *predictor* variable, and y is a dependent or *response* variable.
  - Stats notation: $Y = \beta_0 + \beta_1 X + \epsilon$
- A positive or negative slope corresponds to a positive or negative relationship (respectively) between x and y. The steeper the slope, the stronger the relationship.
- *Simple* linear regression uses only two continuous variables and a straight line. Multiple regression may have several $X_2 \ldots X_p$.

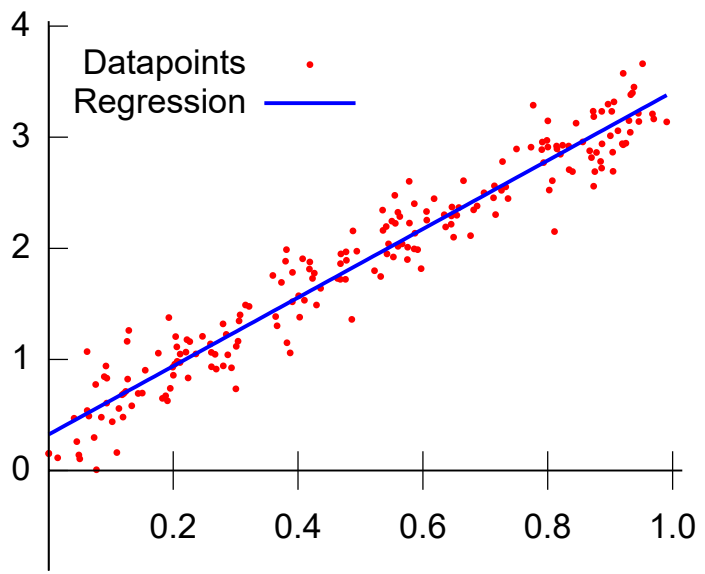
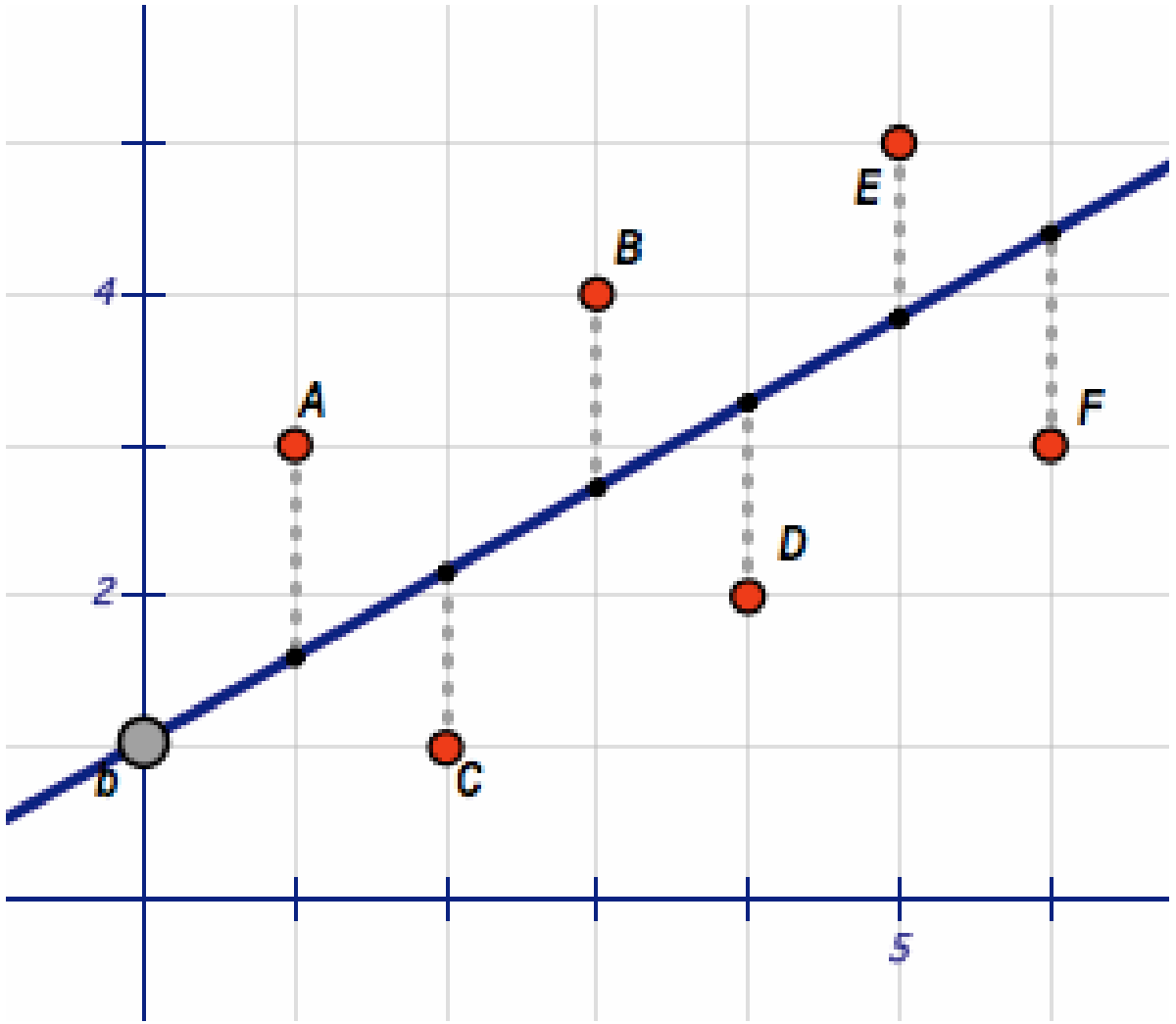
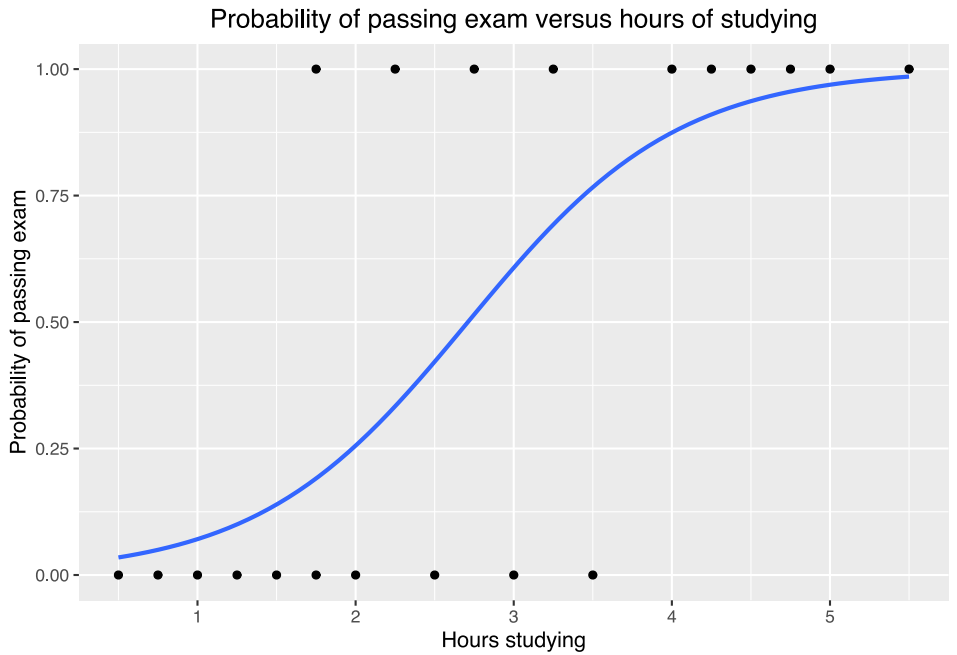


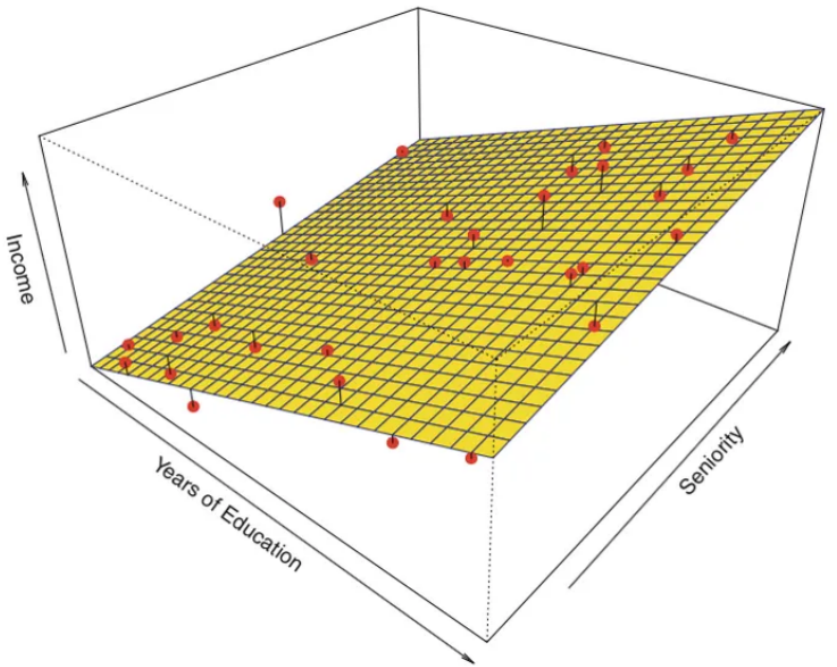
Image source: Berland via Wikimedia Commons (public domain)



Animation of the least-squares method of fitting a linear regression. Data points are red, and their residuals (distance to the regression) are dotted gray lines. The mathematical goal in least-squares fitting is to minimize the sum of squared residuals. Image source: Stephen1729 via Wikimedia Commons (CC BY-SA 3.0)



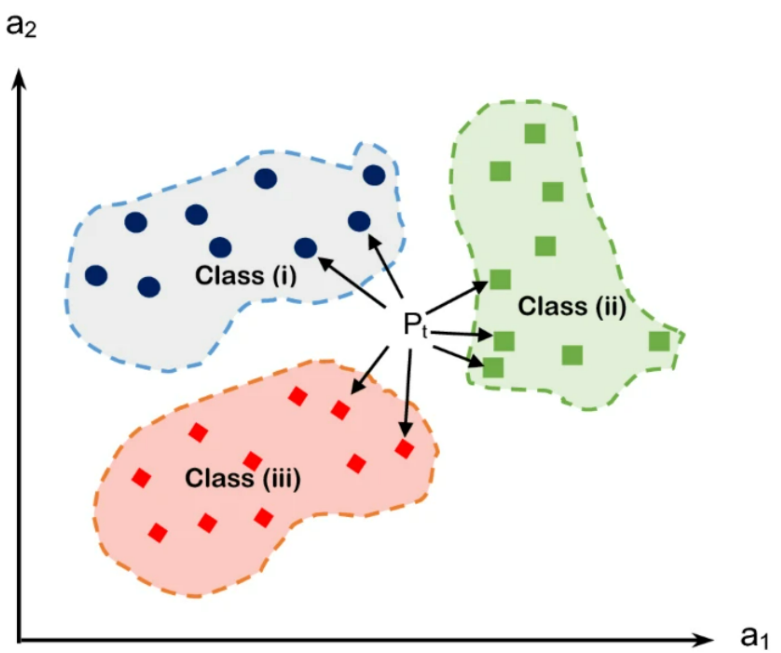
Example of a logistic regression, which predicts a probability between two outcomes. The logistic regression is widely used a binomial classifier. Image source: Canley via Wikimedia Commons (CC BY-SA 4.0)



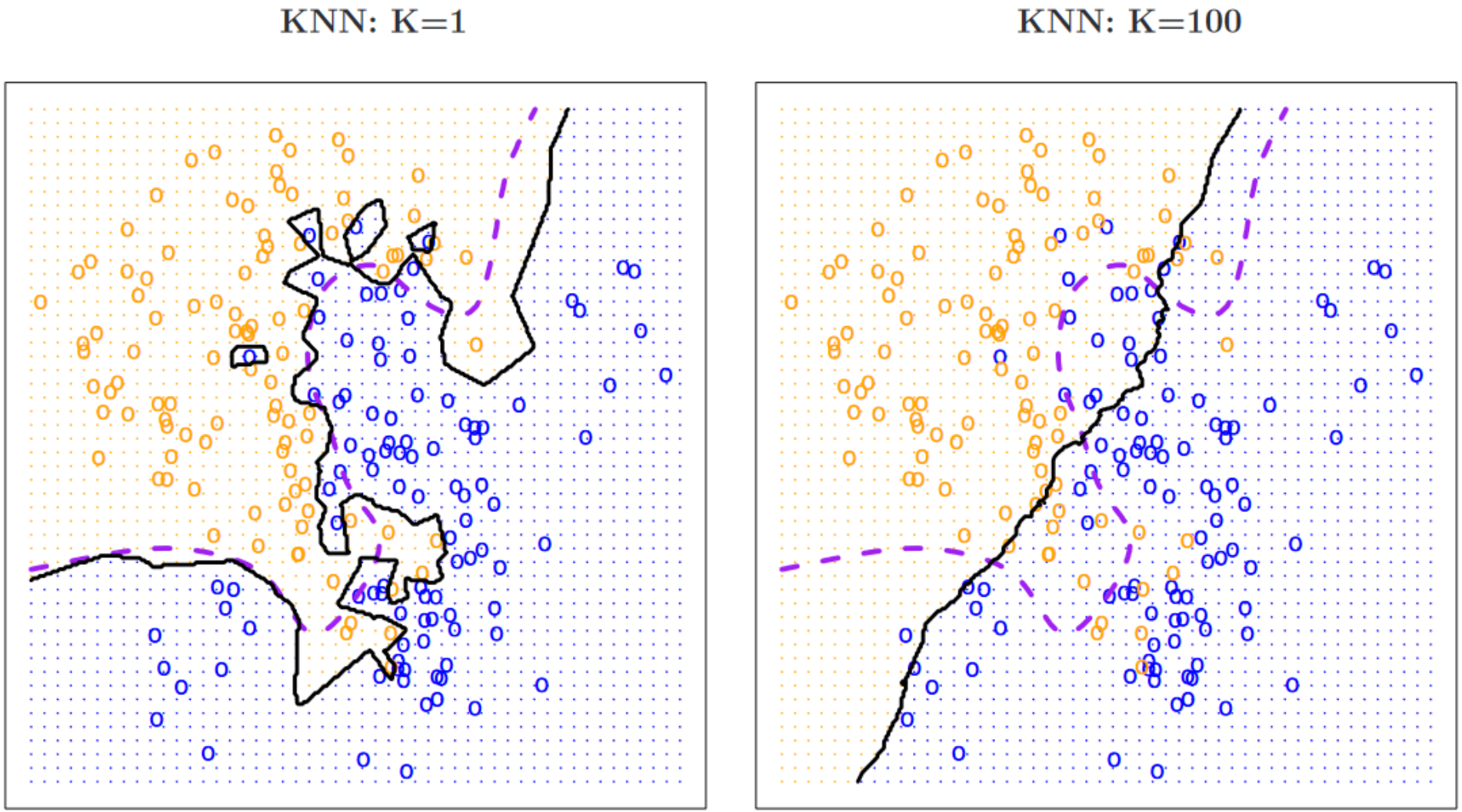
Visualization of a multiple linear regression in three dimensions, resulting in a regression plane. Image source: Christa Dawson, “Understanding Multiple Linear Regression”

K-Nearest Neighbors classifier

- An observation of interest is compared to its K nearest neighbors in p-dimensional space.
  - K is the tuning parameter, chosen by the analyst
  - The observation is classified according to the majority of its neighbors



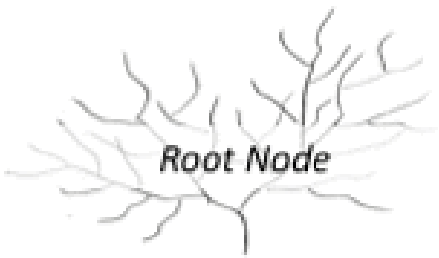
Pt is an observation of unknown category, which we would like to classify. The K nearest neighbors (K = 7 here) are found using a distance function, and the majority class of those neighbors is assigned to Pt. The result would be Class ii in this case. Image source: Atallah, Badawy, and El-Sayed 2019

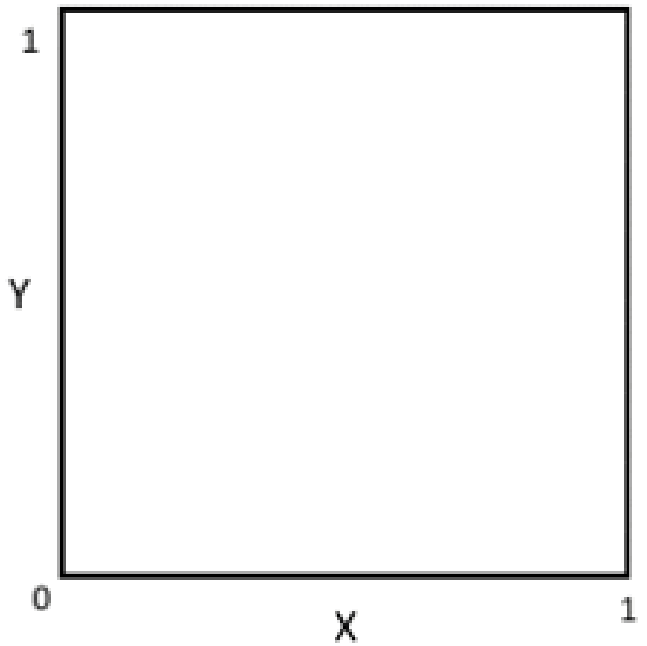


Choice of K has a great effect on the decision boundary (black line). K = 1 will overfit, but K = 100 is far too generalized in this case. The dashed purple line compares a Bayesian classifier fit. Image source: James et al. 2021

Decision trees

- We *stratify* the predictor space using *splitting rules*
  - *Recursive binary splitting*:
    select the X most correlated with $Y \rightarrow$
    find a good "cut point" (decision boundary) to split the data, according to $Y \rightarrow$
    in the two resulting bins, the process is repeated, and so on, until a stop condition is reached.
- May be used for classification but also regression
- Very explainable, but "typically are not competitive with the best supervised learning methods" in terms of accuracy (James et al. 2021); also not very robust on their own
  - Methods like *random forests*, *bagging*, and *boosting* extend the decision tree algorithm and address issues of accuracy and robustness





For more tutorials: annalyzin.wordpress.com

Animation of a simple decision tree example. Each binary branch in the tree on the left corresponds to a partitioning in the x-y space. The response variable (output) of this model is gray/green color classification. Image source: Algobeans

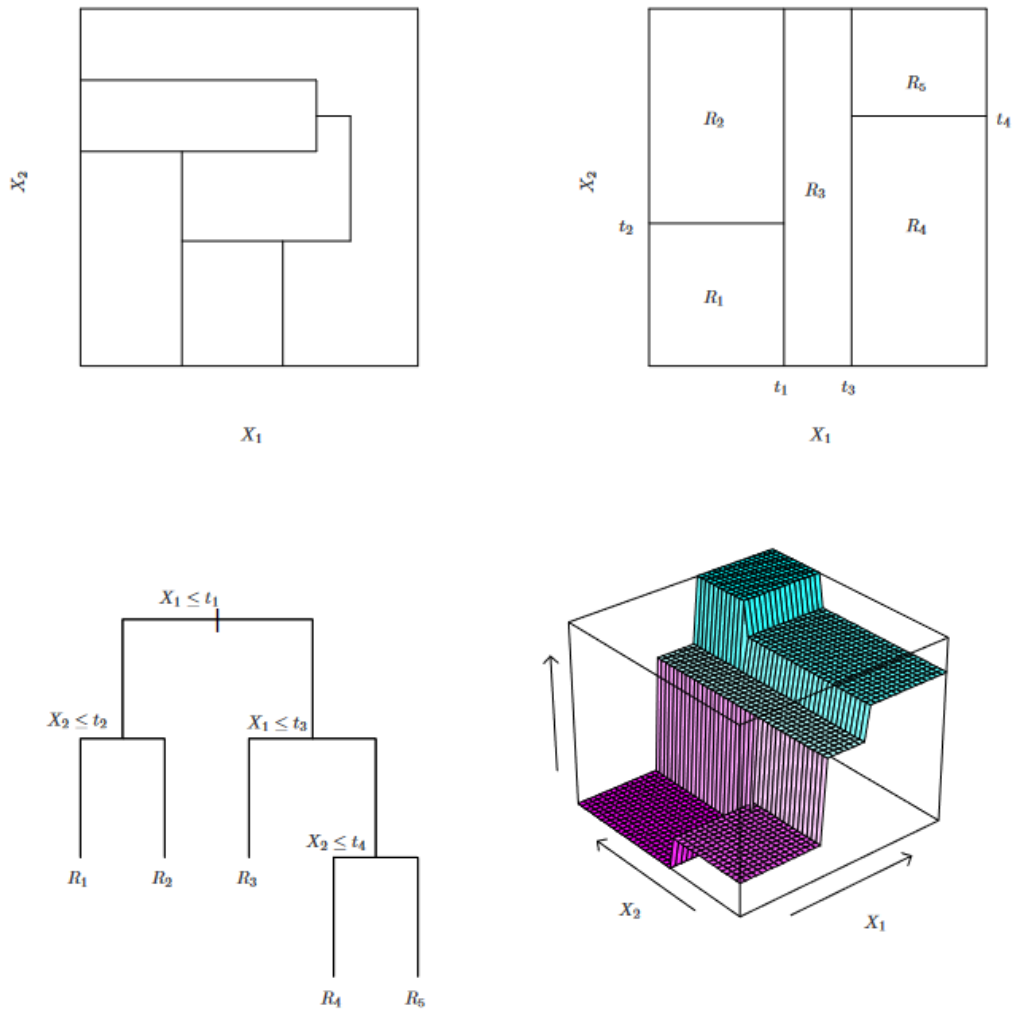


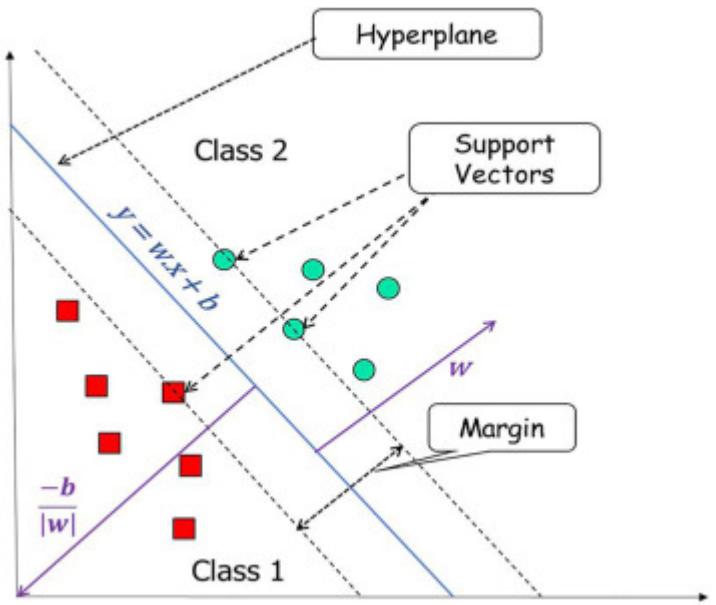
FIGURE 8.3. Top Left: *A partition of two-dimensional feature space that could not result from recursive binary splitting.* Top Right: *The output of recursive binary splitting on a two-dimensional example.* Bottom Left: *A tree corresponding to the partition in the top right panel.* Bottom Right: *A perspective plot of the prediction surface corresponding to that tree.*

Image source: James et al. 2021

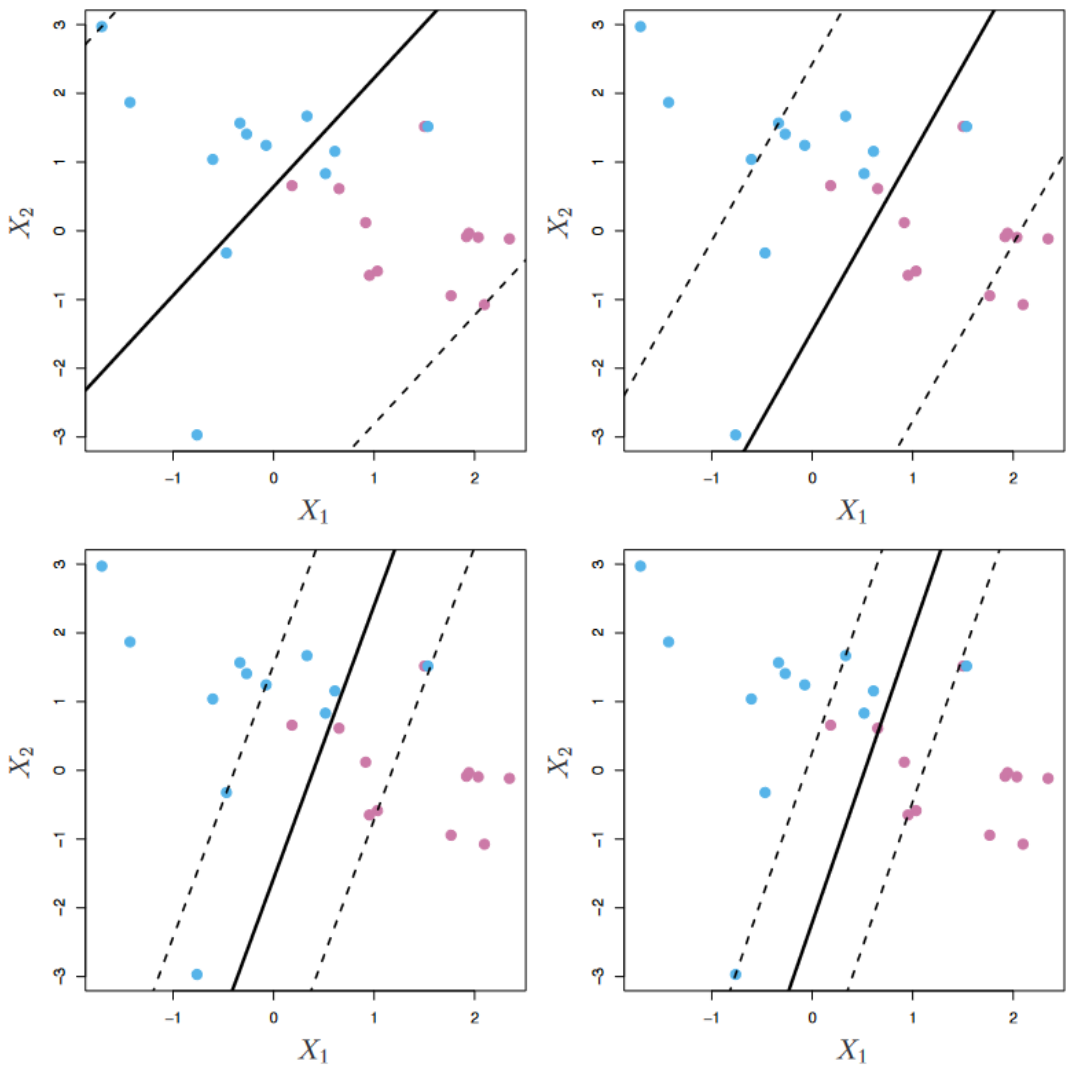
Support vector machines (SVM)

Classifier which fits a *hyperplane*. The hyperplane forms the categorical or decision boundary; observations in the boundary region are *support vectors* pushing against the hyperplane across a distance called the *margin*

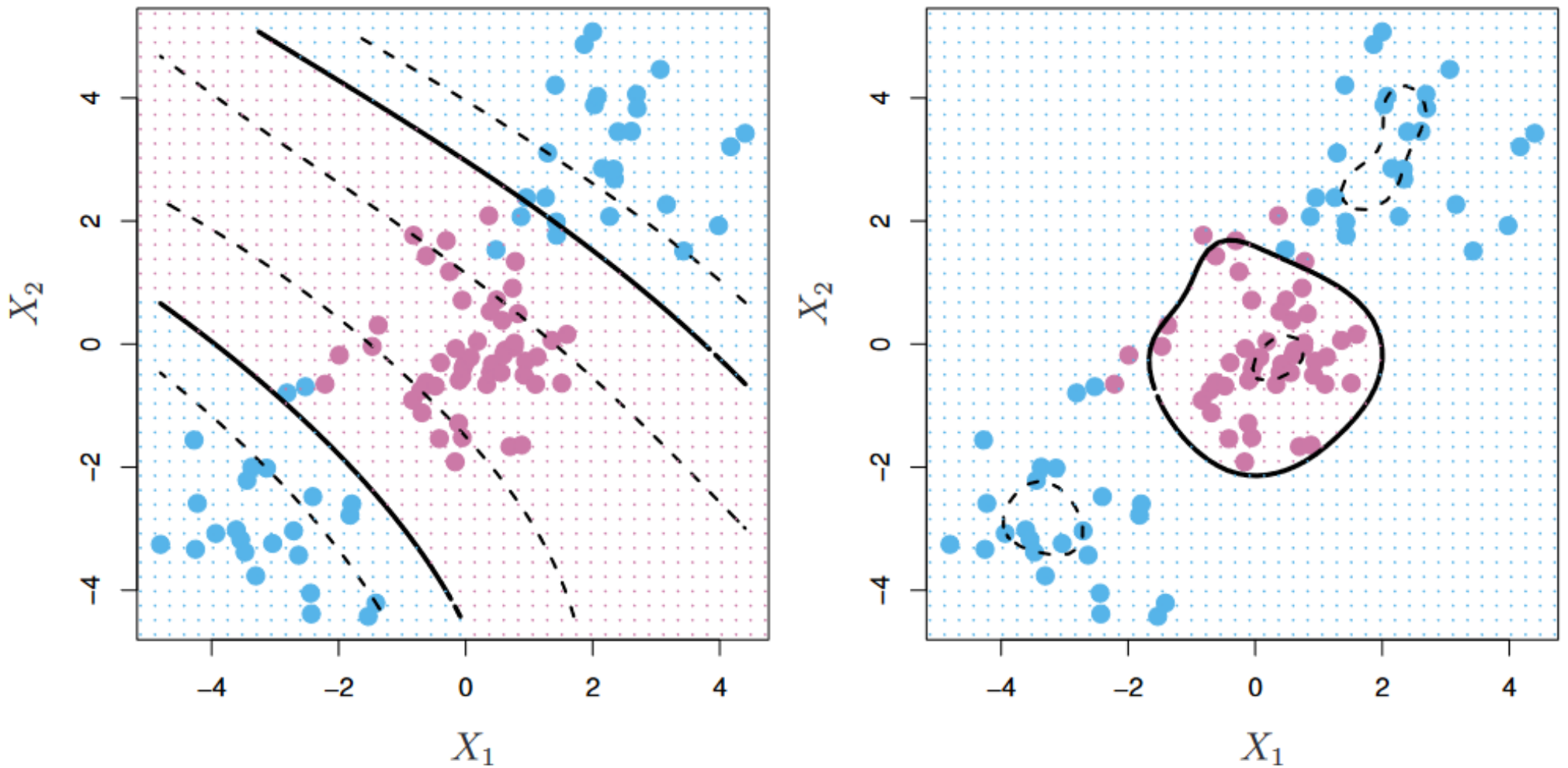
The *margin* between support vectors and hyperplane may need to be a *soft margin* because of overlapping class regions (i.e., observations are mixed)



Support vector classifier. Image source: Rani et al. 2022



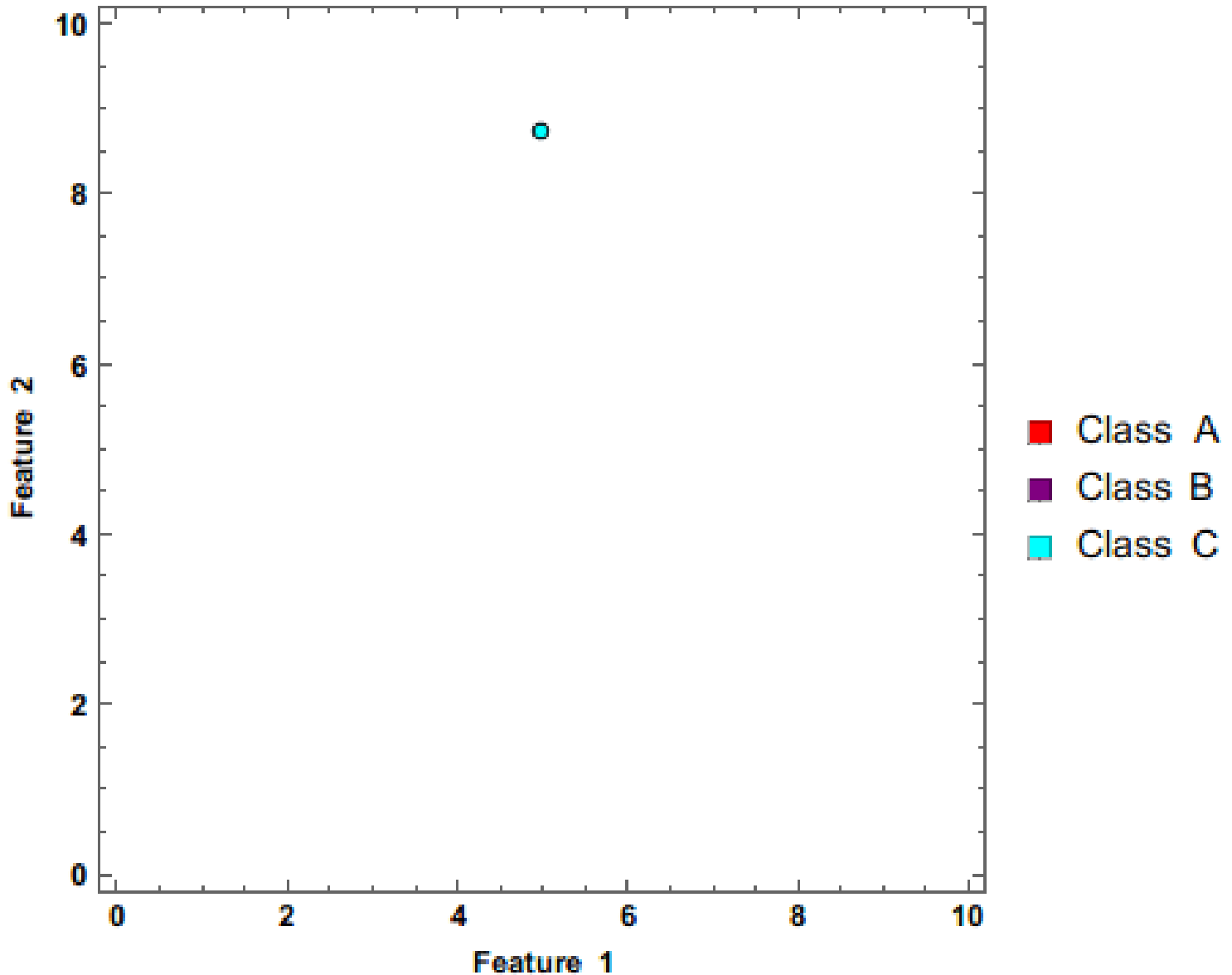
Support vector classifier with four different values for the tuning parameter C. As C gets smaller, the tolerance for observations on the “wrong” side decreases, and margins therefore decrease accordingly. Image source: James et al. 2021



Support vector machines extend the support vector classifier to use non-linear kernels. The data in this figure would fit a poor linear model using the linear classifier. Image source: James et al. 2021

Naive Bayes classifier

- We assume that each predictor variable has a normal (Gaussian) distribution
  - For each variable, the distribution is estimated based on observed data
  - To classify an observation, plot it and check its probability of membership for each class.



Animation of the naive Bayes classifier. Color intensity indicates probability of group membership. Image source: Jacopo Bertolotti via Wikimedia Commons (CC0)

Naive Bayes paper examples

- Predicting water quality (Ilic et al. 2022)

Naive Bayes code example

```r
penguins_clean <- penguins %>%
  filter(species %in% c("Gentoo", "Adelie"),
         !is.na(species),
         !is.na(body_mass_g)) %>%
  mutate(species = fct_drop(species))

species_fit <- naiveBayes(species ~ body_mass_g + flipper_length_mm, data =

species_fit
predict(species_fit, data.frame(body_mass_g=3500, flipper_length_mm=22))
```

Neural networks

- A neural network takes $X_1 \ldots X_p$ variables and builds a nonlinear $f(X)$ to predict the response Y (James et al. 2021)
- Derived features are computed and utilized by *hidden layers*, between the input and output; nodes have different rules for *activating*
- When there is more than one hidden layer, it is called *deep learning*
- **This is the black box method**: very sophisticated, but little or no explainability for an individual given result

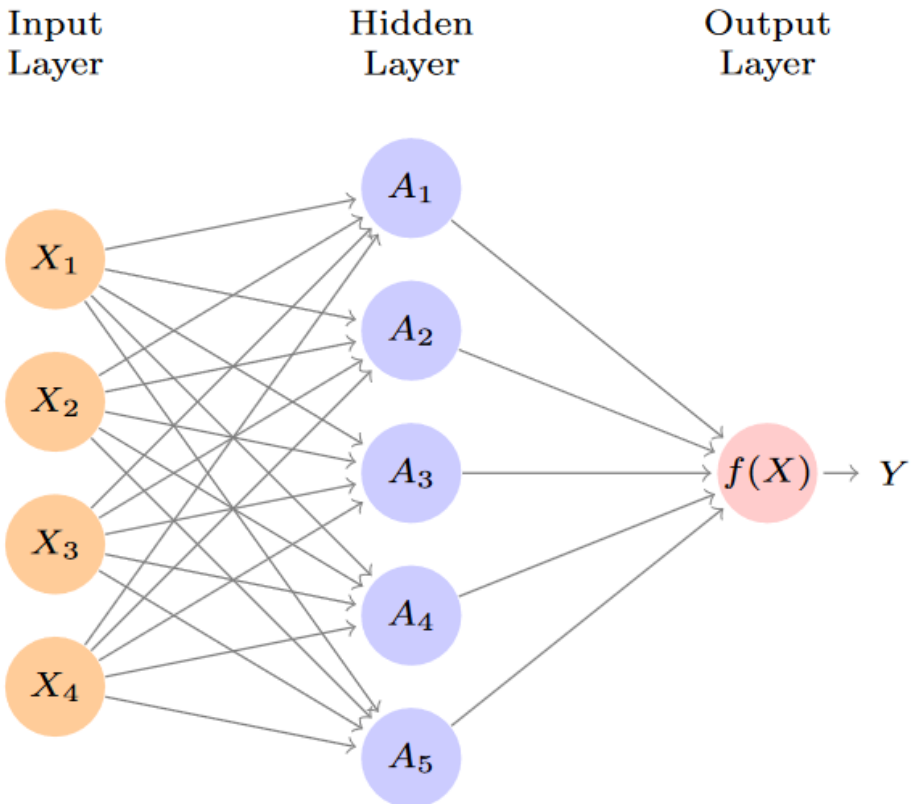


FIGURE 10.1. *Neural network with a single hidden layer. The hidden layer computes activations $A_k = h_k(X)$ that are nonlinear transformations of linear combinations of the inputs $X_1, X_2, \ldots, X_p$. Hence these A_k are not directly observed. The functions $h_k(\cdot)$ are not fixed in advance, but are learned during the training of the network. The output layer is a linear model that uses these activations A_k as inputs, resulting in a function $f(X)$.*

Image source: James et al. 2021

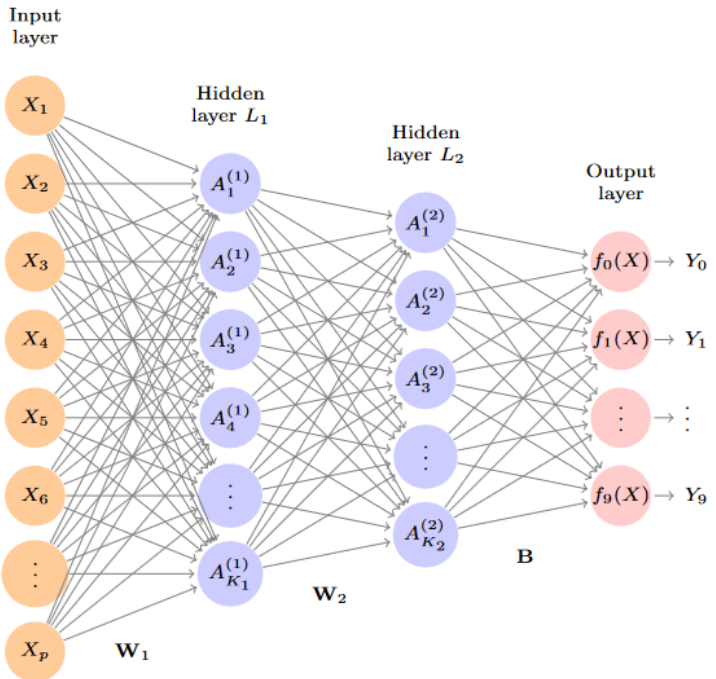


FIGURE 10.4. *Neural network diagram with two hidden layers and multiple outputs, suitable for the* MNIST *handwritten-digit problem. The input layer has* $p = 784$ *units, the two hidden layers* $K_1 = 256$ *and* $K_2 = 128$ *units respectively, and the output layer* 10 *units. Along with intercepts (referred to as* biases *in the deep-learning community) this network has 235,146 parameters (referred to as weights).*

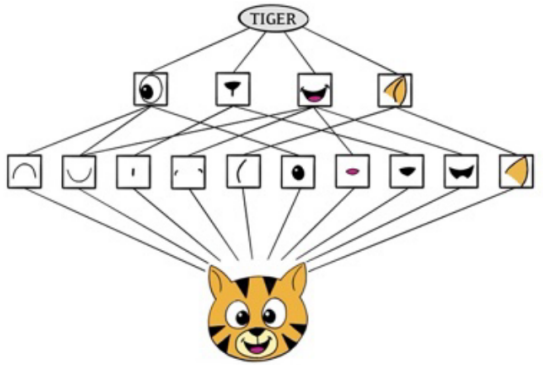
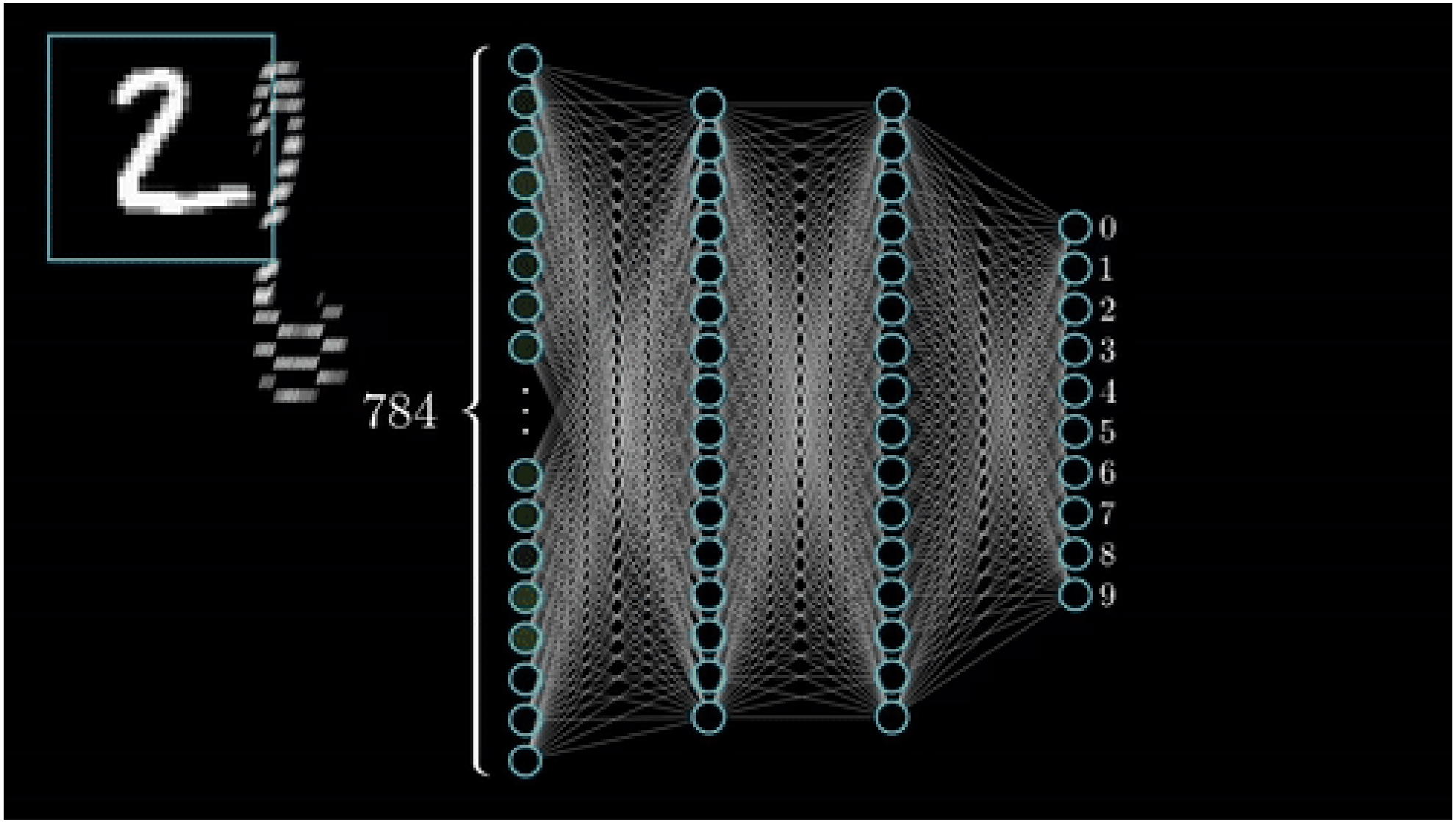


FIGURE 10.6. *Schematic showing how a convolutional neural network classifies an image of a tiger. The network takes in the image and identifies local features. It then combines the local features in order to create compound features, which in this example include eyes and ears. These compound features are used to output the label "tiger".*

Images source: James et al. 2021



Animation demonstrating a neural network for handwriting identification. Note that only certain nodes activate. The class (i.e., Arabic numeral) predicted from the input is “2”. Image source: Suraj Yadav

Model assessment

- There are many tools and possible implementations of supervised methods
- “More data” in the model isn’t automatically better (e.g., can create bias with highly correlated variables)
- Validation
  - Prior to analysis, data are split into *training* and *test* sets
  - The model is fit on the training data
  - Then the test data are submitted to the model, and the test Y is compared to the model’s $\hat{Y}$ prediction—a measure of accuracy (was the model right?)
- Cross-validation: a more sophisticated take
- *Feature selection* is also an important activity, informed by validation as well as techniques such as lasso regression

Unsupervised learning

Unsupervised learning

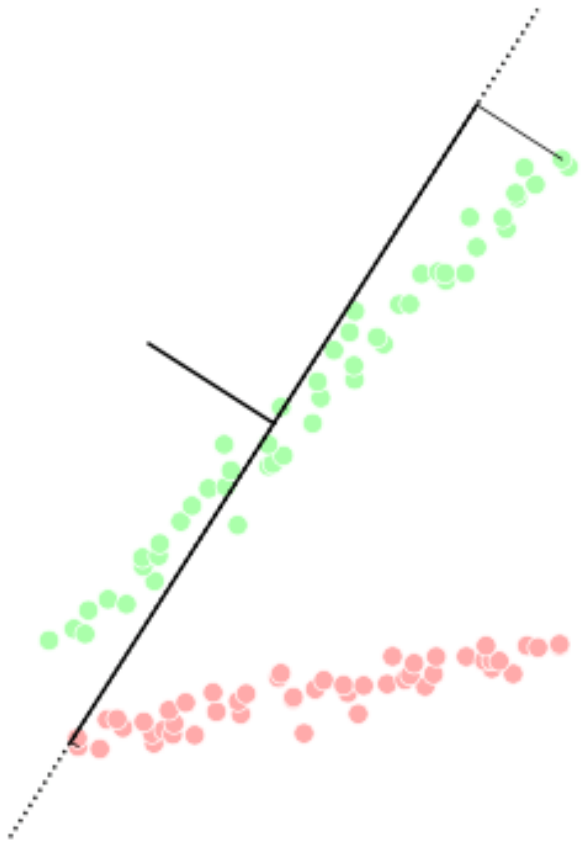
Unsupervised learning has no predictive model: instead it finds previously unknown structure in the data. All variables or features of the data are considered together.

Unsupervised learning tends to be most useful for exploratory data analysis, i.e., prior to having a goal for regression or classification.

- Principal Components Analysis (PCA): dimensionality reduction
- K-Means Clustering

Principal Component Analysis (PCA)

- $X_1, X_2, \ldots, X_p$ features are reduced to a small number of "principal components"
  - Z_1, the first principal component, accounts for most of the variation in the data
  - Z_2 accounts for most of the remaining variation
- Example of **dimensionality reduction** 👇
- Useful for:
  - understanding
  - deriving variables for supervised methods
  - visualizing 3+ variables in a 2D space
  - imputation (guessing empty values)
- Similar to Linear Discriminant Analysis (LDA), a supervised method which incorporates dimensionality reduction among predictors



Animation demonstrating projection of two features onto a single histogram using principal components analysis. Image source: Amélia O. F. da S. via Wikimedia Commons (CC BY-SA 4.0)

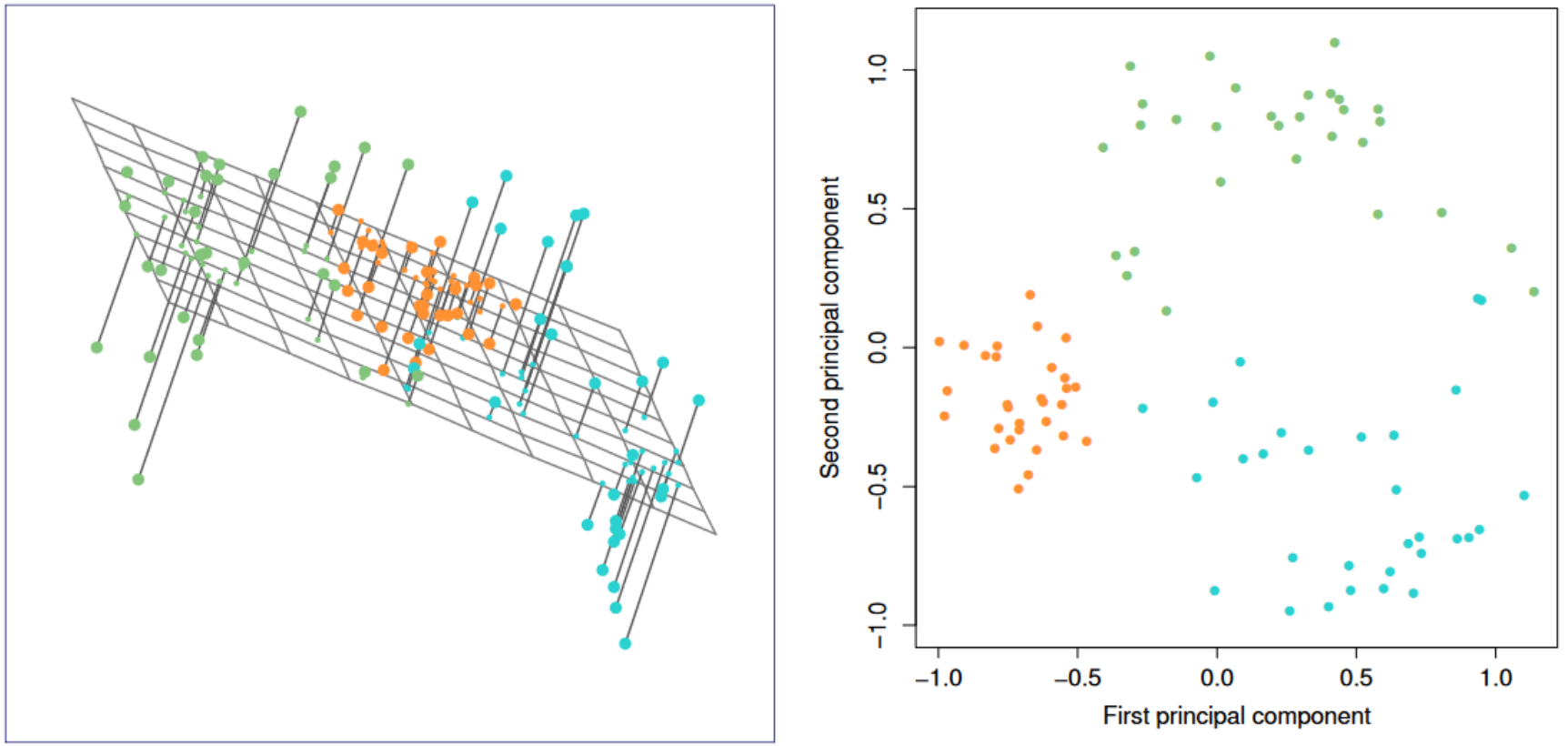


FIGURE 12.2. *Ninety observations simulated in three dimensions. The observations are displayed in color for ease of visualization.* Left: *the first two principal component directions span the plane that best fits the data. The plane is positioned to minimize the sum of squared distances to each point.* Right: *the first two principal component score vectors give the coordinates of the projection of the 90 observations onto the plane.*

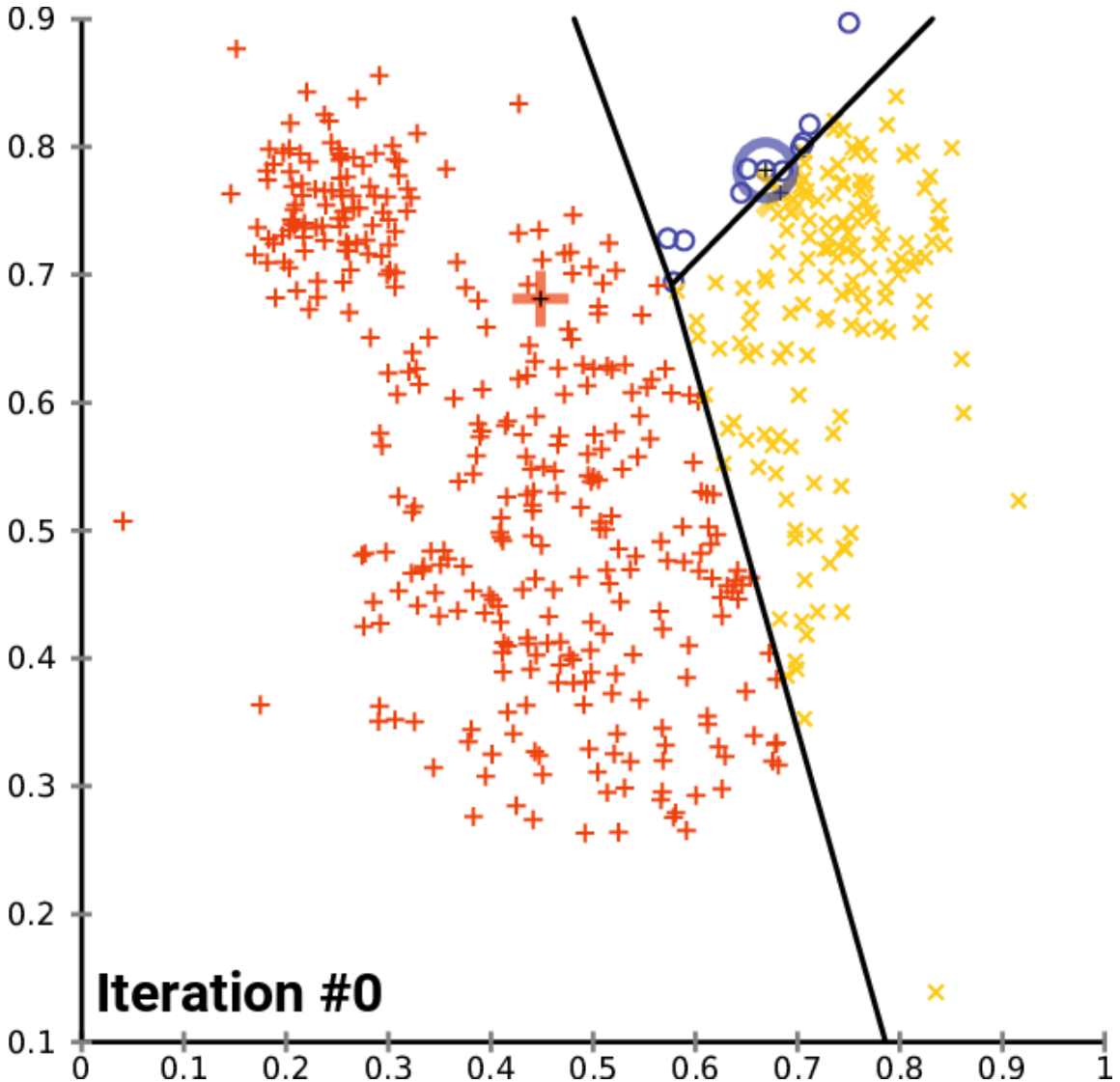
Image source: James et al. 2021

PCA examples

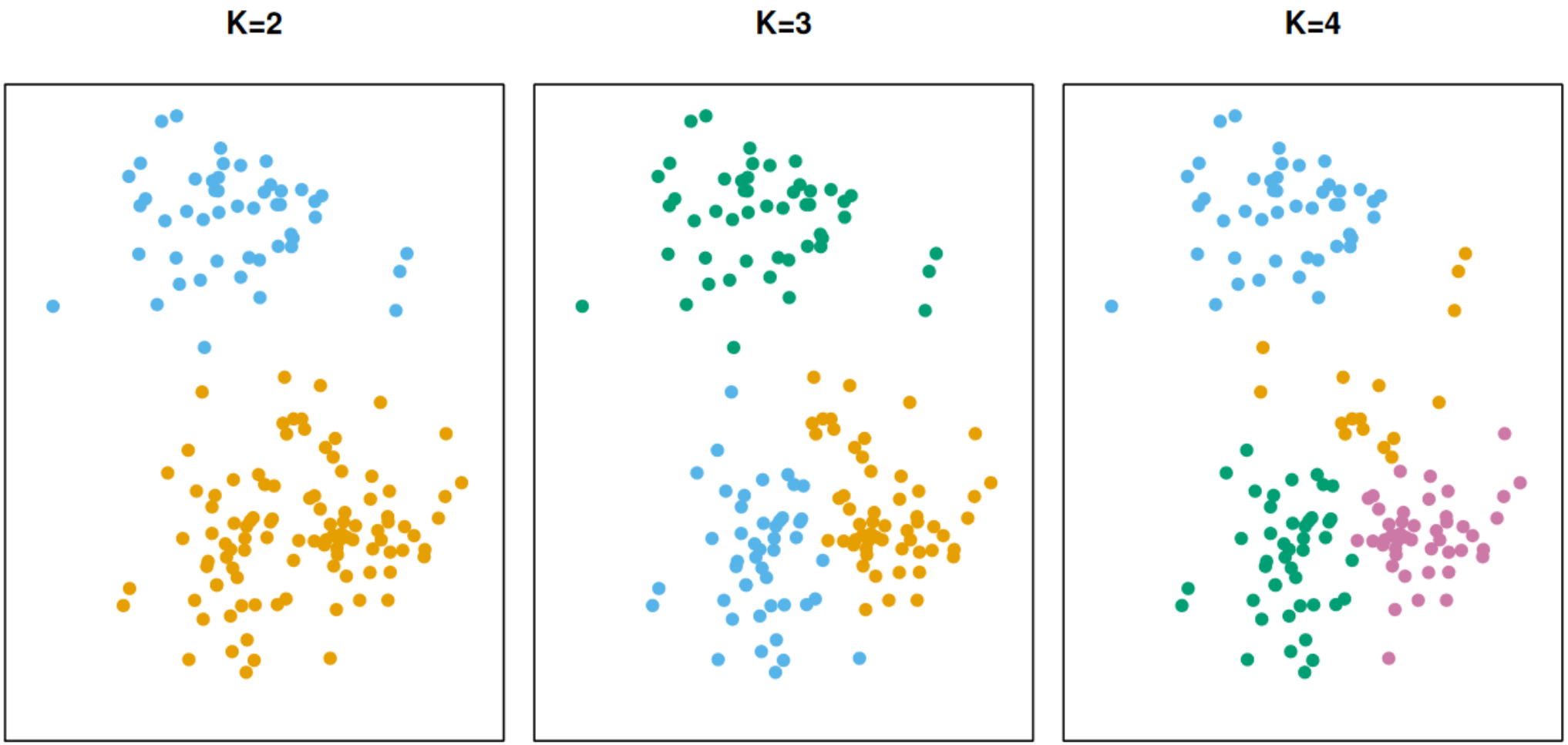
- Characterizing how people perceive themselves in a mirror versus looking at their own body (Jenkinson and Preston 2017)
- Diagnosis of diverse diseases from blood microRNA datasets (Sell et al. 2020)

K-Means Clustering

- A number K of previously unknown clusters are identified; or: we partition the data into K clusters
  - K is chosen by the analyst
  - Clusters aren't completely random, but based on similarities in the data
  - Assign every observation a number 1 through $K \rightarrow$
    calculate each cluster centroid $\rightarrow$
    (re)assign each observation to the closest centroid $\rightarrow$
    continue calculating and reassigning until movement stops
  - Total within-cluster variation is minimized
- *Not* a predictive classifier!
- Useful for:
  - Exploration
  - Identifying potential subpopulations



Animation of the K-means algorithm in action. After initial random group assignment, centroids are randomly placed and used to classify. Then centroids and assignment are iteratively adjusted until movement stops. Image source: Chire on Wikimedia Commons (CC BY-SA 4.0)



150 observations in 2D space, clustered according to different values of K. Prior to clustering, data are not categorized. Colors indicate which group each observation is assigned to by the model. Image source: James et al. 2021

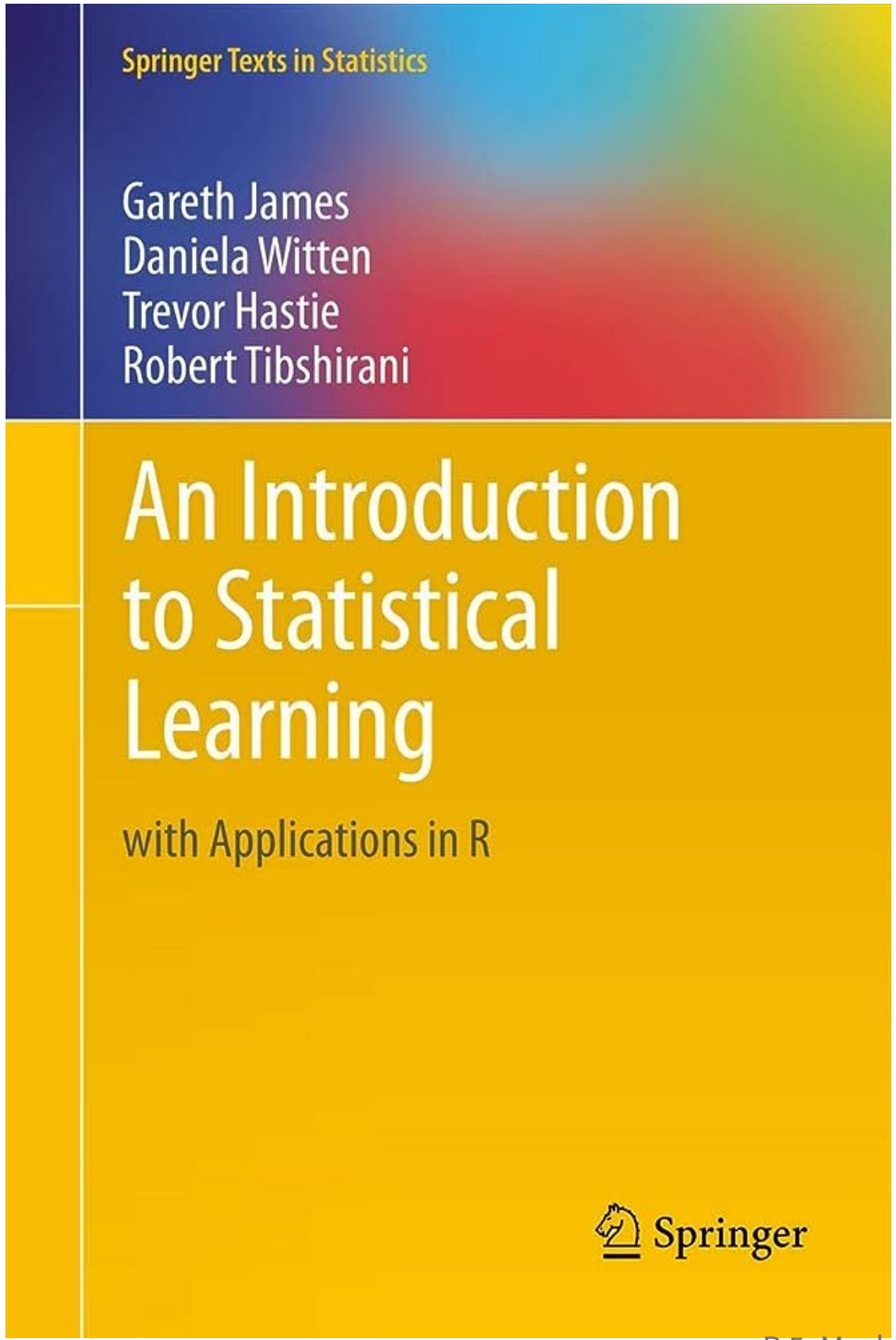
K-Means examples

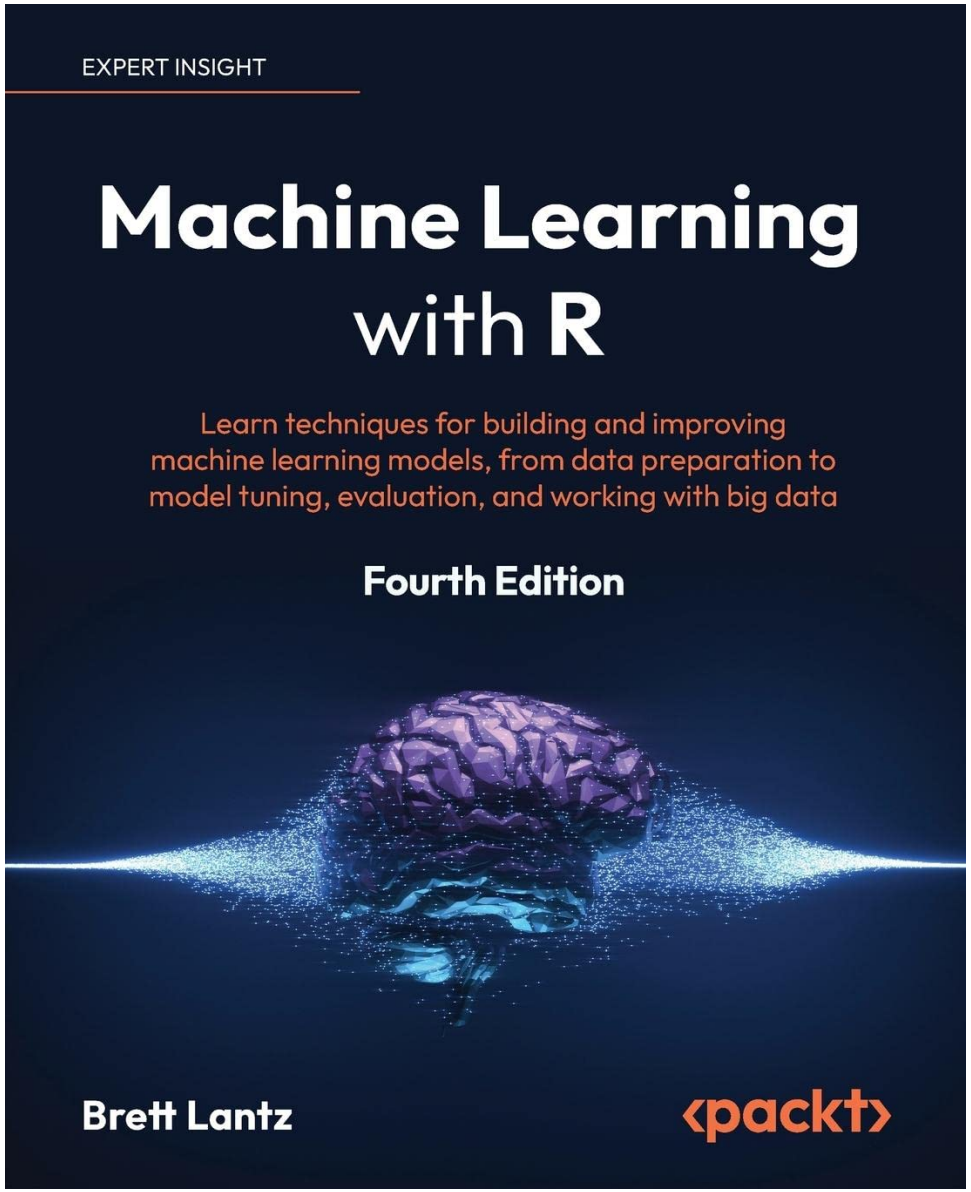
- Identifying patients at risk for paternal age-related schizophrenia (Lee et al. 2011)

Reinforcement learning

- A supervised model is connected to an assessment apparatus
- Model outputs are scored for correctness (e.g., by a human reviewer)
- The model is programmed to seek optimization of this score and may adjust its own machinery to get “better” results

Recommended reading





Lantz 2023

Available in PittCat via

or, looking for discipline-specific R?

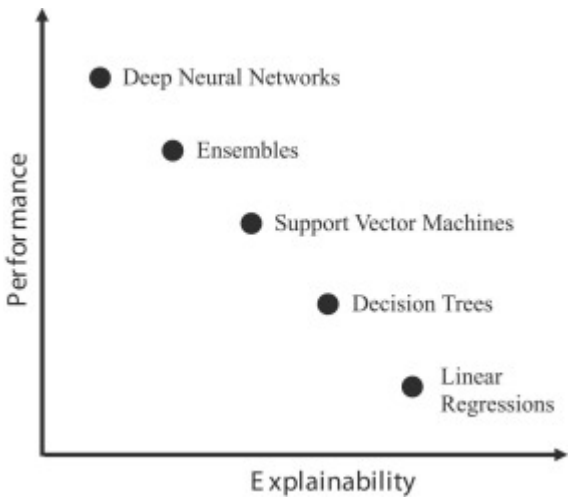
Check out the Big Book of R! An online directory at https://www.bigbookofr.com/ of very many R ebooks, most of them free OER and produced by experts, organized by discipline/topic and searchable.

Look up your discipline (or some topic that interests you, e.g., time series data) and see what applications of R you can find.

Example graphic of a recent update

Concluding thoughts

- Every approach makes certain assumptions and tradeoffs
  - Bias–variance tradeoff: two sources of error; having a model that is not overfit to the data nor too general
  - Consider accuracy vs. explainability ("performance vs. complexity")
- Experiment, but don't jump to conclusions. Consult your discipline's literature. Look for alternative ways to explore and validate unexpected findings.



A common conceptualization in the (applied) ML community. Image source: Herm et al. 2023