



A Conceptual Introduction to **Machine Learning**

Jeffrey M. Girard
University of Kansas
www.jmgirard.com



Overview

Roadmap

1. Define concepts and terminology
2. Describe typical ML workflow
3. Discuss ML usage in psychology



Introduction

- Machine learning is a branch of computer science that develops **algorithms that learn from data**
- Algorithms are tasked with finding **connections and patterns in data**
- It has similar goals to but different values and norms than statistics



Types of Modeling

Inference

- Draw **conclusions** about the data
 - Higher need for model **interpretability**
 - Emphasis on statistical **significance**
-
- *Is self-control associated with truancy?*
 - *Which dosages of a drug are safe?*
 - *Which personality traits predict the amount of positive emotion shown?*

Prediction

- Make **predictions** on new data
 - Higher need for model **flexibility**
 - Emphasis on prediction **accuracy**
-
- *How likely is a child to become truant?*
 - *What dosage is a patient likely to tolerate?*
 - *How much positive emotion is a person expressing in an image, video, or tweet?*



A Tale of Two Traditions

Classical Statistics

- Tend to emphasize **inference**
- Tend to value model **interpretability**
- Tend to use **top-down** assumptions
- *Generalized linear modeling*
- *Linear mixed effects modeling*
- *Structural equation modeling*

Machine Learning

- Tends to emphasize **prediction**
- Tends to value model **flexibility**
- Tends to use **bottom-up** patterns
- *Support vector machines*
- *Decision trees and random forests*
- *Artificial neural networks*



Types of Variable

Labels / Outcomes

- Variables that we **want to predict** and *won't be available* in novel data (e.g., hard to collect, in the future)



Features / Predictors

- Variables that **help predict the labels** and *will be available* in novel data (e.g., easy to collect, in the past)





Types of Learning

Supervised Learning

- Algorithm given **features and labels** and tries to "map" between them
- Can we **predict the labels** from the values that the features take on?



Unsupervised Learning

- Algorithm is provided **features only** and **looks for patterns** within them
- Can we find subgroups/clusters or latent dimensions/embeddings?

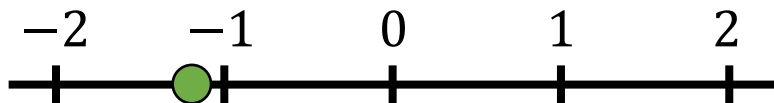




Modes of Supervised Learning

Regression

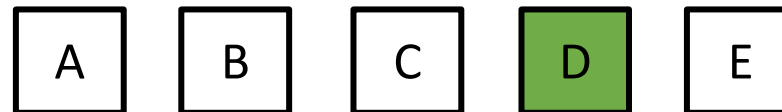
- Predict continuous, numerical values



- *How much will a customer spend?*
- *What GPA will a student achieve?*
- *How long will a patient be hospitalized?*

Classification

- Predict discrete, categorical values

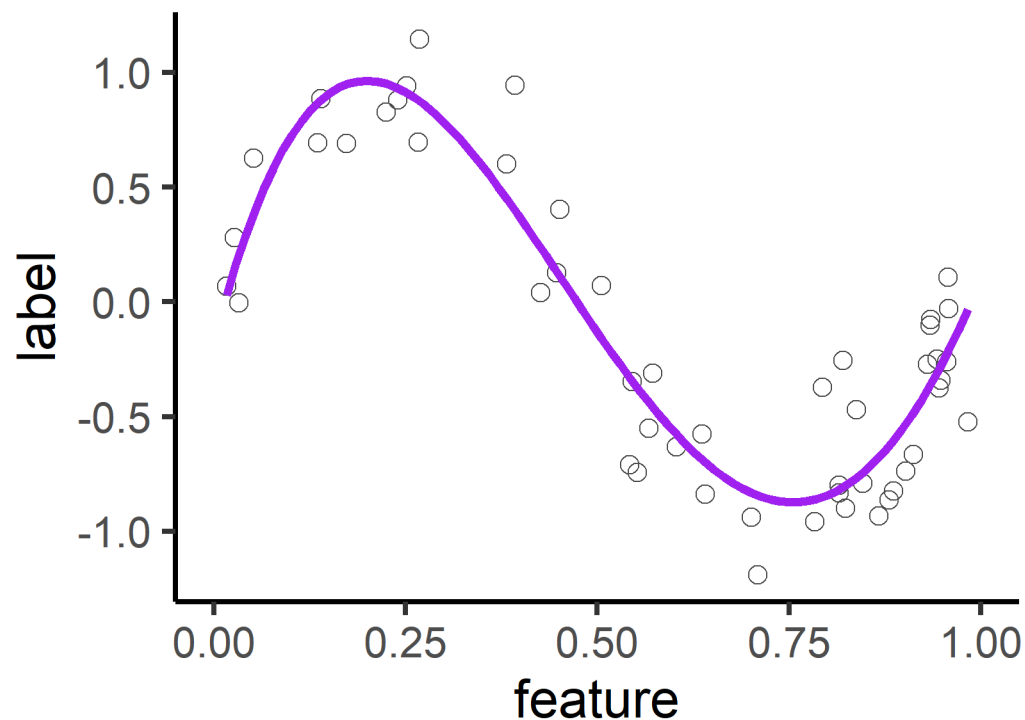


- *Is this email spam or non-spam?*
- *Which candidate will a user vote for?*
- *Is patient's glucose low, normal, or high?*

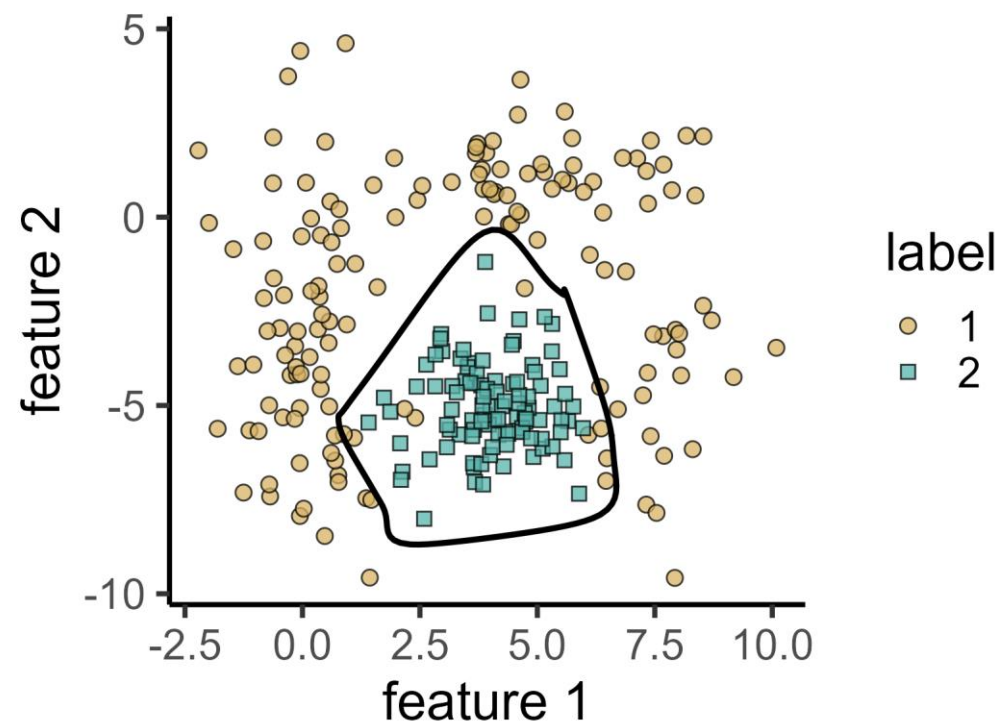


Modes of Supervised Learning

Regression

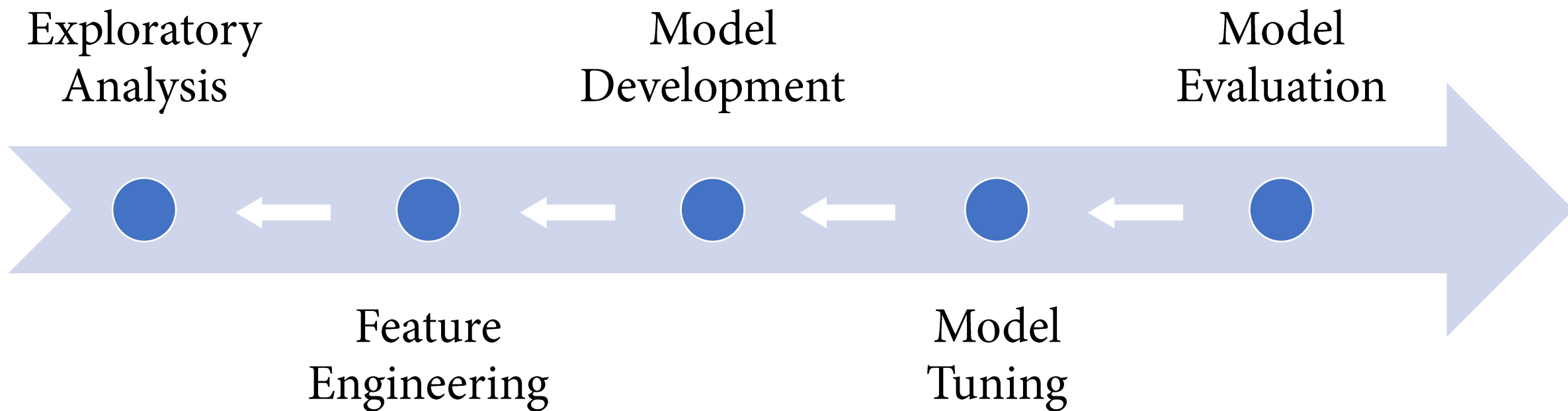


Classification





Typical Modeling Workflow





Exploratory Analysis

Quality Control

- Examine the distributions of variables
- Look for errors, outliers, missing data, etc.

Modeling Inspiration

- Identify relevant features for a label
- Detect highly correlated features
- Determine the "shape" of relationships





Feature Engineering

- Extract features (*e.g., from text, images, audio*)
- Transform features (*e.g., center, normalize, log*)
- Re-encode features (*e.g., dummy code, one hot*)
- Combine features (*e.g., ratios, means, interactions*)
- Reduce dimensionality (*e.g., PCA, EFA, GDA*)
- Address missing values (*e.g., deletion, imputation*)
- Drop features (*e.g., redundant, low variance*)
- Select features (*e.g., wrapper-based, filter-based*)





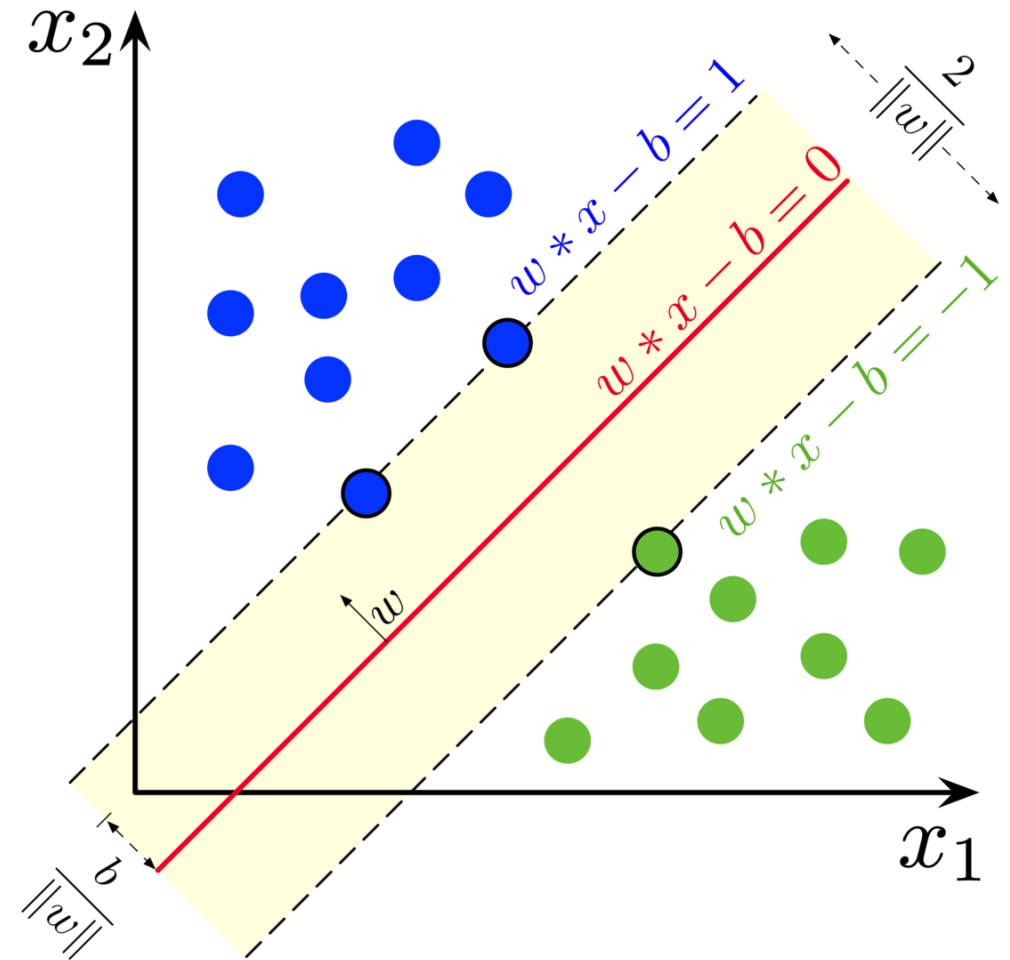
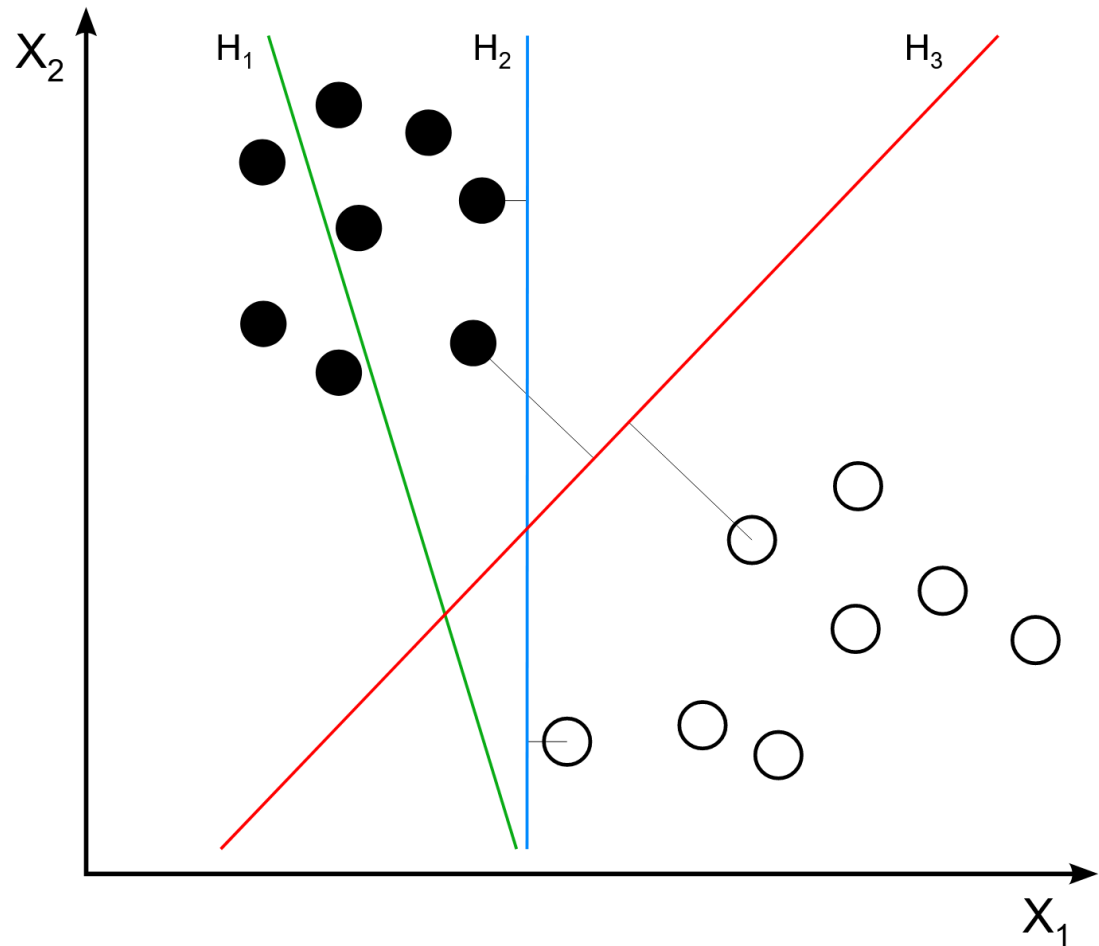
Model Development

- What *type* of model to use?
 - Elastic Net, Random Forest, SVM, MLP?
- What *engine* to use for fitting the model?
 - Which software implementation?
- What *mode* should the model run in?
 - Regression, classification, or ordinal?
- What *formula* should the model fit?
 - Which features and how to combine them?



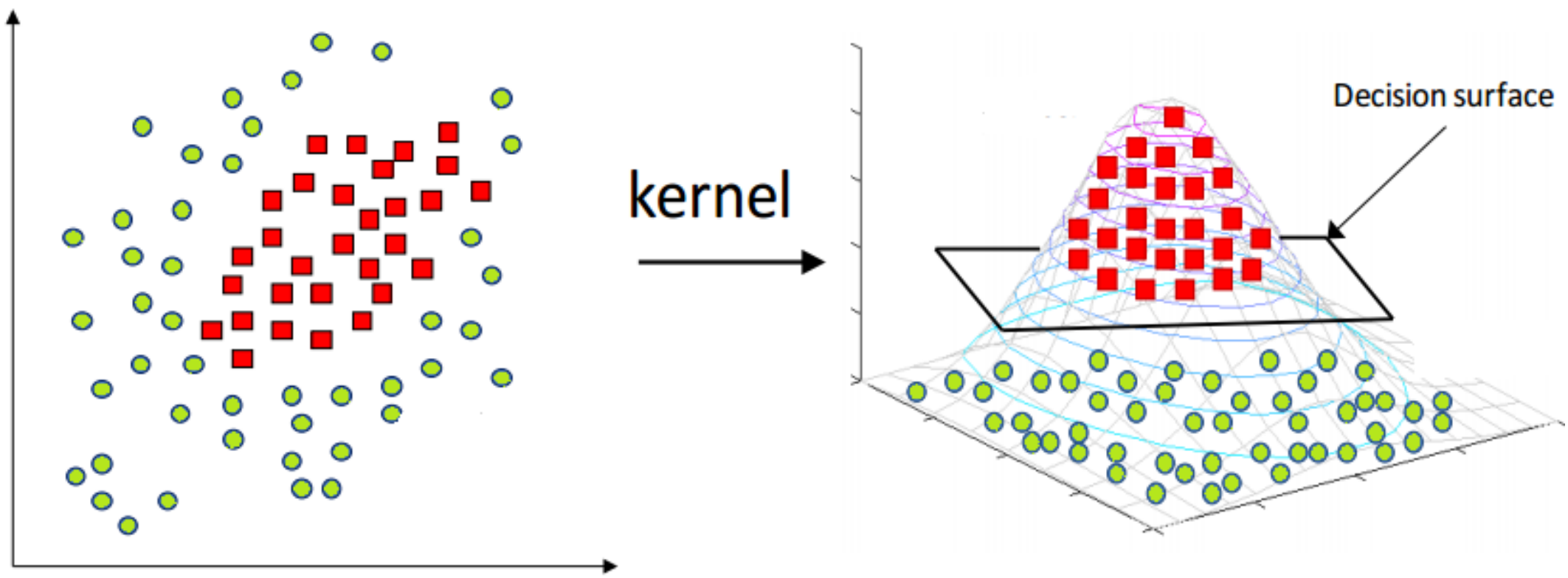


Support Vector Machines





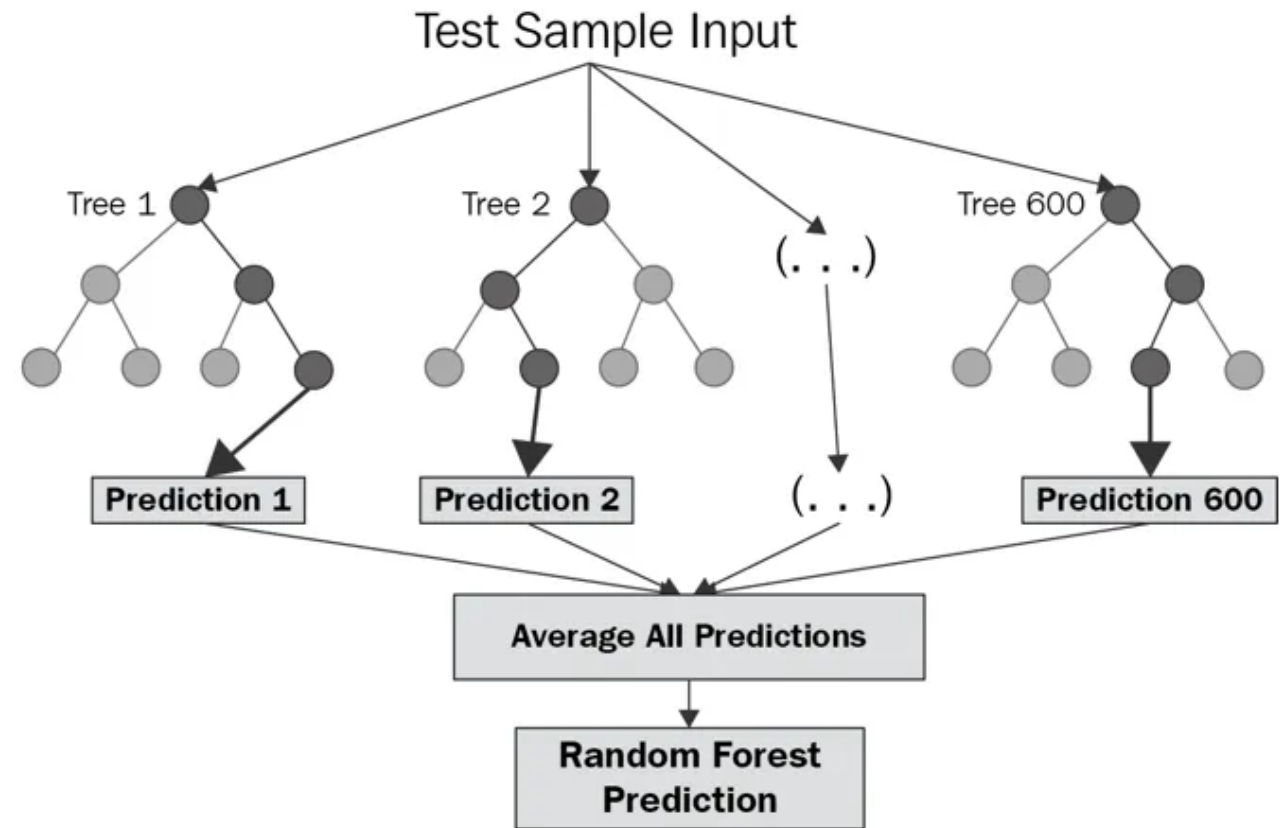
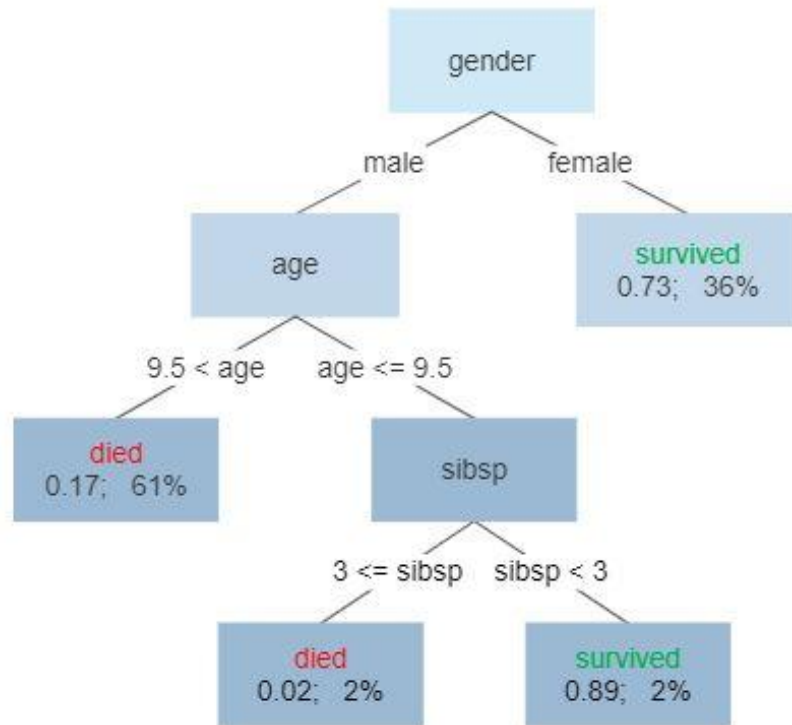
The Kernel Trick





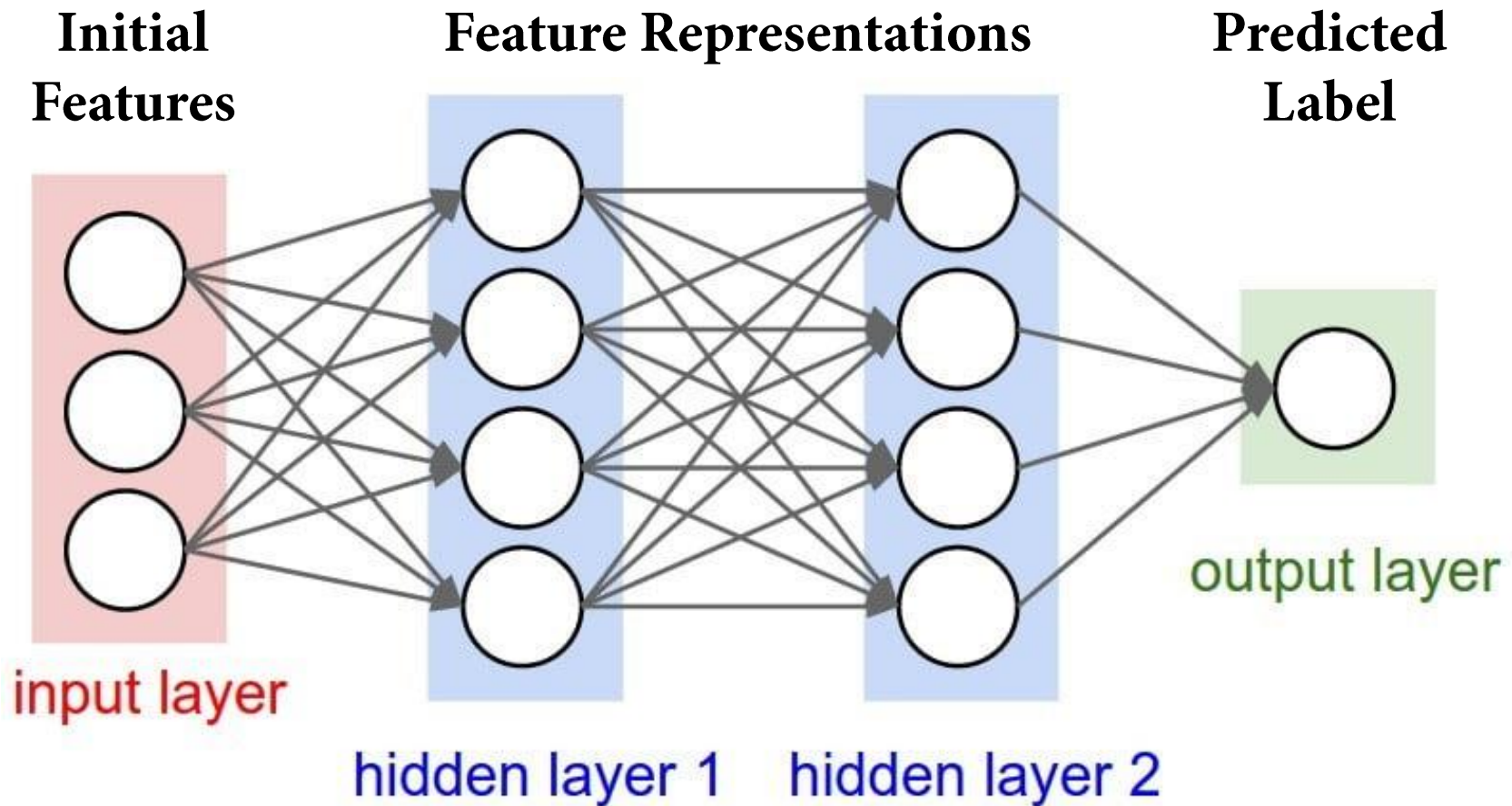
Decision Trees and Random Forests

Survival of passengers on the Titanic





Artificial Neural Networks





Model Tuning

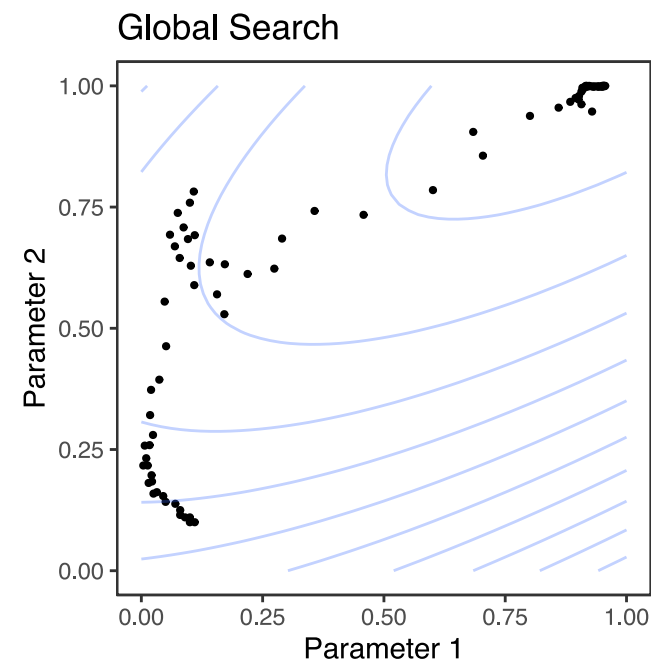
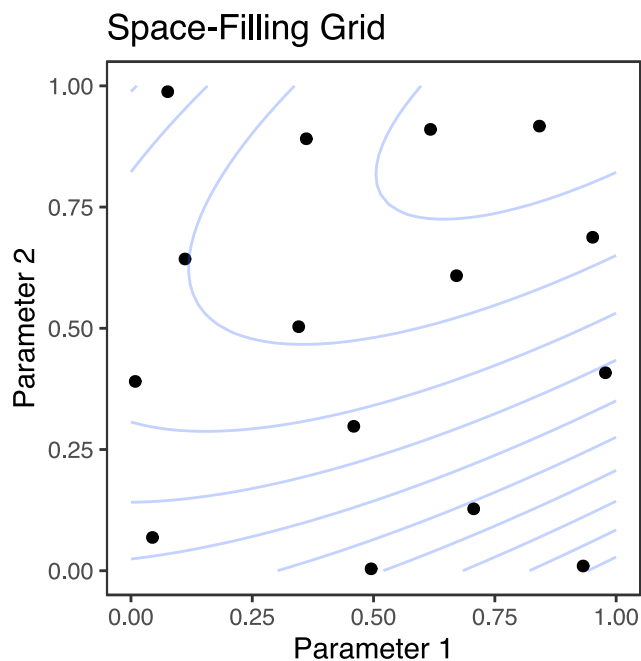
- Models learn by estimating **parameters** from data
 - *where and how to define the margin in an SVM*
 - *which leaves and branches to use in a decision tree*
 - *what weights to use in connecting neurons in an ANN*
- Learning is also influenced by **hyperparameters**
 - *which type of kernel to use in a non-linear SVM*
 - *how many decision trees to include in a random forest*
 - *how many hidden layers to include in an ANN/MLP*
- Hyperparameters often control model flexibility





Model Tuning

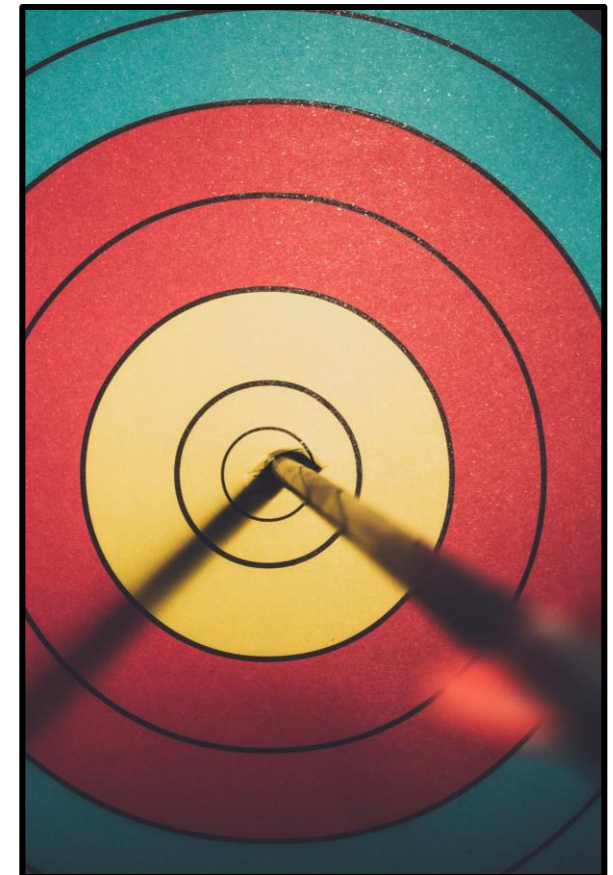
- Unlike parameters, hyperparameters cannot be estimated from the data
- Instead, we must "tune" our hyperparameters by trying / comparing them
- **Grid Search** – Try all values in a pre-defined set (e.g., spaced evenly through likely range)
- **Iterative Search** – Sequentially discover new combinations based on previous results





Model Evaluation

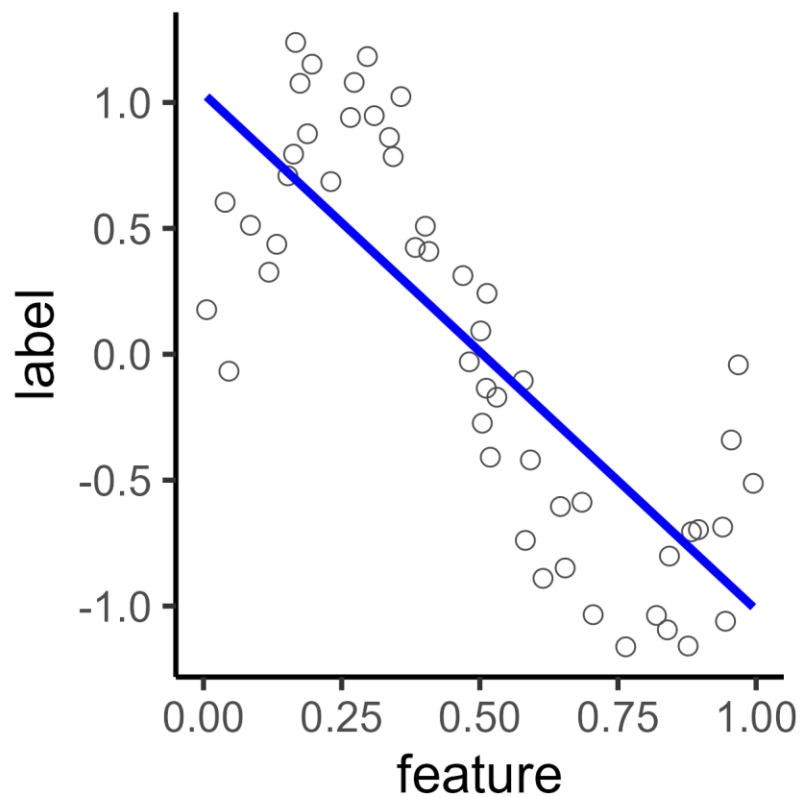
- How to quantify model performance?
 - Compare Predictions to Labels in Test Set
- Regression Metrics
 - Error-based (RMSE, MAE, Huber loss)
 - Correlation-based (CCC, R^2)
- Classification Metrics
 - Class-based (Acc, Sens, Spec, ϕ , F , J)
 - Probability-based (AUC, log loss, cost)
 - Curve Analysis (ROC, P-R, Gain, Lift)
 - Multiclass (macro, micro, specialized)



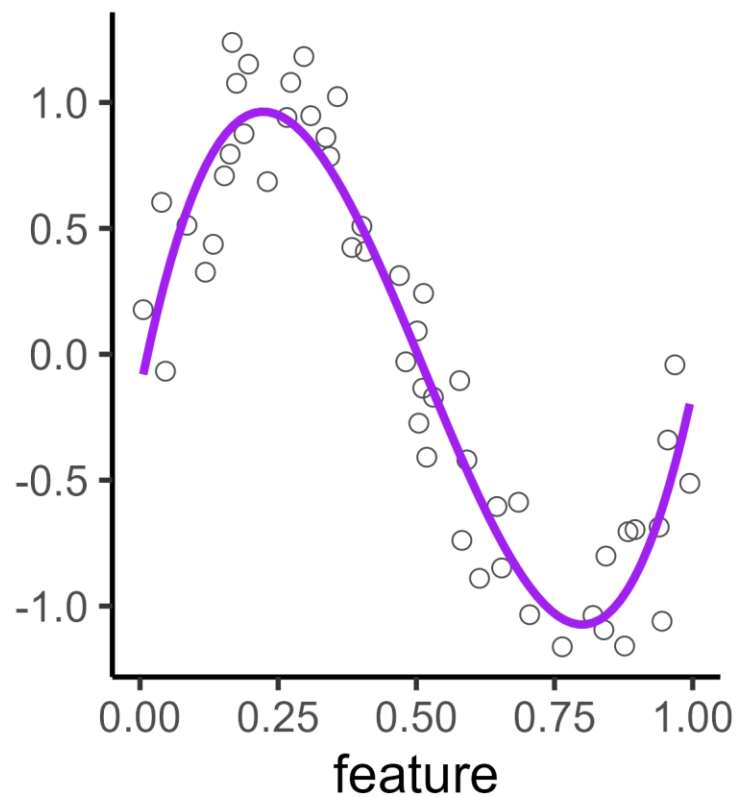


Modeling Flexibility

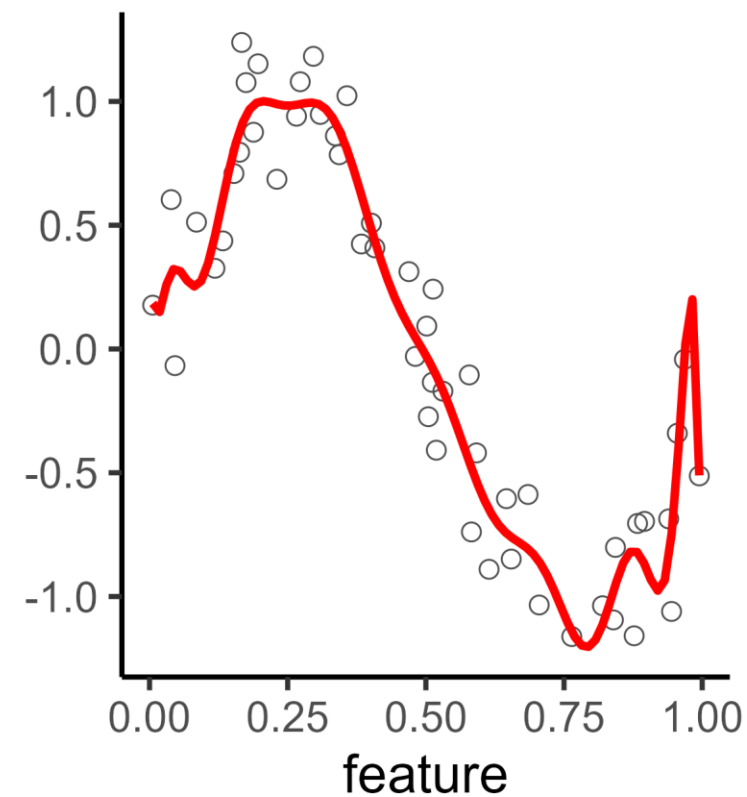
Underfit



Good Fit

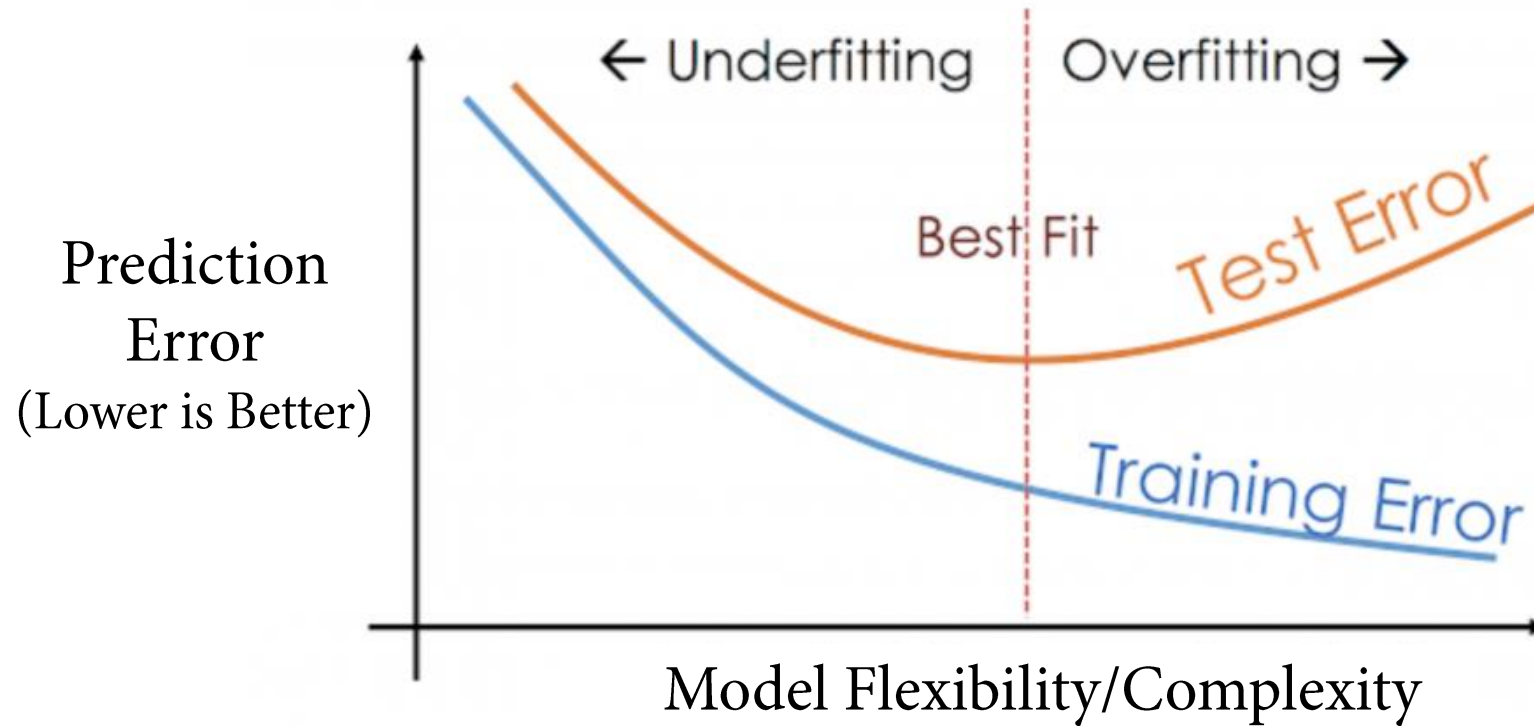


Overfit





A Technical Definition of Overfitting





A Lay Definition of Overfitting



**THE BEST WAY TO
EXPLAIN OVERFITTING**

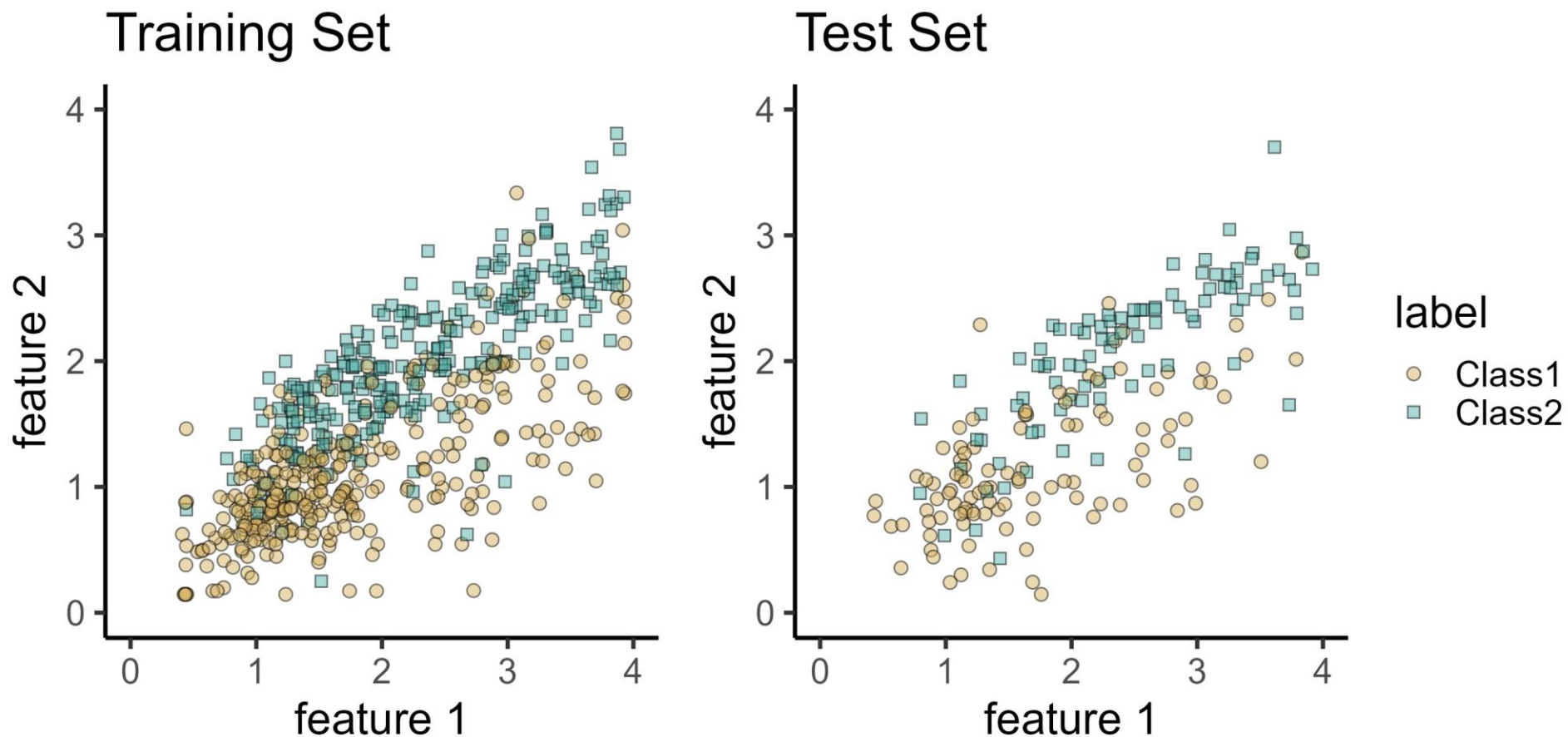


A GOOD EXAMPLE OF

OVERFITTING

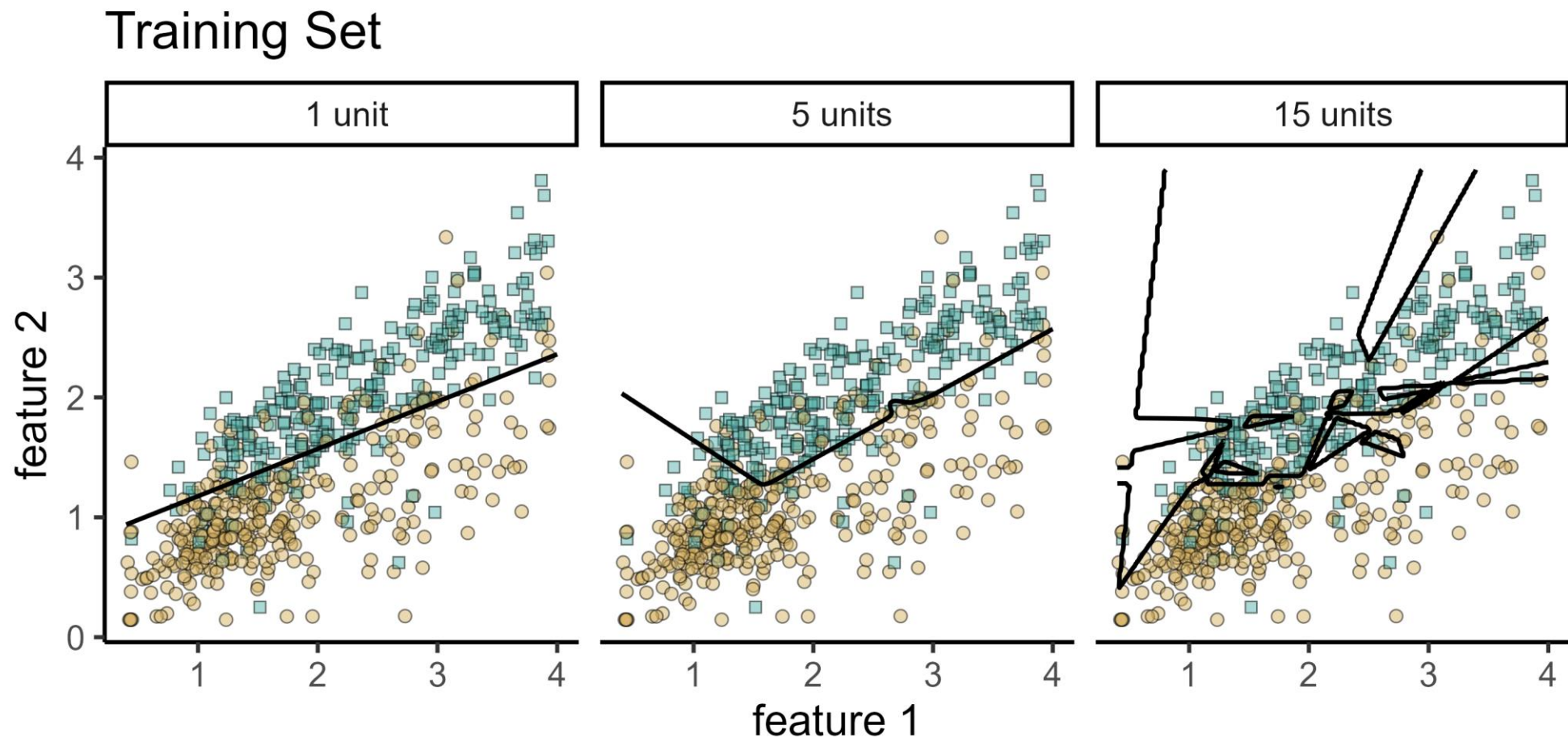


An Example of Overfitting



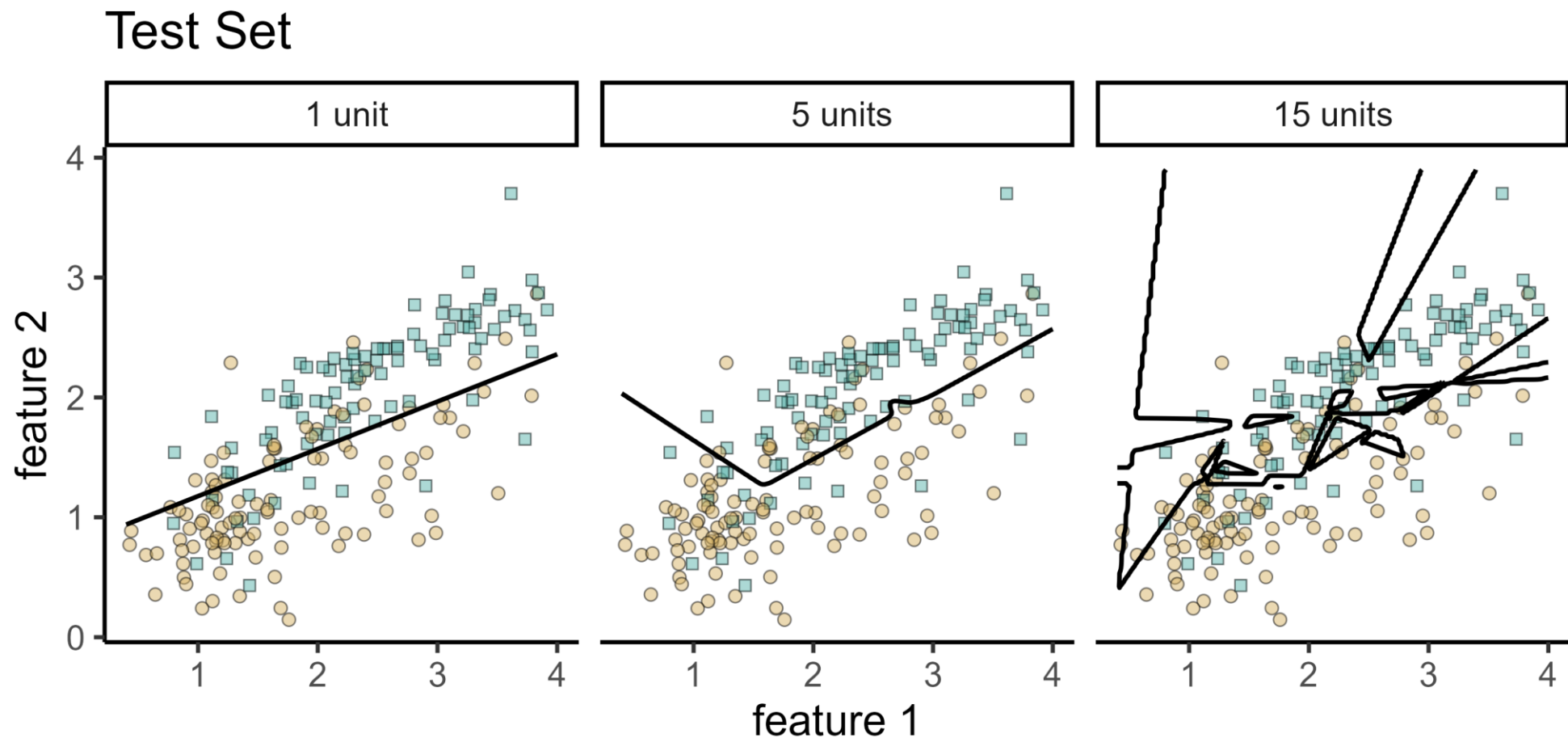


An Example of Overfitting





An Example of Overfitting





A "Solution" to Overfitting

Training Set

- Exploratory Analysis
- Feature Engineering
- Model Development
- Model Tuning

Test Set

- Model Evaluation

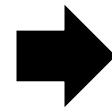
"Oh, East is East, and West is West, and never the twain shall meet"



A "Solution" to Overfitting

All Observations

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30



Training

Test

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30



Cross-Validation

All Observations

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30



Data Partitions / Folds

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30



Cross-Validation

	Fold 1 Iteration	Fold 2 Iteration	Fold 3 Iteration																																																												
Treat as Training Set	<table><tr><td>2</td><td>4</td><td>5</td><td>6</td></tr><tr><td>7</td><td>8</td><td>9</td><td>10</td></tr><tr><td>11</td><td>13</td><td>16</td><td>18</td></tr><tr><td>20</td><td>22</td><td>23</td><td>25</td></tr><tr><td>26</td><td>27</td><td>28</td><td>29</td></tr></table>	2	4	5	6	7	8	9	10	11	13	16	18	20	22	23	25	26	27	28	29	<table><tr><td>1</td><td>3</td><td>5</td><td>6</td></tr><tr><td>8</td><td>9</td><td>12</td><td>13</td></tr><tr><td>14</td><td>15</td><td>16</td><td>17</td></tr><tr><td>19</td><td>20</td><td>21</td><td>24</td></tr><tr><td>26</td><td>28</td><td>29</td><td>30</td></tr></table>	1	3	5	6	8	9	12	13	14	15	16	17	19	20	21	24	26	28	29	30	<table><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>7</td><td>10</td><td>11</td><td>12</td></tr><tr><td>14</td><td>15</td><td>17</td><td>18</td></tr><tr><td>19</td><td>21</td><td>22</td><td>23</td></tr><tr><td>24</td><td>25</td><td>27</td><td>30</td></tr></table>	1	2	3	4	7	10	11	12	14	15	17	18	19	21	22	23	24	25	27	30
2	4	5	6																																																												
7	8	9	10																																																												
11	13	16	18																																																												
20	22	23	25																																																												
26	27	28	29																																																												
1	3	5	6																																																												
8	9	12	13																																																												
14	15	16	17																																																												
19	20	21	24																																																												
26	28	29	30																																																												
1	2	3	4																																																												
7	10	11	12																																																												
14	15	17	18																																																												
19	21	22	23																																																												
24	25	27	30																																																												
Treat as Test Set	<table><tr><td>1</td><td>3</td></tr><tr><td>12</td><td>14</td></tr><tr><td>15</td><td>17</td></tr><tr><td>19</td><td>21</td></tr><tr><td>24</td><td>30</td></tr></table>	1	3	12	14	15	17	19	21	24	30	<table><tr><td>2</td><td>4</td></tr><tr><td>7</td><td>10</td></tr><tr><td>11</td><td>18</td></tr><tr><td>22</td><td>23</td></tr><tr><td>25</td><td>27</td></tr></table>	2	4	7	10	11	18	22	23	25	27	<table><tr><td>5</td><td>6</td></tr><tr><td>8</td><td>9</td></tr><tr><td>13</td><td>16</td></tr><tr><td>20</td><td>26</td></tr><tr><td>28</td><td>29</td></tr></table>	5	6	8	9	13	16	20	26	28	29																														
1	3																																																														
12	14																																																														
15	17																																																														
19	21																																																														
24	30																																																														
2	4																																																														
7	10																																																														
11	18																																																														
22	23																																																														
25	27																																																														
5	6																																																														
8	9																																																														
13	16																																																														
20	26																																																														
28	29																																																														



Pitfalls to Avoid / Practical Advice

- **Information Leakage** Don't use *any* info about test set during training
For clustered data, create data partitions by cluster
- **Biased/Flawed Data** Evaluate your data for systematic bias and noise
Test sets should represent the applied population
- **Insufficient Data** Modeling complexity often requires lots of data
Machine learning isn't appropriate for some samples
- **Ignored Uncertainty** Provide prediction intervals in applied settings
Compare models using inferential statistics
- **Magical Thinking** Don't expect machine learning to "fix" research mistakes
Attend to research design, sampling, measurement, etc.



Where to Learn More

Free Online Textbook

- Tidy Modeling with R
Kuhn & Silge (2021)
www.tmwr.org

Online Summer Camp

- Applied Machine Learning in R
Girard & Wang (July 19-23, 2021)
www.pittmethods.com/applied-ml

The screenshot shows a web browser displaying the 'Applied Machine Learning in R' website. The browser's address bar shows the URL 'https://www.pittmethods.com/applied-ml'. The website has a dark header with the text 'PITT METHODS' and navigation links for 'HOME', 'COURSES', 'INSTRUCTORS', 'LOCATION', and a 'REGISTRATION' button. Below the header is a large image of a city skyline at night. The main content area is divided into two columns. The left column contains the title 'Applied Machine Learning in R', a subtitle 'A hands-on introduction for social and behavioral scientists', the names of the instructors 'JEFFREY GIRARD, PH.D., UNIVERSITY OF KANSAS' and 'SHIRLEY WANG, A.M., HARVARD UNIVERSITY', and sections for 'Dates' (July 19th-23rd, 2021), 'Format' (Online workshop, 5x half-days), and 'Pricing' (Trainees: \$700, Professionals: \$1000). The right column contains a 'Description' section explaining the workshop's focus on machine learning methods and their application in social and behavioral sciences.

Applied Machine Learning in R

A hands-on introduction for social and behavioral scientists

JEFFREY GIRARD, PH.D., UNIVERSITY OF KANSAS
SHIRLEY WANG, A.M., HARVARD UNIVERSITY

Dates

July 19th-23rd, 2021 – Workshop days will involve online lectures, live coding, and exercises completed in small groups. Opportunities for methodological consulting with the instructors will also be provided. All days are anticipated to start at 1:00pm and end by 5:00pm (UTC-05:00) with breaks.

Format

Online workshop (5x half-days)

Pricing

Trainees (students and post-docs) – \$700
Professionals (including faculty) – \$1000

Description

Whereas statistical methods traditionally used in the social and behavioral sciences emphasize interpretability and quantification of uncertainty, machine learning methods emphasize complexity and accuracy of predictions. Machine learning methods are thus particularly well-suited for applications where (1) there are nonlinear and complex relationships among a large number of predictor variables and (2) accurately predicting the outcome variable is more important than fully understanding the relationships between variables.

This workshop will provide a hands-on introduction to the application of machine learning techniques in R using the {caret} package. It will emphasize practical knowledge and conceptual intuitions (e.g., teaching you how to drive a car) rather than technical and theoretical mastery (e.g., teaching you how to build a car). In addition, rather than briefly surveying the full breadth of available machine learning techniques, this workshop will provide a deep dive into two supervised learning methods with broad applicability in the social and behavioral sciences: regularized regression models (e.g., GLMNET) and random forest ensembles. The final day of the workshop will also provide an