

# Introduction to Classification

*Chris Connell*

*Data Scientist*

# Learning Objectives

After this lesson, you should be able to:

- Define class label and classification
- Build a K-Nearest Neighbors using the scikit-learn library
- Evaluate and tune model by using metrics such as classification accuracy/error

# Outline

- **Final Project 1 Presentations** 😊
- Review
- Types of machine learning problems
- What is classification?
- What is binary classification?
- Iris dataset and exploratory analysis
- Hand-coded classifiers
- Classification metrics
- K-Nearest Neighbors (KNN)
- High dimensionality
- What is the best value for k?
- Validation and cross-validation
- Advantages and disadvantages of KNN
- Lab
- Review
- Assigned
  - Final Project 2 (due in 3 weeks)
- In-flight
  - **Unit Project 3 (due next session on 3/24)**

DS

# Pre-Work

# Pre-Work

Before this lesson, you should already be able to:

- Understand how to optimize for error in a model
- Understand the concept of iterations to solve problems
- Measure basic probability

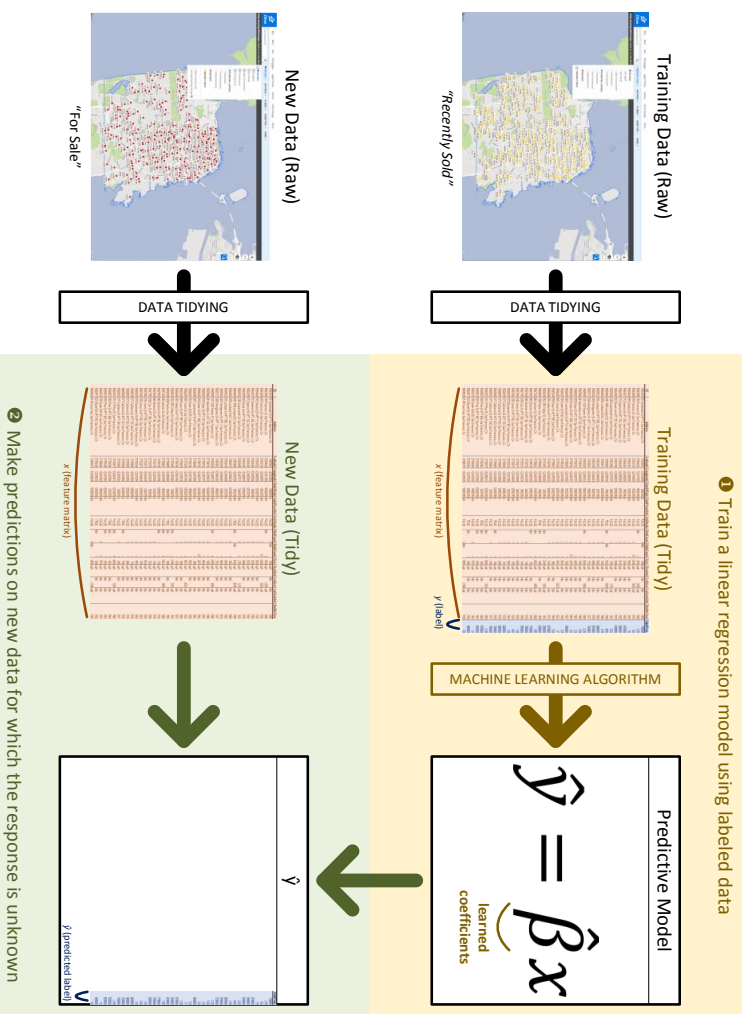
DS

# Review

# Linear Regression Review

- Linear regression assumes that the dependence of  $y$  (your response variable) on  $x$  (your input variables) is linear. Linear regressions are:
  - Highly interpretable and simple to explain
  - Model training and prediction are fast
  - No tuning is required (excluding regularization)
  - (Input) Features don't need scaling
  - Can perform well with a small number of observations
  - Well-understood

# Linear regression is a simple approach to supervised learning





# Linear regression is a simple approach to supervised learning (cont.)

- A supervised machine learning model learns the relationship between the feature variables and the response variable (also called the labeled data)
- The primary goal of supervised learning is to build a model that “generalizes” so as to accurately predicts the future (rather than the past)
- We’ve focused so far on predicting a continuous set of values
  - That means that we’ve been able to use distance to measure how accurate our predictions are
- However, for other problems, we need to predict binary responses. E.g., Will a loan default? Is an email spam or ham?



DS

# Types of Machine Learning Problems

# Types of Machine Learning Problems

	Continuous	Categorical
<b>Supervised</b> (a.k.a., predictive modeling)	<b>Linear Regression</b> (sessions 6 & 7)	<b>K-Nearest Neighbors</b> (session 8) <b>Logistic Regression</b> (session 9)
<b>Unsupervised</b>	A machine learning model that doesn't use labeled data is called unsupervised. It extract structure from the data. Goal is “representation”	

DS

# What is Classification?

# What is classification?

- Classification is a machine learning problem for solving a set of categorical values ( $y$ ; the response variable) given the knowledge we have about these values ( $x$ ; the feature matrix)
  - E.g., what if you are predicting whether an image is of a *human*, *dog*, or *cat*?
- The possible values of the response variable are called *class labels*
  - E.g., “*human*”, “*dog*”, and “*cat*”

# What is binary classification?

- Binary classification is the simplest form of classification
  - I.e., the response is a *boolean* value (true/false)
- Many classification problems are binary in nature
  - E.g., we may be using patient data (medical history) to predict whether a patient smokes or not

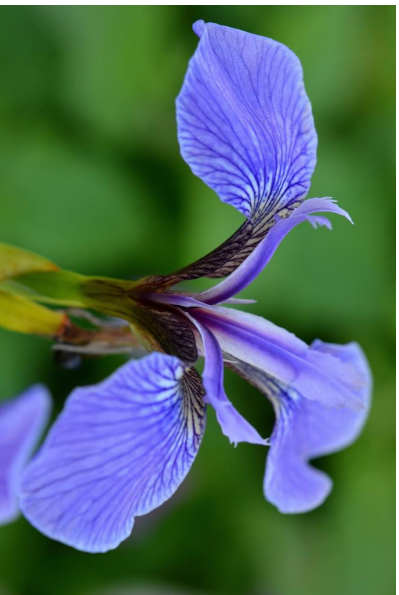
- At first, many problems don't appear to be binary; however, you can usually transform them into binary problems
  - E.g., what if you are predicting whether an image is of a "human", "dog", or "cat"?
  - You can transform this non-binary problem into three binary problems
    - 1. Will it be "human" or "not human"?
    - 2. Will it be "dog" or "not dog"?
    - 2. Will it be "cat" or "not cat"?
- This is similar to the concept of dummy variables

DS

# Iris Dataset

The Iris dataset contains 3 classes of 50 instances each, each class referencing a type of iris plant (Setosa, Versicolor, or Virginica)

**Iris Setosa**



**Iris Versicolor**



**Iris Virginica**



Source: Flickr



# Iris dataset (cont.)

- Can you identify the type of iris based on the following four attributes?
  - Sepal length and width
  - Petal length and width



Source: Flickr



DS

# Codealong & Activity – Part A

## Iris Dataset Exploratory Analysis

# Activity: Iris Dataset Exploratory Analysis



## EXERCISE

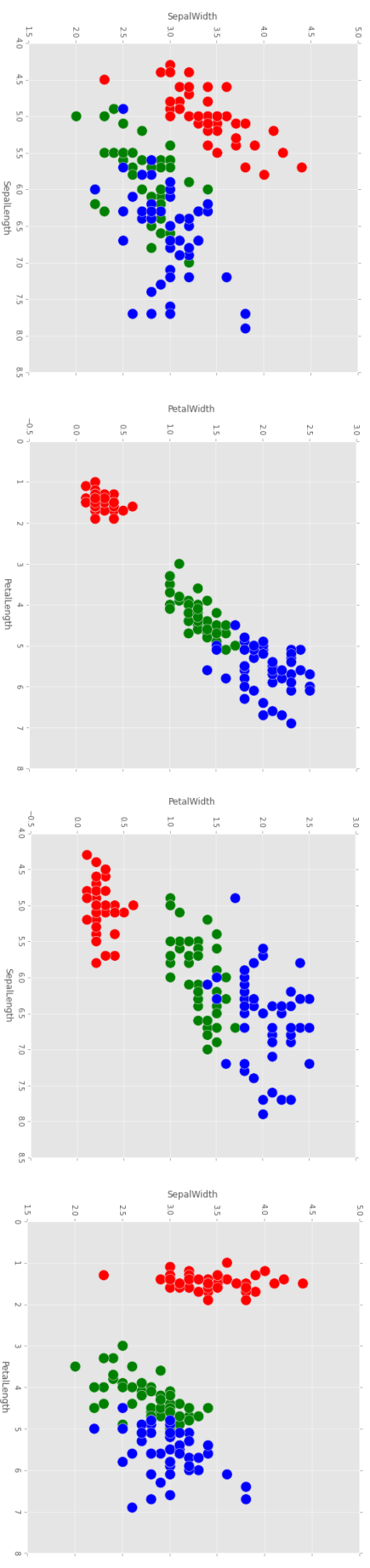
### ANSWER THE FOLLOWING QUESTIONS (10 minutes)

1. Using the Iris dataset (`iris.csv` in the datasets folder), perform exploratory analysis between *SepalLength*, *SepalWidth*, *PetalLength*, and *PetalWidth* (the *feature* variables) and *Species* (the *class* variable). How can you use these features to separate one species from the other two?
2. When finished, share your answers with your table

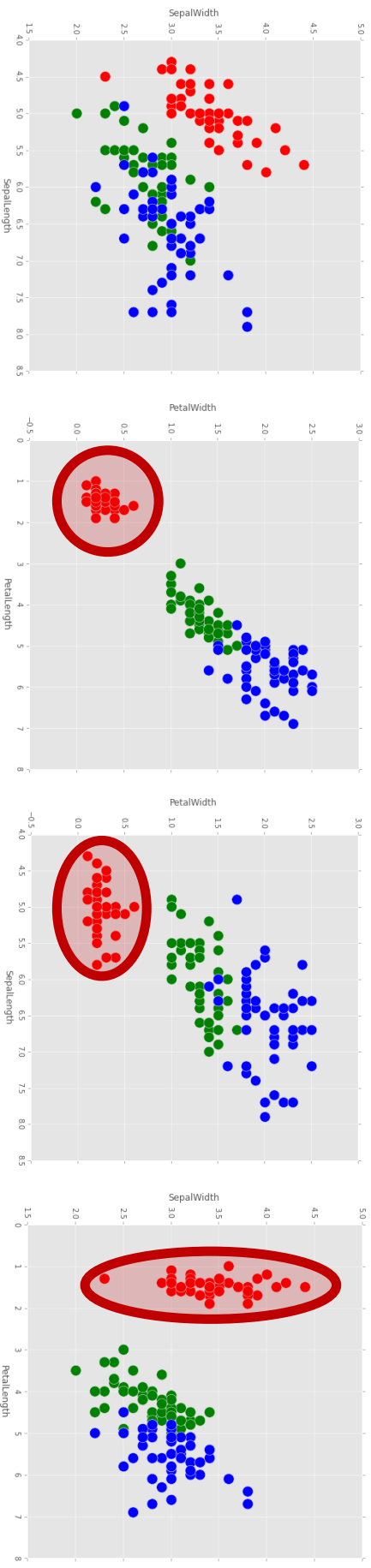
### DELIVERABLE

Answers to the above questions

# Iris Dataset Exploratory Analysis

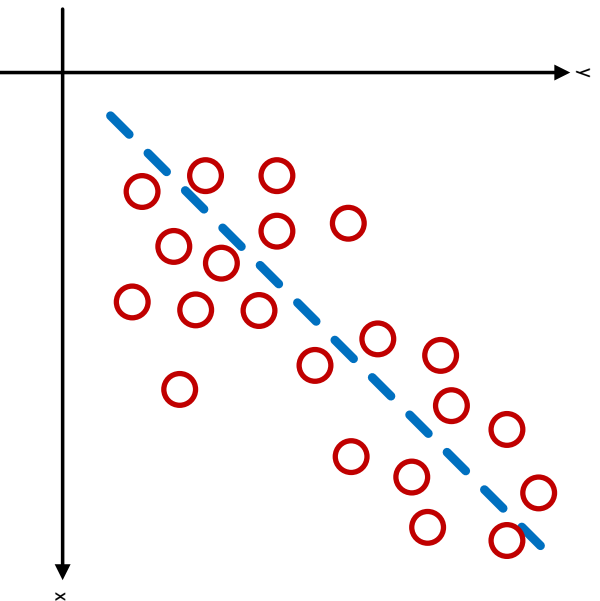


The setosa class is linearly separable from the other two

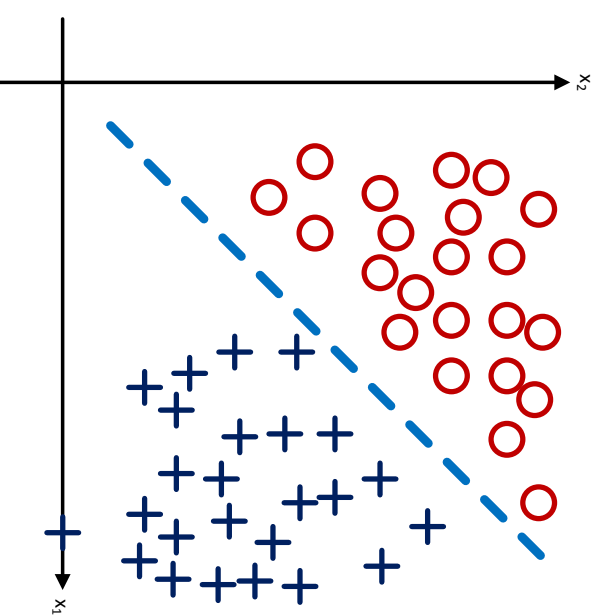


Classification and regression differ in what they are trying to predict

**Regression**



**Classification**



The logo consists of a solid black circle containing the white letters 'DS' in a bold, sans-serif font.

# Codealong – Part B

## First Hand-Coded Classifier

DS

# Classification Metrics



# Classification Metrics

- The metrics we've used for regressions do not apply for classification
  - We could measure distance between the probability of a given class and an item being in the class. E.g., guessing .6 for a 1 is a .4 error, while guessing .99 for 1 is .01 error...
  - but this overly complicates our current goal: understanding binary classifications, like whether something is right or wrong

# Classification Metrics (cont.)

- Instead, let's start with two new metrics, which are inverses of each other: accuracy and misclassification rate
- Since they are opposite of each other, you can pick one or the other; effectively they will be the same. But when coding, do make sure that you are using a classification metric when solving a classification problem!
- *sklearn* will not intuitively understand if you are doing classification or regression, and accidentally using mean squared error for classification, or accuracy for regression, is a common programming pitfall

## ▸ Accuracy

- How many observations that we predicted were correct? This is a value we'd want to increase (like  $R^2$ )

## ▸ Misclassification rate

- Directly opposite of accuracy
- Of all the observations we predicted, how many were incorrect? This is a value we'd want to decrease (like the mean squared error)



DS

# Codealong – Part C

## Classification Metrics



DS

# Codealong & Activity – Part D

## Second Hand-Coded Classifier

# Activity: Second hand-coded classifier



## EXERCISE

### ANSWER THE FOLLOWING QUESTIONS (10 minutes)

1. Improve the first hand-coded classifier to further separate the remaining classes of iris
2. When finished, share your answers with your table

### DELIVERABLE

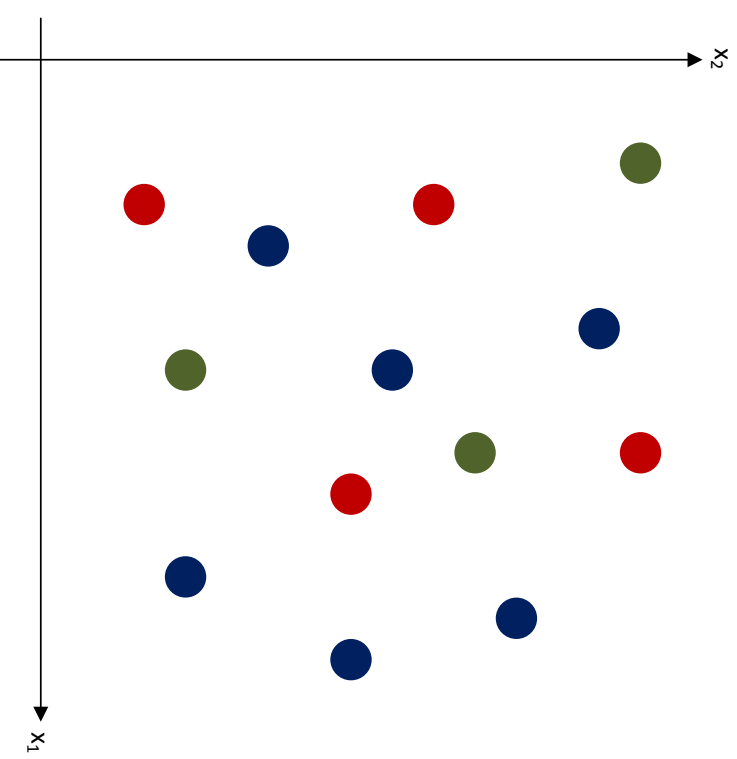
Answers to the above questions

DS

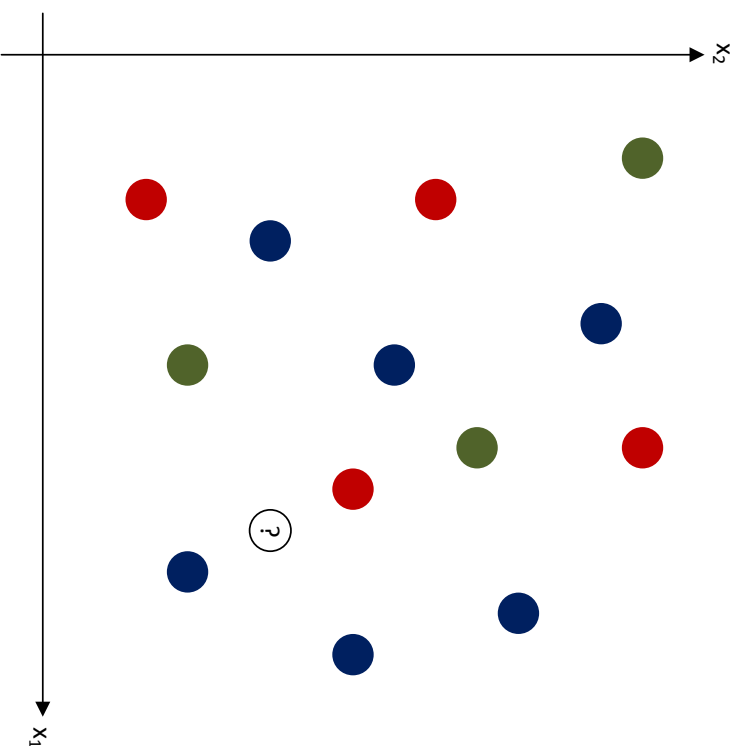
# K-Nearest Neighbors (KNN)

# K-Nearest Neighbors

- K-Nearest Neighbors (KNN) is a classification algorithm that makes a prediction based upon the closest data points

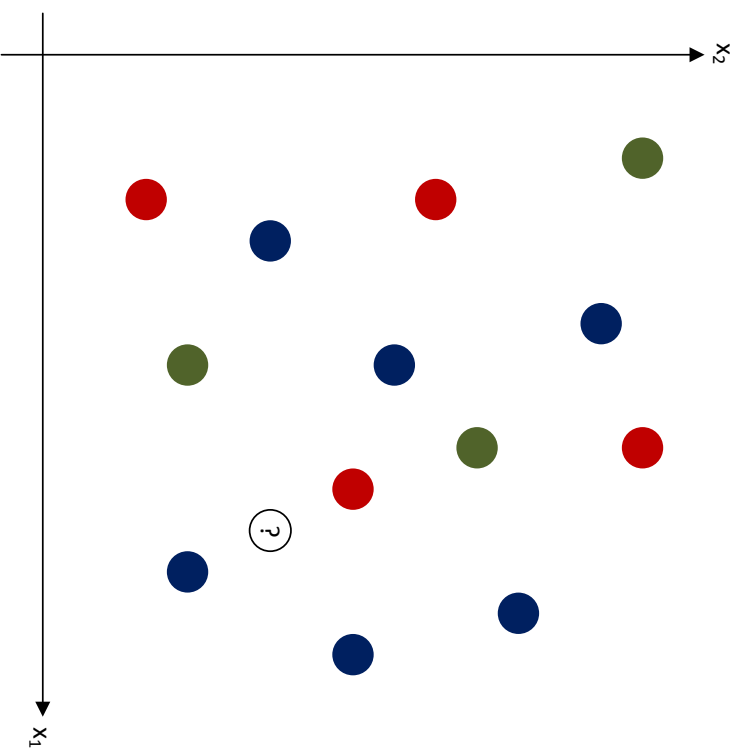


How would you predict the color of the “question mark” point?

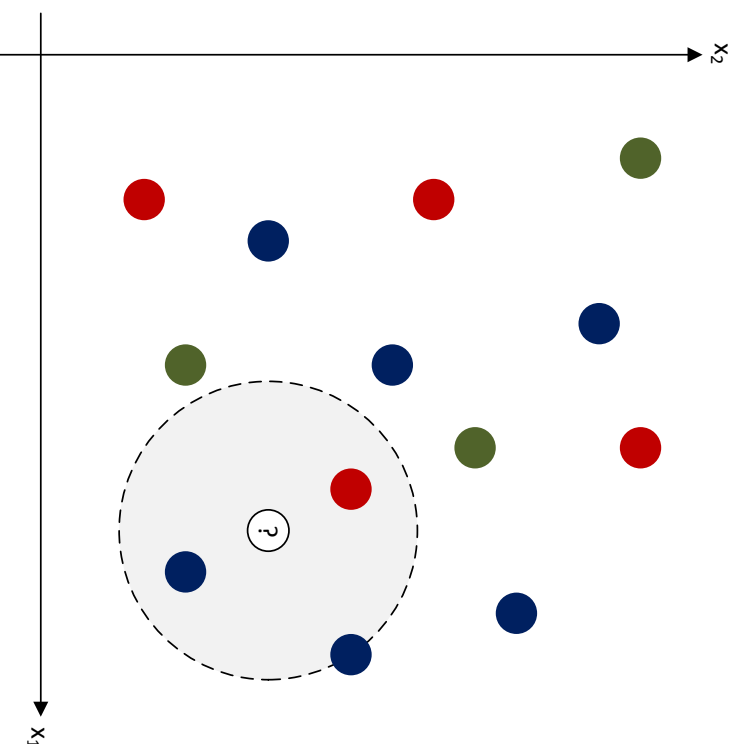




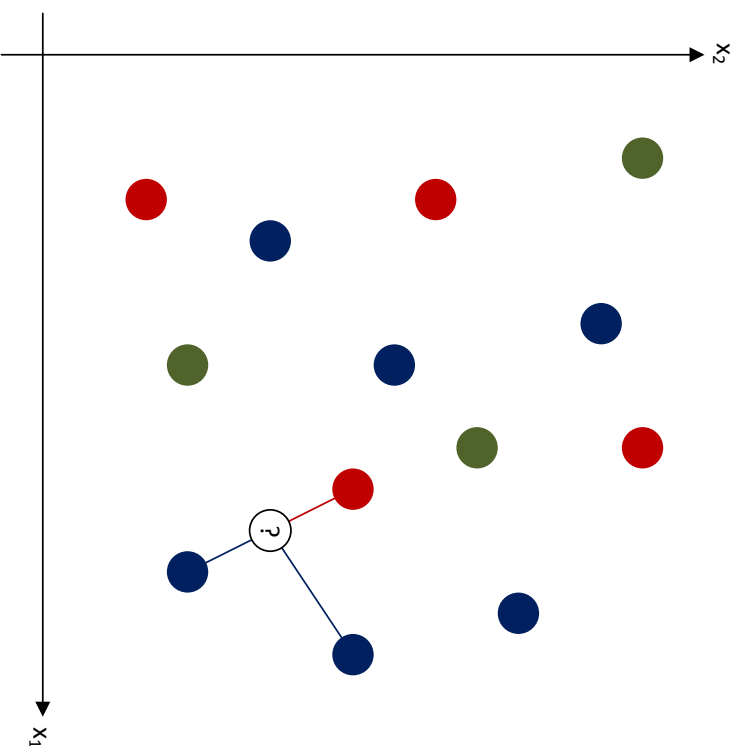
1 Pick a value for  $k$ , e.g.,  $k = 3$



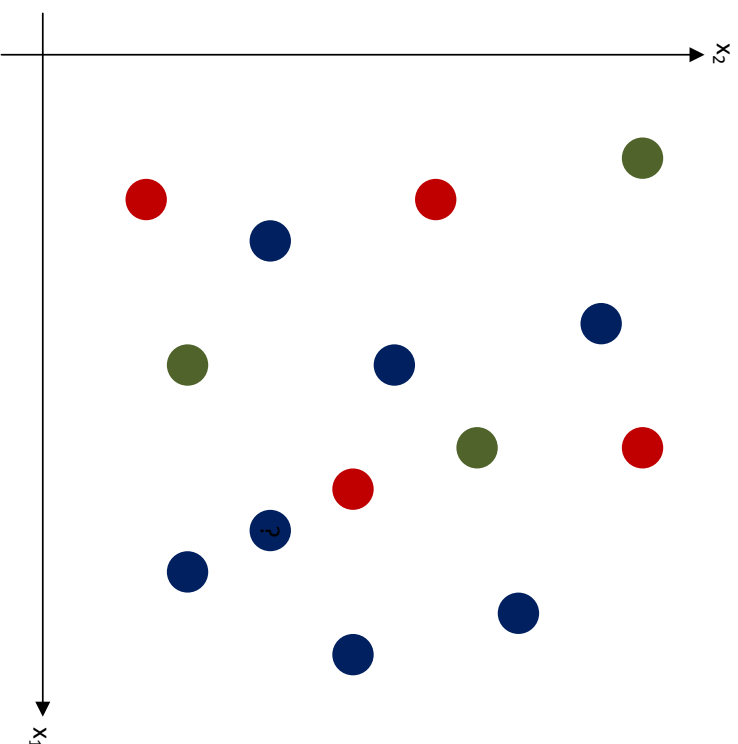
**2** Calculate the distance to all other points; given those distances, pick the  $k$  closest points



- ③ Calculate the probabilities of each class label given those points:  $\frac{1}{3}$  “red”,  $\frac{2}{3}$  “blue”

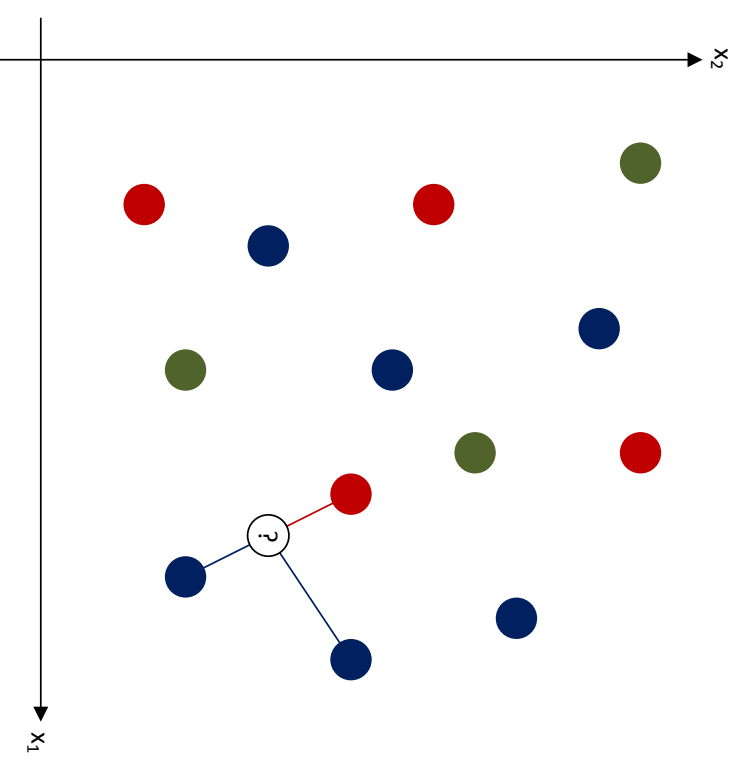


- 4 The original point is classified as the class label with the largest probability (“votes”): “blue”



# K-Nearest Neighbors (cont.)

- KNN uses distance to predict a class label
- This application of distance is used as a measure of similarity between classifications
  - We are using shared traits to identify the most likely class label



# What happens if two classes get the same number of votes?

- *Scikit-learn* will choose the class it first saw in the training set

- We could also implement a weight, taking into account the distance between a point and its neighbors

- This can be done in *sklearn* by changing the *weights* parameter to *'distance'*



DS

# Codealong – Part E

## K-Nearest Neighbors

DS

# High Dimensionality



# What happens in high dimensionality?

- Since KNN works with distance, higher dimensionality of data (i.e., more features) requires significantly more samples in order to have the same predictive power
  - With more dimensions, all points slowly start averaging out to be equally distant; this causes significant issues for KNN
- Keep the feature space limited and KNN will do well; exclude extraneous features when using KNN



DS

# Codealong – Part F

## What is the best value for $k$ ?

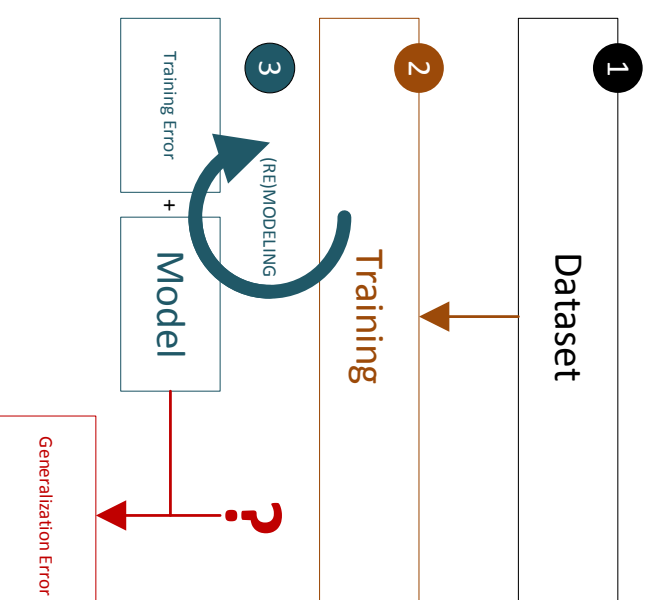
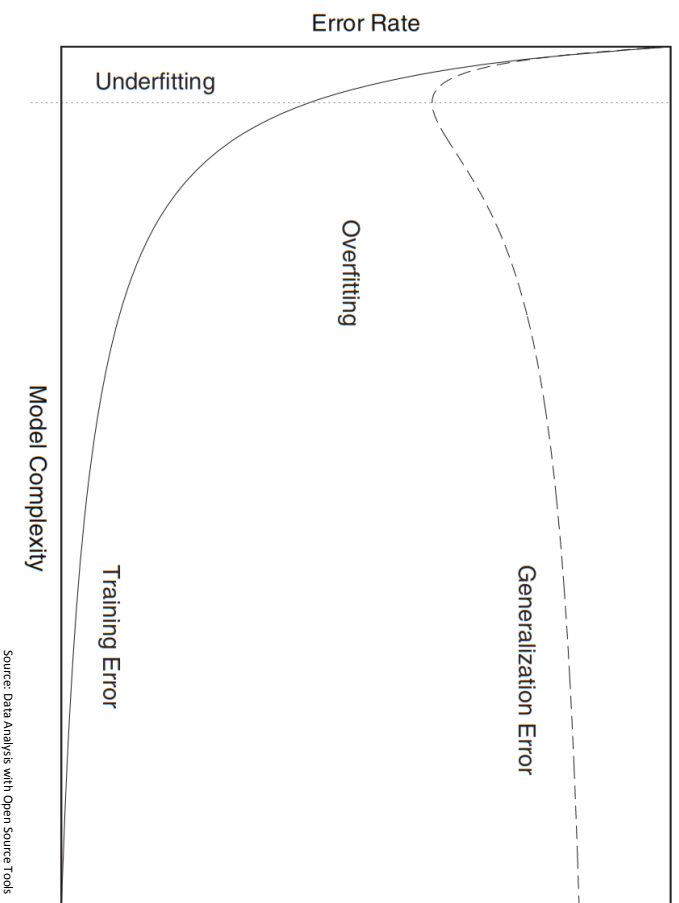
DS

What is the best value for  $k$ ?



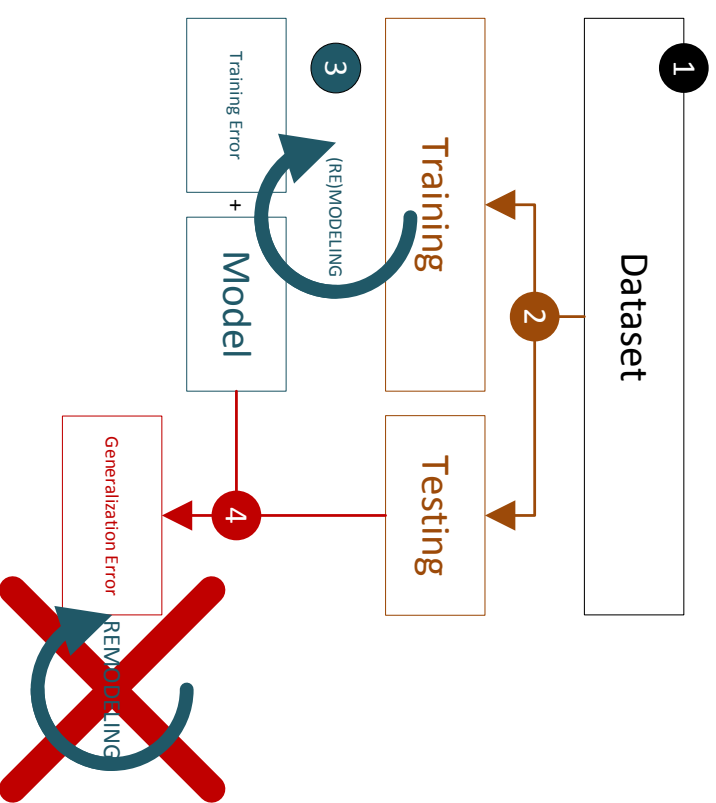
# Validation

So far, we used the entire dataset to train the models.  
How can we estimate the generalization error?



# Validation is an answer

- Answer: (Randomly) divide the dataset into a training set and a testing set
  - Set aside the testing set; don't look at it
- Train the models with the training set
  - Compute the training set and remodel as needed
- Once you are happy with your model, use the testing set to compute the generalization error
  - But you cannot go back and remodel; otherwise these previously unknown data points are not longer unseen





DS

# Codealong – Part G

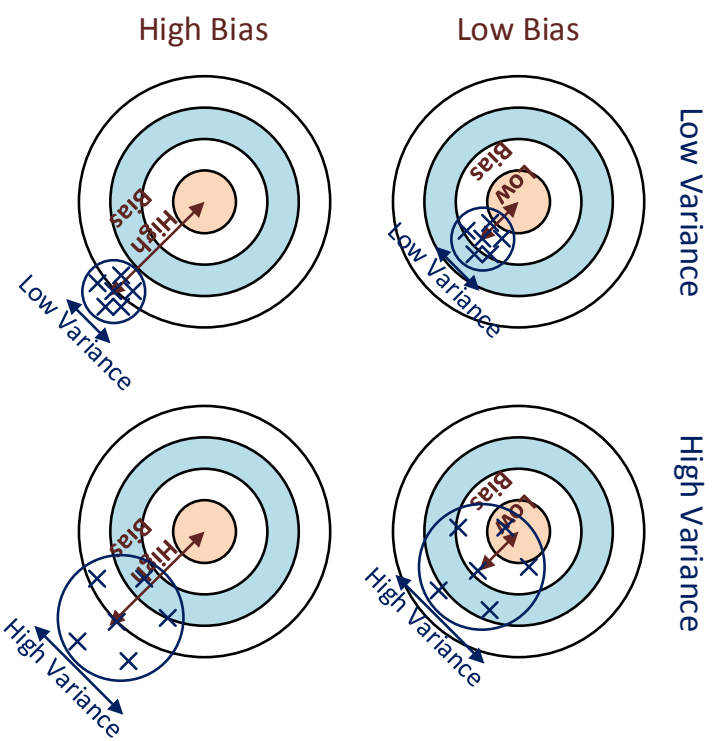
## Validation

DS

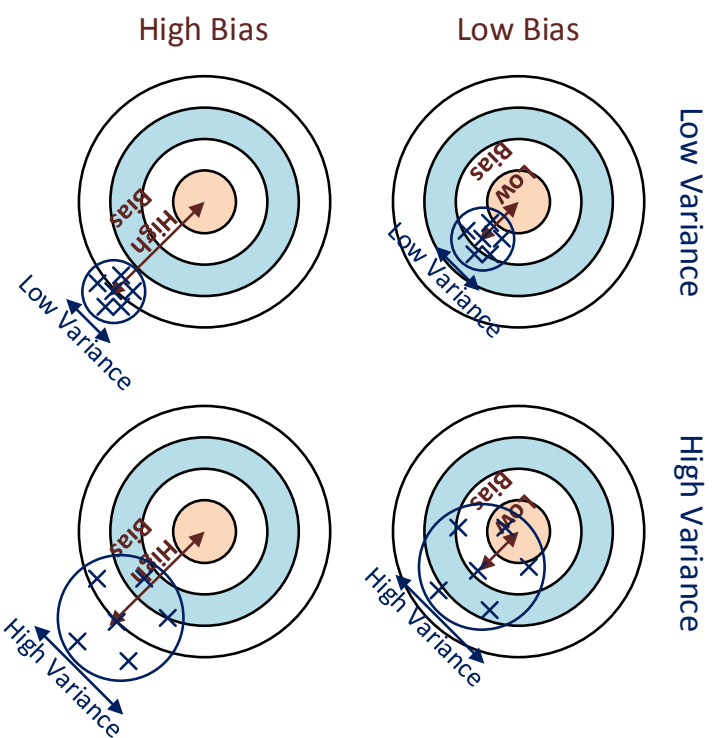
# Cross-Validation



Recall our conversation about bias and variance, a.k.a., *systematic and random errors*? (session 3)

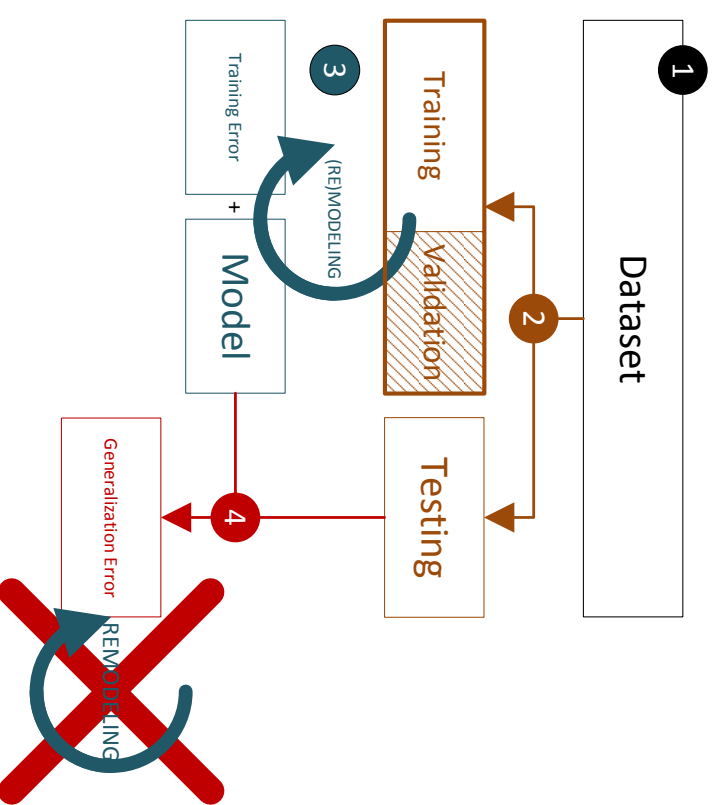


The generalization error has a bias component (systematic; non-random) and a variance component (idiosyncratic; random). Can we lower the bias error?



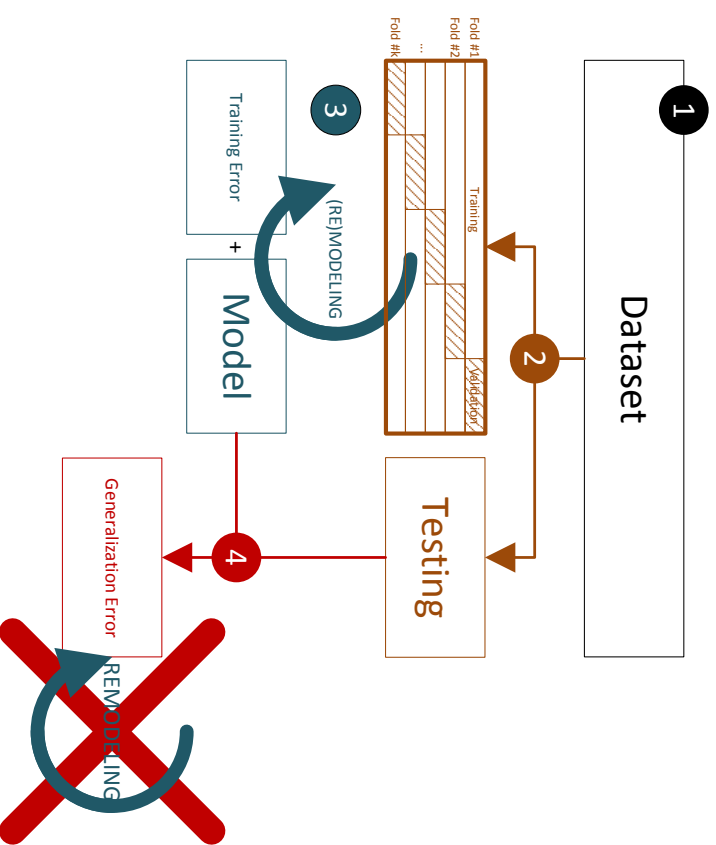
# Cross-validation (CV) is a technique to lower the bias error

- Cross-validation
  - Another technique to validate models
  - Used to estimate how accurately the model generalize to unseen data
  - You can iterate as much as you want with the data
  - You then build a final model that uses all the data (cross-validation is used for model checking, not model building)
- [You still create an unseen testing set to estimate how well your model generalize to unseen data (and you stop there; no remodeling)]



# (k-fold) cross-validation

- k-fold cross-validation
  - Quite popular
  - Typically,  $k = 5$  or  $10$  with each sample being used both for training ( $k - 1$  times) and validation (1 time)
  - The training error is the average training error of all folds
  - Again, after selecting the model that minimize the training error, you then build a final model that uses all the data
- You still create an unseen testing set to estimate how well your model generalize to unseen data (and you stop there; no remodeling)





DS

# Codealong – Part H

## Cross-Validation



DS

# Advantages and Disadvantages of KNN

# Advantages and disadvantages of KNN

## Advantages

- Simple to understand and explain
- Model training phase is fast
- Non-parametric (does not presume a “form” of the “decision boundary”)

## Disadvantages

- Prediction phase can be slow when  $n$  is large
- Sensitive to irrelevant features
- Very sensitive to feature scaling



# Review



# Review

- What are class labels? What does it mean to classify?
- How is a classification problem different from a regression problem?  
How are they similar?
- How does the KNN algorithm work?
- What primary parameters are available for tuning a KNN estimator?
- How do you define accuracy and misclassification?

## Review (cont.)

You should now be able to:

- Define class label and classification
- Build a K-Nearest Neighbors using the scikit-learn library
- Evaluate and tune model by using metrics such as classification accuracy/error

DS

# Pre-Work

# Pre-Work

Before the next lesson, you should already be able to:

- Implement a linear model (`LinearRegression`) with *scikit-learn*
- Define the concept of coefficients
- Recall metrics for accuracy and misclassification
- Recall the differences between L1 and L2 regularization



# Q & A



DS

# Exit Ticket

*Don't forget to fill out your exit ticket [here](#)*