Introduction to Classification

Ivan Corneillet

Data Scientist

GENERAL ASSEMBLY

Learning Objectives

After this lesson, you should be able to:

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

Announcements and Exit Tickets

Q & A

Review

Review

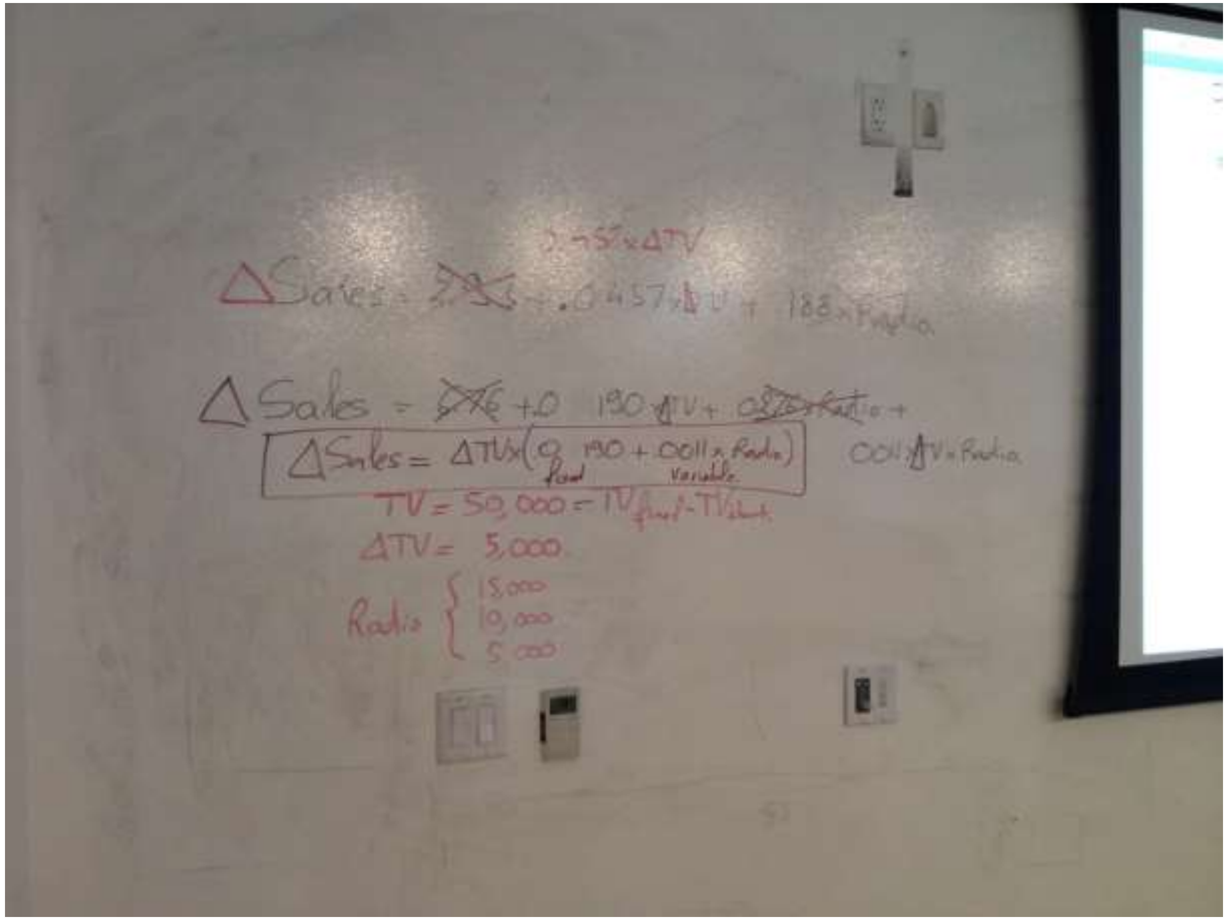
Linear Regression

Linear regression is a simple approach to supervised learning. Pros and Cons are:

- Pros
  - Intuitive and well-understood
  - Can perform well with a small number of observations
  - Highly interpretable and simple to explain
  - Model training and prediction are fast
  - No need to standardize your data (features don’t need scaling)
  - No tuning is required (excluding regularization which is a topic we won’t discuss)

- Cons
  - Assumes linear association among variables
  - Assumes normally distributed residuals
  - Outliers can easily affect coefficients

Linear Regression | Interaction effects



Review

Activity | Linear Regression | Customer Retention Rates

Activity | Linear Regression | Customer Retention Rates

EXERCISE

DIRECTIONS (20 minutes)

1. The following dataset documents the “survival” pattern over seven years for a sample of 1000 customers who were all “acquired” in the same period
2. Build one or more models to capture this pattern, then use each model to project the survival curve over the next five years
3. When finished, share your answers with your table

DELIVERABLE

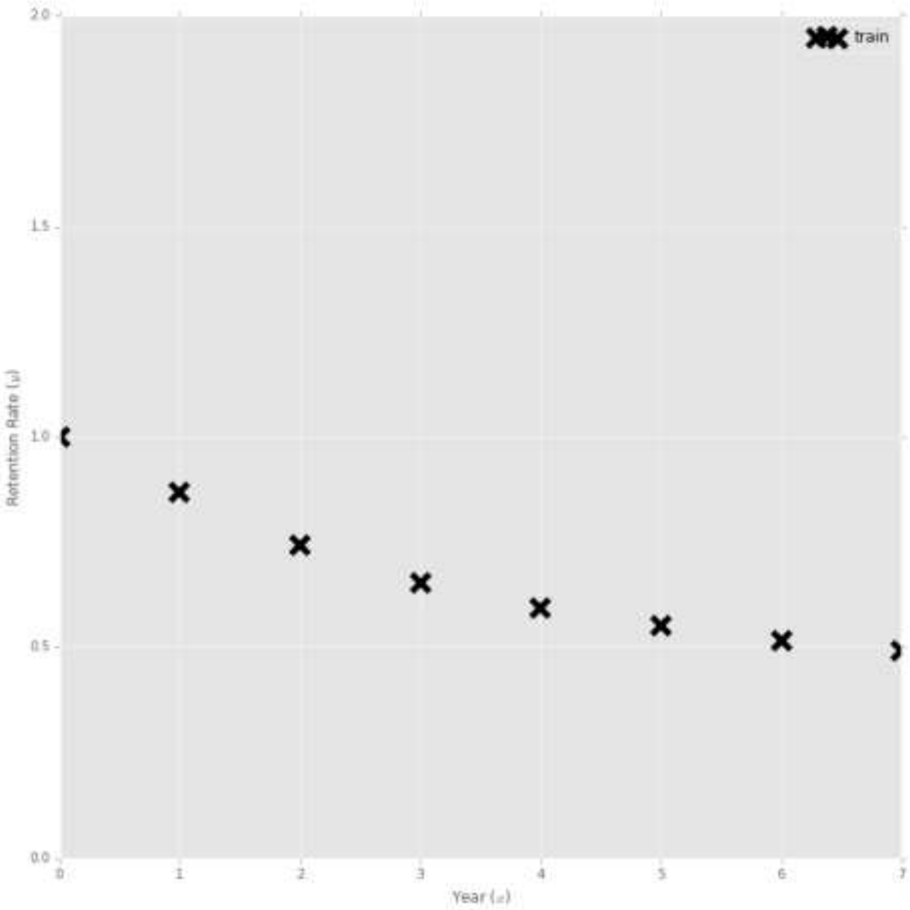
Answers to the above questions

Year	Retention Rate
0	1
1	.869
2	.743
3	.653
4	.593
5	.551
6	.517
7	.491

Source: Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management

Activity | Retention rate (y) as a function of the year (x)

Year (x)	Retention Rate (y)
0	1
1	.869
2	.743
3	.653
4	.593
5	.551
6	.517
7	.491

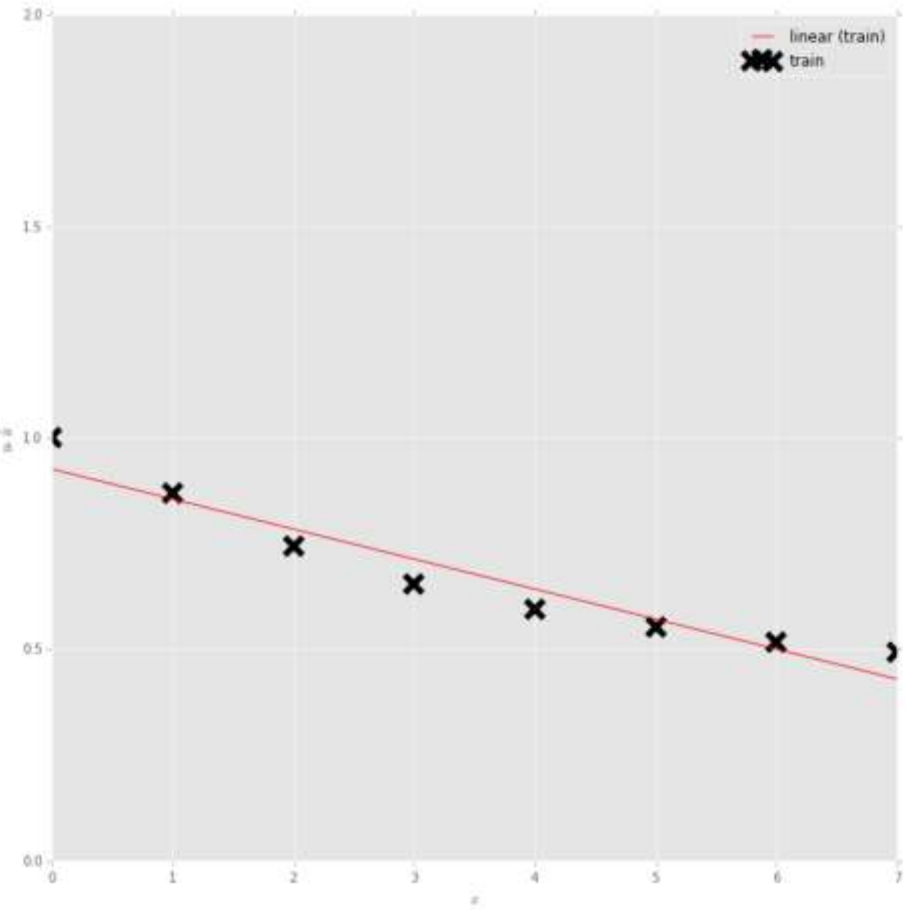


Activity | Linear Model: $y = .9254 - .0709t$

Dep. Variable:	y	**R-squared:**	0.922
Model:	OLS	**Adj. R-squared:**	0.909
Method:	Least Squares	**F-statistic:**	70.91
Date:		**Prob (F-statistic):**	0.000153
Time:		**Log-Likelihood:**	13.061
No. Observations:	8	**AIC:**	-22.12
Df Residuals:	6	**BIC:**	-21.96
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[95.0% Conf. Int.]**
Intercept	0.9254	0.035	26.258	0.000	0.839 1.012
x	-0.0709	0.008	-8.421	0.000	-0.092 -0.050

Omnibus:	1.277	**Durbin-Watson:**	0.634
Prob(Omnibus):	0.528	**Jarque-Bera (JB):**	0.711
Skew:	0.310	**Prob(JB):**	0.701
Kurtosis:	1.678	**Cond. No.**	7.95

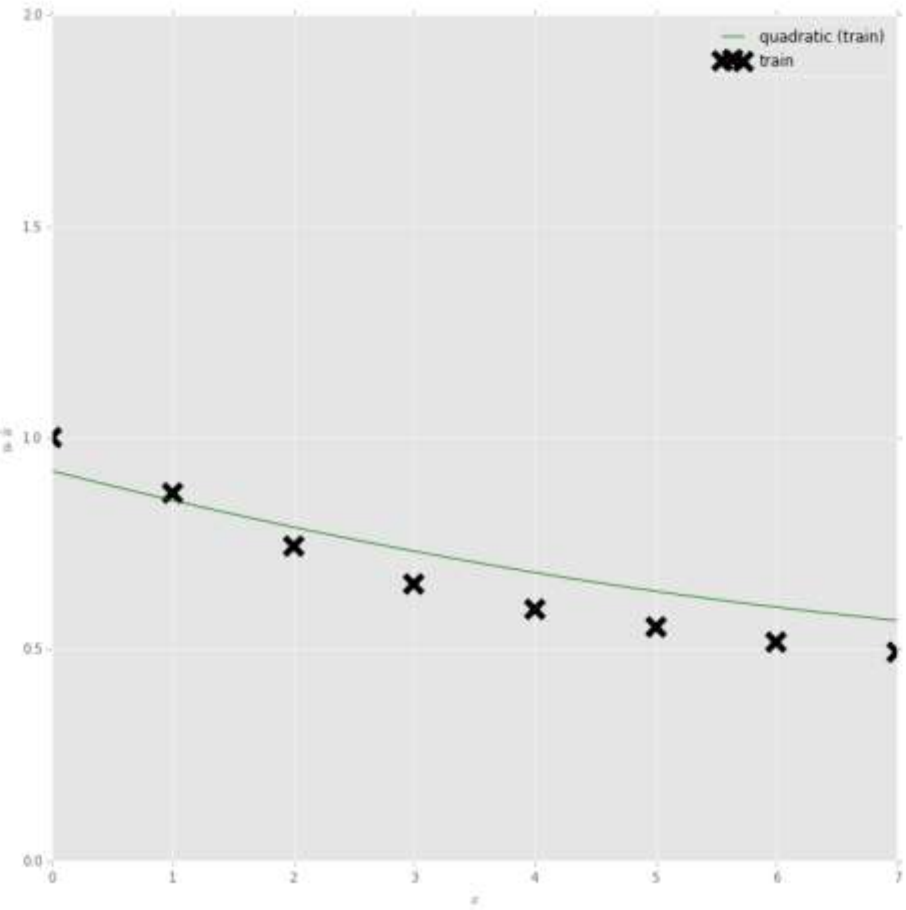


Activity | Quadratic Model: $y = .9211 - .0729t + .0032t^2$

Dep. Variable:	y	R-squared:	0.923
Model:	OLS	Adj. R-squared:	0.892
Method:	Least Squares	F-statistic:	30.03
Date:		Prob (F-statistic):	0.00164
Time:		Log-Likelihood:	13.121
No. Observations:	8	AIC:	-20.24
Df Residuals:	5	BIC:	-20.00
Df Model:	2		
Covariance Type:	nonrobust		

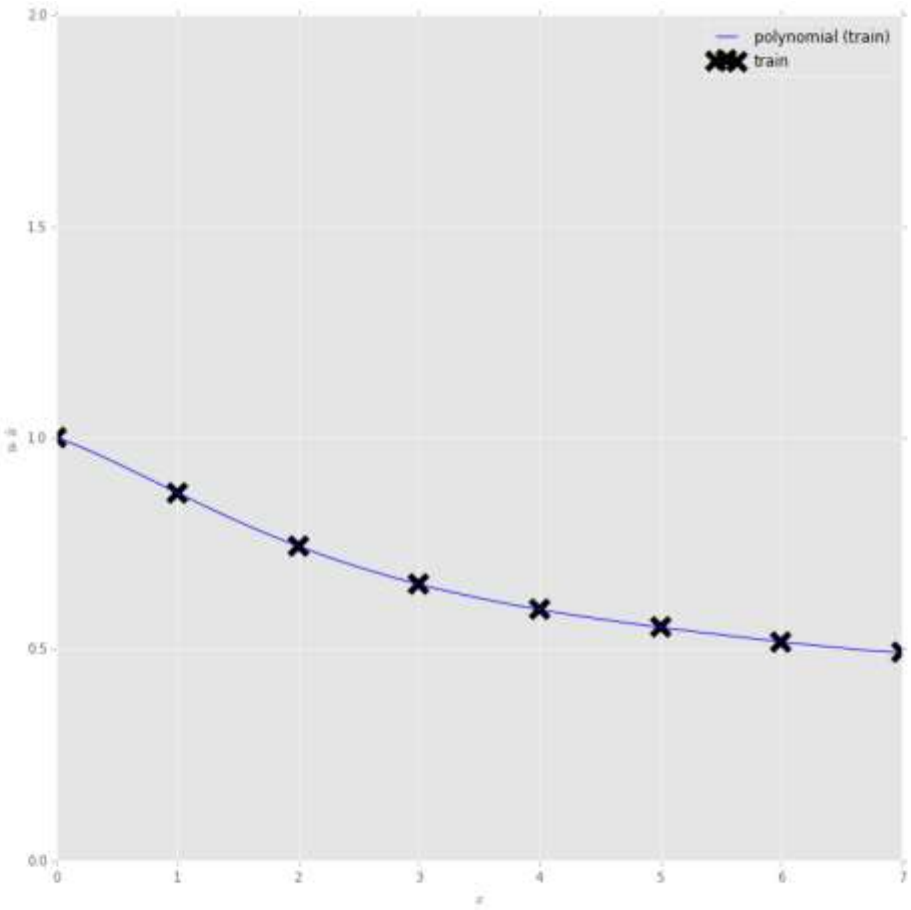
	coef	std err	t	P>|t|	[95.0% Conf. Int.]
Intercept	0.9211	0.041	22.274	0.000	0.815 1.027
x	-0.0729	0.012	-6.252	0.002	-0.103 -0.043
x ^ 2	0.0032	0.012	0.275	0.795	-0.027 0.033

Omnibus:	1.491	Durbin-Watson:	0.630
Prob(Omnibus):	0.474	Jarque-Bera (JB):	0.769
Skew:	0.342	Prob(JB):	0.681
Kurtosis:	1.644	Cond. No.	11.6

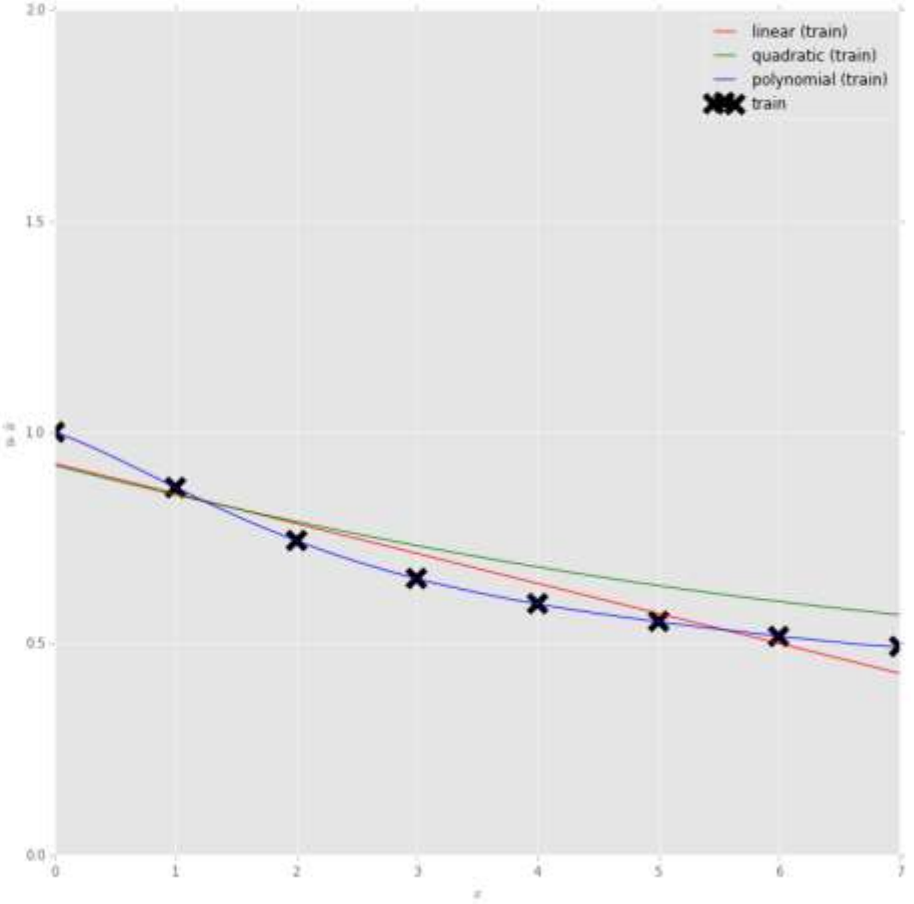


Activity | Polynomial of degree 7

$$y = 1 - .100597619t - .0596777778t^2 + .0380569444t^3 - .010194444t^4 + .00153611111t^5 - .0001277777t^6 + .0000456349206t^7$$

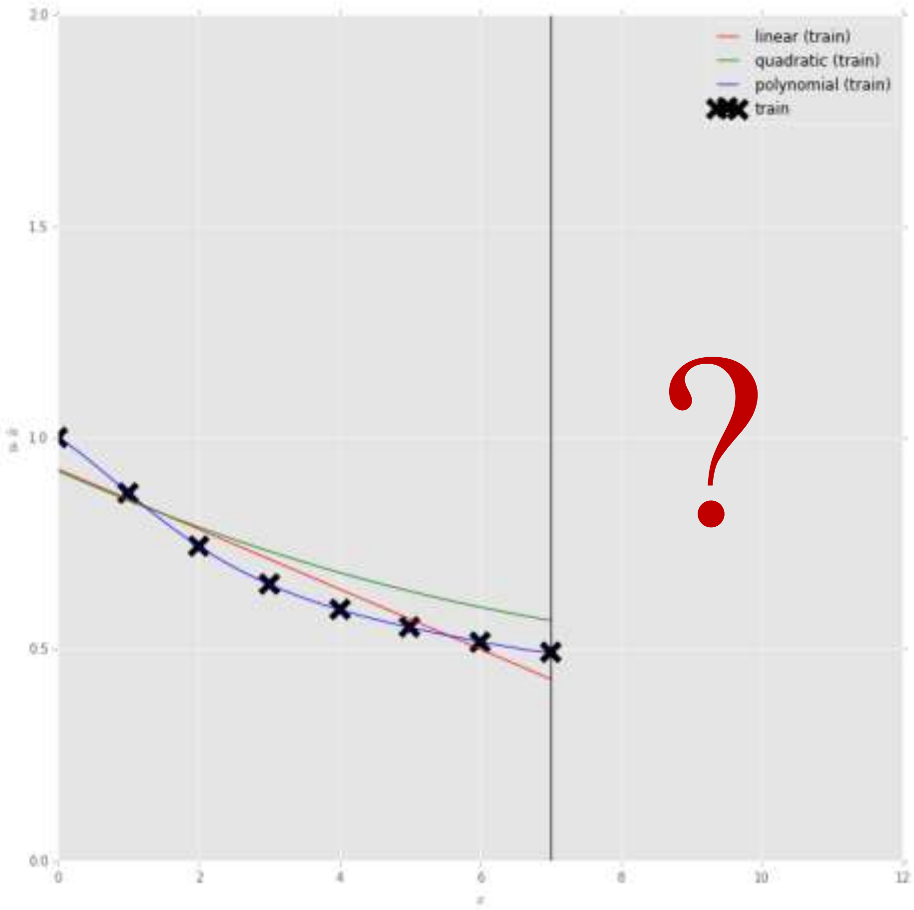


Activity | Training data's R^2



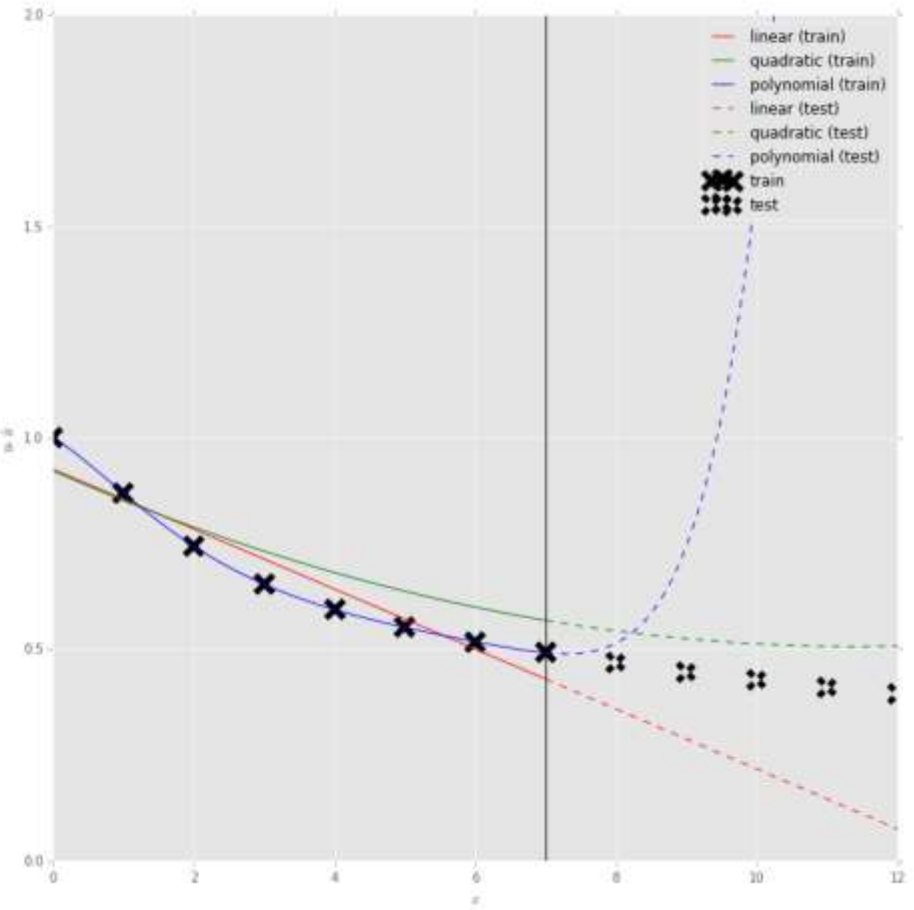
Model	R^2
Linear	.922
Quadratic	.923
Polynomial	1

Activity | Test data and the models' R^2?



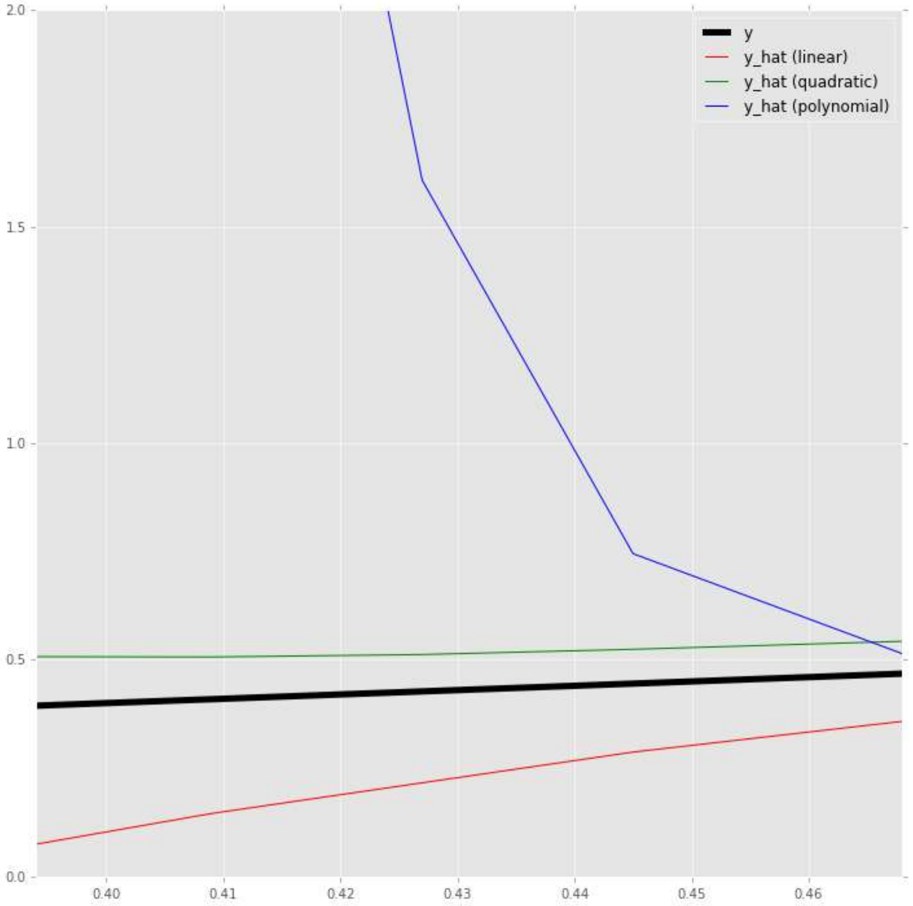
Model	R^2
Linear	?
Quadratic	
Polynomial	

Activity | Test data and the models' R^2 (cont.)



Model	R^2
Linear	.994
Quadratic	.895
Polynomial	.739

Activity | Why are the models' R^2 so high on the test data?



- In the interval $8 \le t \le 12$, the test data, the linear and quadratic models are "close to be linear"
  - The linear and quadratic models are pretty linear against the test data (within $8 \le t \le 12$)
- Then recall that (1) $R^2 = \rho_{X,Y}^2$ and (2) $\rho_{X,Y}$ measures the strength for a linear association between X and Y

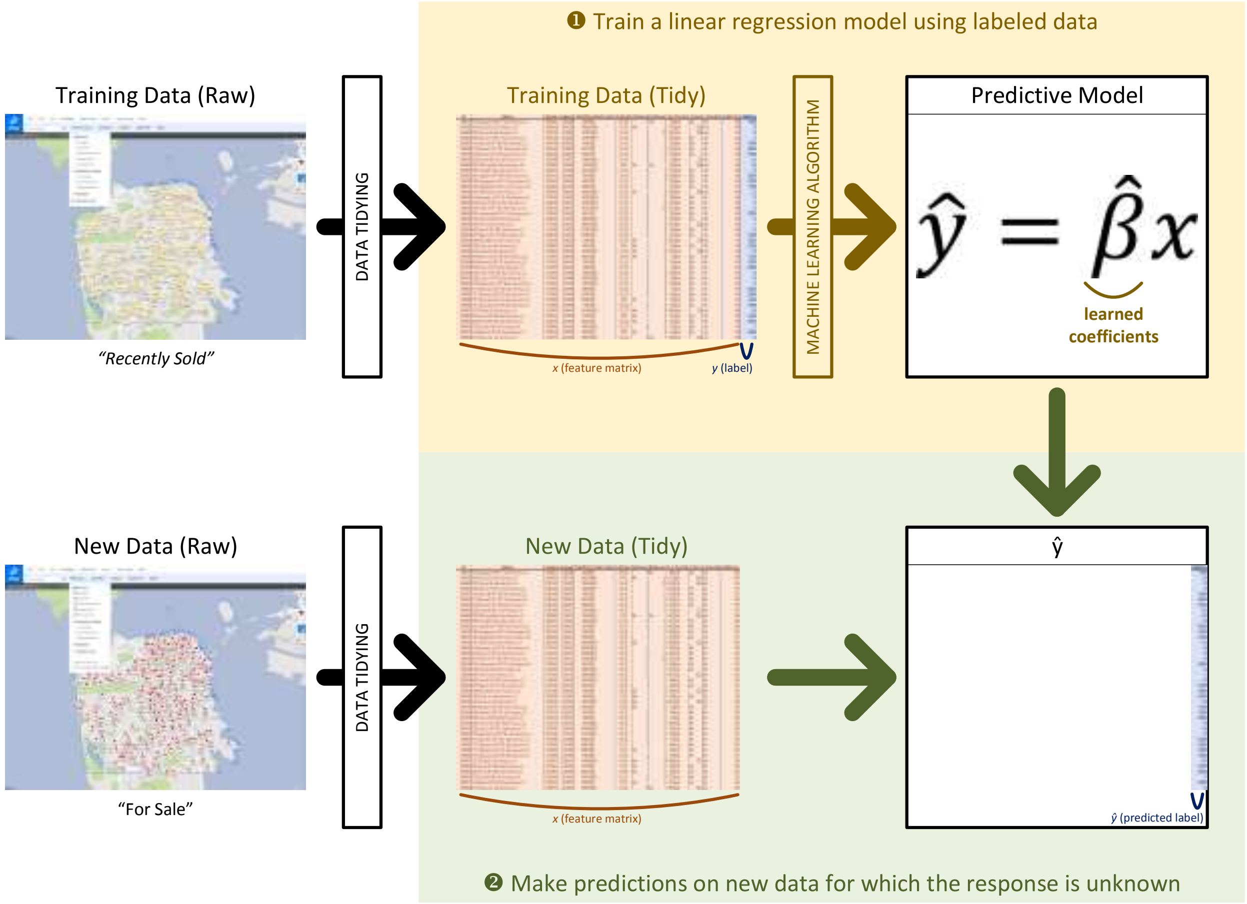
Activity | Takeaways

- Scoring a model (e.g., with R^2 for a linear regression model) using training data ("seen" data) is not a guarantee that the model will generalize well on "unseen" data (e.g., test data, data that hasn't been used to train the model)
  - E.g., the polynomial of degree 7 memorized perfectly the training data but did poorly on new data (overfitting)
- A high R^2, on training data but also on testing data, is not a guarantee that your model accurately fit the training or testing data

Review

Linear Regression (cont.)

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)

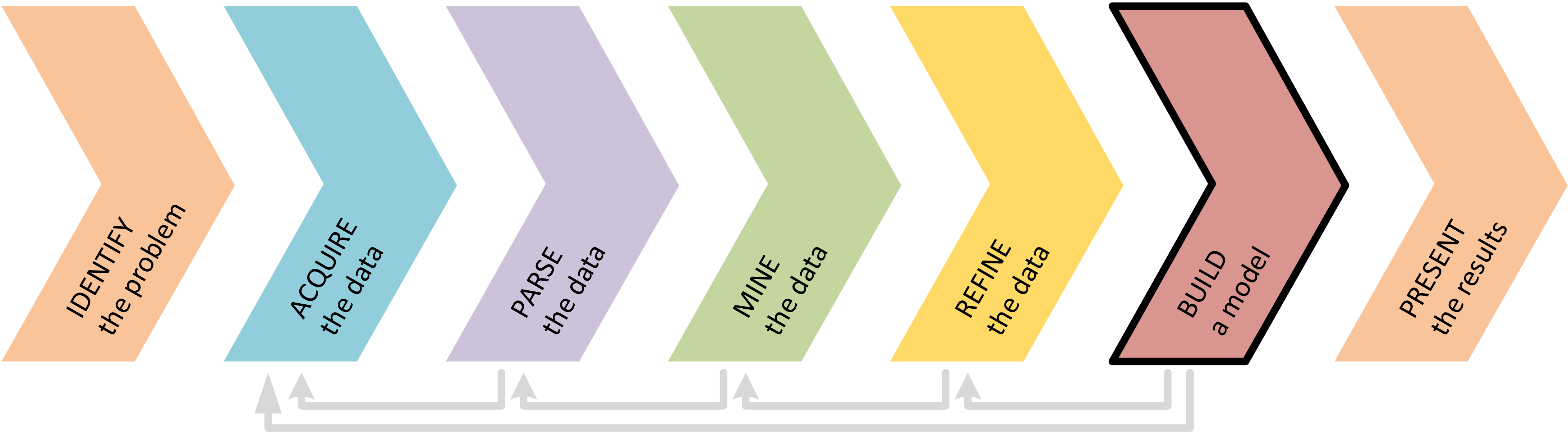
- We've focused so far on predicting a continuous set of values
  - That means that we've been able to use distance to measure how accurate our predictions are
- However, for other problems, we need to predict binary responses. E.g., will a loan default? Is an email spam or ham?

Today

Today, we are introducing what classification is and what classification models are

Research Design and Data Analysis	Research Design	Data Visualization in *pandas*	Statistics	Exploratory Data Analysis in *pandas*
Foundations of Modeling	Linear Regression	Classification Models	Evaluating Model Fit	Presenting Insights from Data Models
Data Science in the Real World	Decision Trees and Random Forests	Time Series Data	Natural Language Processing	Databases

Today, we keep our focus on the BUILD a model step but with a focus on classification



Here’s what’s happening today:

- Announcements and Exit Tickets
- Review
- ❻ Build a Model | Classification
  - Types of machine learning problems
  - What’s classification; what’s binary classification?
  - Classification vs. regression
  - Iris dataset
    - Exploratory data analysis
    - Hand-coded classifiers
  - Classification metrics
  - K-Nearest Neighbors (KNN)
    - High dimensionality
    - What’s the best value for k?
  - Validation and cross-validation
- Lab – Introduction to Logistic Regression
- Review
- **Final Project 2 (due next session on 6/2)**

❻ Build a Model

Types of Machine Learning Problems

Types of Machine Learning Problems

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

❻ Build a Model

What is Classification and what is Binary 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 binary variables

Iris Dataset

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

Iris Setosa

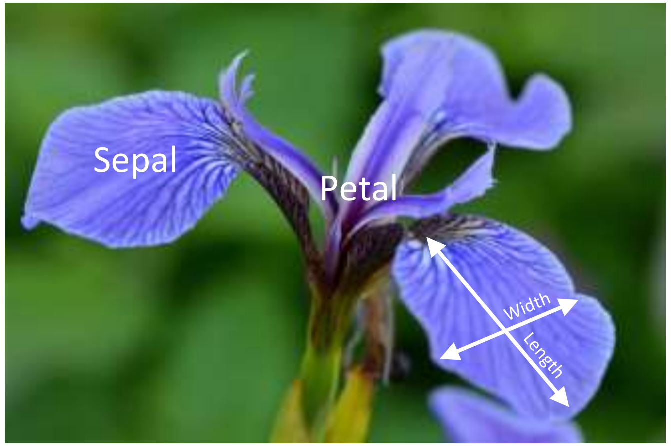
Iris Versicolor

Iris Virginica

Source: Flickr

Iris dataset (cont.)

- Can we teach a machine to identify the type of iris based on the following four attributes?
  - Sepal length and width
  - Petal length and width



Source: Flickr

Iris Dataset

Activity & Codealong – Part A

Exploratory Data Analysis

Activity | Iris Dataset | Exploratory Data Analysis

EXERCISE

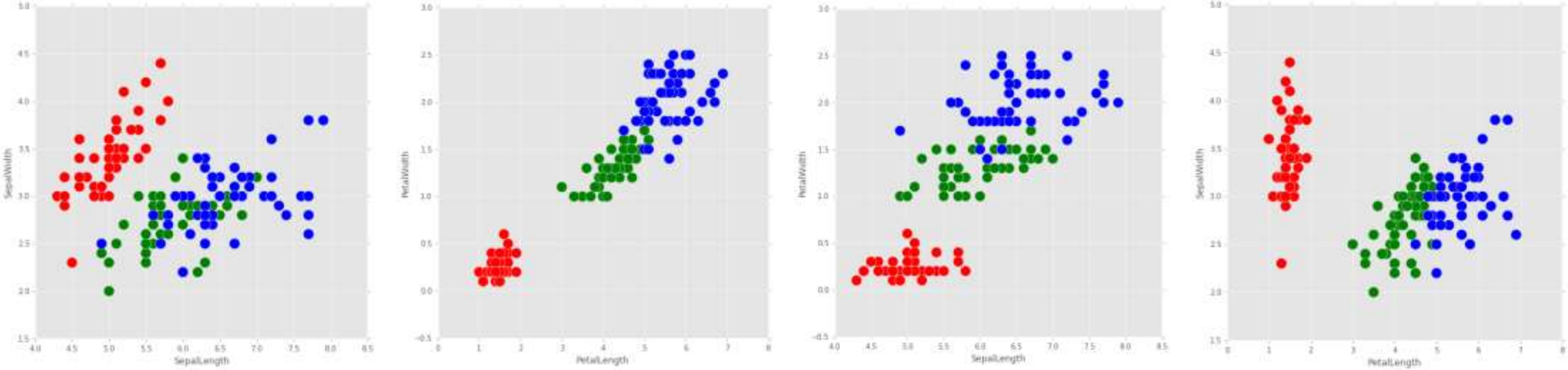
DIRECTIONS (10 minutes)

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

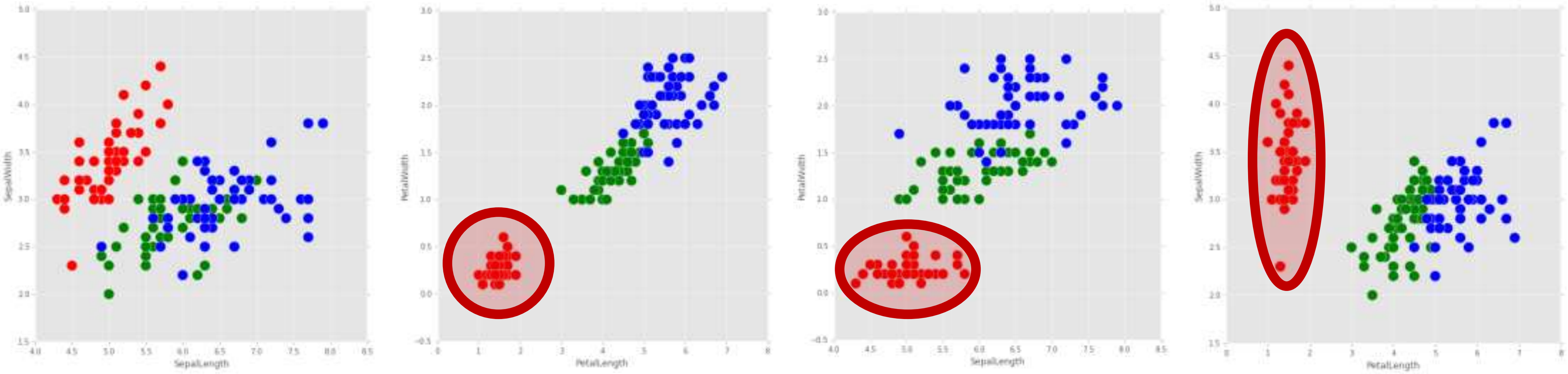
DELIVERABLE

Answers to the above questions

Activity | Iris Dataset | Exploratory Data Analysis (cont.)



Iris Dataset | The *Setosa* class (in red) is linearly separable from the other two (*Versicolor* in green and *Virginica* in blue)

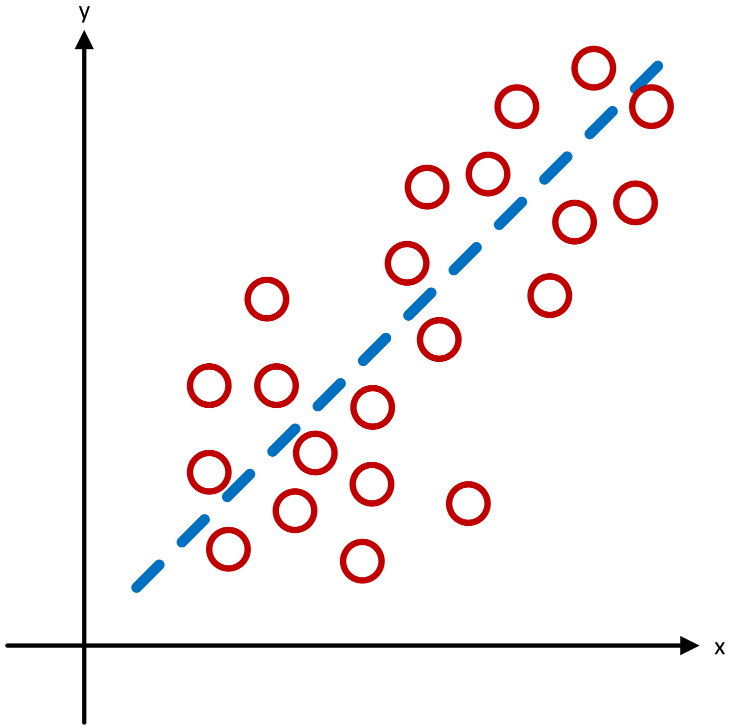


❻ Build a Model

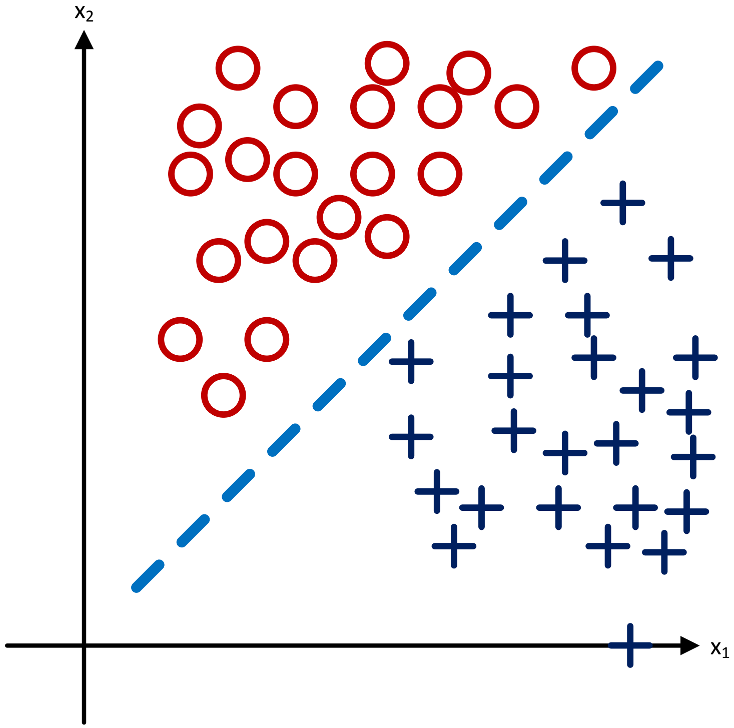
Classification vs. Regression

Classification and regression differ in what they are trying to predict

Regression



Classification



Iris Dataset

Codealong – Part B

First Hand-Coded Classifier

❻ Build a Model

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)

Iris Dataset

Codealong – Part C

Classification Metrics

Iris Dataset

Activity & Codealong – Part D

Second Hand-Coded Classifier

Activity | Iris Dataset | Second hand-coded classifier

EXERCISE

DIRECTIONS (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

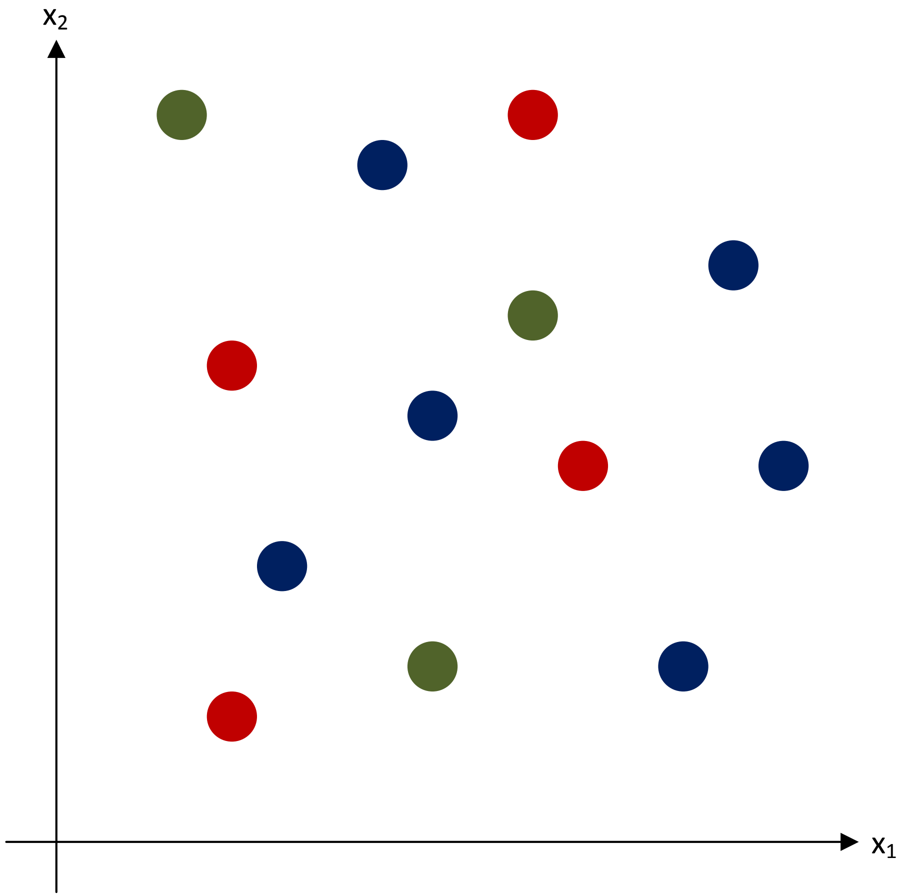
DELIVERABLE

Answers to the above questions

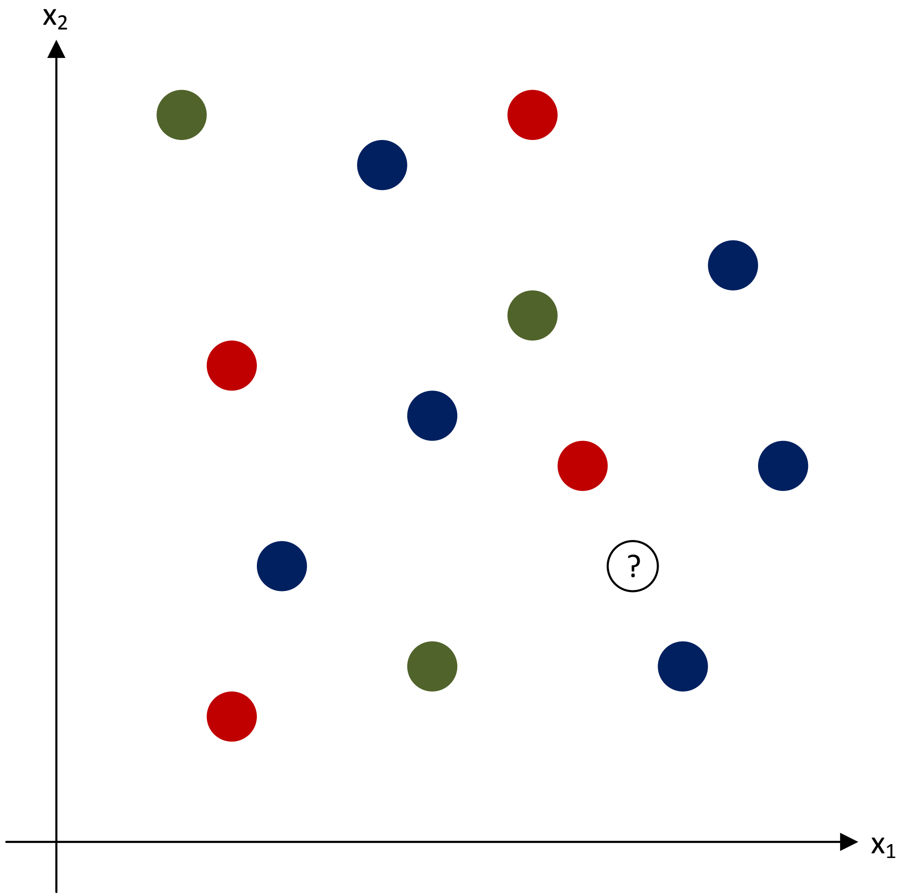
K-Nearest Neighbors (KNN)

K-Nearest Neighbors

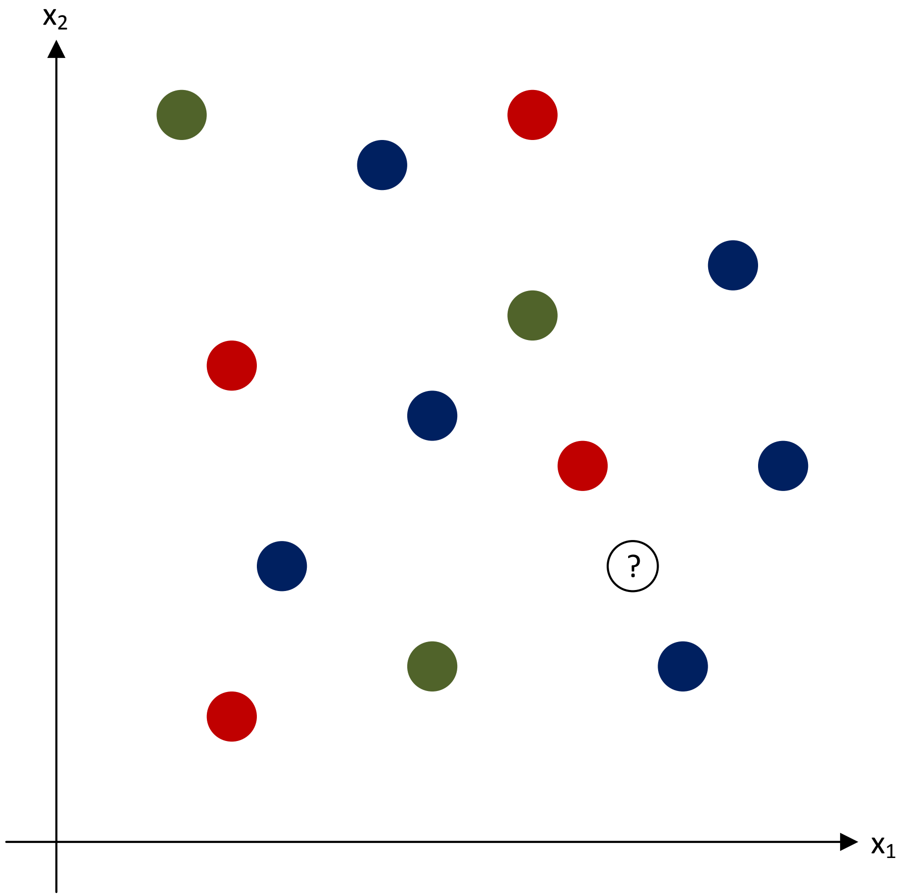
- K-Nearest Neighbors (KNN) is a classification algorithm that makes a prediction based upon the closest data points



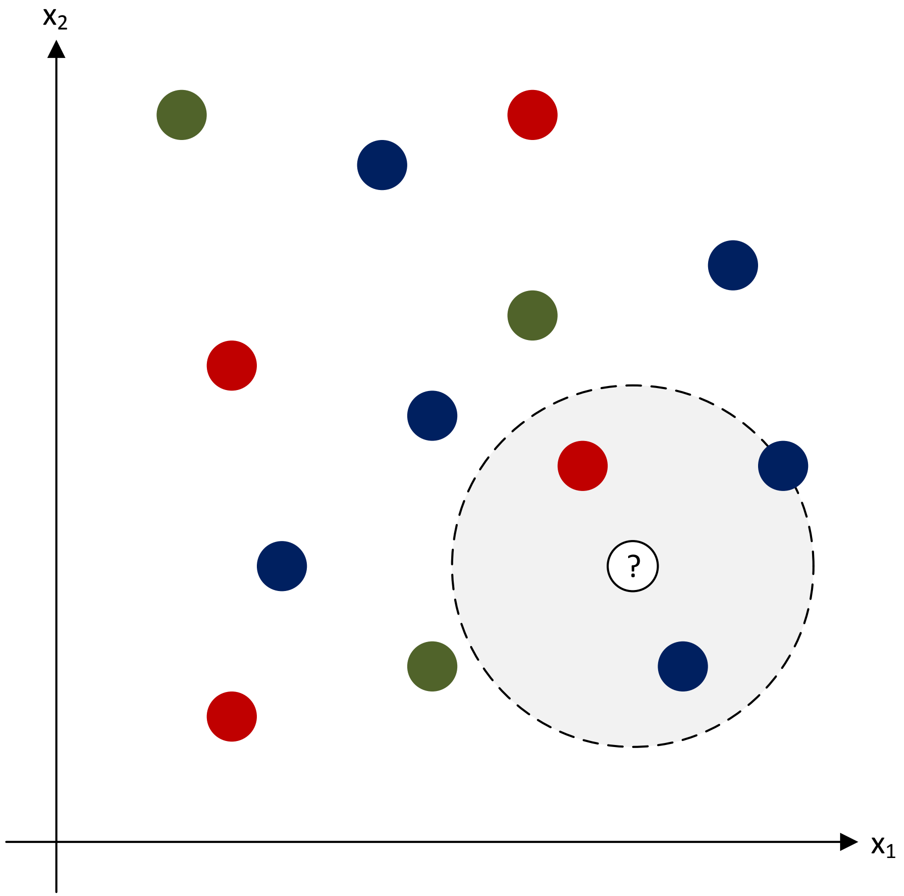
KNN | How would you predict the color of the "question mark" point?



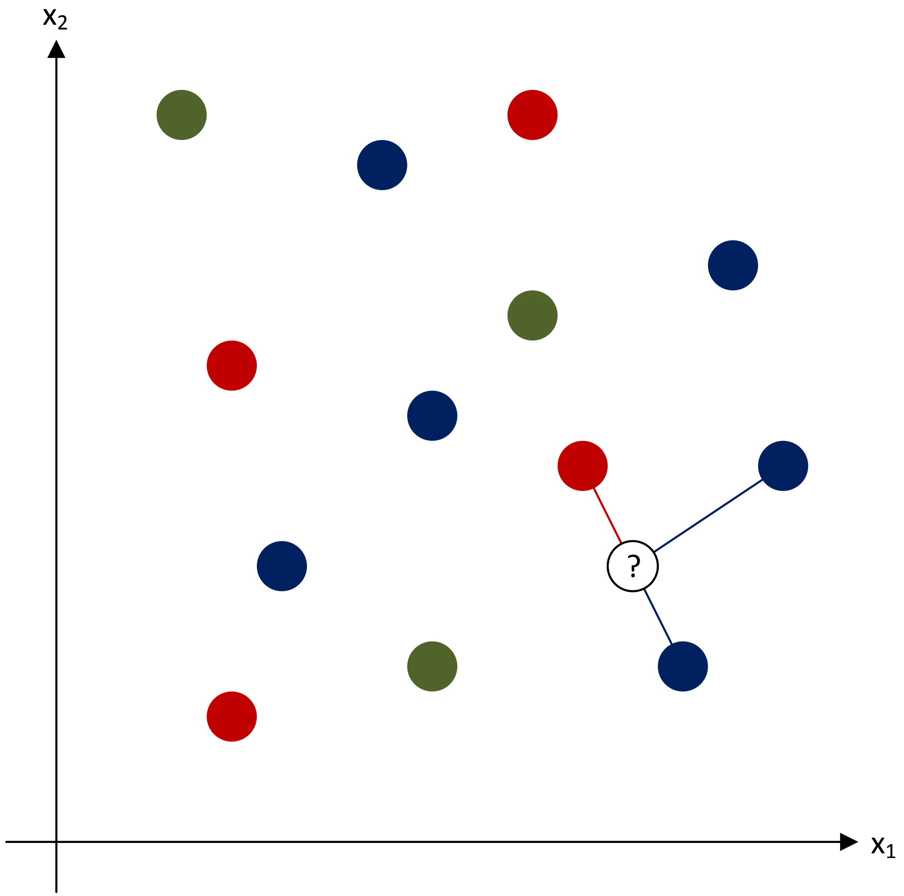
KNN | ❶ Pick a value for k, e.g., $k = 3$



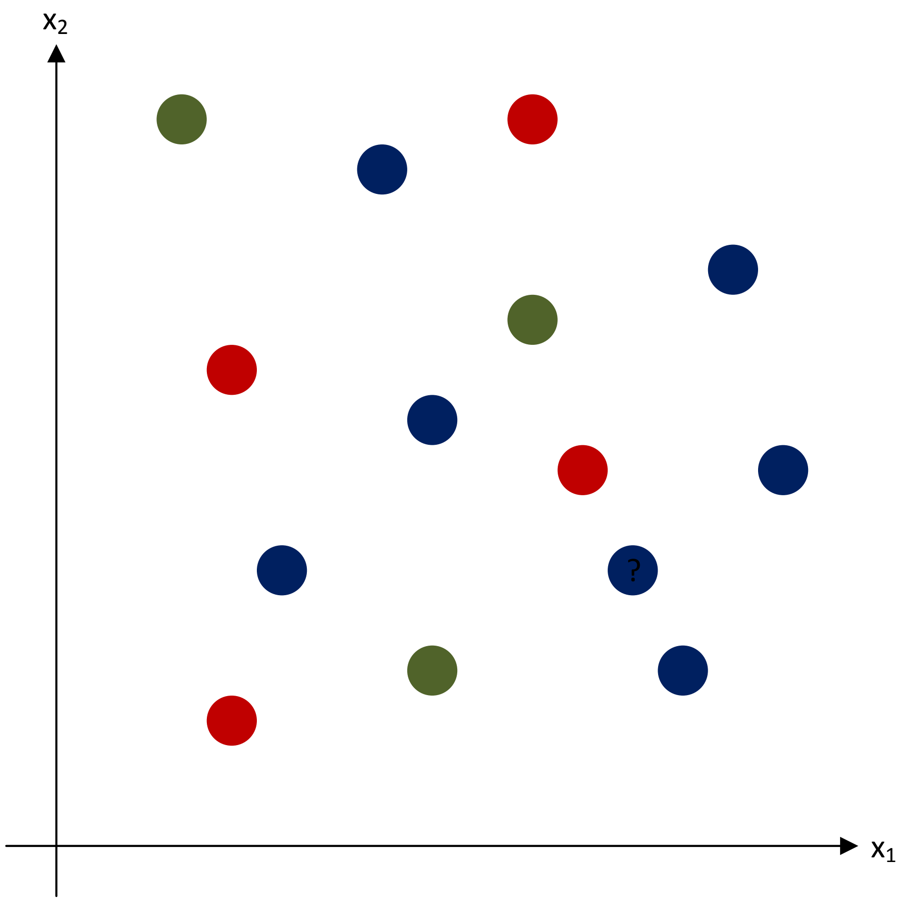
KNN | ❷ Calculate the distance to all other points; given those distances, pick the k closest points



KNN | ❸ Calculate the probabilities of each class label given those points: $\frac{1}{3}$ "red", $\frac{2}{3}$ "blue"

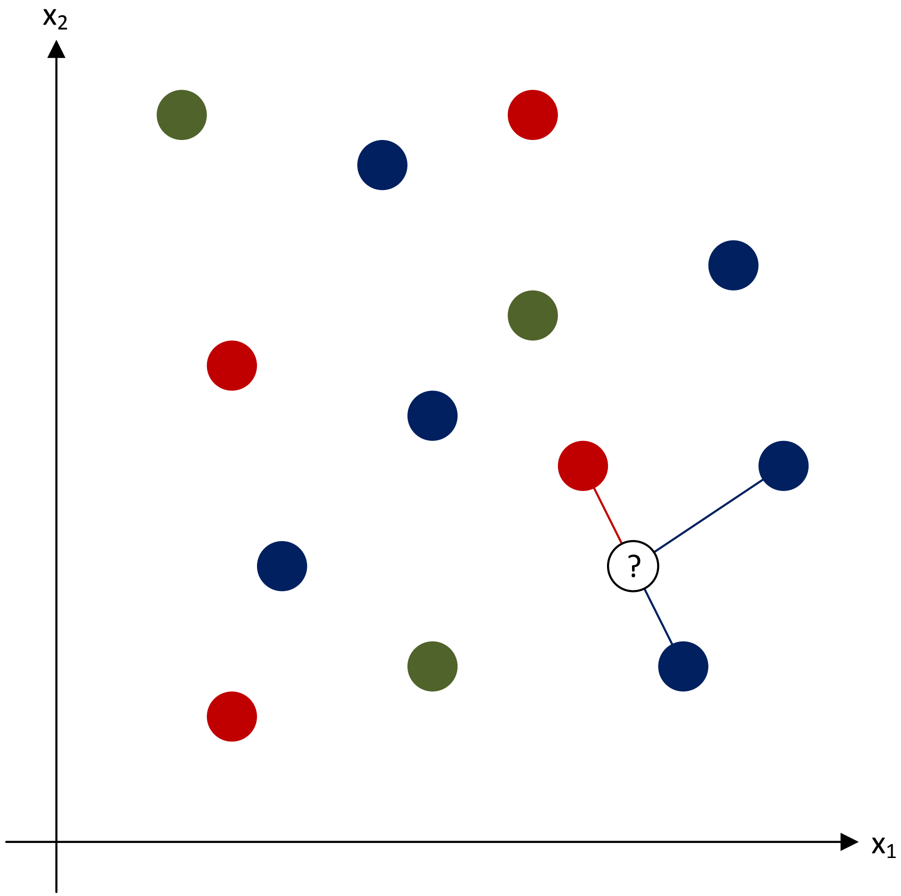


KNN | ❹ The original point is classified as the class label with the largest probability (“votes”): “blue”



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



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

- *sklearn* 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'*

Iris Dataset

Codealong – Part E

K-Nearest Neighbors

K-Nearest Neighbors (KNN)

High Dimensionality

KNN | 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

K-Nearest Neighbors (KNN)

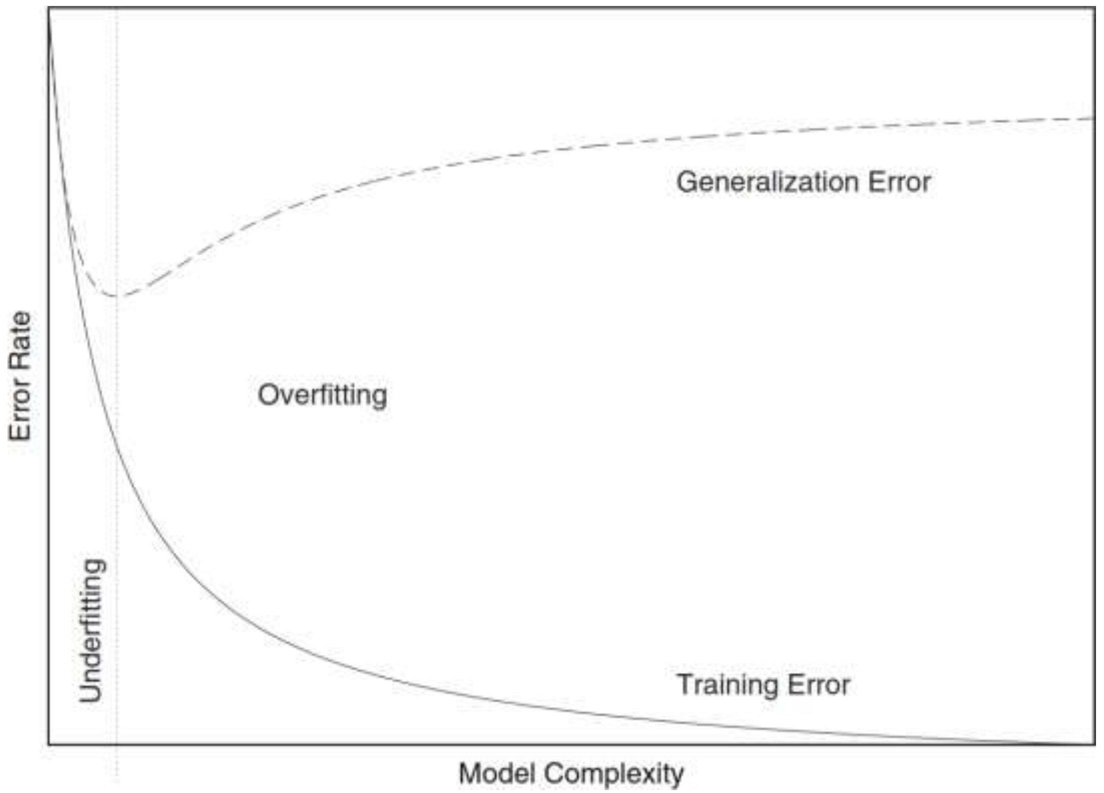
Codealong – Part F

What is the best value for k?

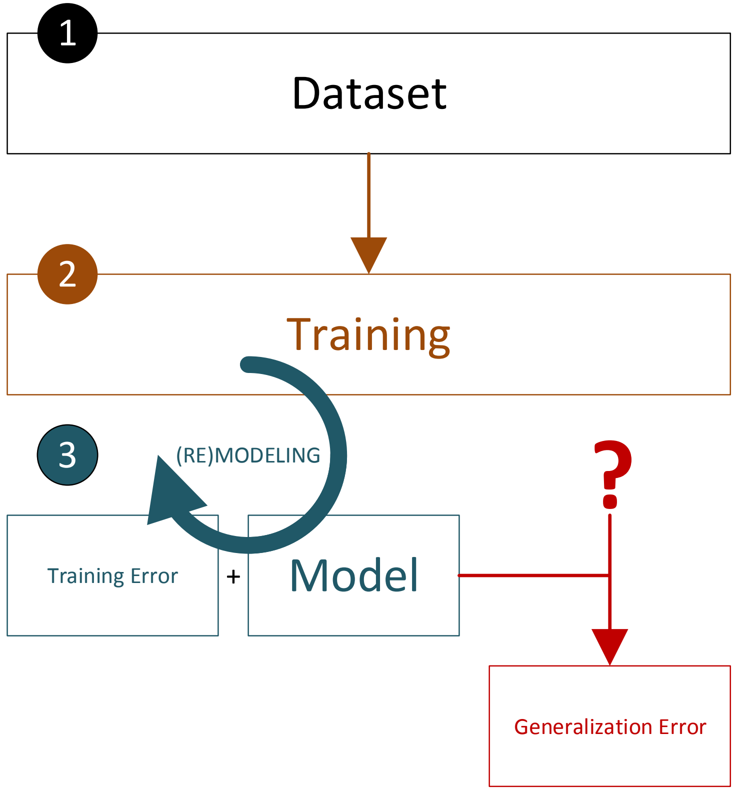
❻ Build a Model

Validation

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

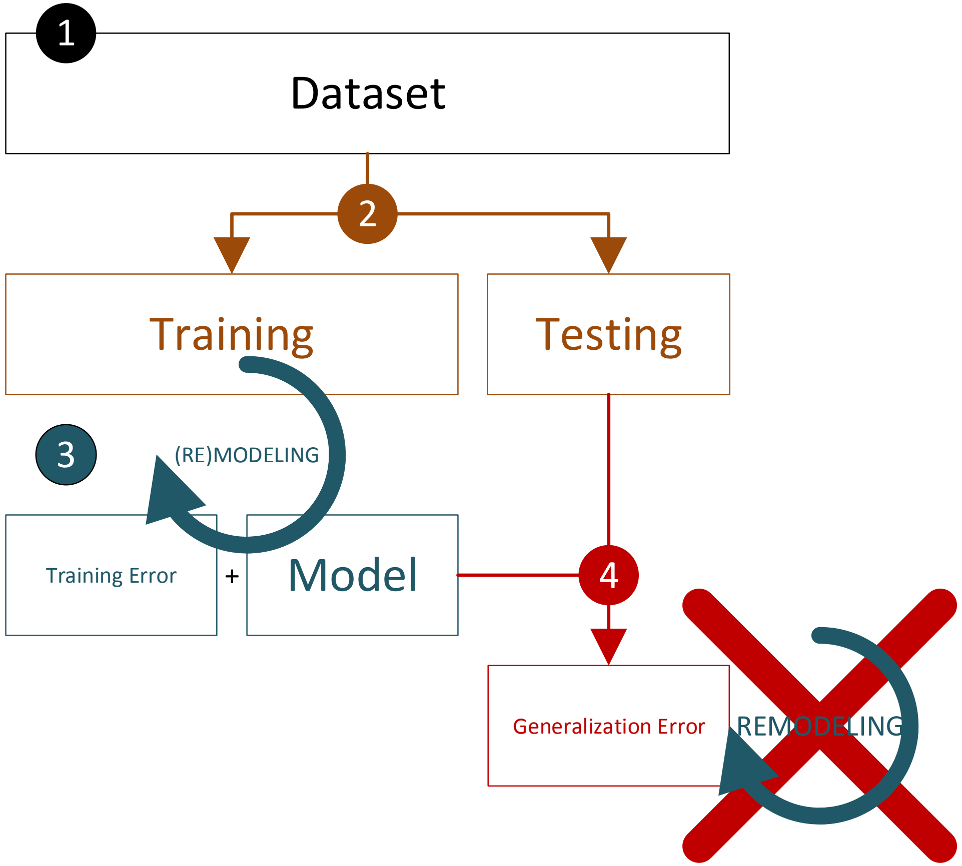


Source: Data Analysis with Open Source Tools



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



Iris Dataset

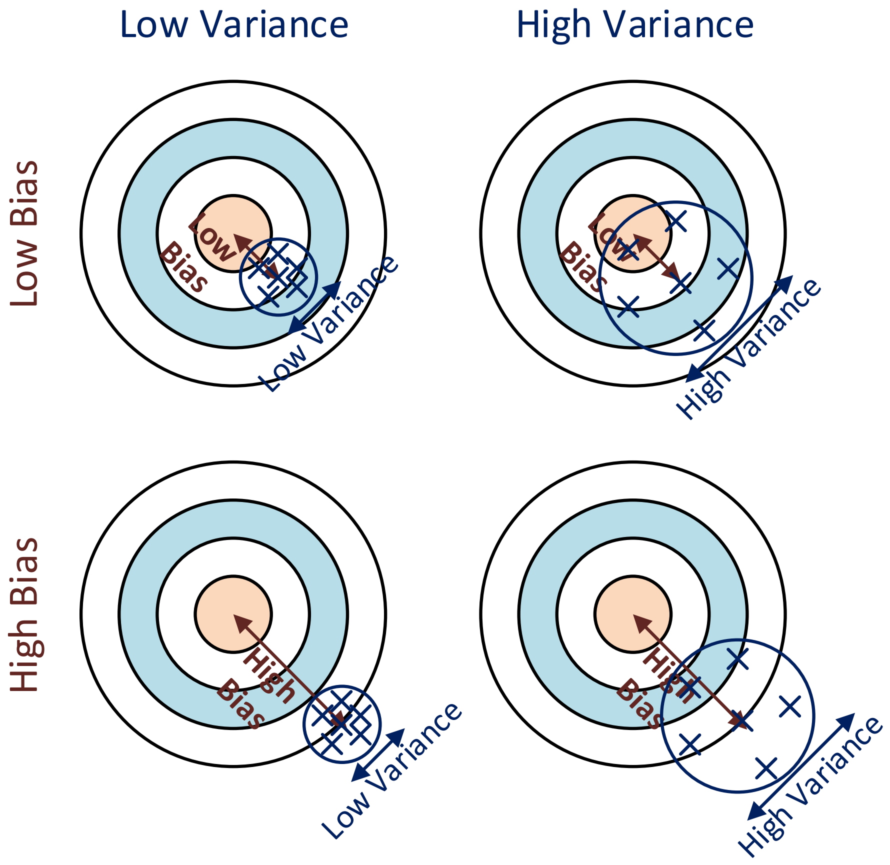
Codealong – Part G

Validation

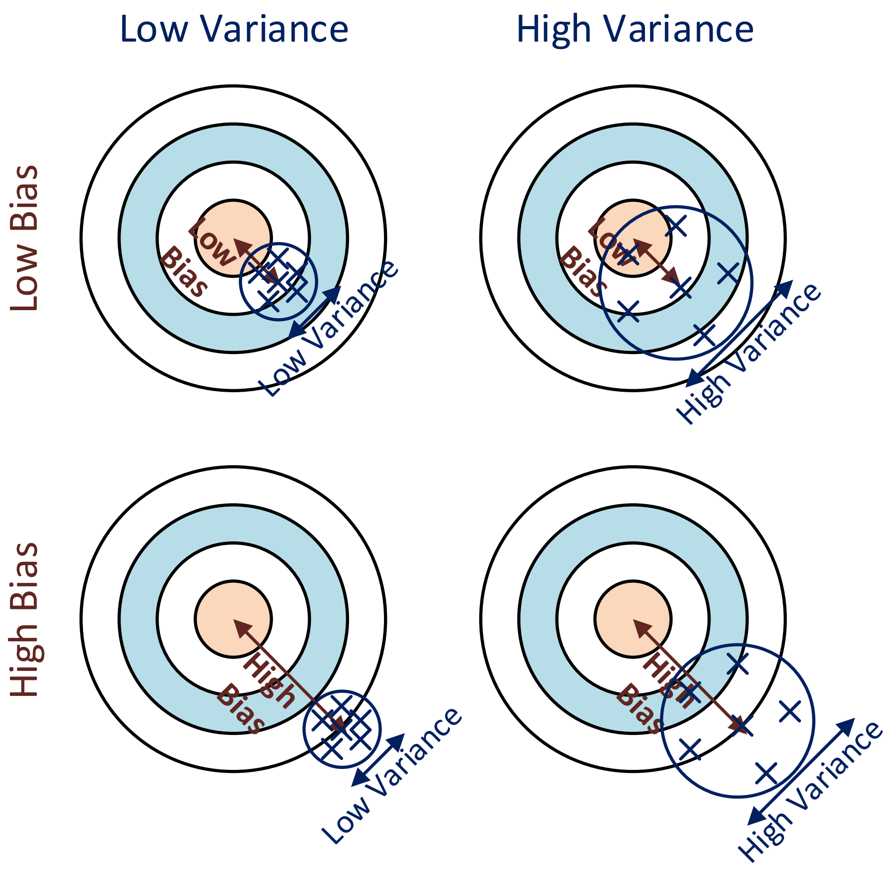
❻ Build a Model

Cross-Validation

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

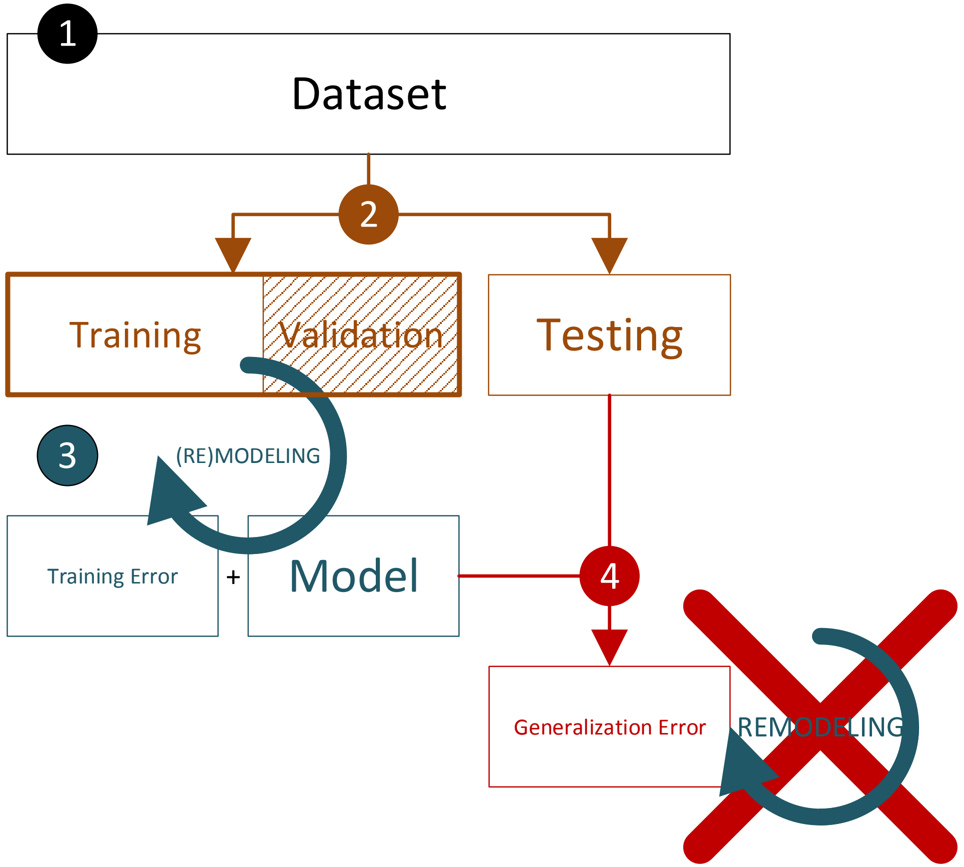


The generalization error has a bias component (systematic; non-random) and a variance component (idiosyncratic; random). Can we lower the bias error?



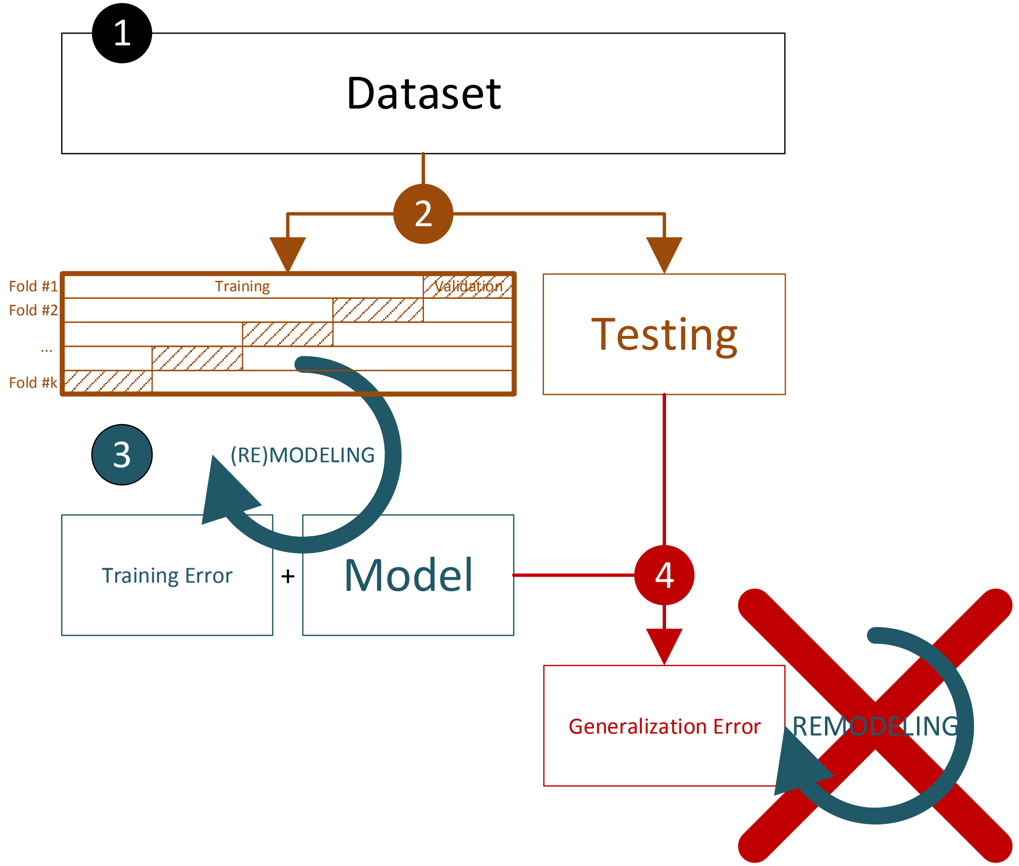
Cross-validation (CV) is a technique to lower the bias error

- 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)



Iris Dataset

Codealong – Part H

Cross-Validation

K-Nearest Neighbors (KNN)

Pros and Cons

KNN | Pros and cons

- Pros
  - Intuitive and simple to explain
  - Training phase is fast
  - Non-parametric (does not presume a “form” of the “decision boundary”)
  - Easily capture non-linearity

- Cons
  - Not interpretable
  - Prediction phase can be slow when n (number of observations) is large
  - Very sensitive to feature scaling; need to standardize the data
  - Sensitive to irrelevant features
  - Cannot be used if you have sparse data and feature space with dimension ≥ 4

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 (KNN) model using the *sklearn* library
- Evaluate and tune model by using metrics such as classification accuracy/error

Q & A

Before Next Class

Before Next Class

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

- Implement a linear model (`LinearRegression`) with *sklearn*
- Define the concept of coefficients
- Recall metrics for accuracy and misclassification

Next Class

Introduction to Logistic Regression

Learning Objectives

After the next lesson, you should be able to:

- Build a logistic regression classification model using *sklearn*
- Describe the logit and sigmoid functions, odds, and odds ratios as well as how they relate to logistic regression
- Evaluate a model using metrics such as classification accuracy/error

DS

Exit Ticket

Don't forget to fill out your exit ticket here