GENERAL ASSEMBLY

INTRO TO DATA SCIENCE

REVIEW

supervised	*making predictions*
unsupervised	*discovering patterns*

supervised	*labeled examples*
unsupervised	*no labeled examples*

	continuous	categorical
supervised	regression	classification
unsupervised	dimension reduction	clustering

QUESTION

HOW
DO YOU
REPRESENT
YOUR
DATA?

	continuous	***categorical***
	quantitative	*qualitative*

	continuous	*categorical*
color	*RGB-values*	*{red, blue}*
ratings	*1 – 10 rating*	*Good / Bad*

QUESTION

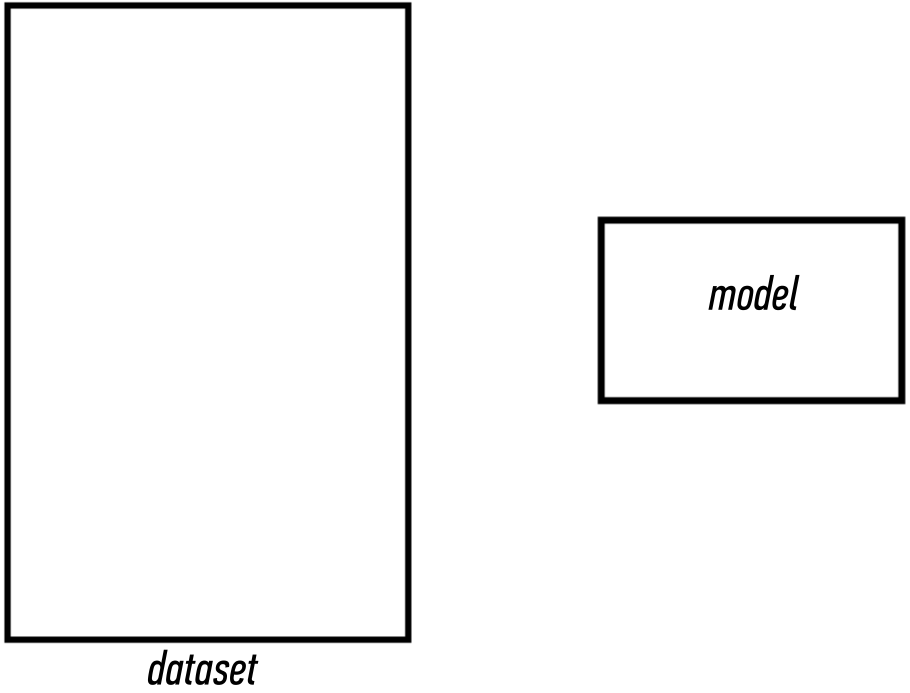
HOW DO YOU MEASURE OF QUALITY?

supervised	*test out your predictions*
unsupervised	*...*

supervised	*Accuracy, MSE, MAE, AUC*
unsupervised	*...*

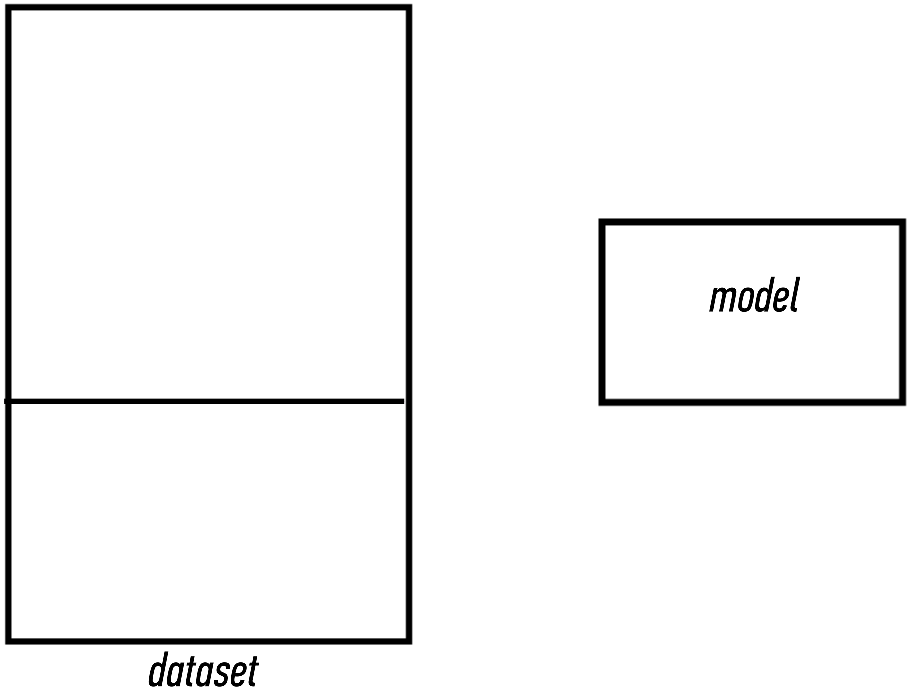
III. SUPERVISED LEARNING

Q: What steps does a classification problem require?



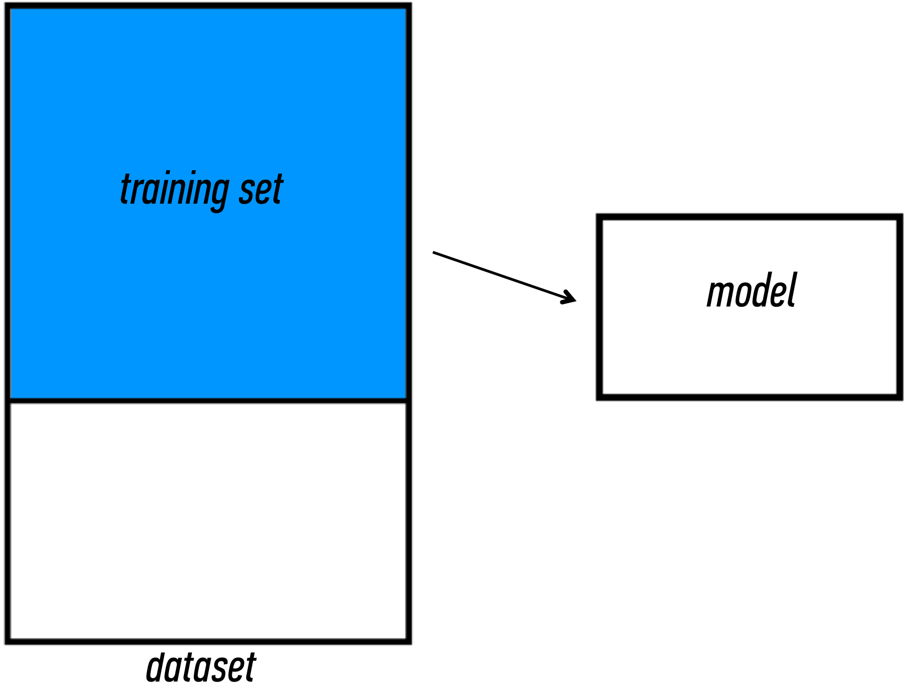
Q: What steps does a classification problem require?

1) *split dataset*



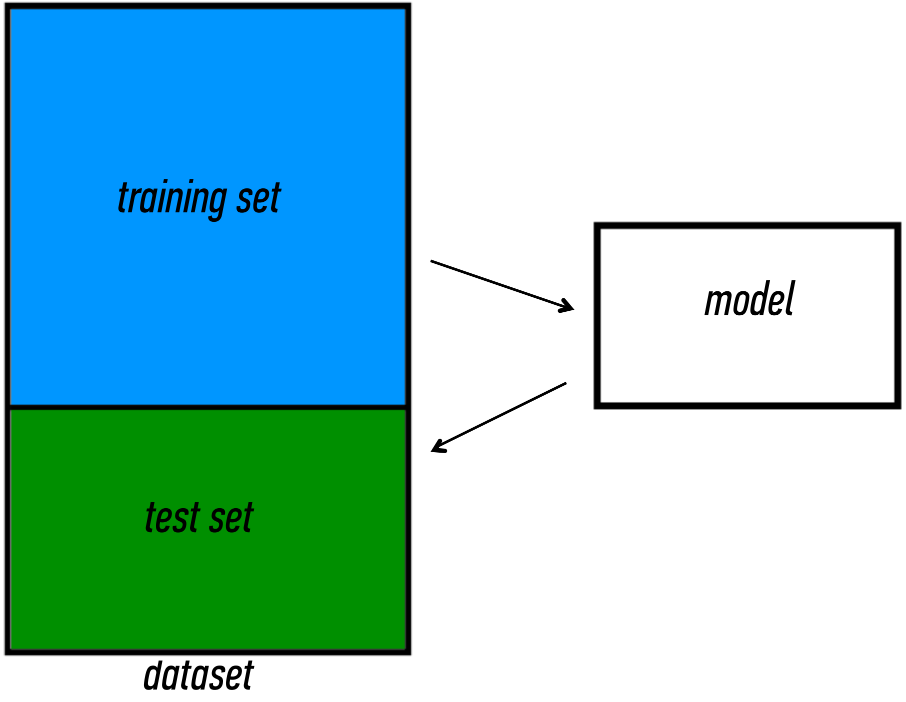
Q: What steps does a classification problem require?

1) *split dataset*
2) *train model*



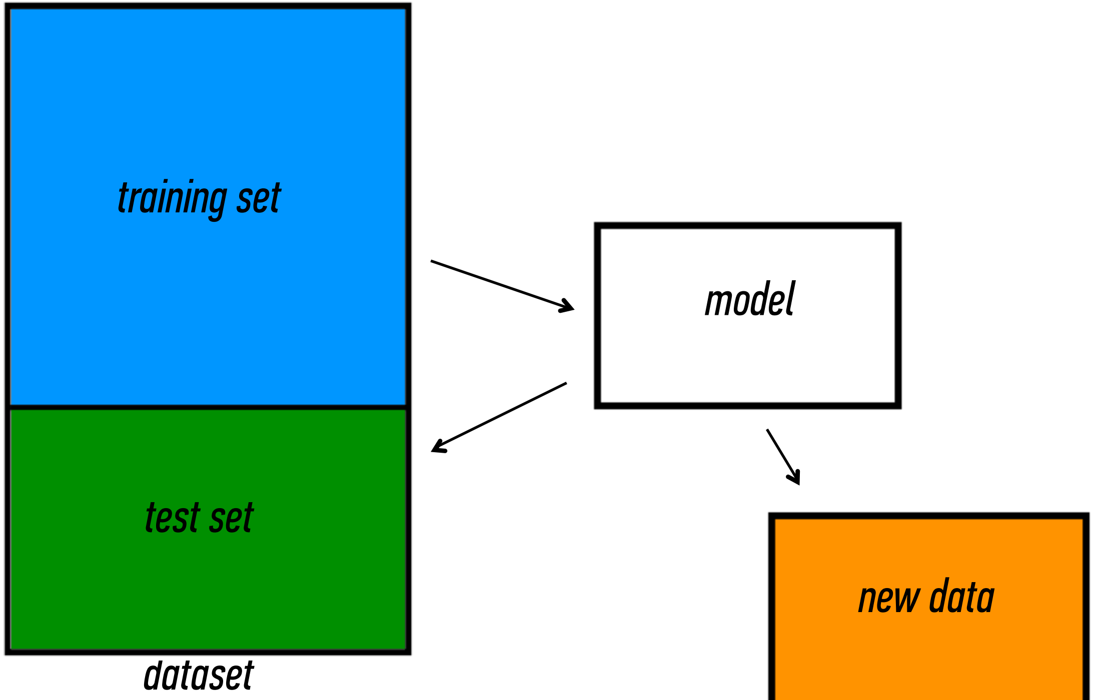
Q: What steps does a classification problem require?

1) split dataset
2) train model
3) test model



Q: What steps does a classification problem require?

1) *split dataset*
2) *train model*
3) *test model*
4) *make predictions*



III. LINEAR REGRESSION

	continuous	categorical
supervised	regression	classification
unsupervised	dimension reduction	clustering

Q: What is a **regression** *model?*

A: A functional relationship between input & response variables

The **simple linear regression** *model captures a linear relationship between a single input variable* x *and a response variable* y:

$$y = \alpha + \beta x + \varepsilon$$

Q: What do the terms in this model mean?

$$y = \alpha + \beta x + \varepsilon$$

A: y = **response variable** *(the one we want to predict)*

x = **input variable** *(the one we use to train the model)*

α = **intercept** *(where the line crosses the y-axis)*

β = **regression coefficient** *(the model "parameter")*

ε = **residual** *(the prediction error)*

OLS: $\min(\|y - x\beta\|^2)$

L1 regularization: $\min(\|y - x\beta\|^2 + \lambda\|\beta\|)$

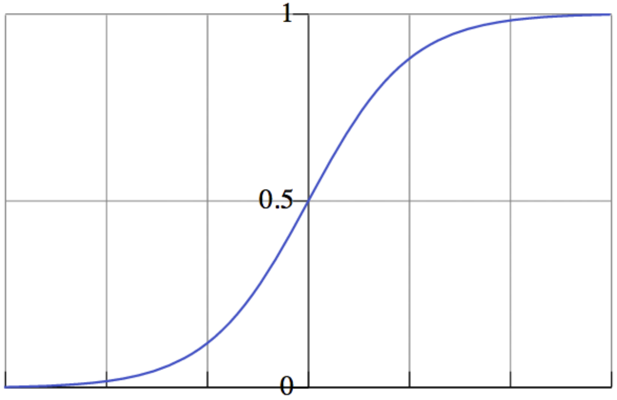
L2 regularization: $\min(\|y - x\beta\|^2 + \lambda\|\beta\|^2)$

IV. LOGISTIC REGRESSION

$$E(y|x) = \pi(x) = \frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}}$$

$$E(y|x) = \pi(x) = \frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}}$$

We've already seen what this looks like:



The **logit function** *is an important transformation of the logistic function. Notice that it returns the linear model!*

$$g(x) = ln(\frac{\pi(x)}{1-\pi(x)}) = \alpha + \beta x$$

The **logit function** *is an important transformation of the logistic function. Notice that it returns the linear model!*

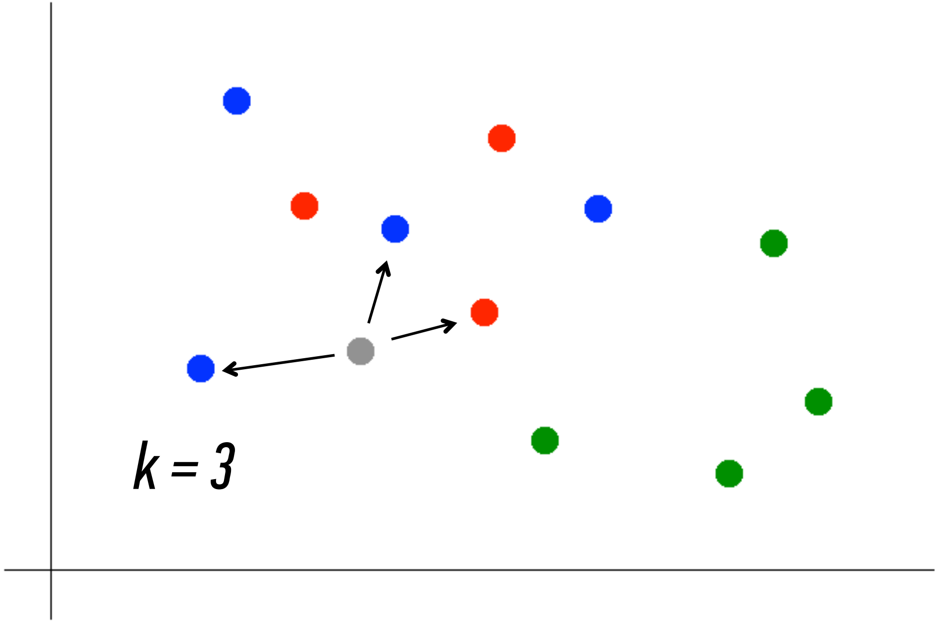
$$g(x) = ln(\frac{\pi(x)}{1-\pi(x)}) = \alpha + \beta x$$

The logit function is also called the **log-odds function***.*

V. KNN CLASSIFICATION

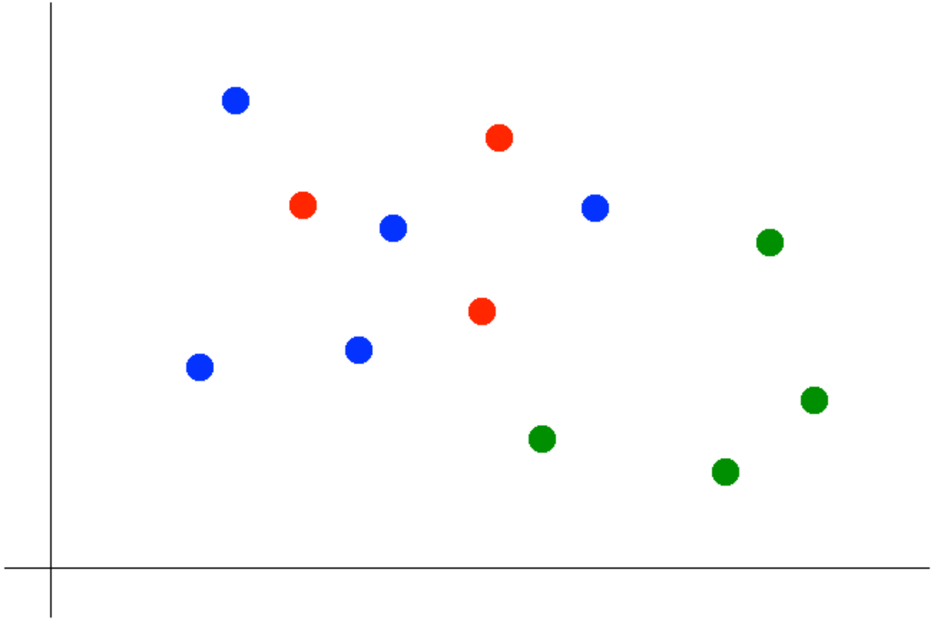
Suppose we want to predict the color of the grey dot.

1) *Pick a value for k.*
2) *Find colors of k nearest neighbors.*



Suppose we want to predict the color of the grey dot.

1) *Pick a value for k.*
2) *Find colors of k nearest neighbors.*
3) *Assign the most common color to the grey dot.*



VI. NAÏVE BAYES

Suppose we have a dataset with features $x_1, \ldots, x_n$ and a class label c. What can we say about classification using Bayes' theorem?

$$P(\text{class } C \mid \{x_i\}) = \frac{P(\{x_i\} \mid \text{class } C) \cdot P(\text{class } C)}{P(\{x_i\})}$$

Bayes' theorem can help us to determine the probability of a record belonging to a class, given the data we observe.

source: Data Analysis with Open Source Tools, by Philipp K. Janert. O'Reilly Media, 2011.

This term is the **likelihood function**. *It represents the joint probability of observing features* $\{x_i\}$ *given that that record belongs to class* C.

$$P(\text{class } C \mid \{x_i\}) = \frac{P(\{x_i\} \mid \text{class } C) \cdot P(\text{class } C)}{P(\{x_i\})}$$

This term is the **prior probability** *of* C*. It represents the probability of a record belonging to class* C *before the data is taken into account.*

$$P(\text{class } C \mid \{x_i\}) = \frac{P(\{x_i\} \mid \text{class } C) \cdot P(\text{class } C)}{P(\{x_i\})}$$

This term is the **normalization constant.** *It doesn't depend on* C*, and is generally ignored until the end of the computation.*

$$P(\text{class } C \mid \{x_i\}) = \frac{P(\{x_i\} \mid \text{class } C) \cdot P(\text{class } C)}{P(\{x_i\})}$$

This term is the **posterior probability** *of* C*. It represents the probability of a record belonging to class* C *after the data is taken into account.*

$$P(\text{class } C \mid \{x_i\}) = \frac{P(\{x_i\} \mid \text{class } C) \cdot P(\text{class } C)}{P(\{x_i\})}$$

This term is the **posterior probability** *of* C*. It represents the probability of a record belonging to class* C *after the data is taken into account.*

$$P(\text{class } C \mid \{x_i\}) = \frac{P(\{x_i\} \mid \text{class } C) \cdot P(\text{class } C)}{P(\{x_i\})}$$

The goal of any Bayesian computation is to find ("learn") the posterior distribution of a particular variable.

The idea of Bayesian inference, then, is to **update** *our beliefs about the distribution of* C *using the data ("evidence") at our disposal.*

$$P(\text{class } C \mid \{x_i\}) = \frac{P(\{x_i\} \mid \text{class } C) \cdot P(\text{class } C)}{P(\{x_i\})}$$

Then we can use the posterior for prediction.

	continuous	categorical
supervised	regression	classification
unsupervised	dimension reduction	clustering

V. COMPARISON

KNN

	KNN
linear	N

	KNN
linear	N
scalability	+/-

	KNN
linear	N
scalability	+/-
interpretation	-

	KNN
linear	N
scalability	+/-
interpretation	-
configuration	+

	KNN
linear	N
scalability	+/-
interpretation	-
configuration	+
specification	-

	KNN
linear	N
scalability	+/-
interpretation	-
configuration	+
feature-select	-
overfitting	< K

	KNN	Logistic
linear	N	Y
scalability	+/-	+
interpretation	-	+
configuration	+	+
feature-select	-	+
overfitting	< K	L1 / L2

	KNN	Logistic	NB
linear	N	Y	Y
scalability	+/-	+	+
interpretation	-	+	+
configuration	+	+	+
feature-select	-	+	+
overfitting	< K	L1 / L2	Prior

	KNN	Logistic	NB	RF
linear	N	Y	Y	N
scalability	+/-	+	+	-
interpretation	-	+	+	-
configuration	+	+	+	+
feature-select	-	+	+	+
overfitting	< K	L1 / L2	Prior	n tree

	KNN	Logistic	NB	RF	SVM
linear	N	Y	Y	N	Y/N
scalability	+/-	+	+	-	-
interpretation	-	+	+	-	-
configuration	+	+	+	+	-
feature-select	-	+	+	+	-
overfitting	< K	L1 / L2	Prior	n tree	C-cost