GENERAL ASSEMBLY

# INTRO TO DATA SCIENCE:

## KNN CLASSIFICATION

# What's big data?

The practical viewpoint:

1. $O(n^2)$ algorithm feasible: small data
2. Fits on one machine: medium data
3. Doesn't fit on one machine: big data

*source: http://people.cs.umass.edu/~mcgregor/stocworkshop/langford.pdf*

# I. WHAT IS MACHINE LEARNING?
# II. CLASSIFICATION PROBLEMS
# III. BUILDING EFFECTIVE CLASSIFIERS
# IV. THE KNN CLASSIFICATION MODEL

# EXERCISES:
# IV. LAB: KNN CLASSIFICATION IN PYTHON
# V. BONUS LAB: VISUALIZATION WITH MATPLOTLIB (IF TIME ALLOWS)

# I. WHAT IS MACHINE LEARNING?

## WHAT IS MACHINE LEARNING?

"A field of study that gives computers the ability to learn without being explicitly programmed." (1959)

Arthur Samuel, AI pioneer
Source: Stanford

## WHAT IS MACHINE LEARNING?

"A computer program is said to learn from experience *E* with respect to some set of tasks *T* and performance measure *P*, if its performance at tasks in *T*, as measured by *P*, improves with experience *E*". (1989)

Tom Mitchell, Professor, CMU
(Source: CMU)

## WHAT IS MACHINE LEARNING?

"A computer program is said to learn from experience *E* with respect to some set of tasks *T* and performance measure *P*, if its performance at tasks in *T*, as measured by *P*, improves with experience *E*".

A person is said to learn from a college course E with respect to some set of readings and midterms T and grades P, if its performance at tasks in T, as measured by P, improves with E.

## WHAT IS MACHINE LEARNING?

from Wikipedia:

“Machine learning, a branch of artificial intelligence, is about the construction and study of systems that can learn from data.”

source: http://en.wikipedia.org/wiki/Machine_learning

## WHAT IS MACHINE LEARNING?

from Wikipedia:

“Machine learning, a branch of artificial intelligence, is about the construction and study of systems that can learn from data.”

“The core of machine learning deals with representation and generalization…”

source: http://en.wikipedia.org/wiki/Machine_learning

## WHAT IS MACHINE LEARNING?

from Wikipedia:

“Machine learning, a branch of artificial intelligence, is about the construction and study of systems that can learn from data.”

“The core of machine learning deals with representation and generalization…”

- representation – extracting structure from data

source: http://en.wikipedia.org/wiki/Machine_learning

# WHAT IS MACHINE LEARNING?

from Wikipedia:

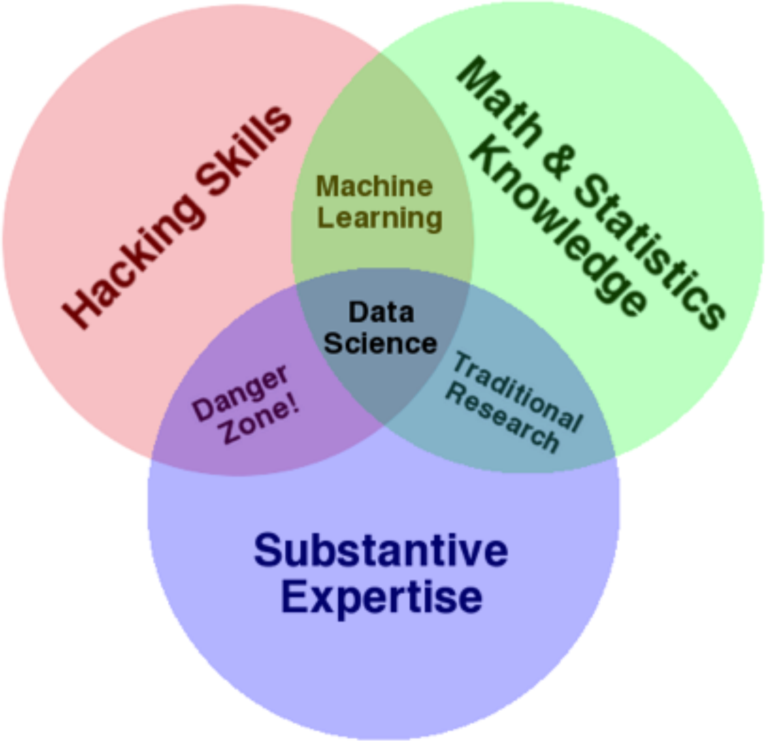
“Machine learning, a branch of artificial intelligence, is about the construction and study of systems that can learn from data.”

“The core of machine learning deals with representation and generalization…”

- representation – extracting structure from data
- generalization – making predictions from data

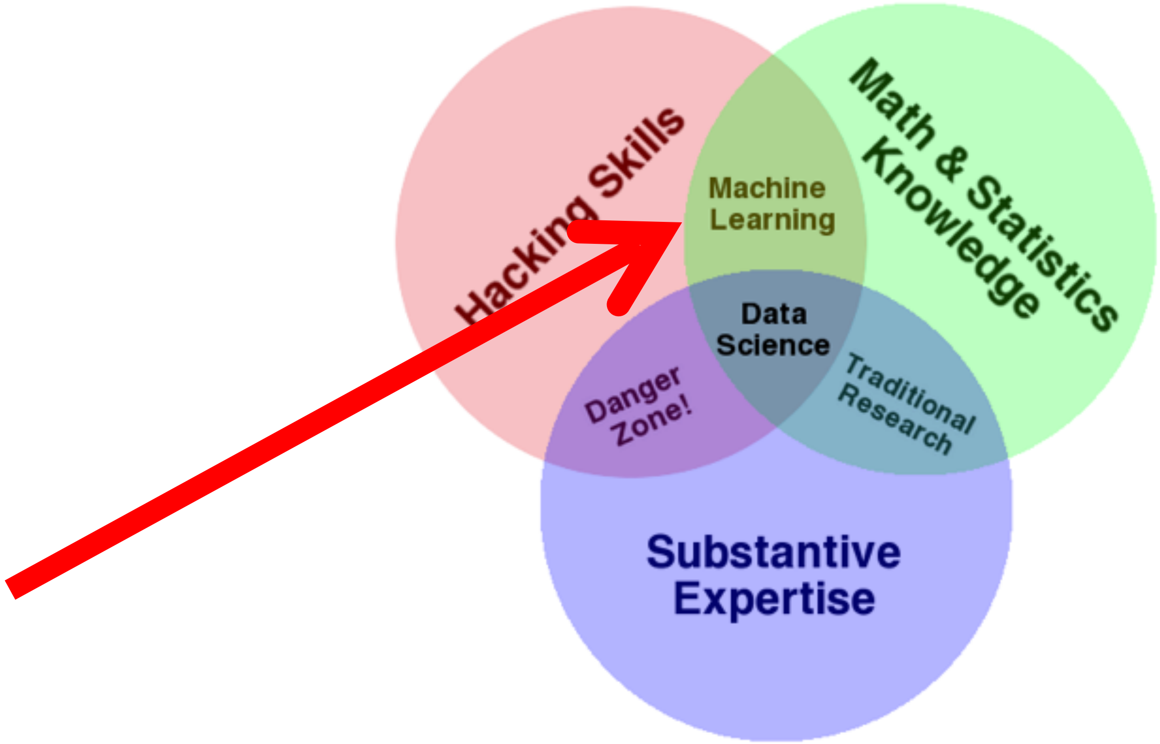
source: http://en.wikipedia.org/wiki/Machine_learning

## REMEMBER THIS?



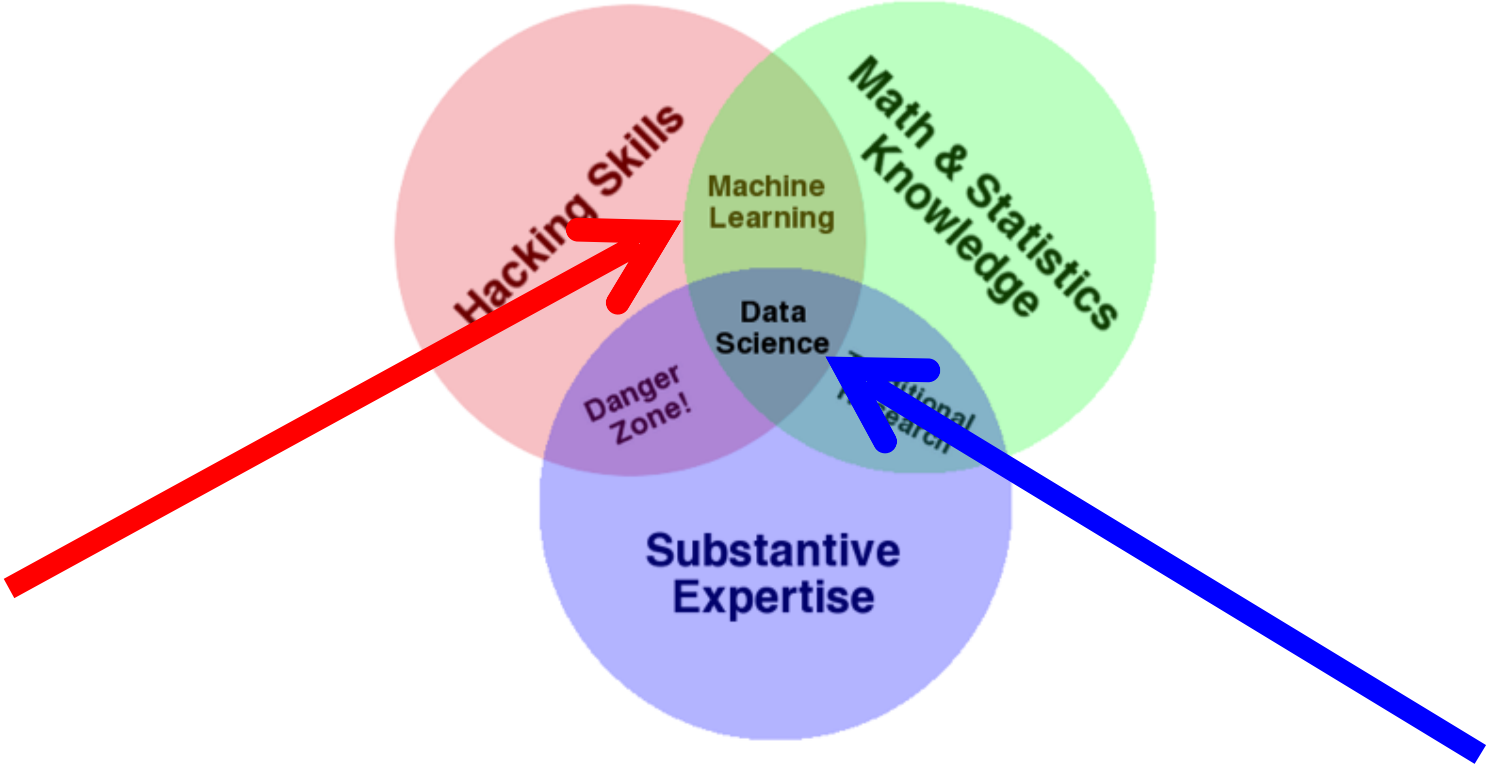

*source: http://www.dataists.com/2010/09/the-data-science-venn-diagram/*

## WE ARE NOW HERE



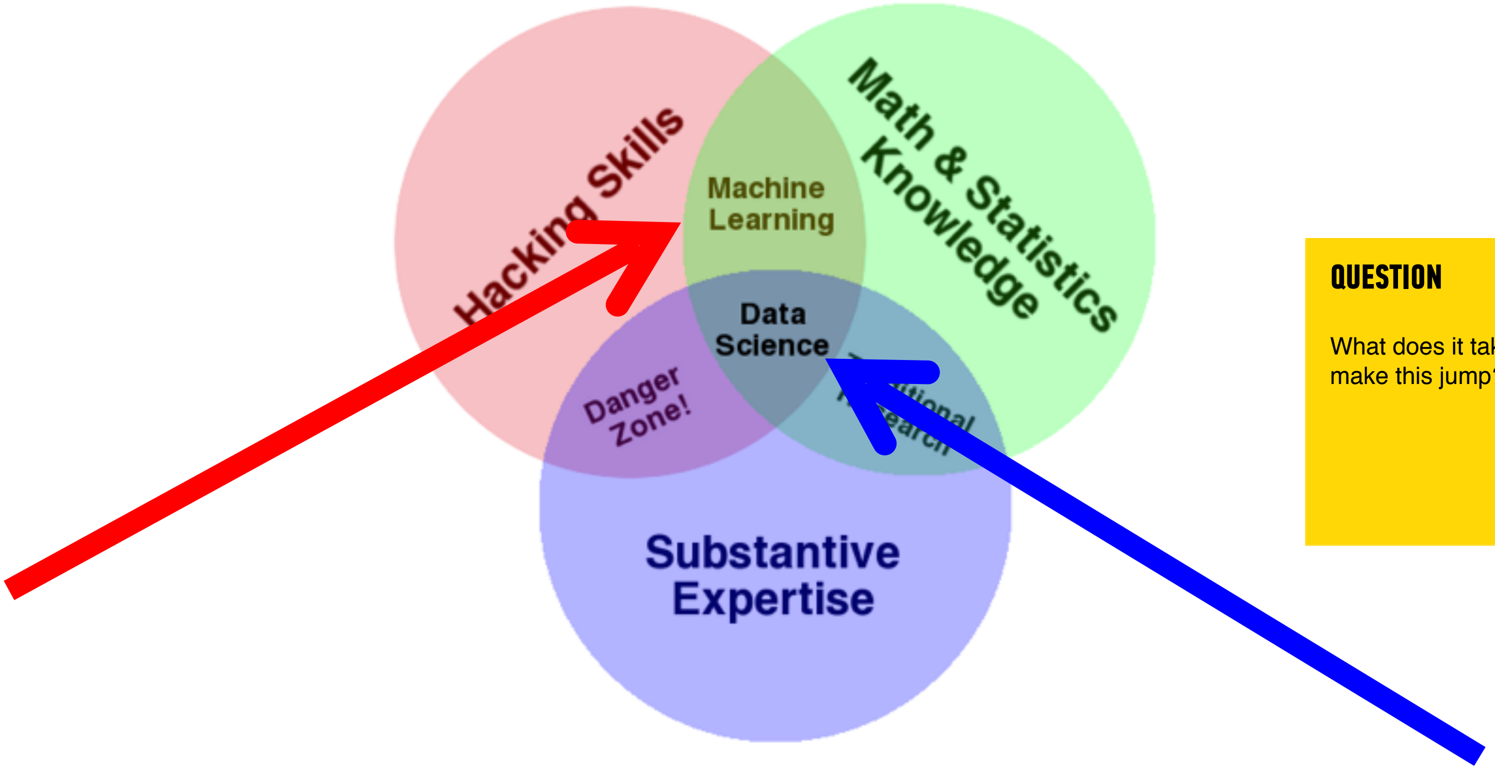

*source: http://www.dataists.com/2010/09/the-data-science-venn-diagram/*

# WE WANT TO GO HERE




*source: http://www.dataists.com/2010/09/the-data-science-venn-diagram/*

## WE WANT TO GO HERE

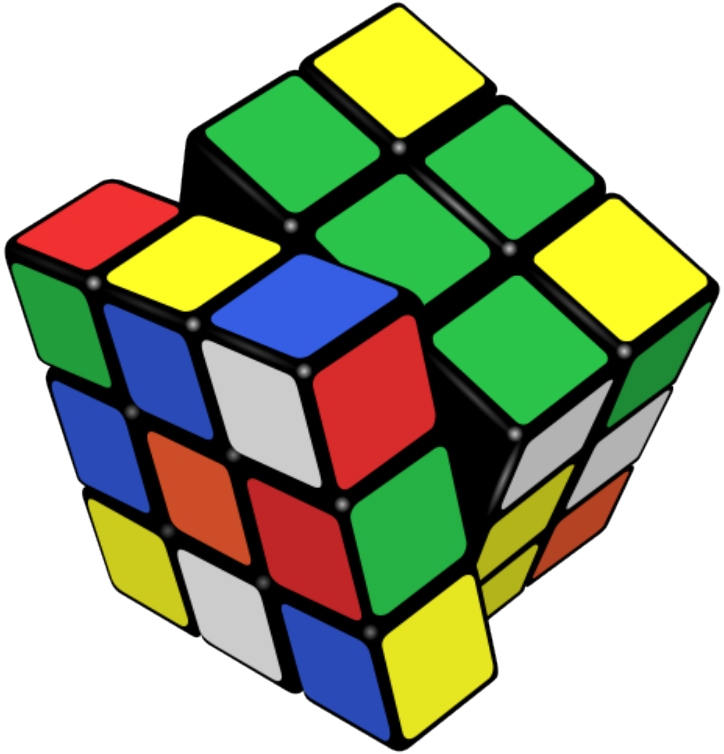



**QUESTION**

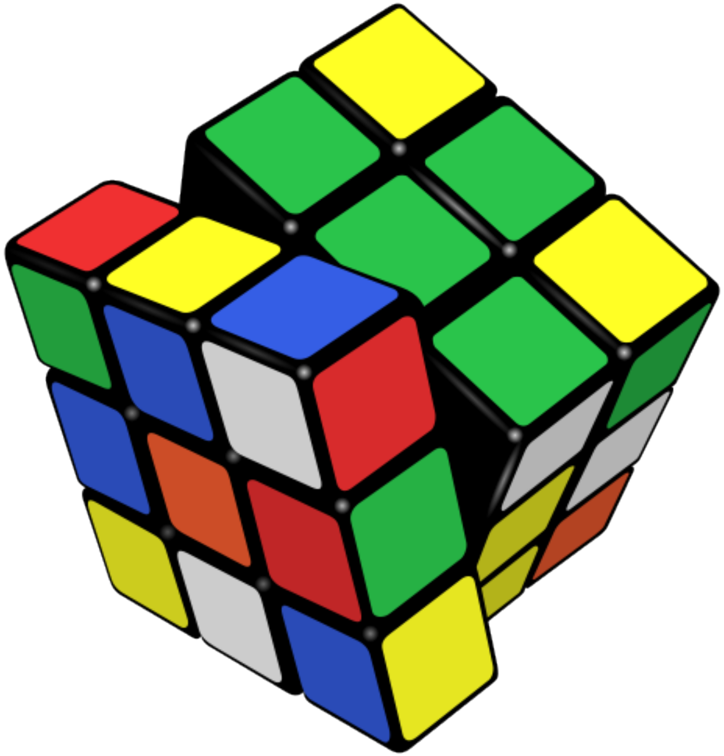
What does it take to make this jump?

*source: http://www.dataists.com/2010/09/the-data-science-venn-diagram/*

## ANSWER: PROBLEM SOLVING!



# ANSWER: PROBLEM SOLVING!



**NOTE**

Implementing solutions to ML problems is the focus of this course!

# THE STRUCTURE OF MACHINE LEARNING PROBLEMS

## TYPES OF LEARNING PROBLEMS

| | |
|---|---|
| ***supervised*** | *making predictions* |
| ***unsupervised*** | *extracting structure* |

## REMEMBER WHAT WE SAID BEFORE?

*generalization*

| | |
|---|---|
| ***supervised*** | *making predictions* |
| ***unsupervised*** | *extracting structure* |

*representation*

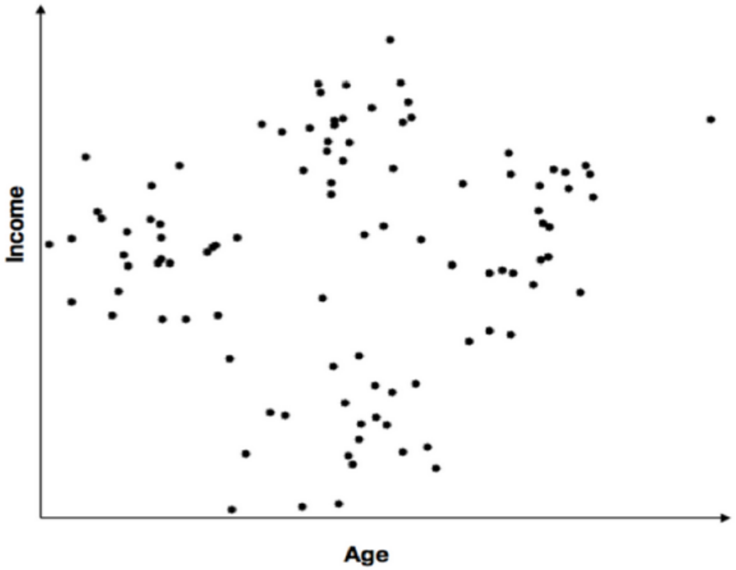
# TYPES OF LEARNING PROBLEMS - SUPERVISED EXAMPLE

Supervised Learning - Can we create a function that predicts a value based on labeled training data?

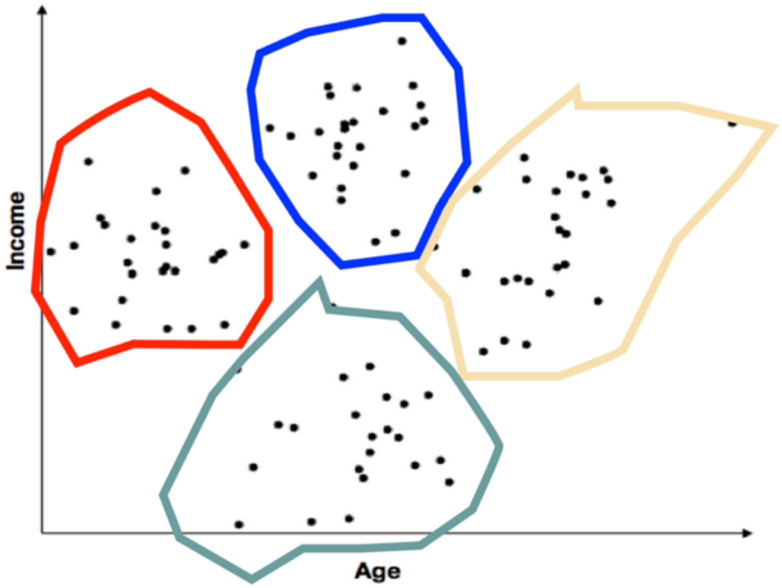
Regression example: Alan is 30 years old and can eat *four donuts an hour*. Betty is 60 years old, and can eat *two donuts an hour*. Cameron is 15 years old--how many donuts an hour eaten would be a good guess? This prediction is a regression model.

Classification example: Let' s use the same data above. What is the probability that Cameron will eat eight donuts? Here, we have an answer and am now calculating the probability that an outcome has occurred.

Unsupervised Learning - Can we find structure to unlabeled data?



Unsupervised Learning - Can we find structure to unlabeled data?



## TYPES OF DATA

| | *continuous* | *categorical* |
|---|---|---|
| | *quantitative* | *qualitative* |

## TYPES OF DATA

| | |
|---|---|
| *continuous* | *categorical* |
| *quantitative* | *qualitative* |

**NOTE**

The space where data live is called the *feature space.*

Each point in this space is called a *record*.

## TYPES OF ML SOLUTIONS

| | continuous | categorical |
|---|---|---|
| supervised | regression | classification |
| unsupervised | dimension reduction | clustering |

## TYPES OF ML SOLUTIONS

| | *continuous* | *categorical* |
|---|---|---|
| *supervised* | *regression* | *classification* |
| *unsupervised* | *dimension reduction* | *clustering* |

**NOTE**

We will implement solutions using *models* and *algorithms.*

Each will fall into one of these four buckets.

QUESTION

# WHAT
# IS THE
# GOAL
# OF
# MACHINE LEARNING?

# GOALS OF ML

| | |
|---|---|
| *supervised* | *making predictions* |
| *unsupervised* | *extracting structure* |

**ANSWER**

The goal is determined by the type of problem.

QUESTION

# HOW DO YOU DETERMINE THE RIGHT APPROACH?

## APPROACHES TO ML PROBLEMS

| | continuous | categorical |
|---|---|---|
| supervised | regression | classification |
| unsupervised | dimension reduction | clustering |

**ANSWER**

The right approach is determined by the desired solution.

## APPROACHES TO ML PROBLEMS

| | continuous | categorical |
|---|---|---|
| supervised | regression | classification |
| unsupervised | dimension reduction | clustering |

**ANSWER**

The
is d
des

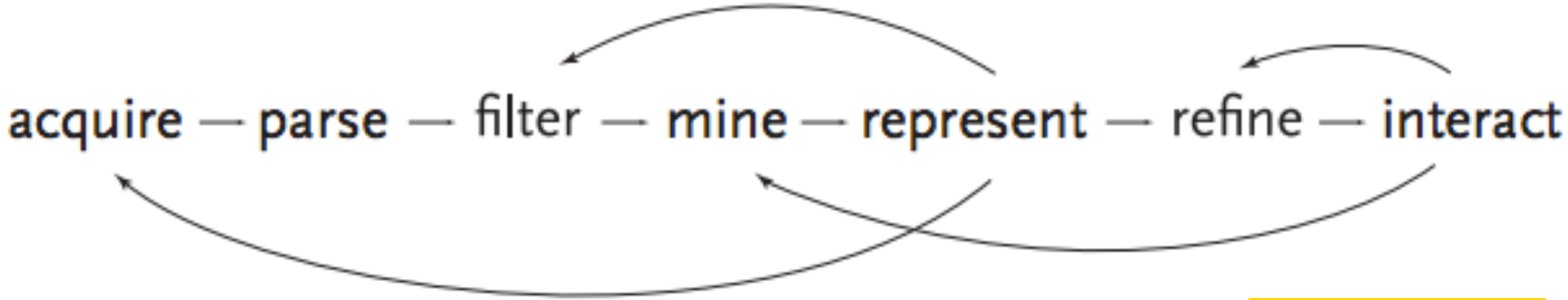
**NOTE**

All of this depends on your data!

QUESTION

# WHAT
DO YOU
# DO
WITH YOUR
# RESULTS?

## THE DATA SCIENCE WORKFLOW

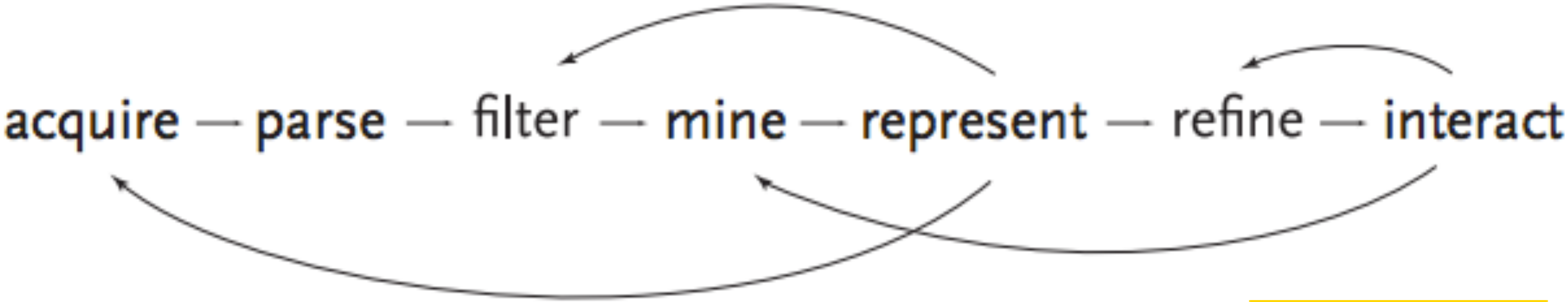



### ANSWER

Interpret them and react accordingly.

source: http://benfry.com/phd/dissertation-110323c.pdf

# THE DATA SCIENCE WORKFLOW




**ANSWER**

Int
re

**NOTE**

This also relies on your problem solving skills!

*source: http://benfry.com/phd/dissertation-110323c.pdf*

# II. CLASSIFICATION PROBLEMS

## CLASSIFICATION PROBLEMS

| | *continuous* | *categorical* |
|---|---|---|
| ***supervised*** | ??? | ??? |
| ***unsupervised*** | ??? | ??? |

## CLASSIFICATION PROBLEMS

| | continuous | categorical |
|---|---|---|
| supervised | regression | classification |
| unsupervised | dimension reduction | clustering |

# *Here's (part of) an example dataset:*

**Fisher's *Iris* Data**

| Sepal length | Sepal width | Petal length | Petal width | Species |
|---|---|---|---|---|
| 5.1 | 3.5 | 1.4 | 0.2 | *I. setosa* |
| 4.9 | 3.0 | 1.4 | 0.2 | *I. setosa* |
| 4.7 | 3.2 | 1.3 | 0.2 | *I. setosa* |
| 4.6 | 3.1 | 1.5 | 0.2 | *I. setosa* |
| 5.0 | 3.6 | 1.4 | 0.2 | *I. setosa* |
| 5.4 | 3.9 | 1.7 | 0.4 | *I. setosa* |
| 4.6 | 3.4 | 1.4 | 0.3 | *I. setosa* |
| 5.0 | 3.4 | 1.5 | 0.2 | *I. setosa* |

# *Here's (part of) an example dataset:*

**Fisher's *Iris* Data**

| Sepal length | Sepal width | Petal length | Petal width | Species |
|---|---|---|---|---|
| 5.1 | 3.5 | 1.4 | 0.2 | *I. setosa* |
| 4.9 | 3.0 | 1.4 | 0.2 | *I. setosa* |
| 4.7 | 3.2 | 1.3 | 0.2 | *I. setosa* |
| 4.6 | 3.1 | 1.5 | 0.2 | *I. setosa* |
| 5.0 | 3.6 | 1.4 | 0.2 | *I. setosa* |
| 5.4 | 3.9 | 1.7 | 0.4 | *I. setosa* |
| 4.6 | 3.4 | 1.4 | 0.3 | *I. setosa* |
| 5.0 | 3.4 | 1.5 | 0.2 | *I. setosa* |

*independent variables*

*Here's (part of) an example dataset:*

**Fisher's *Iris* Data**

| Sepal length | Sepal width | Petal length | Petal width | Species |
|---|---|---|---|---|
| 5.1 | 3.5 | 1.4 | 0.2 | *I. setosa* |
| 4.9 | 3.0 | 1.4 | 0.2 | *I. setosa* |
| 4.7 | 3.2 | 1.3 | 0.2 | *I. setosa* |
| 4.6 | 3.1 | 1.5 | 0.2 | *I. setosa* |
| 5.0 | 3.6 | 1.4 | 0.2 | *I. setosa* |
| 5.4 | 3.9 | 1.7 | 0.4 | *I. setosa* |
| 4.6 | 3.4 | 1.4 | 0.3 | *I. setosa* |
| 5.0 | 3.4 | 1.5 | 0.2 | *I. setosa* |

*independent variables*

*class labels (qualitative)*

*Q: What does "supervised" mean?*

## CLASSIFICATION PROBLEMS

*Q: What does "supervised" mean?*

*A: We know the labels.*

```
Welcome to R! Thu Feb 28 13:07:25 2013
> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
 Median :5.800   Median :3.000   Median :4.350   Median :1.300  
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
       Species  
 setosa    :50  
 versicolor:50  
 virginica :50  
```

*Q: How does a classification problem work?*

*Q: How does a classification problem work?*

*A: Data in, predicted labels out.*

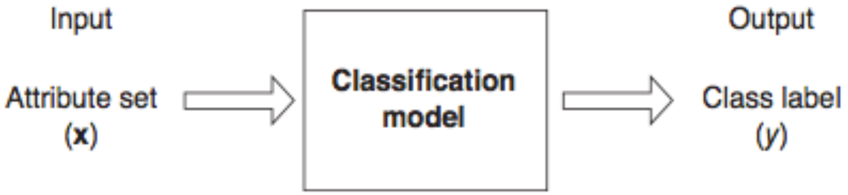
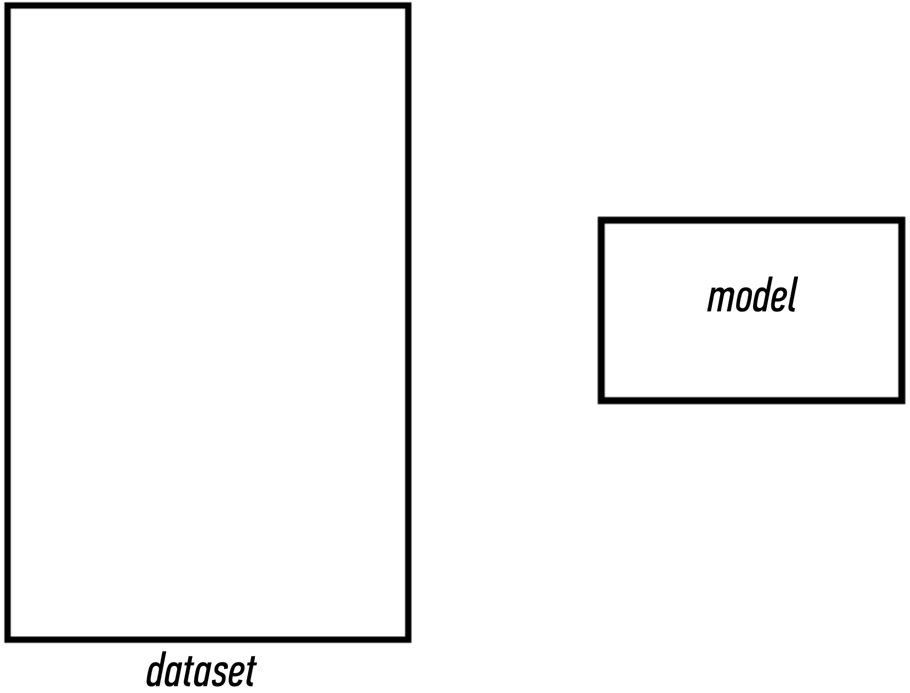



**Figure 4.2.** Classification as the task of mapping an input attribute set $\mathbf{x}$ into its class label $y$.

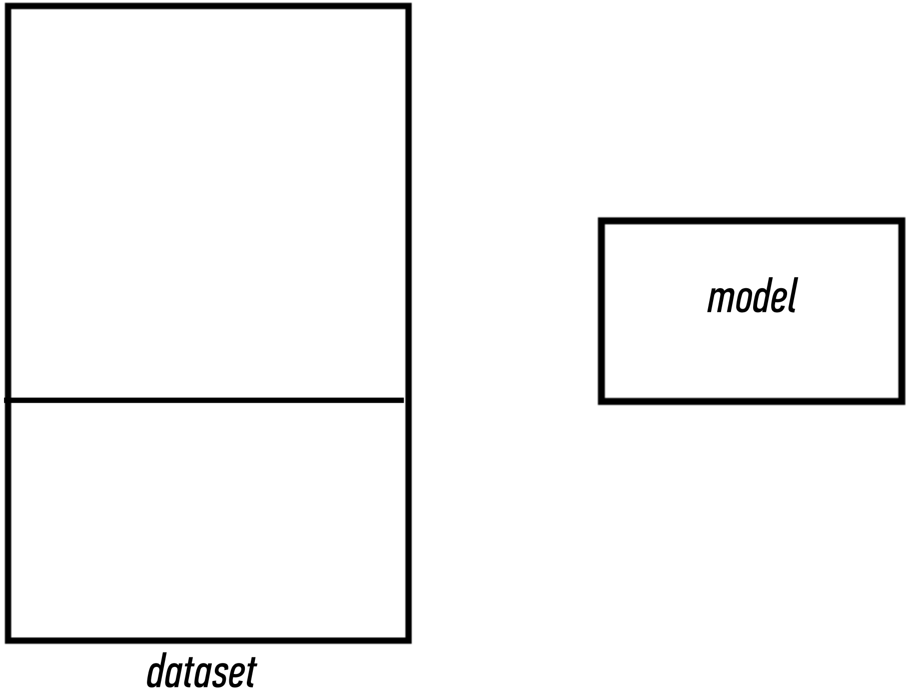
*source: http://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf*

## Q: What steps does a classification problem require?



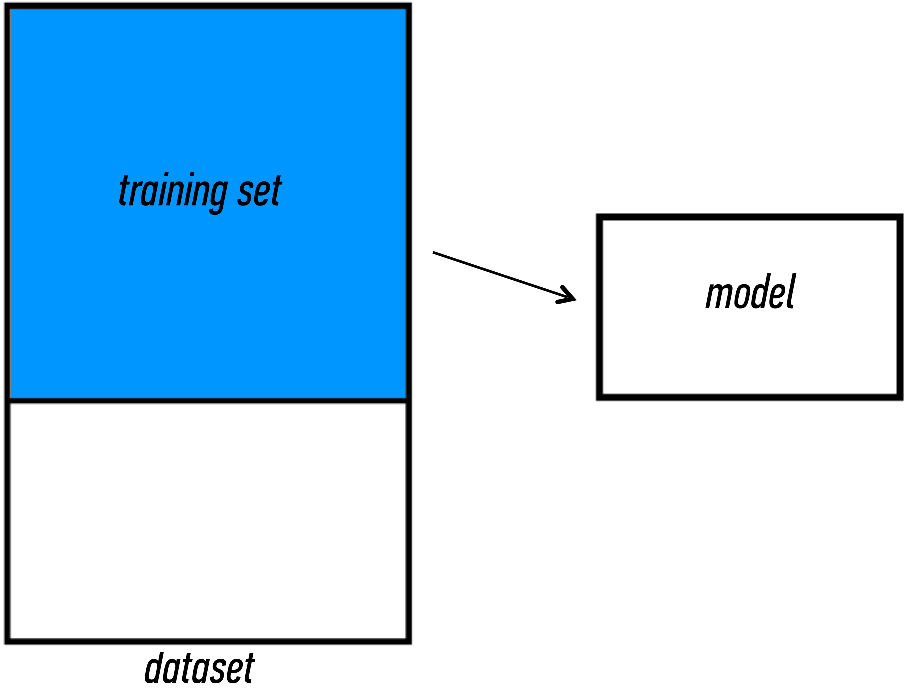
# Q: What steps does a classification problem require?

1) split dataset



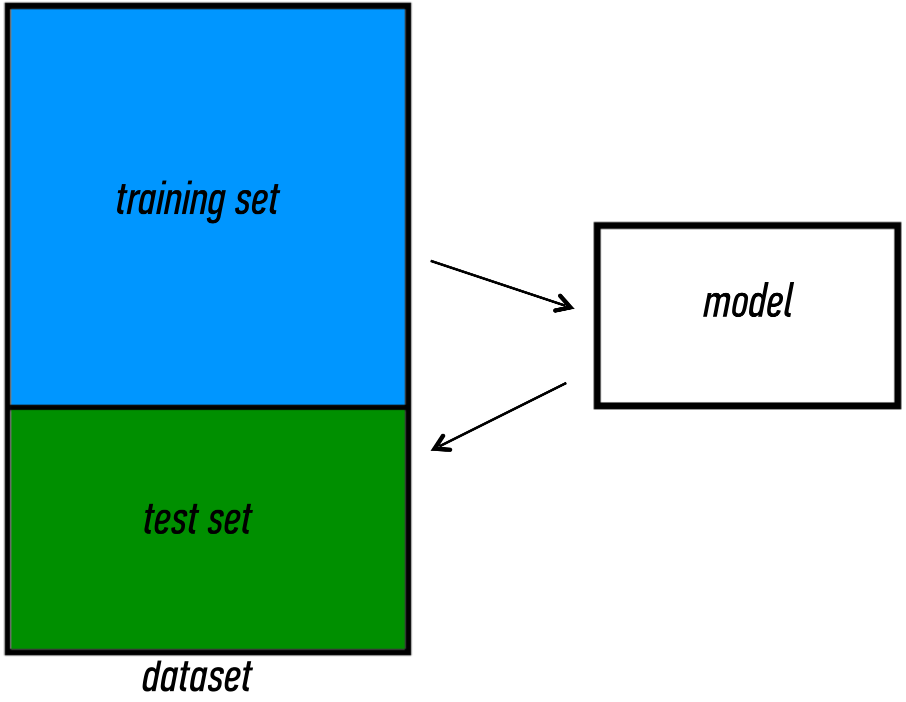
## Q: What steps does a classification problem require?

1) split dataset
2) train model



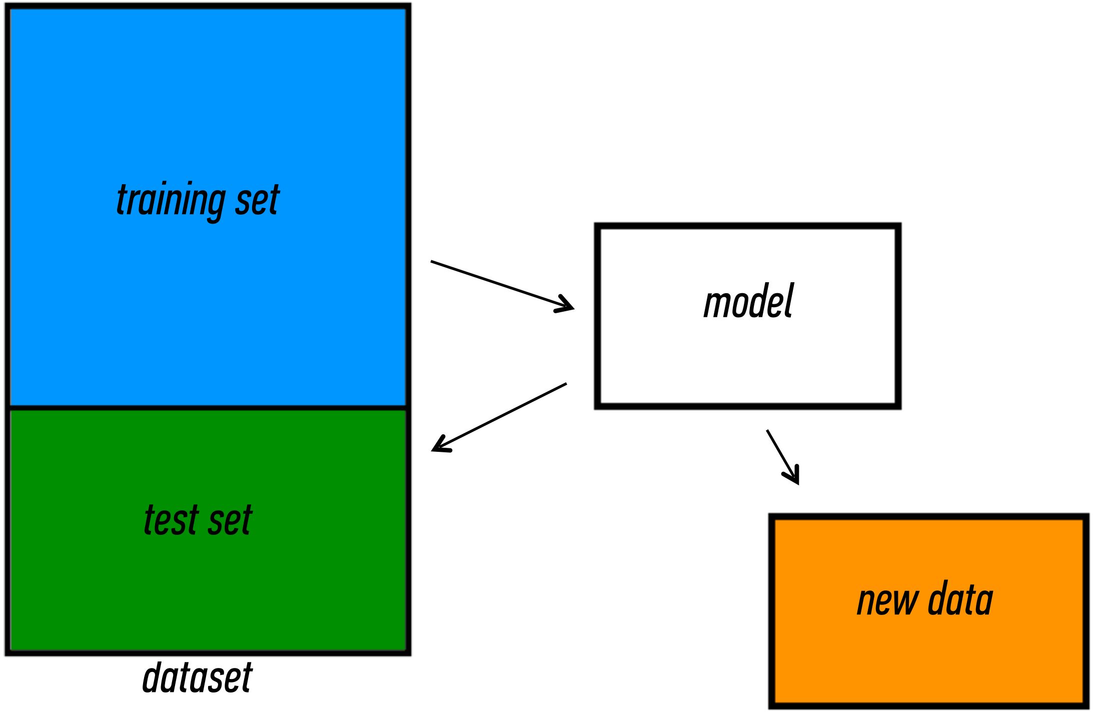
# Q: What steps does a classification problem require?

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



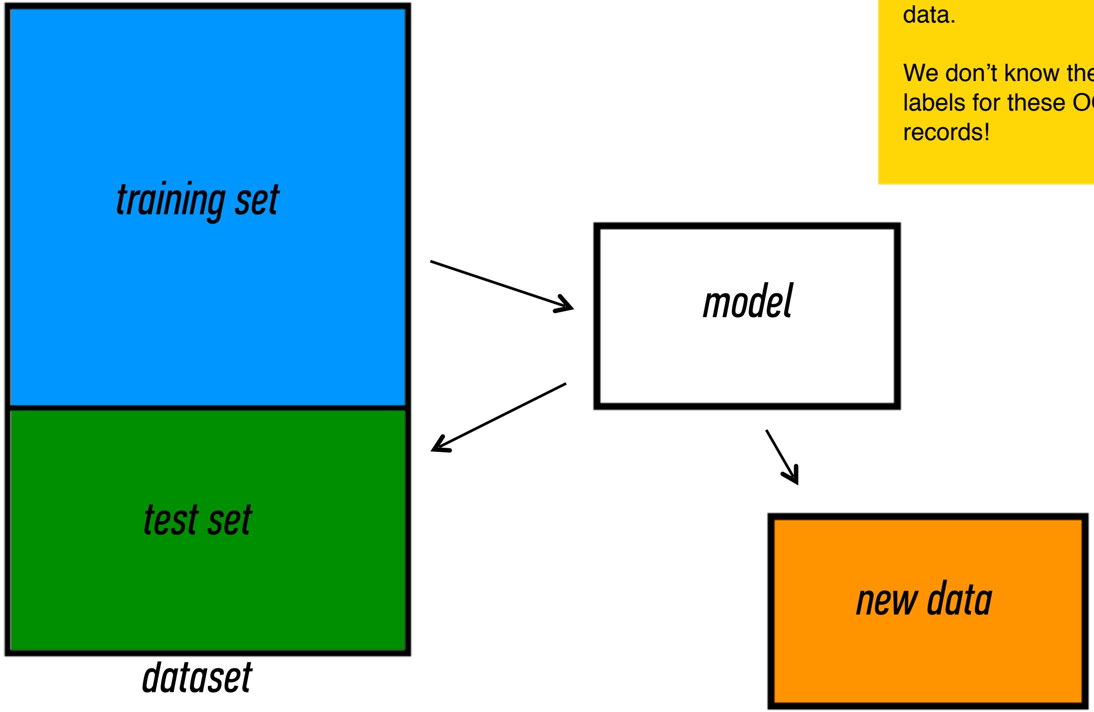
*Q: What steps does a classification problem require?*

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



## *Q: What steps does a classification problem require?*

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

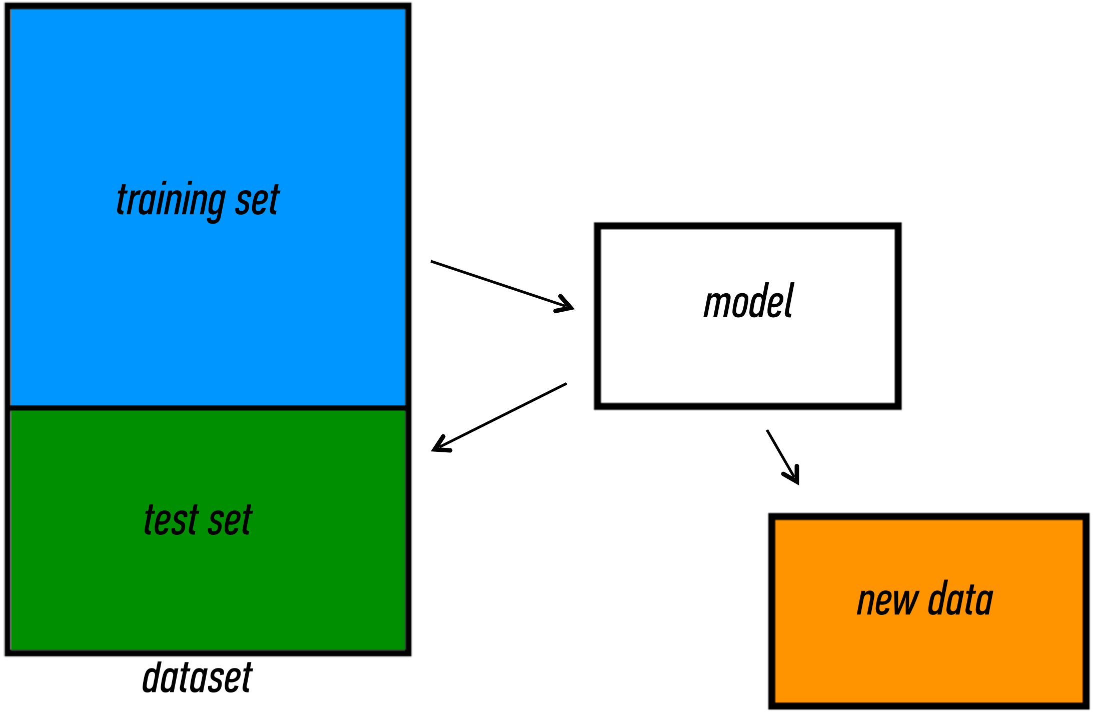



**NOTE**
This new data is called *out of sample* data.

We don't know the labels for these OOS records!

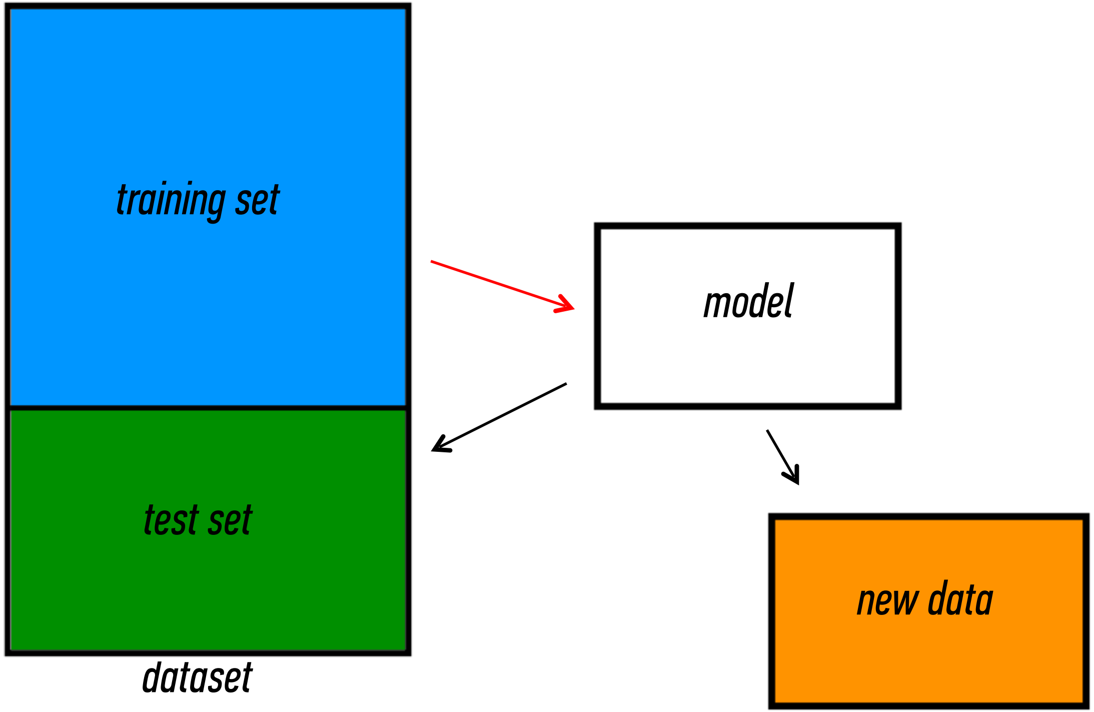
# III. BUILDING EFFECTIVE CLASSIFIERS

# Q: What types of prediction error will we run into?



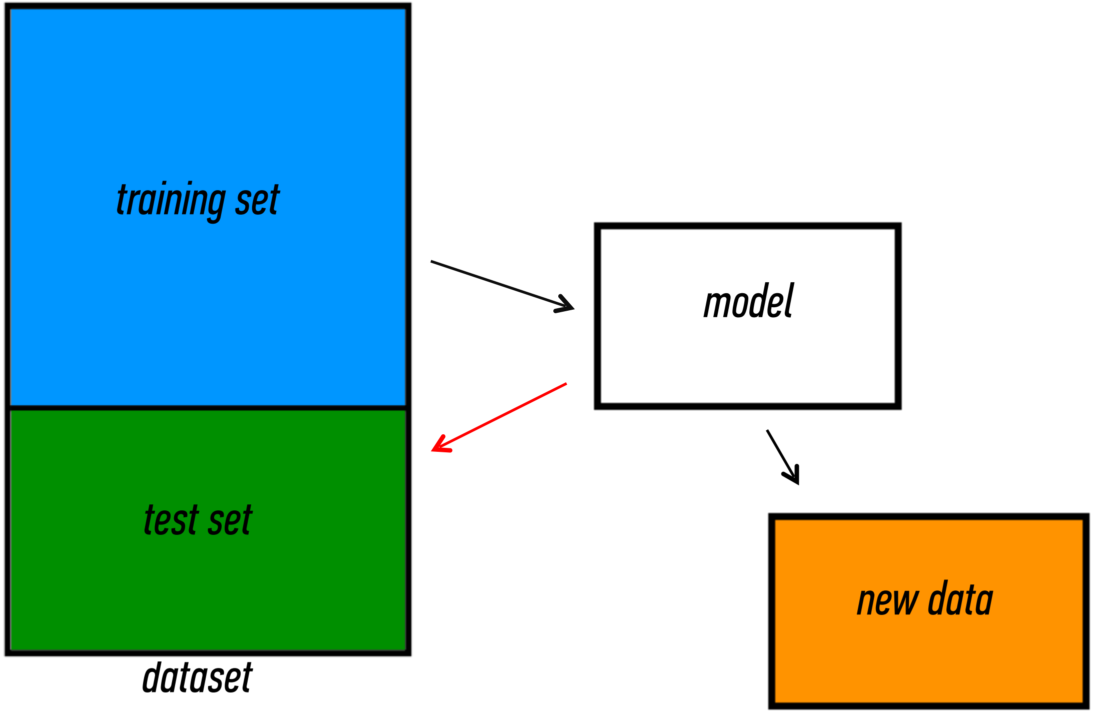
# Q: What types of prediction error will we run into?

1) training error



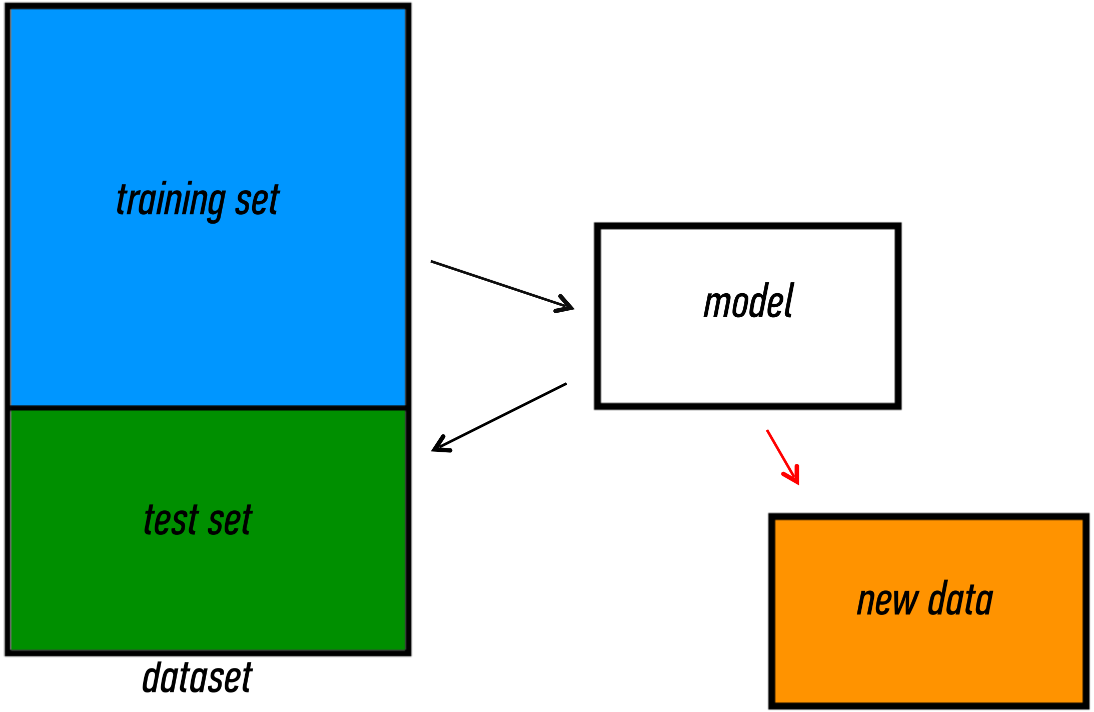
## Q: What types of prediction error will we run into?

1) training error
2) generalization error



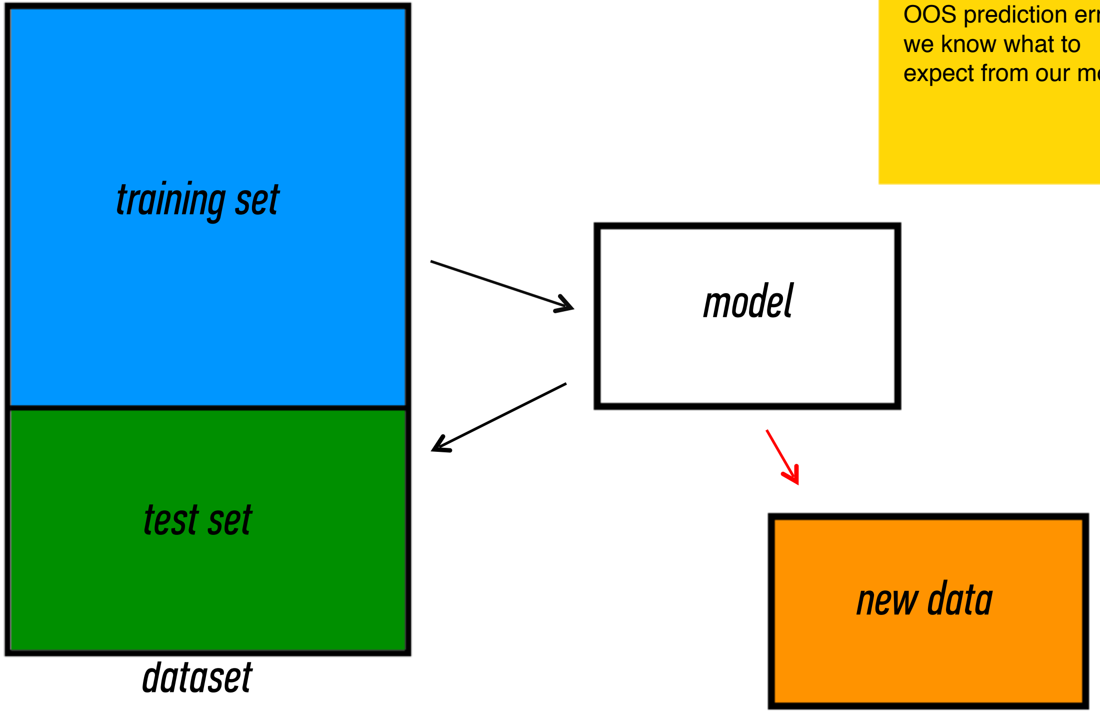
# Q: What types of prediction error will we run into?

1) training error
2) generalization error
3) OOS error



# Q: What types of prediction error will we run into?

1) training error
2) generalization error
3) OOS error




**NOTE**

We want to estimate OOS prediction error so we know what to expect from our model.

*Q: Why should we use training & test sets?*

## *Q: Why should we use training & test sets?*

*Thought experiment:*

*Suppose instead, we train our model using the entire dataset.*

*Q: Why should we use training & test sets?*

*Thought experiment:*
*Suppose instead, we train our model using the entire dataset.*
*Q: How low can we push the training error?*

# *Q: Why should we use training & test sets?*

*Thought experiment:*

*Suppose instead, we train our model using the entire dataset.*

*Q: How low can we push the training error?*

- *We can make the model arbitrarily complex (effectively "memorizing" the entire training set).*

# *Q: Why should we use training & test sets?*

*Thought experiment:*

*Suppose instead, we train our model using the entire dataset.*

*Q: How low can we push the training error?*

- *We can make the model arbitrarily complex (effectively "memorizing" the entire training set).*

*A: Down to zero!*

*Q: Why should we use training & test sets?*

*Thought experiment:*

*Suppose instead, we train our model using the entire dataset.*

*Q: How low can we push the training error?*

- *We can make the model arbitrarily complex (effectively "memorizing" the entire training set).*

*A: Down to zero!*

**NOTE**

This phenomenon is called *overfitting*.

## OVERFITTING

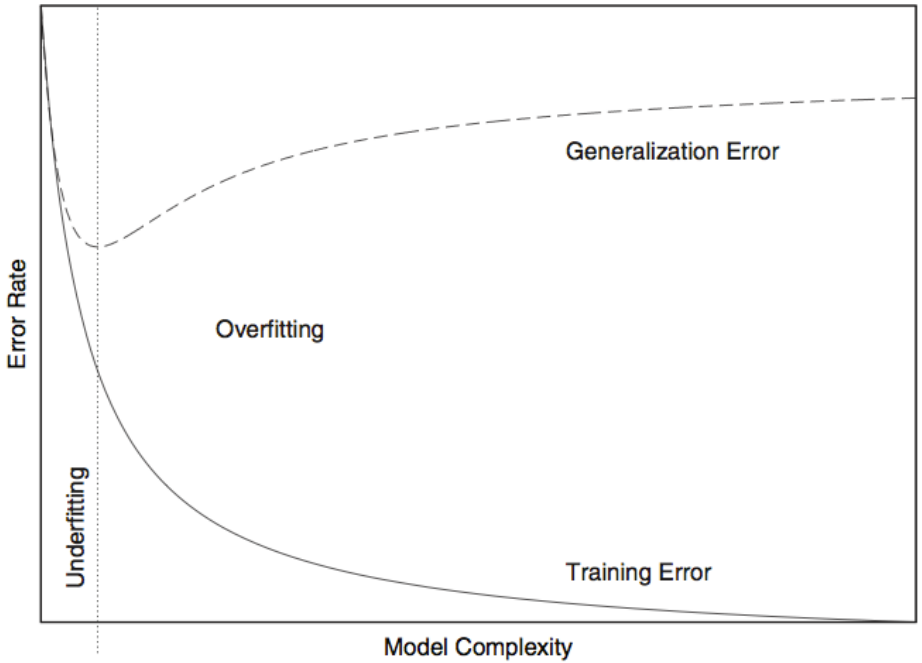
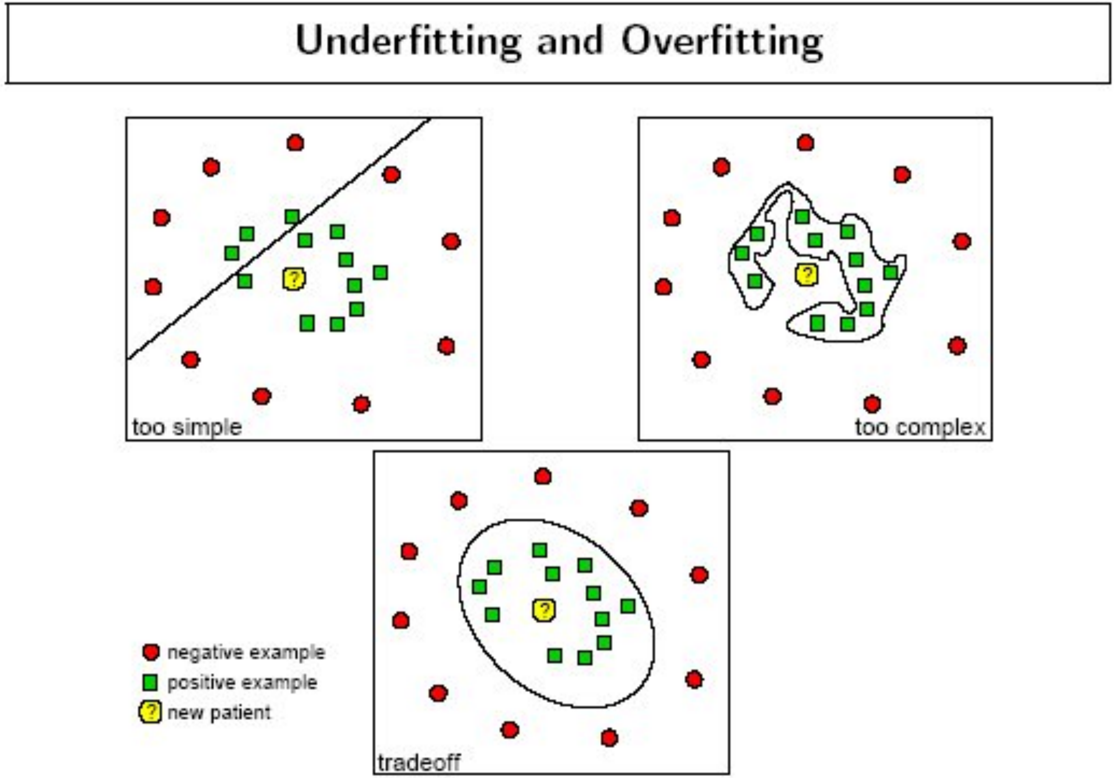



*FIGURE 18-1. Overfitting: as a model becomes more complex, it becomes increasingly able to represent the training data. However, such a model is overfitted and will not generalize well to data that was not used during training.*

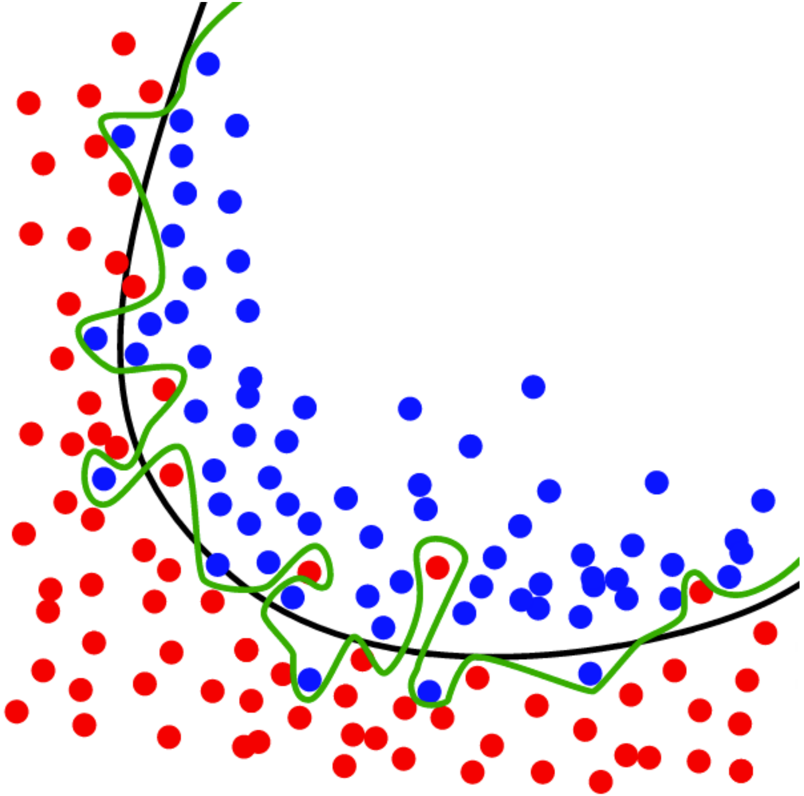
*source: Data Analysis with Open Source Tools, by Philipp K. Janert. O'Reilly Media, 2011.*

# OVERFITTING - EXAMPLE




source: http://www.dtreg.com

## OVERFITTING - EXAMPLE



source: http://www.dtreg.com

## TRAINING ERROR

# *Q: Why should we use training & test sets?*

*Thought experiment:*

*Suppose instead, we train our model using the entire dataset.*

*Q: How low can we push the training error?*

- *We can make the model arbitrarily complex (effectively "memorizing" the entire training set).*

*A: Down to zero!*

**NOTE**

This phenomenon is called *overfitting*.

## *A: Training error is not a good estimate of OOS accuracy.*

## GENERALIZATION ERROR

*Suppose we do the train/test split.*

*Suppose we do the train/test split.*

*Q: How well does generalization error predict OOS accuracy?*

## GENERALIZATION ERROR

*Suppose we do the train/test split.*

*Q: How well does generalization error predict OOS accuracy?*

*Thought experiment:*

*Suppose we had done a different train/test split.*

*Suppose we do the train/test split.*

*Q: How well does generalization error predict OOS accuracy?*

*Thought experiment:*

*Suppose we had done a different train/test split.*

*Q: Would the generalization error remain the same?*

## GENERALIZATION ERROR

*Suppose we do the train/test split.*

*Q: How well does generalization error predict OOS accuracy?*

*Thought experiment:*

*Suppose we had done a different train/test split.*

*Q: Would the generalization error remain the same?*

*A: Of course not!*

*Suppose we do the train/test split.*

*Q: How well does generalization error predict OOS accuracy?*

*Thought experiment:*

*Suppose we had done a different train/test split.*

*Q: Would the generalization error remain the same?*

*A: Of course not!*

*A: On its own, not very well.*

*Suppose we do the train/test split.*

*Q: How well does generalization error predict OOS accuracy?*

*Thought experiment:*

*Suppose we had done a different train/test split.*

*Q: Would the generalization error remain the same?*

*A: Of course not!*

*A: On its own, not very well.*

**NOTE**

The generalization error gives a *high-variance estimate* of OOS accuracy.

## GENERALIZATION ERROR

*Something is still missing!*

## GENERALIZATION ERROR

*Something is still missing!*

*Q: How can we do better?*

*Something is still missing!*

*Q: How can we do better?*

*Thought experiment:*

*Different train/test splits will give us different generalization errors.*

*Something is still missing!*

*Q: How can we do better?*

*Thought experiment:*

*Different train/test splits will give us different generalization errors.*

*Q: What if we did a bunch of these and took the average?*

*Something is still missing!*

*Q: How can we do better?*

*Thought experiment:*

*Different train/test splits will give us different generalization errors.*

*Q: What if we did a bunch of these and took the average?*

*A: Now you're talking!*

*Something is still missing!*

*Q: How can we do better?*

*Thought experiment:*

*Different train/test splits will give us different generalization errors.*

*Q: What if we did a bunch of these and took the average?*

*A: Now you're talking!*

*A: Cross-validation.*

*Steps for n-fold cross-validation:*

## *Steps for n-fold cross-validation:*

1) *Randomly split the dataset into n equal partitions.*

# *Steps for n-fold cross-validation:*

1) *Randomly split the dataset into n equal partitions.*
2) *Use partition 1 as test set & union of other partitions as training set.*

# *Steps for n-fold cross-validation:*

1) *Randomly split the dataset into n equal partitions.*
2) *Use partition 1 as test set & union of other partitions as training set.*
3) *Find generalization error.*

# *Steps for n-fold cross-validation:*

1) *Randomly split the dataset into n equal partitions.*
2) *Use partition 1 as test set & union of other partitions as training set.*
3) *Find generalization error.*
4) *Repeat steps 2-3 using a different partition as the test set at each iteration.*

# *Steps for n-fold cross-validation:*

1) *Randomly split the dataset into n equal partitions.*
2) *Use partition 1 as test set & union of other partitions as training set.*
3) *Find generalization error.*
4) *Repeat steps 2-3 using a different partition as the test set at each iteration.*
5) *Take the average generalization error as the estimate of OOS accuracy.*

*Features of n-fold cross-validation:*

# *Features of n-fold cross-validation:*

1) *More accurate estimate of OOS prediction error.*

# *Features of n-fold cross-validation:*

1) *More accurate estimate of OOS prediction error.*
2) *More efficient use of data than single train/test split.*
   - *Each record in our dataset is used for both training and testing.*

# *Features of n-fold cross-validation:*

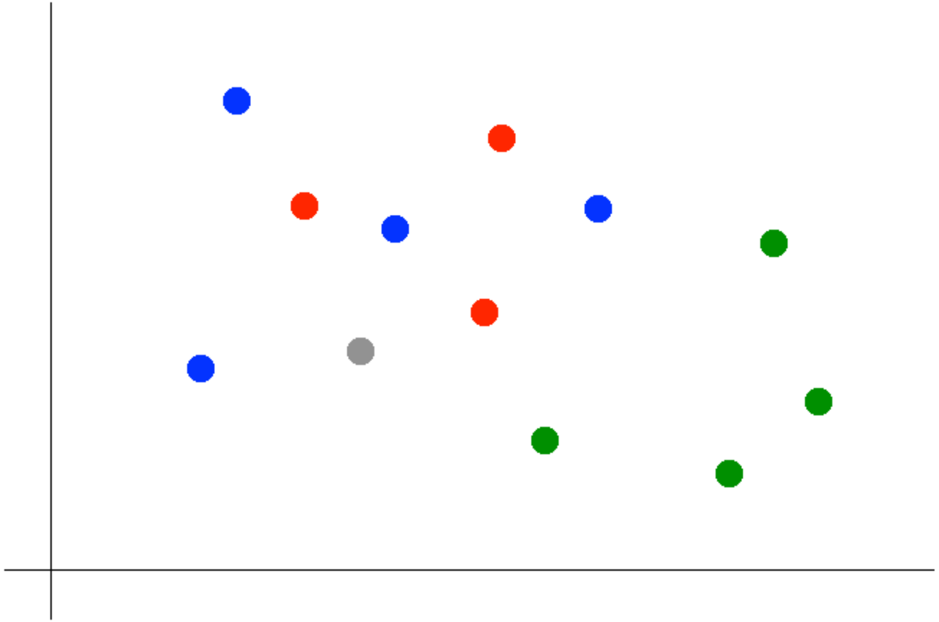
1) *More accurate estimate of OOS prediction error.*
2) *More efficient use of data than single train/test split.*
   - *Each record in our dataset is used for both training and testing.*
3) *Presents tradeoff between efficiency and computational expense.*
   - *10-fold CV is 10x more expensive than a single train/test split*

# *Features of n-fold cross-validation:*

1) *More accurate estimate of OOS prediction error.*
2) *More efficient use of data than single train/test split.*
   - *Each record in our dataset is used for both training and testing.*
3) *Presents tradeoff between efficiency and computational expense.*
   - *10-fold CV is 10x more expensive than a single train/test split*
4) *Can be used for model selection.*

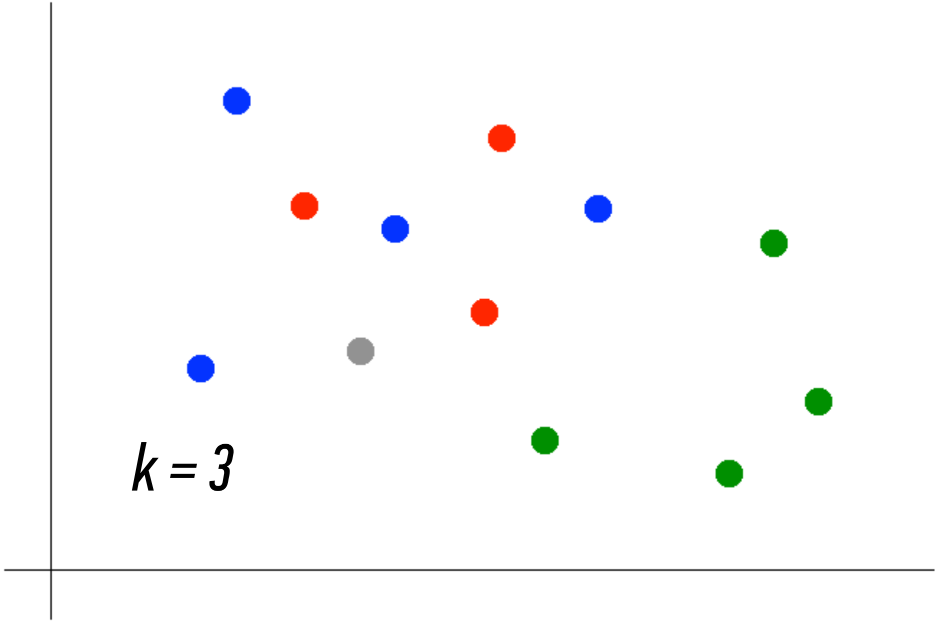
# IV. KNN CLASSIFICATION

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



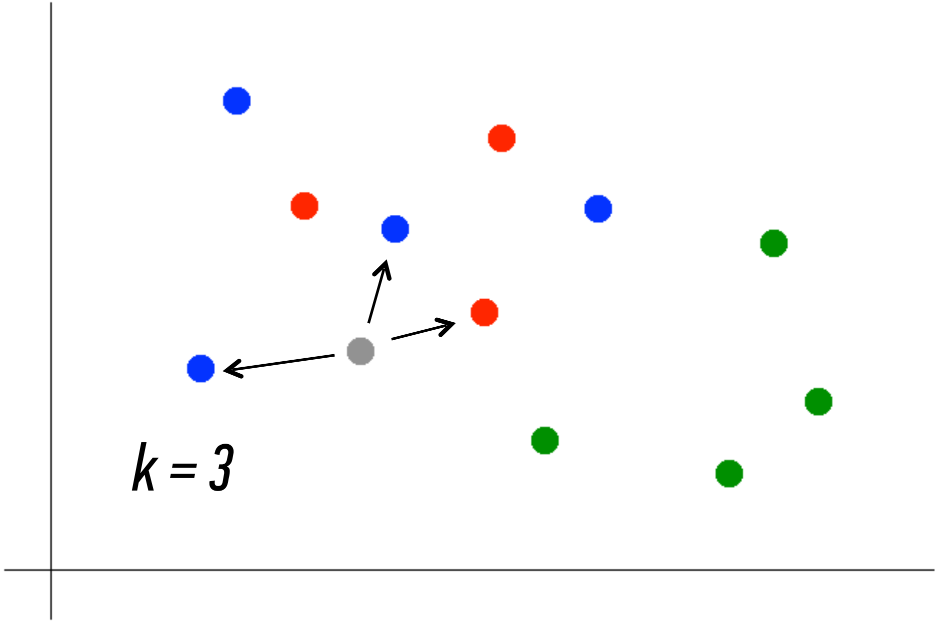
*Suppose we want to predict the color of the grey dot.*

1) *Pick a value for k.*



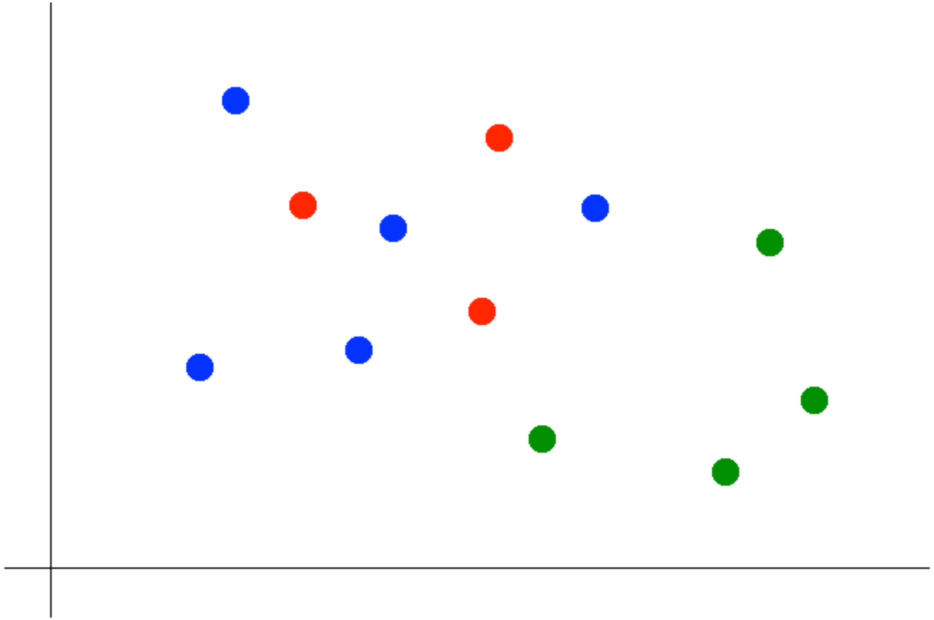
*Suppose we want to predict the color of the grey dot.*

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



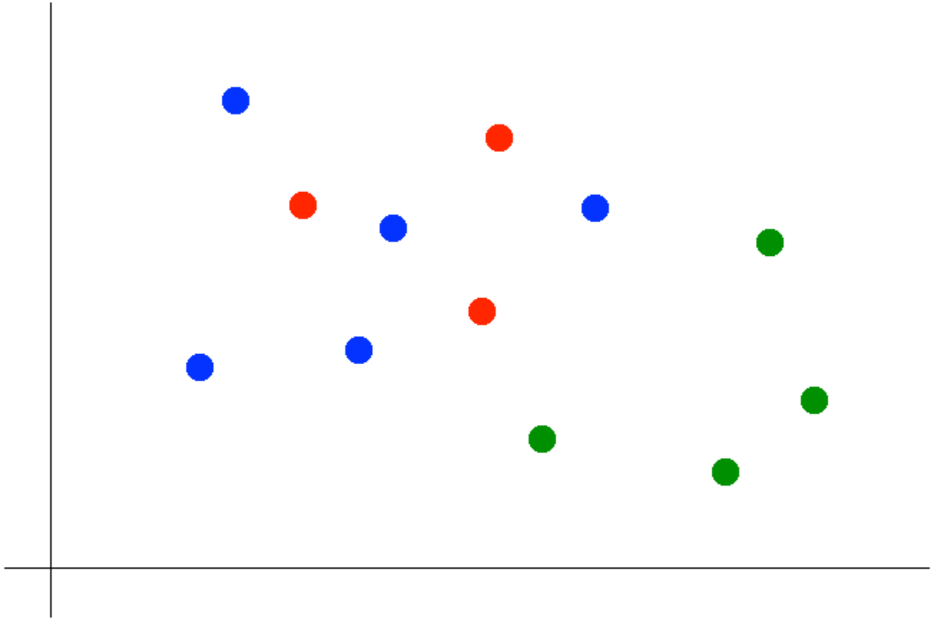
*Suppose we want to predict the color of the grey dot.*

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



# *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.*



**OPTIONAL NOTE**

Our definition of "nearest" implicitly uses the *Euclidean distance function.*

# LABS

# ASSIGNMENT – KNN WITH N-FOLD CROSS-VALIDATION

## KEY OBJECTIVES

*Extend the script we used in class to implement knn classification on the iris dataset using n-fold cross-validation.*

*(bonus: split code into functions)*

*for example:*

```
knn.nfold <- function(n, … ) {
   # create n-fold partition of dataset
   # perform knn classification n times
   # n-fold generalization error = average over all iterations
}
```