DATA SCIENCE
MODEL EVALUATION PROCEDURES

Q: What's wrong with training error?

Thought experiment:
Suppose 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: What's wrong with training error?

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

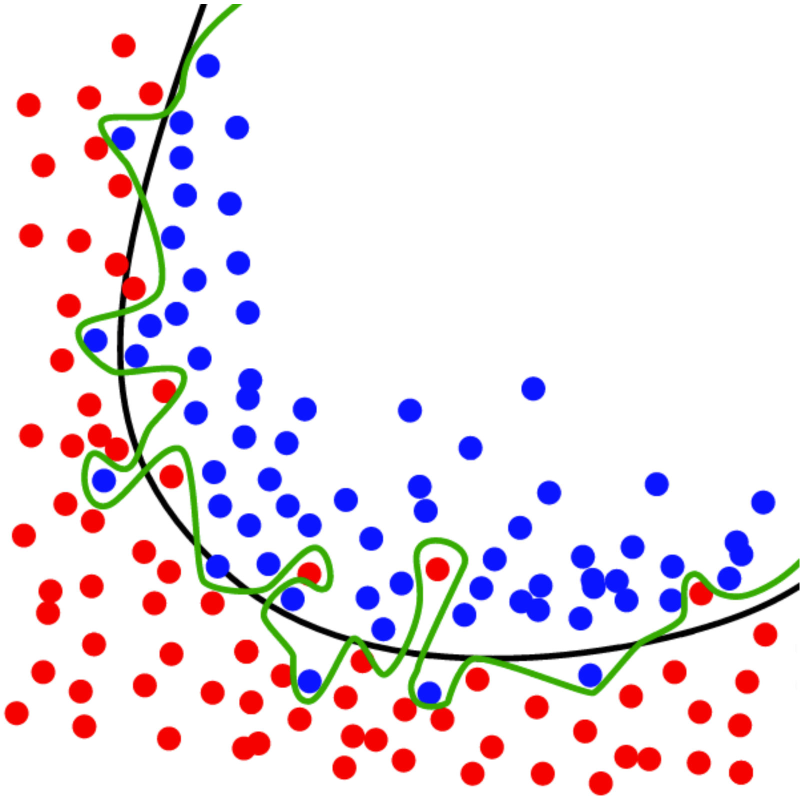
- *We can make the model arbitrarily complex (effectively "memorizing" the entire training set).*

A: Down to zero!

NOTE

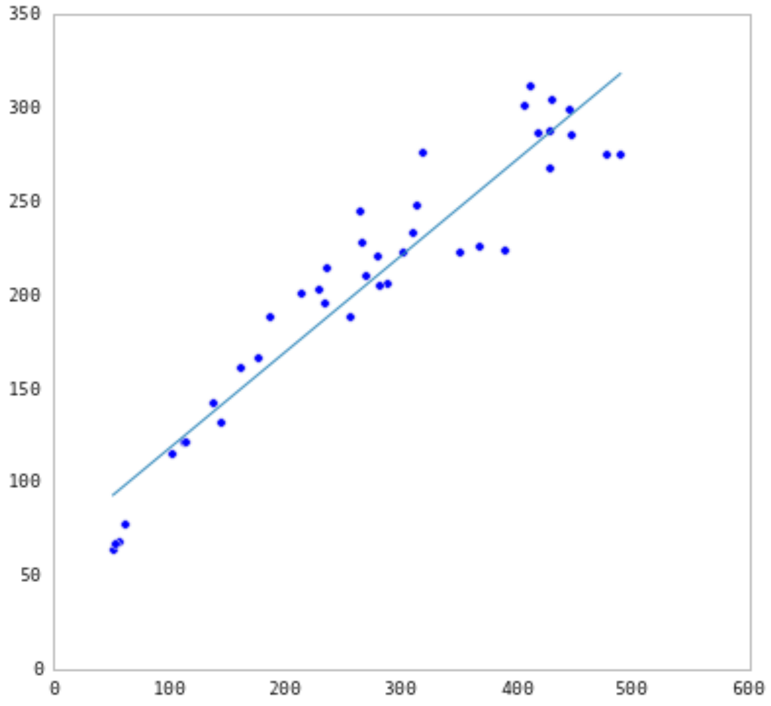
This phenomenon is called *overfitting*.

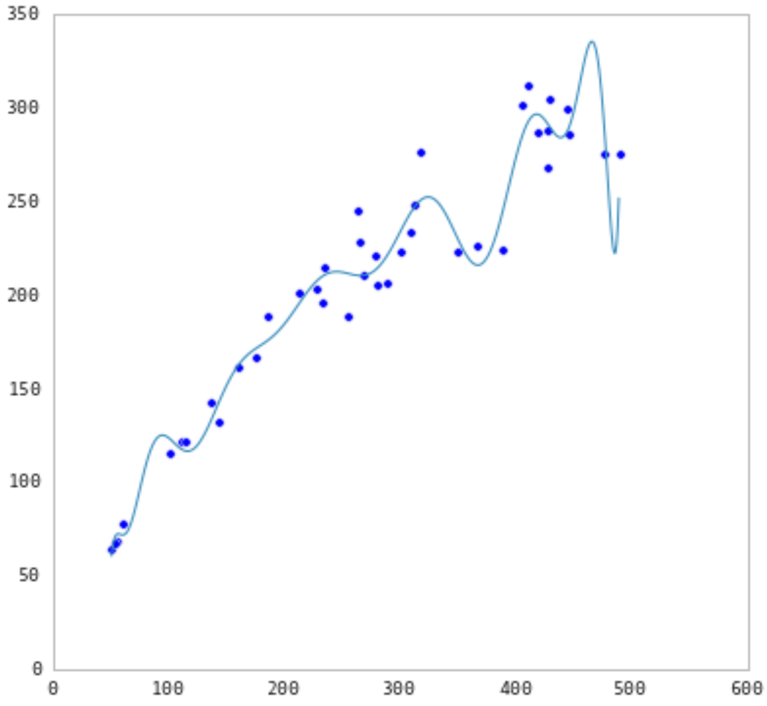
OVERFITTING



Source: http://www.dtreg.com

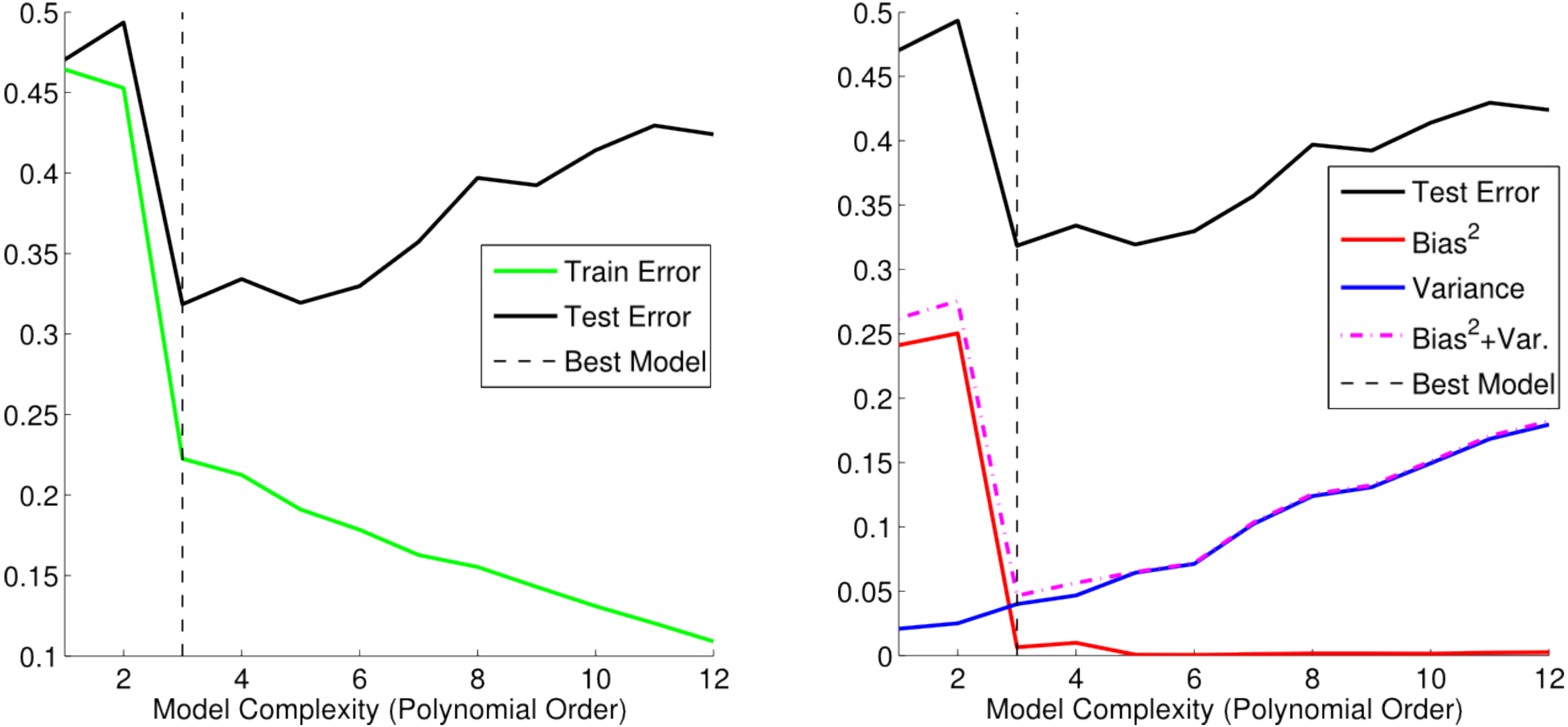
UNDERFITTING AND OVERFITTING





Source: http://nbviewer.ipython.org/github/fonnesbeck/Bios366/blob/master/notebooks/Section6_3-Model-Selection-and-Validation.ipynb

UNDERFITTING AND OVERFITTING



Source: http://theclevermachine.wordpress.com/2013/04/21/model-selection-underfitting-overfitting-and-the-bias-variance-tradeoff/

Q: What's wrong with training error?

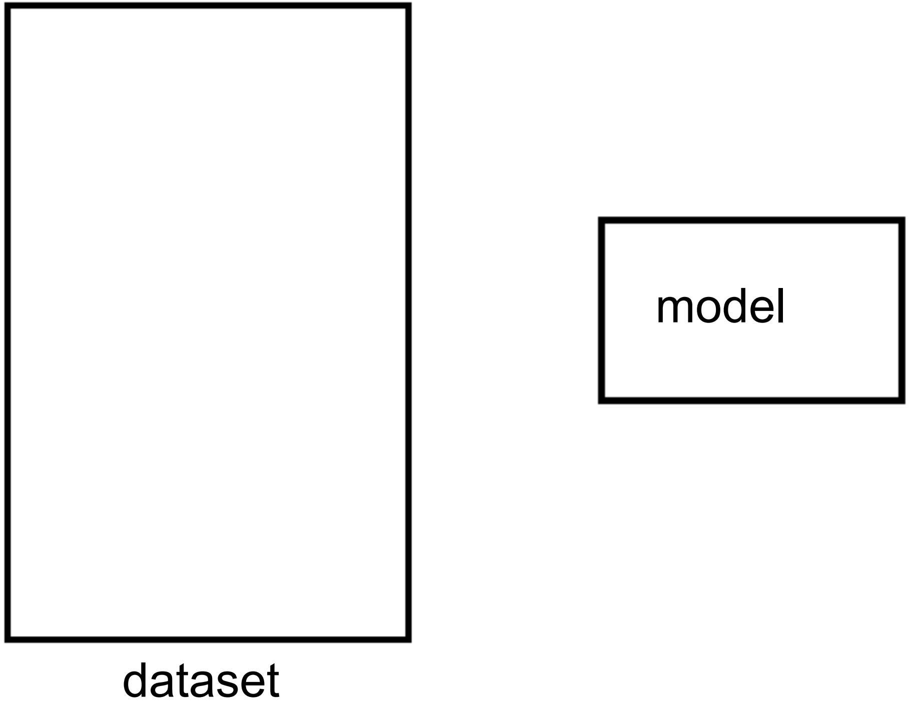
Thought experiment:
Suppose we train our model using the entire dataset.
Q: How low can we push the training error?

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

A: Down to zero!

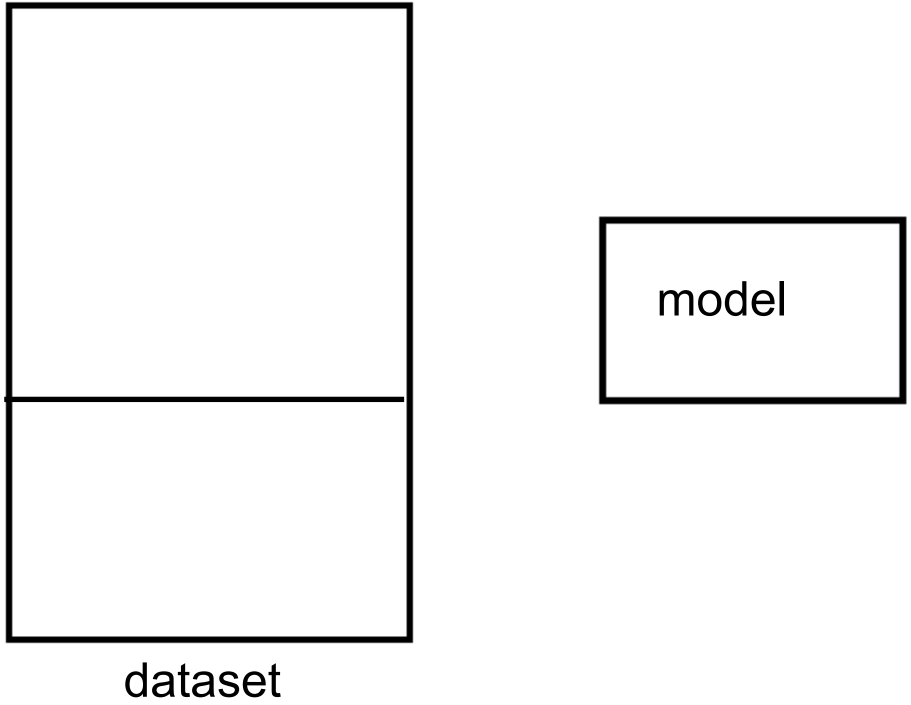
A: Training error is not a good estimate of accuracy beyond training data.

Q: How can we make a model that generalizes well?



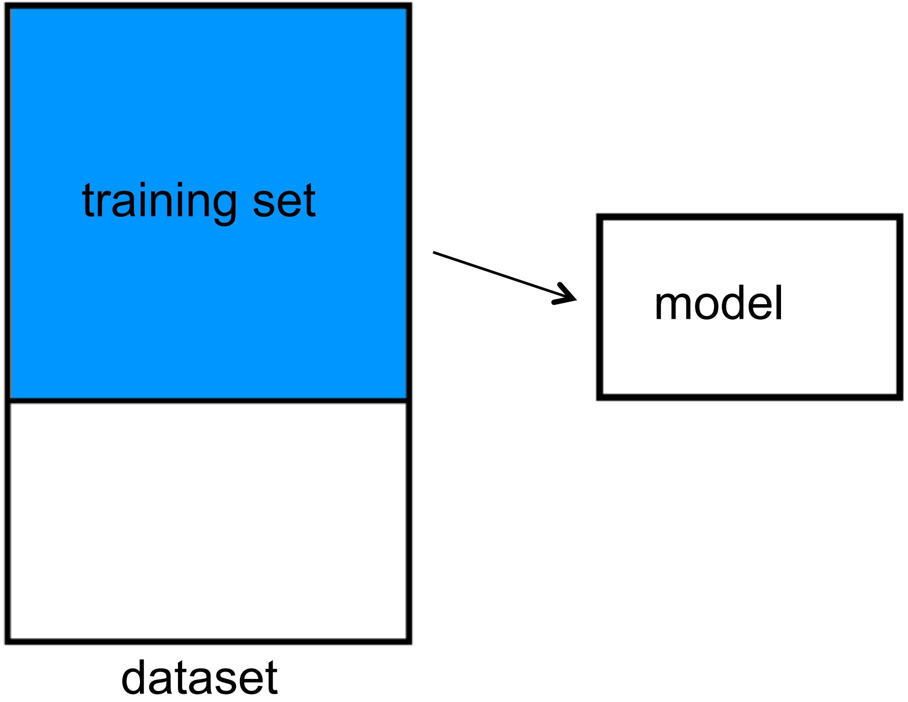
Q: How can we make a model that generalizes well?

1) split dataset



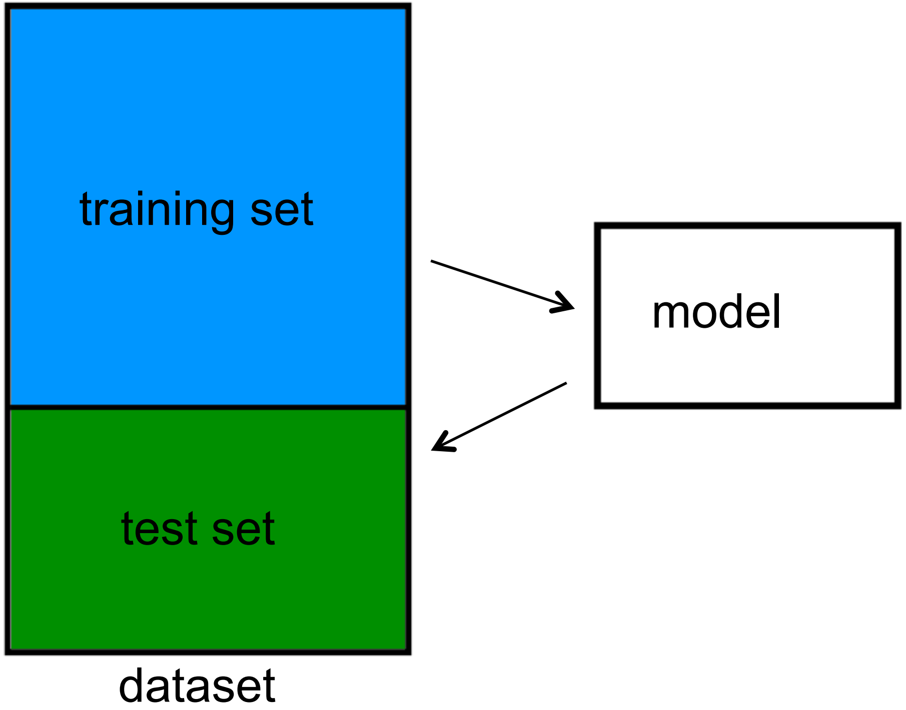
Q: How can we make a model that generalizes well?

1) split dataset
2) train model



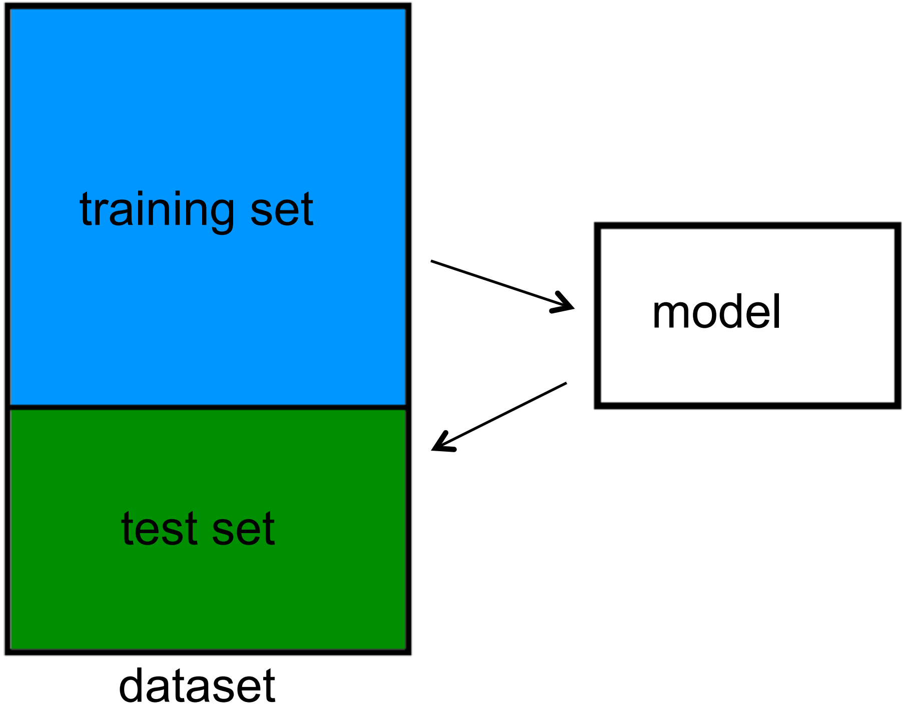
Q: How can we make a model that generalizes well?

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



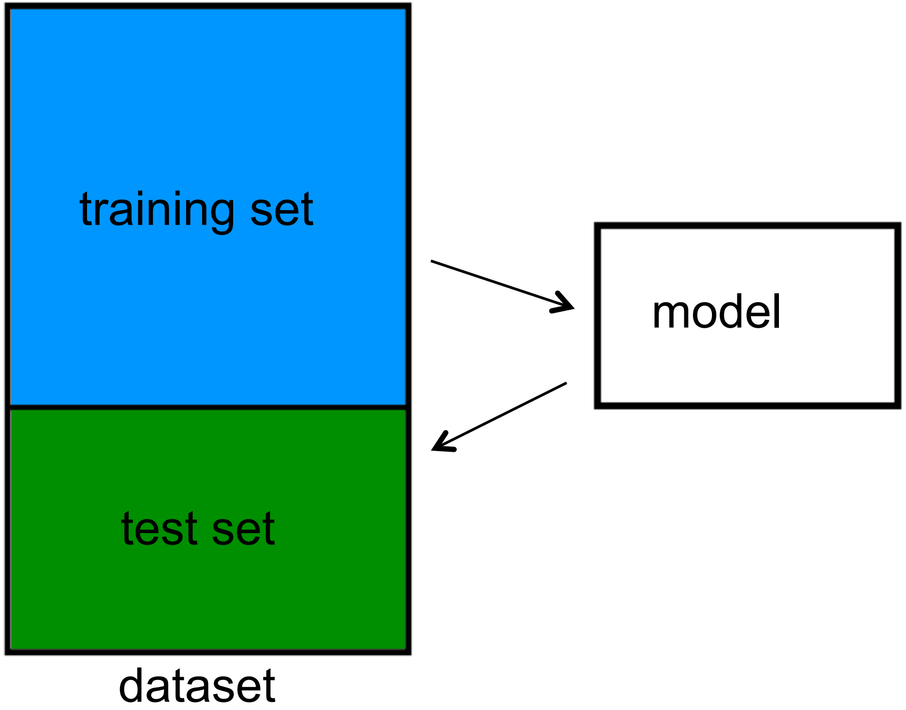
Q: How can we make a model that generalizes well?

1) split dataset
2) train model
3) test model
4) parameter tuning



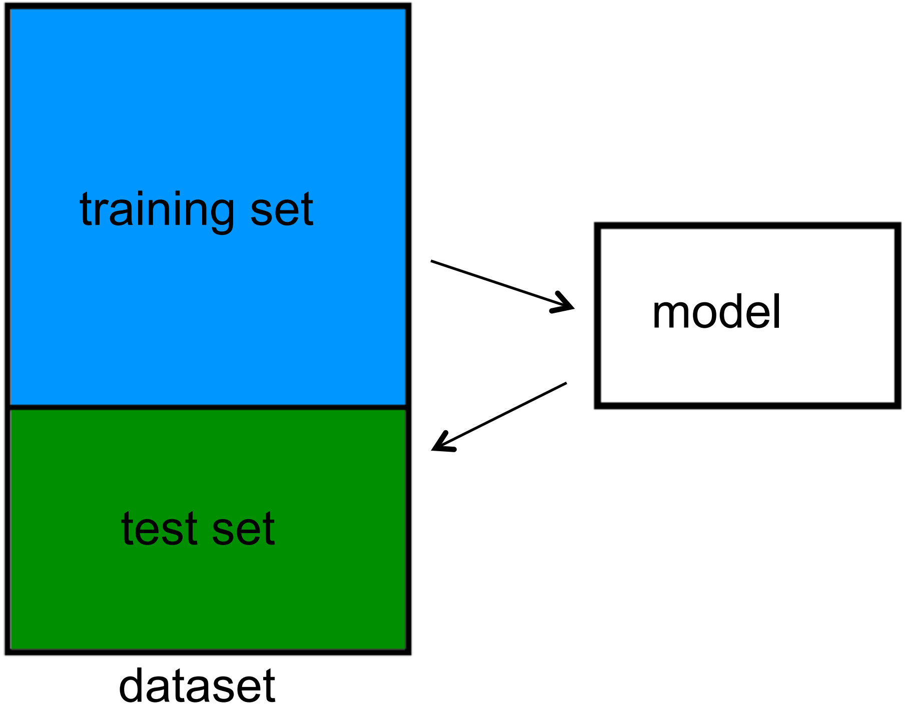
Q: How can we make a model that generalizes well?

1) split dataset
2) train model
3) test model
4) parameter tuning
5) choose best model



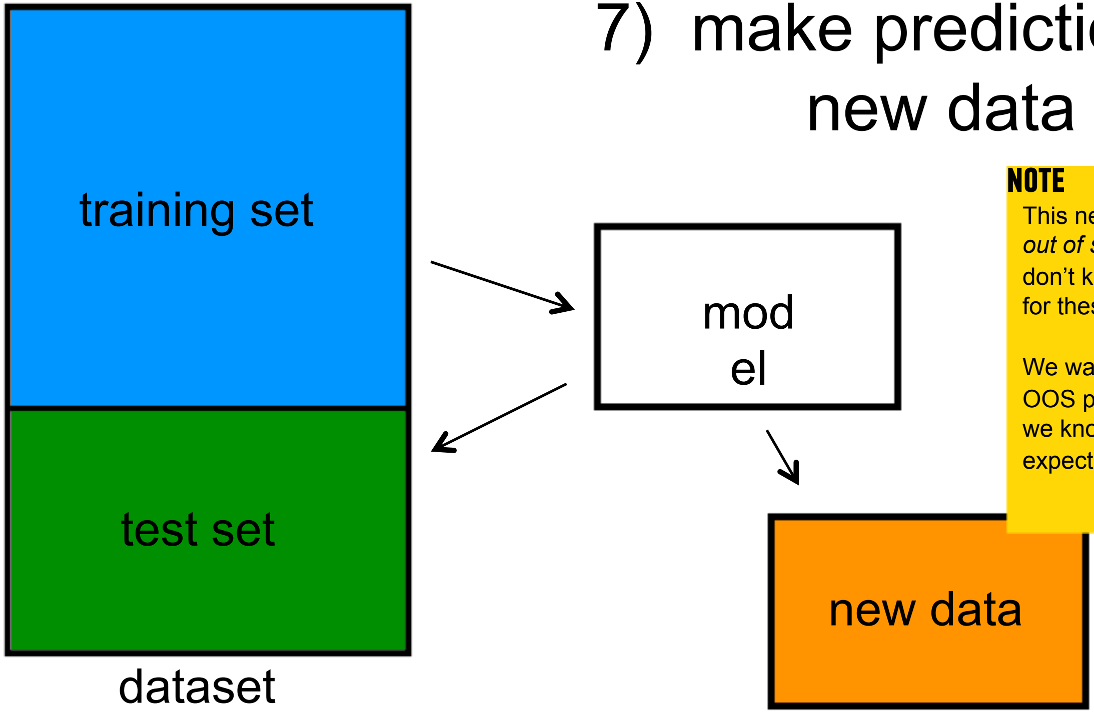
Q: How can we make a model that generalizes well?

1) split dataset
2) train model
3) test model
4) parameter tuning
5) choose best model
6) train on **all** data



Q: How can we make a model that generalizes well

1) split dataset
2) train model
3) test model
4) parameter tuning
5) choose best model
6) train on **all** data
7) make predictions on new data



NOTE
This new data is called *out of sample* data. We don't know the labels for these OOS records!

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

Suppose we do the train/test split.
Q: How well does test set error predict OOS accuracy?

Thought experiment:
Suppose we had done a different train/test split.
Q: Would the test set error remain the same?
A: Of course not!

A: On its own, not very well.

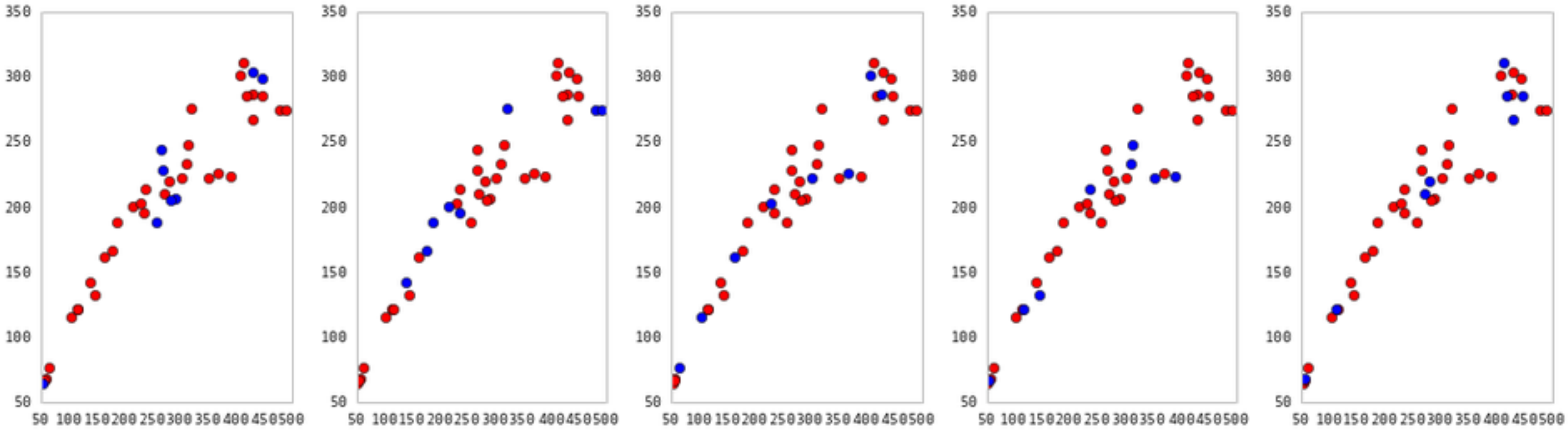
Something is still missing!
Q: How can we do better?

Thought experiment:
Different train/test splits will give us different test set errors.
Q: What if we did a bunch of these and took the average?
A: Now you’re talking!

A: Cross-validation.

Steps for K-fold cross-validation:

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

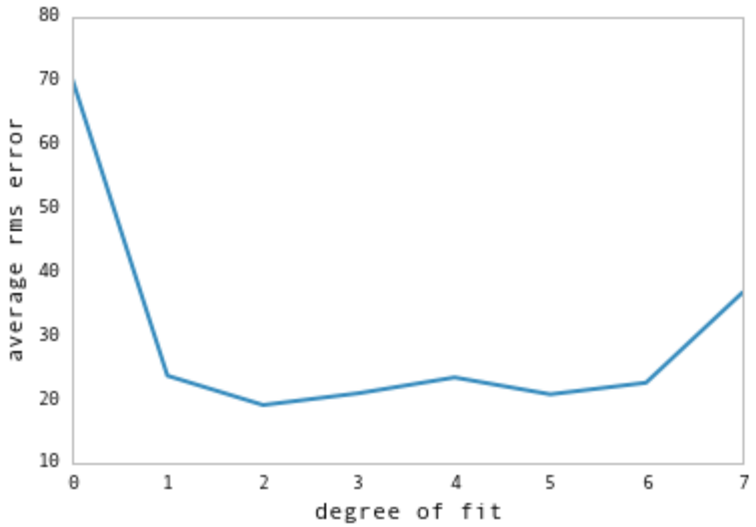


5-fold cross-validation: red = training folds, blue = test fold

Source: http://nbviewer.ipython.org/github/fonnesbeck/Bios366/blob/master/notebooks/Section6_3-Model-Selection-and-Validation.ipynb

Features of K-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 ex
    - 10-fold CV is 10x more expensive than a single train/test split
4) Can be used for parameter tuning and model selection.



Model selection using cross-validation:
lowest predicted OOS error at degree = 2

Source: http://nbviewer.ipython.org/github/fonnesbeck/Bios366/blob/master/notebooks/Section6_3-Model-Selection-and-Validation.ipynb

DATA SCIENCE