GENERAL ASSEMBLY

DATA SCIENCE

SYD DAT 6

Week 3 - Model Evaluation
Wednesday 26th October

AGENDA

1. Evaluating machine learning models
2. Why is this important?
3. Correctly assessing the accuracy of a model
4. Lab
5. Review

DATA SCIENCE PART TIME COURSE

THE POINT OF EVALUATING MODELS

Why do we need to evaluate models?

Why might we need to be rigorous in evaluating models?

DATA SCIENCE PART TIME COURSE

ESSENTIALS OF MODEL EVALUATION

Q: What's wrong with training error?

Thought experiment:

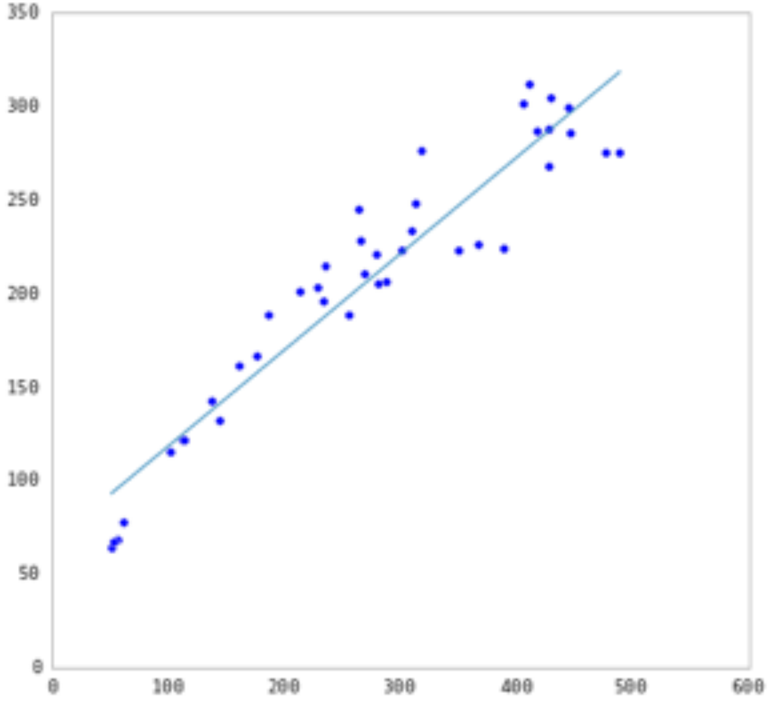
Suppose we train our model using the entire dataset.

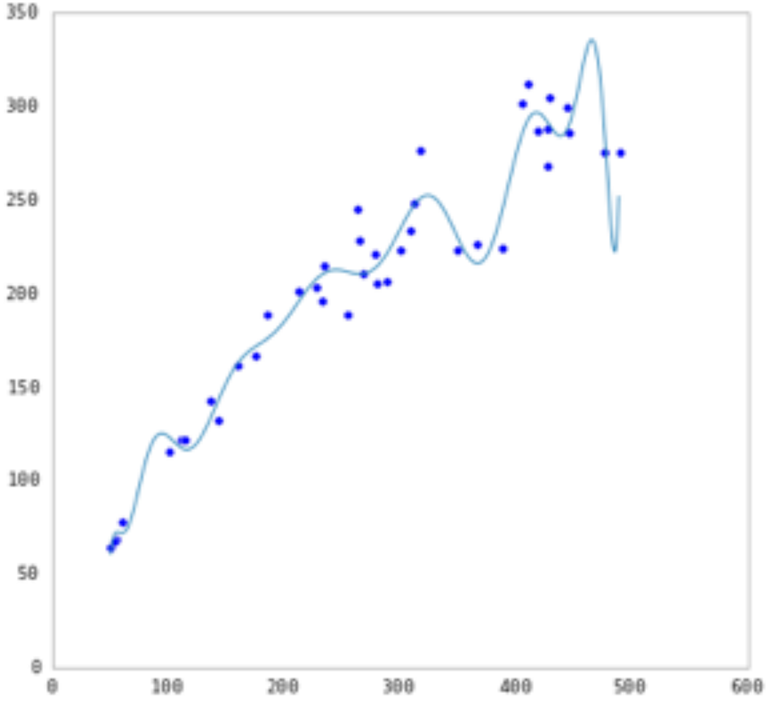
Q: How low can we push the training error?

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

A: Down to zero!

TRAINING ERROR





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!

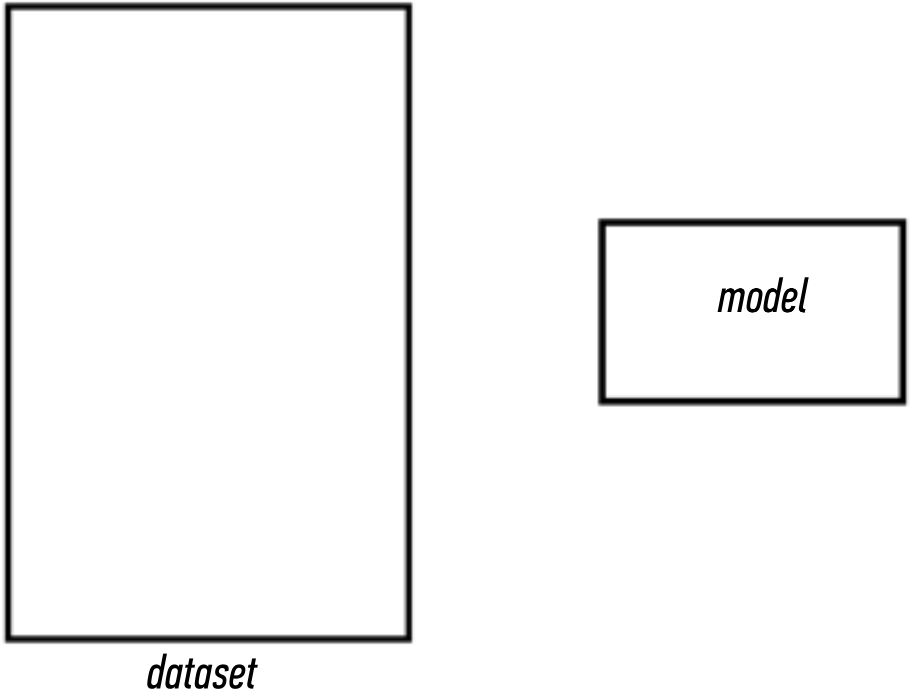
A: Training error is not a good estimate of accuracy beyond training data.

WHY THIS MATTERS

The data that we are given for prediction won’t always be the end of the data we are interested in! We may not have access to all the data of interest

We will gather data and build and iterate over models however a main reason for building the model was to predict unseen test cases.

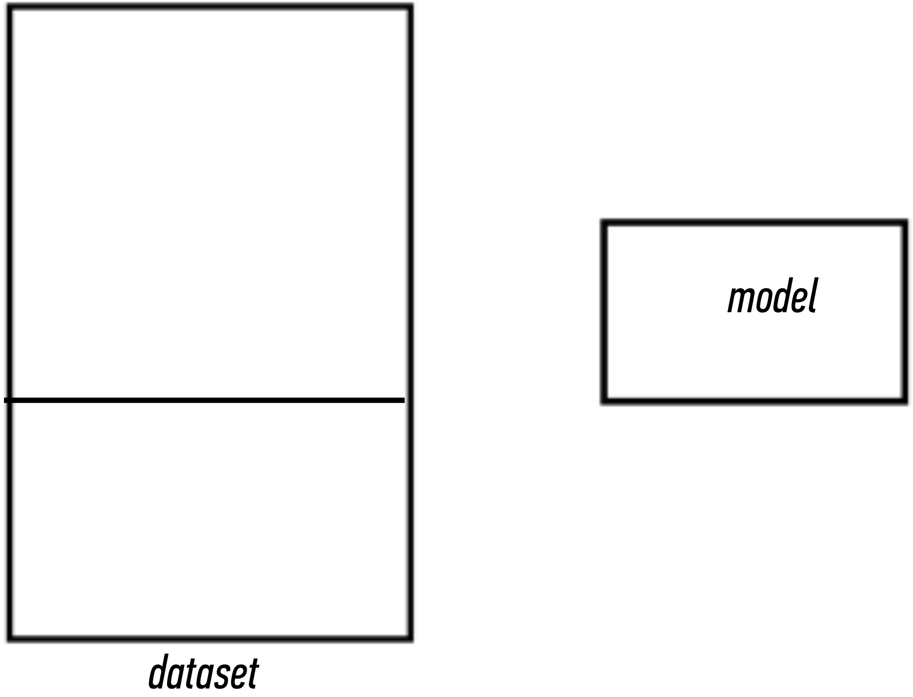
Q: How can we make a model that generalizes well?



TEST SET APPROACH

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

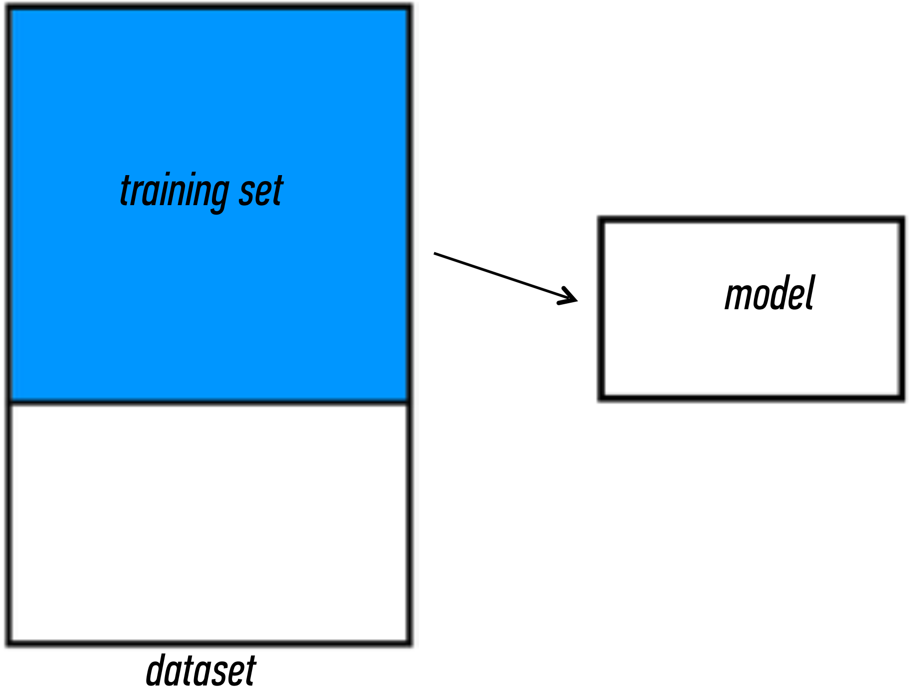
1) split dataset



TEST SET APPROACH

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

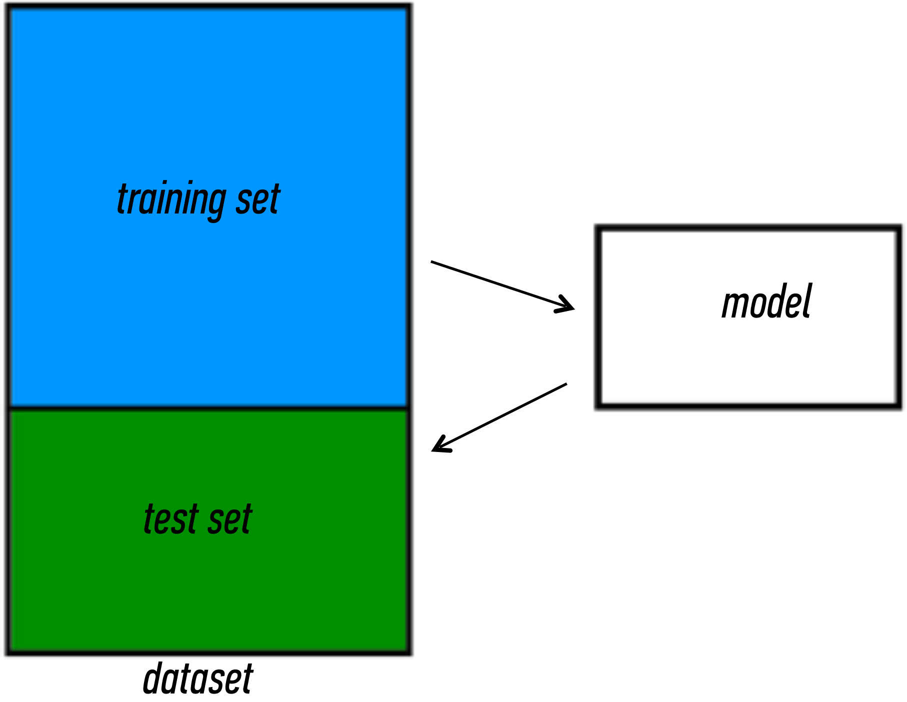
1) split dataset
2) train model



TEST SET APPROACH

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

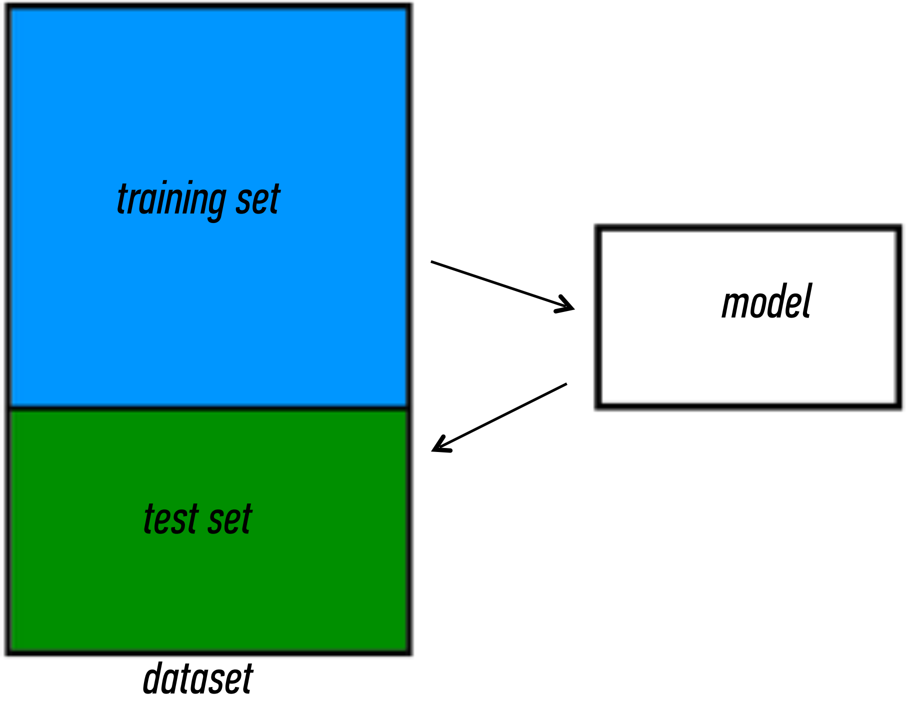
1) split dataset
2) train model
3) test model



TEST SET APPROACH

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

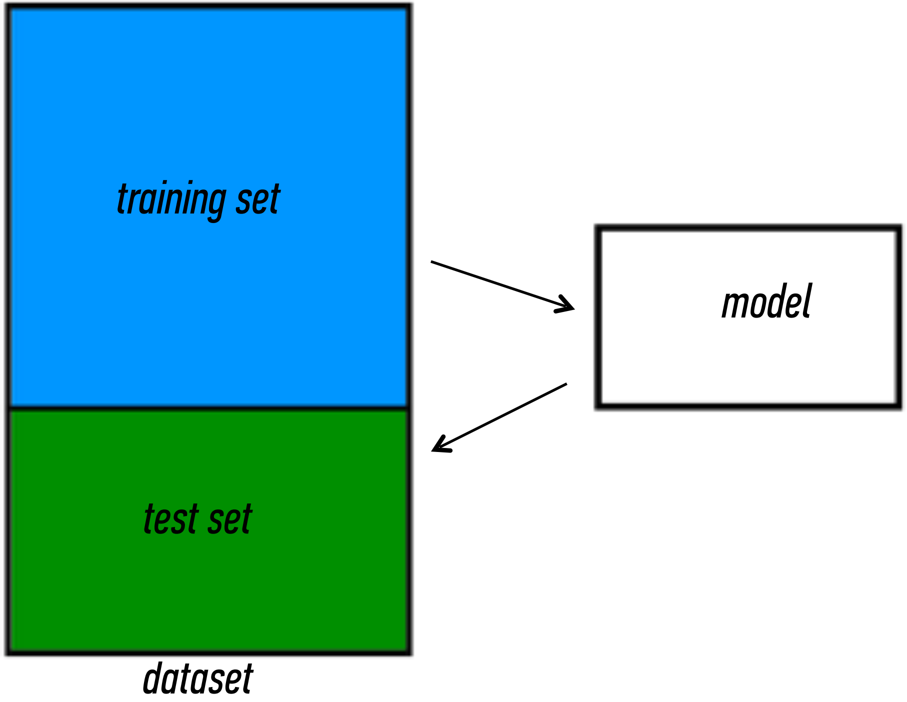
1) split dataset
2) train model
3) test model
4) parameter tuning



TEST SET APPROACH

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

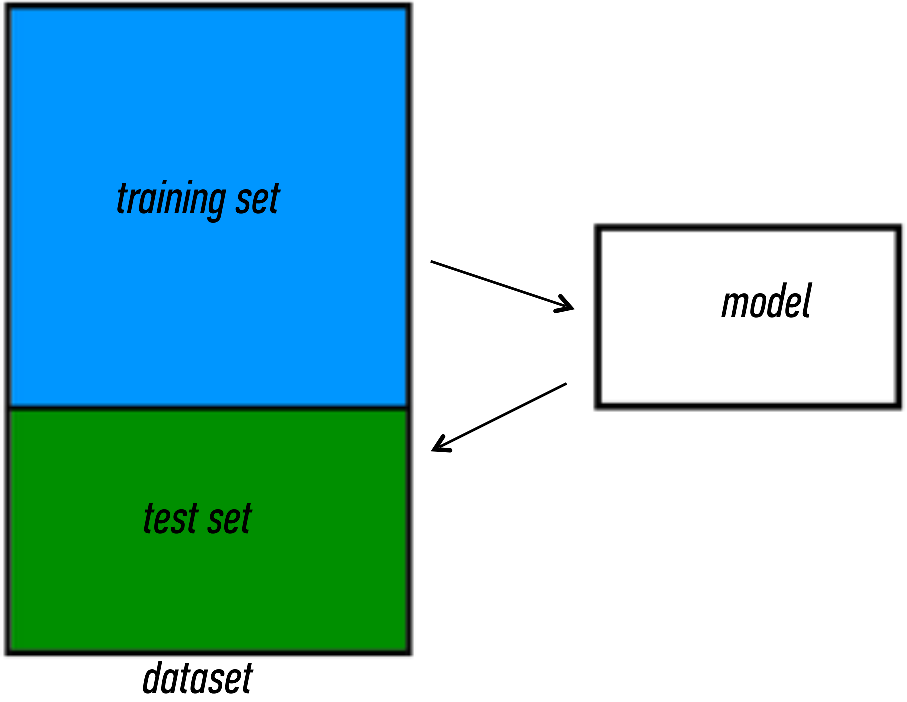
1) split dataset
2) train model
3) test model
4) parameter tuning
5) choose best model



TEST SET APPROACH

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

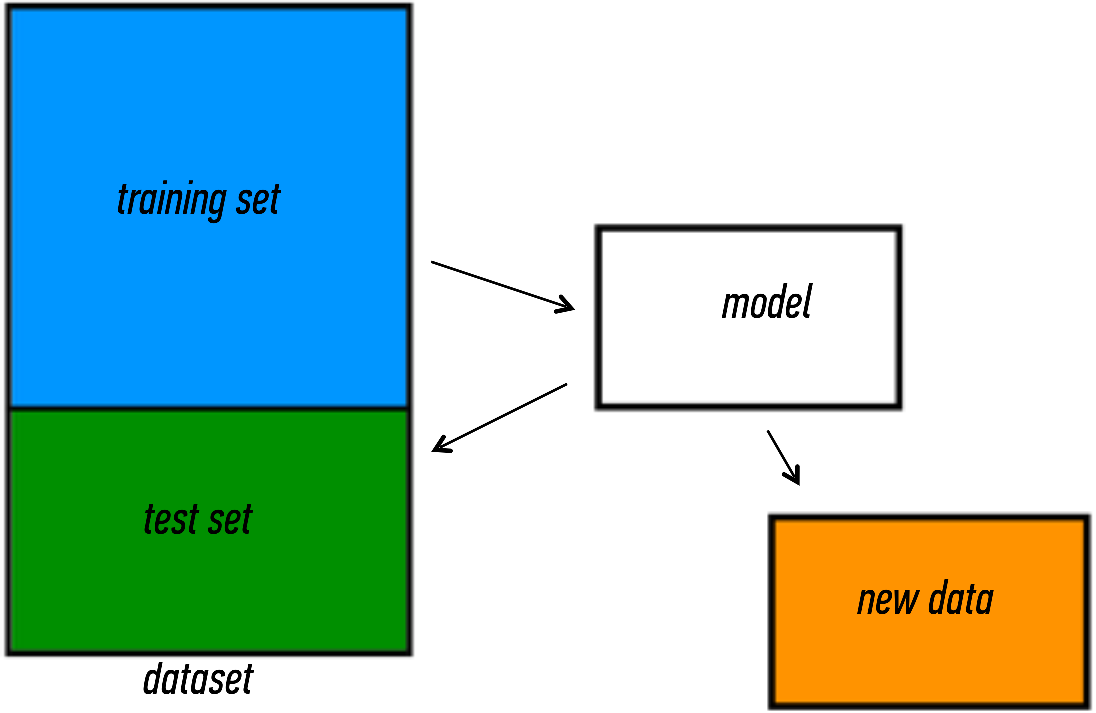
1) split dataset
2) train model
3) test model
4) parameter tuning
5) choose best model
6) train on all data



TEST SET APPROACH

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

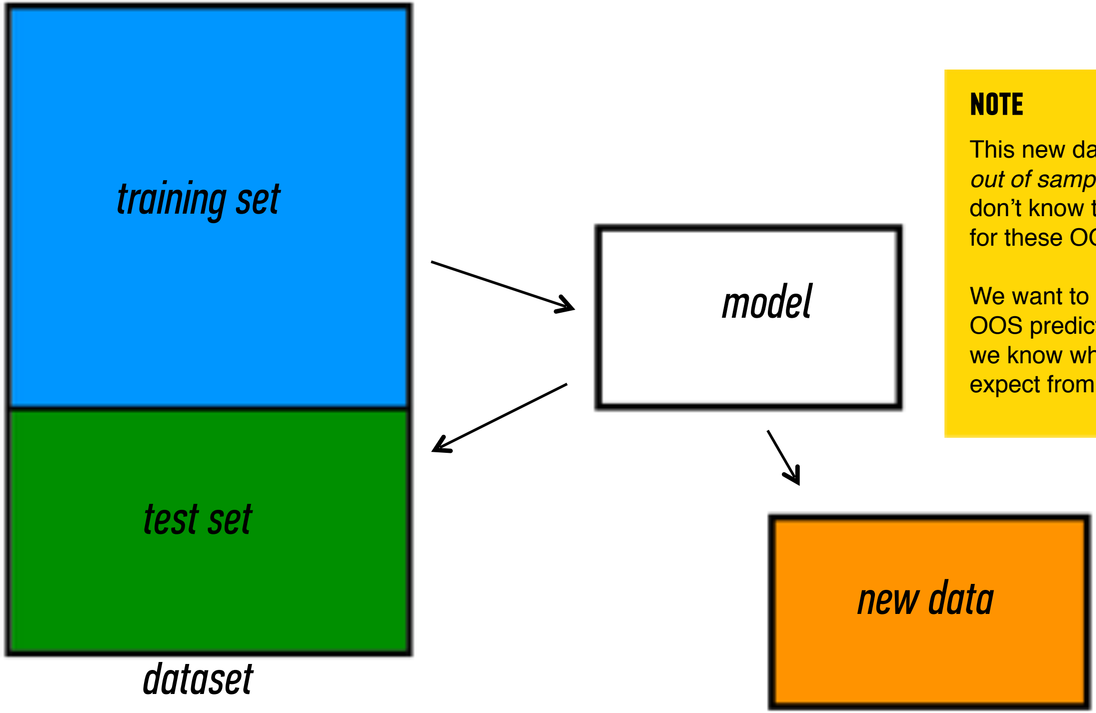
1) split dataset
2) train model
3) test model
4) parameter tuning
5) choose best model
6) train on all data
7) make predictions on new data



TEST SET APPROACH

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.

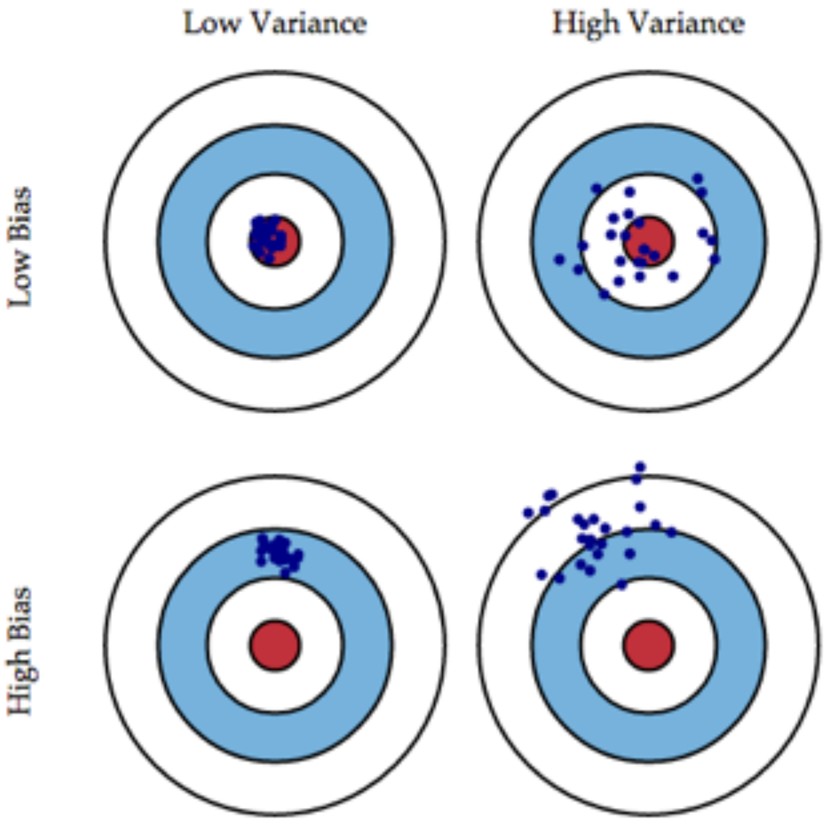
Errors due to Bias

When we are training over multiple data sets we will have different errors. Bias measures how far off in general the predictions are from the actual values

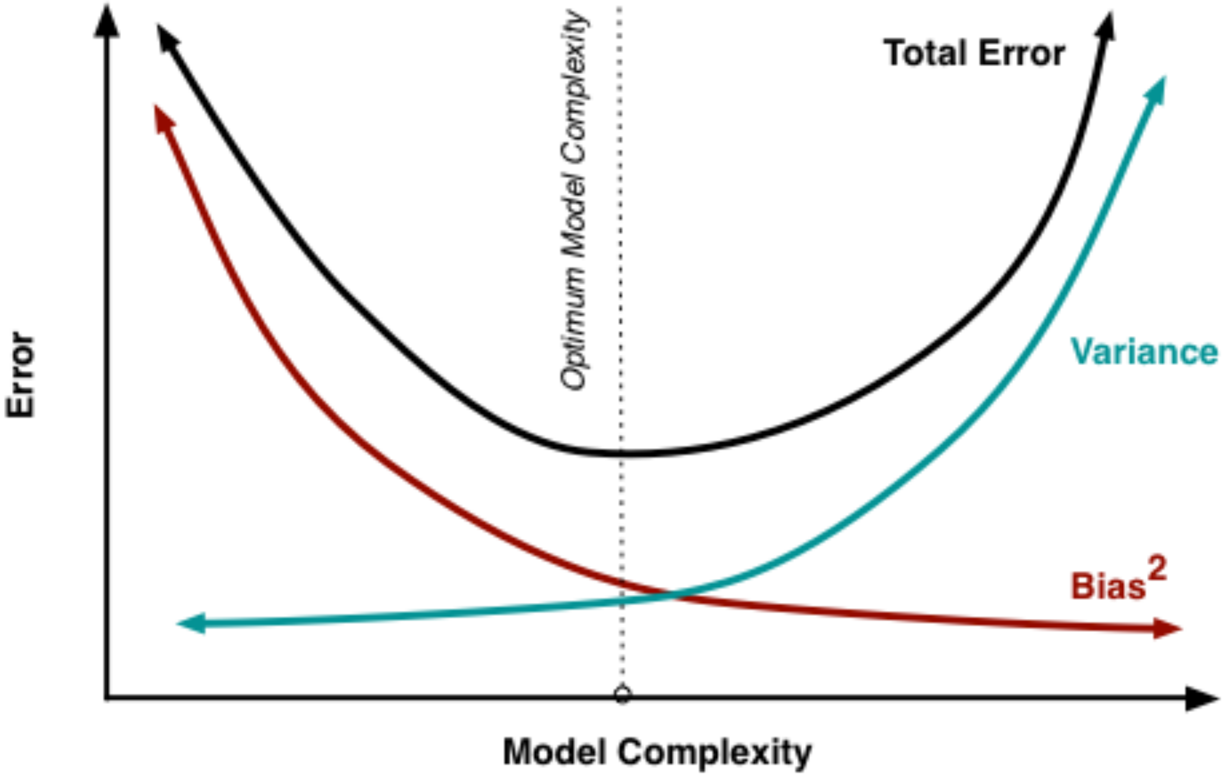
Errors due to Variance

This is how variable our model is for a given data point. The variance calculates how much the predicted are from the actual values

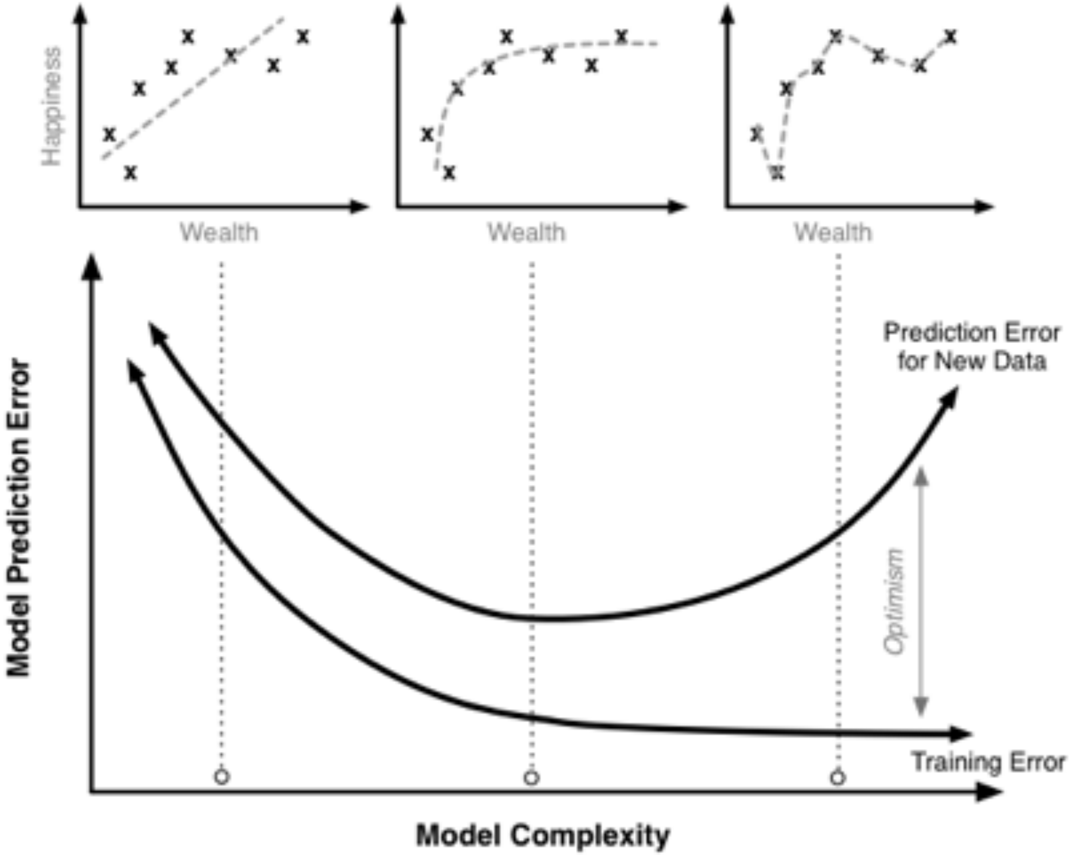
BIAS – VARIANCE TRADEOFF



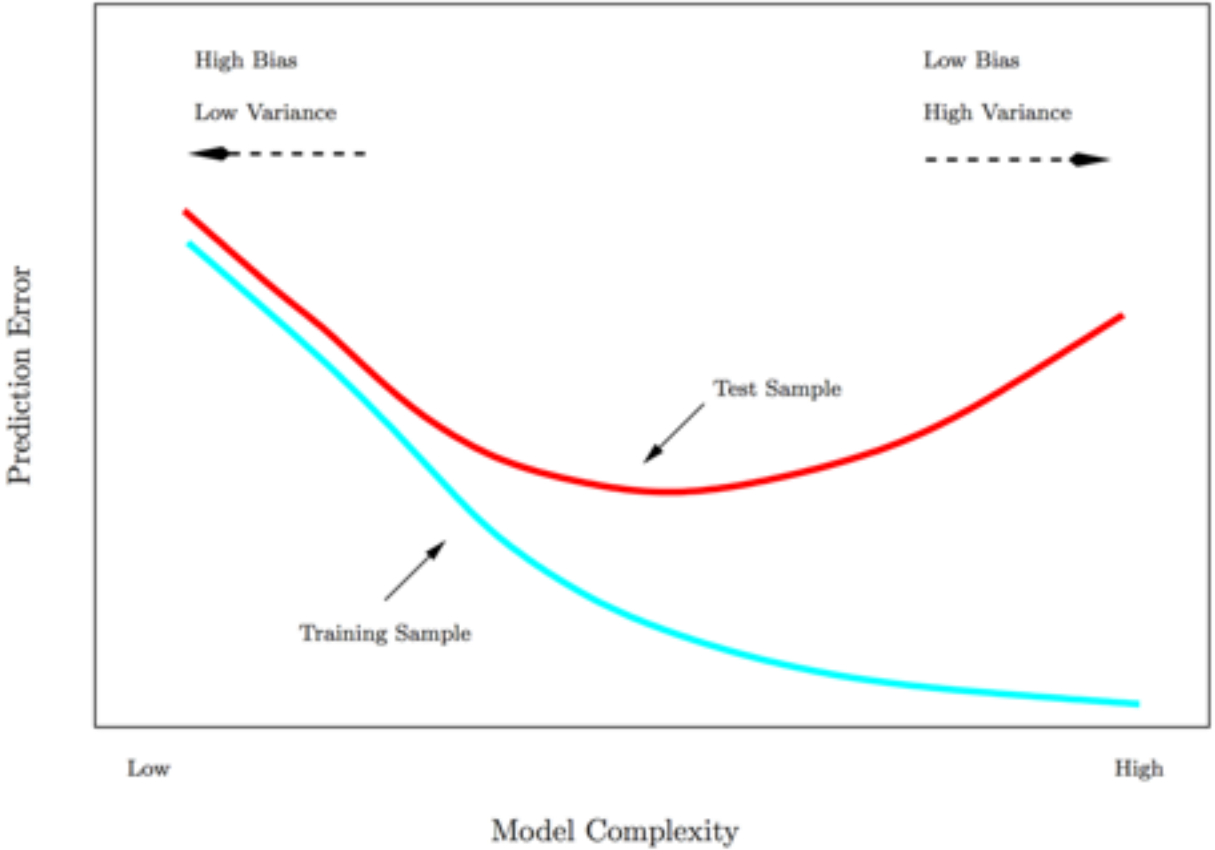
BIAS – VARIANCE TRADEOFF



BIAS – VARIANCE TRADEOFF



BIAS – VARIANCE TRADEOFF



LAB1 - bias / variance

SYNCHING YOUR FORK WITH THE COURSE REPO

1. re-name your labs with lab_name.<yourname>.ipynb (to prevent a conflict)
2. cd <path to the root of your SYD_DAT_6 local repo>
3. commit your changes ahead of sync
   - git status
   - git add .
   - git commit -m “descriptive label for the commit”
   - git status
4. download new material from official course repo (upstream) and merge it
   - git checkout master (ensures you are in the master branch)
   - git fetch upstream
   - git merge upstream/master

DATA SCIENCE PART TIME COURSE

CROSS VALIDATION

Suppose we do the train/test split.

Q: How well does test set error predict OOS?

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

Suppose we do the train/test split.

Q: How well does test set error predict OOS?

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.

Suppose we do the train/test split.

Q: How well does test set error predict OOS?

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.

NOTE

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

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!

Something is still missing!

Q: How can we do better?

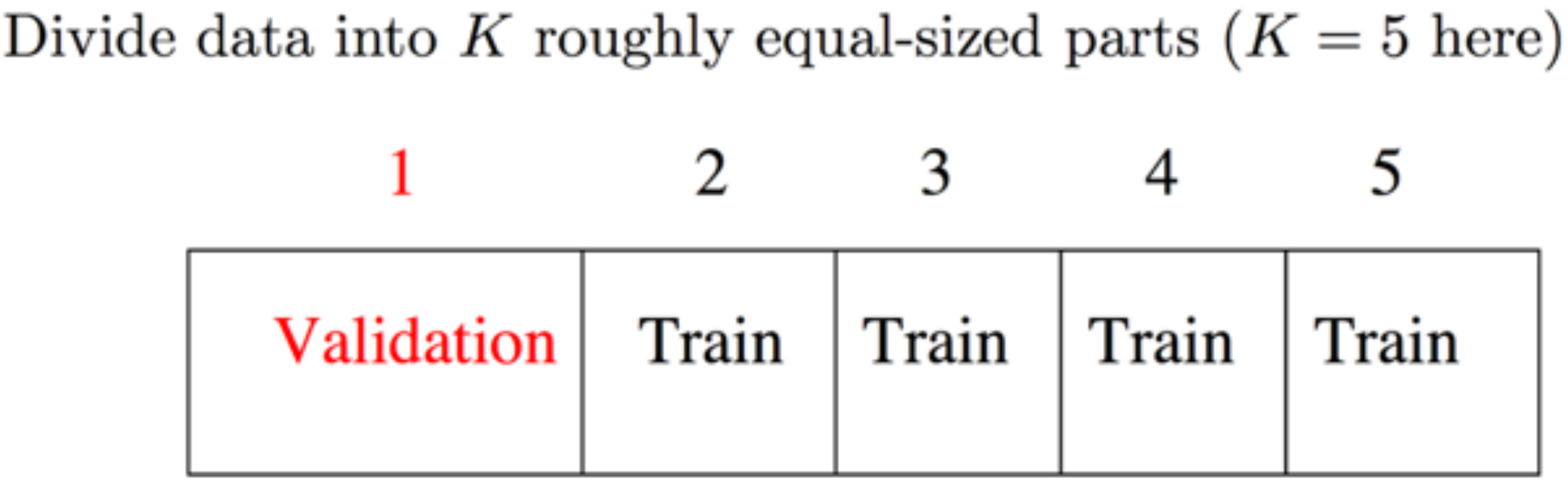
Thought experiment:
Different train/test splits will give us different test set errors.
Q: What if we did a bunch of these and took the average?
A: Now you're talking!

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.

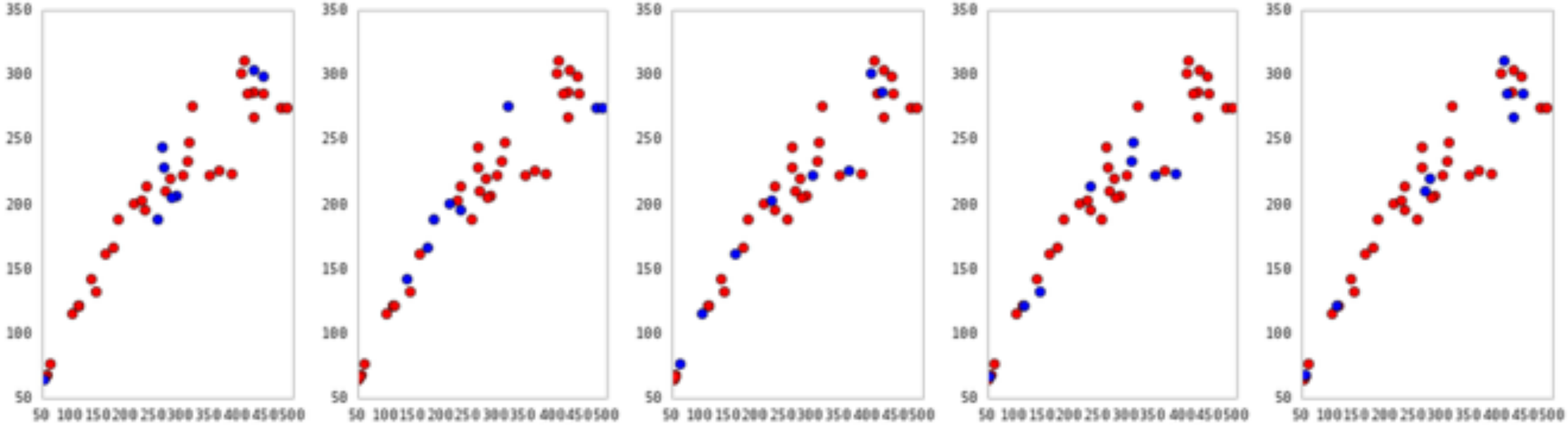
CROSS VALIDATION



Divide data into K roughly equal-sized parts ($K = 5$ here)

1	2	3	4	5
Validation	Train	Train	Train	Train

CROSS VALIDATION



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

CROSS VALIDATION

Features of K-fold cross-validation:

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

CROSS VALIDATION – TAKE CARE

Consider a simple classifier applied to some two-class data:

1. Starting with 5000 predictors and 50 samples, find the 100 predictors having the largest correlation with the class labels.

2. We then apply a classifier such as logistic regression, using only these 100 predictors.

How do we estimate the test set performance of this classifier?

Can we apply cross-validation in step 2, forgetting about step 1?

Can we apply cross-validation in step 2, forgetting about step 1?

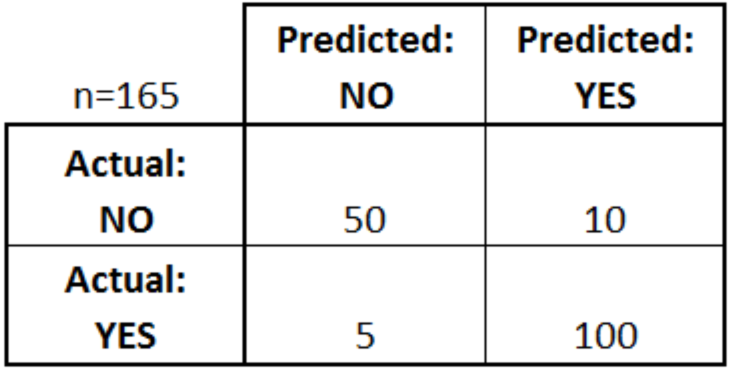
NO

- This would ignore the fact that in Step 1, the procedure has already seen the labels of the training data, and made use of them. This is a form of training and must be included in the validation process.

DATA SCIENCE PART TIME COURSE

CONFUSION MATRIX

Confusion Matrix: table to describe the performance of a classifier



n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

Example: Test for presence of disease
NO = negative test = False = 0
YES = positive test = True = 1

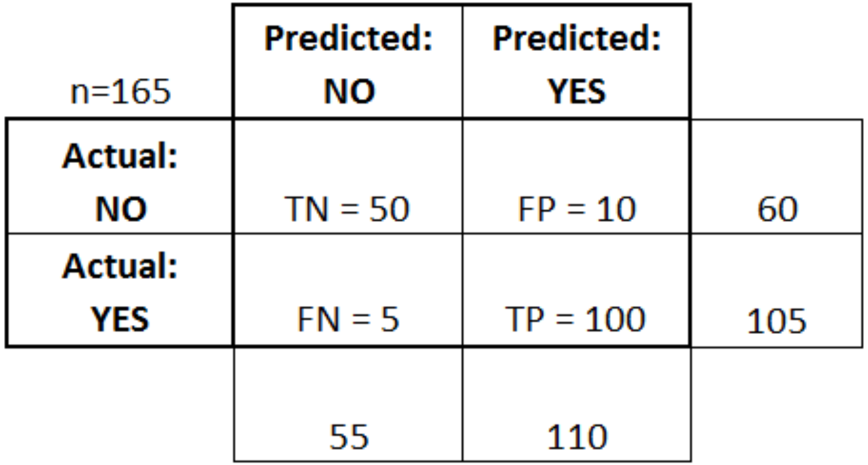
- *How many classes are there?*
- *How many patients?*
- *How many times is disease predicted?*
- *How many patients actually have the disease?*

CONFUSION MATRIX

$$\text{sensitivity} = \frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false negatives}}$$

$$\text{specificity} = \frac{\text{number of true negatives}}{\text{number of true negatives} + \text{number of false positives}}$$

CONFUSION MATRIX



n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

False Positive Rate:

- *When actual value is **negative**, how often is prediction **wrong**?*
- *FP / actual no = 10/60 = 0.17*

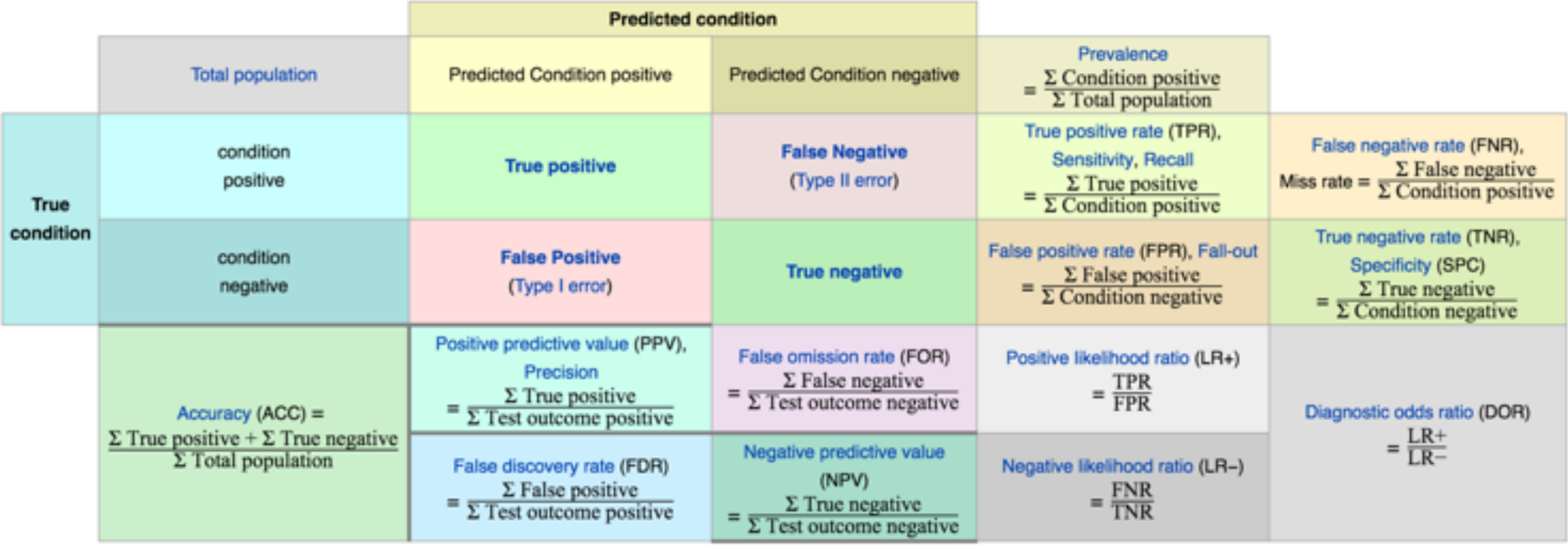
Sensitivity:

- *When actual value is **positive**, how often is prediction **correct**?*
- *TP / actual yes = 100/105 = 0.95*
- *“True Positive Rate” or “Recall”*

Specificity:

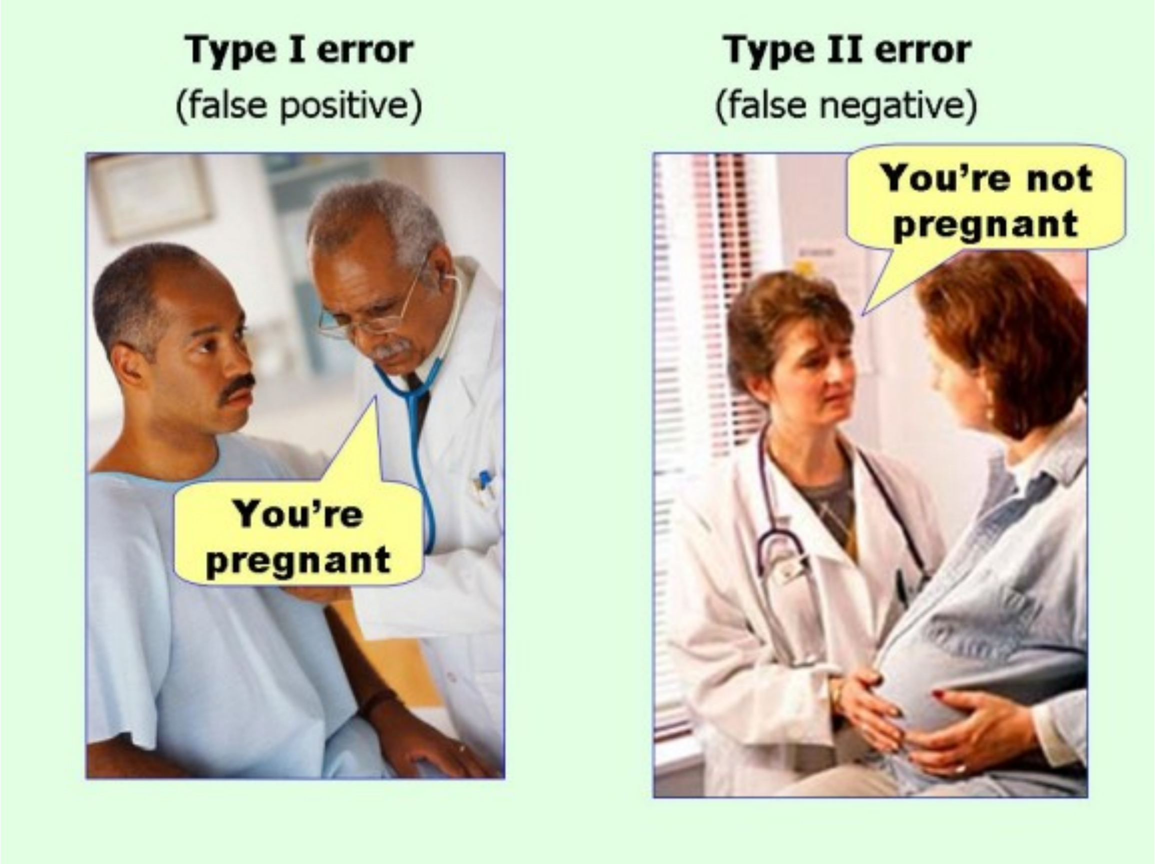
- *When actual value is **negative**, how often is prediction **correct**?*
- *TN / actual no = 50/60 = 0.83*

CONFUSION MATRIX



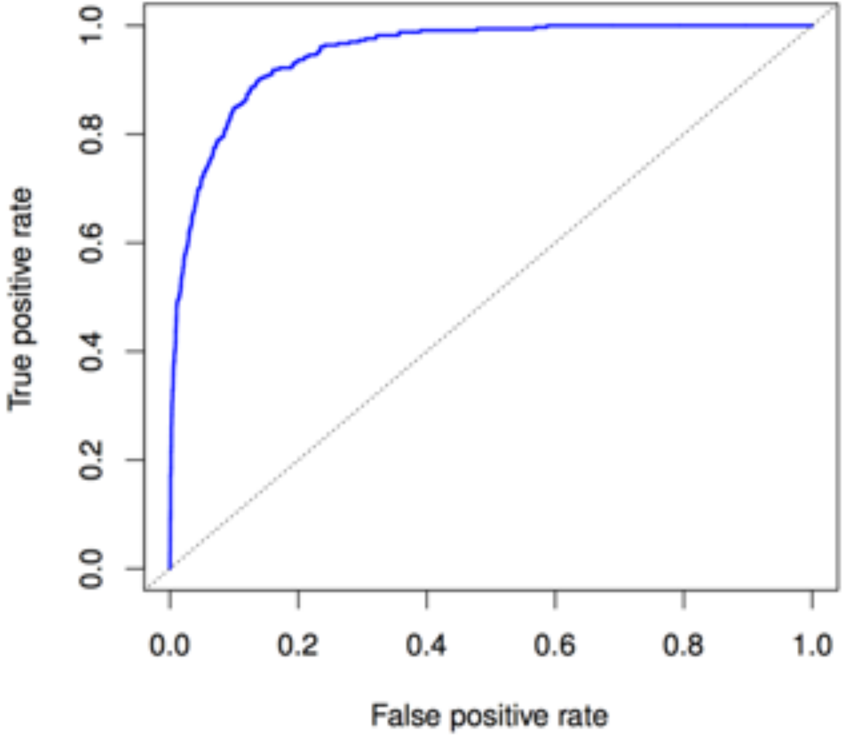
		Predicted condition			
	Total population	Predicted Condition positive	Predicted Condition negative	Prevalence $= \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	
True condition	condition positive	True positive	False Negative (Type II error)	True positive rate (TPR), Sensitivity, Recall $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$
	condition negative	False Positive (Type I error)	True negative	False positive rate (FPR), Fall-out $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	True negative rate (TNR), Specificity (SPC) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$
	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	Positive predictive value (PPV), Precision $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Test outcome positive}}$	False omission rate (FOR) $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Test outcome negative}}$	Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR−}}$
		False discovery rate (FDR) $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Test outcome positive}}$	Negative predictive value (NPV) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Test outcome negative}}$	Negative likelihood ratio (LR–) $= \frac{\text{FNR}}{\text{TNR}}$	

CONFUSION MATRIX



SETTING THE CLASSIFICATION THRESHOLD

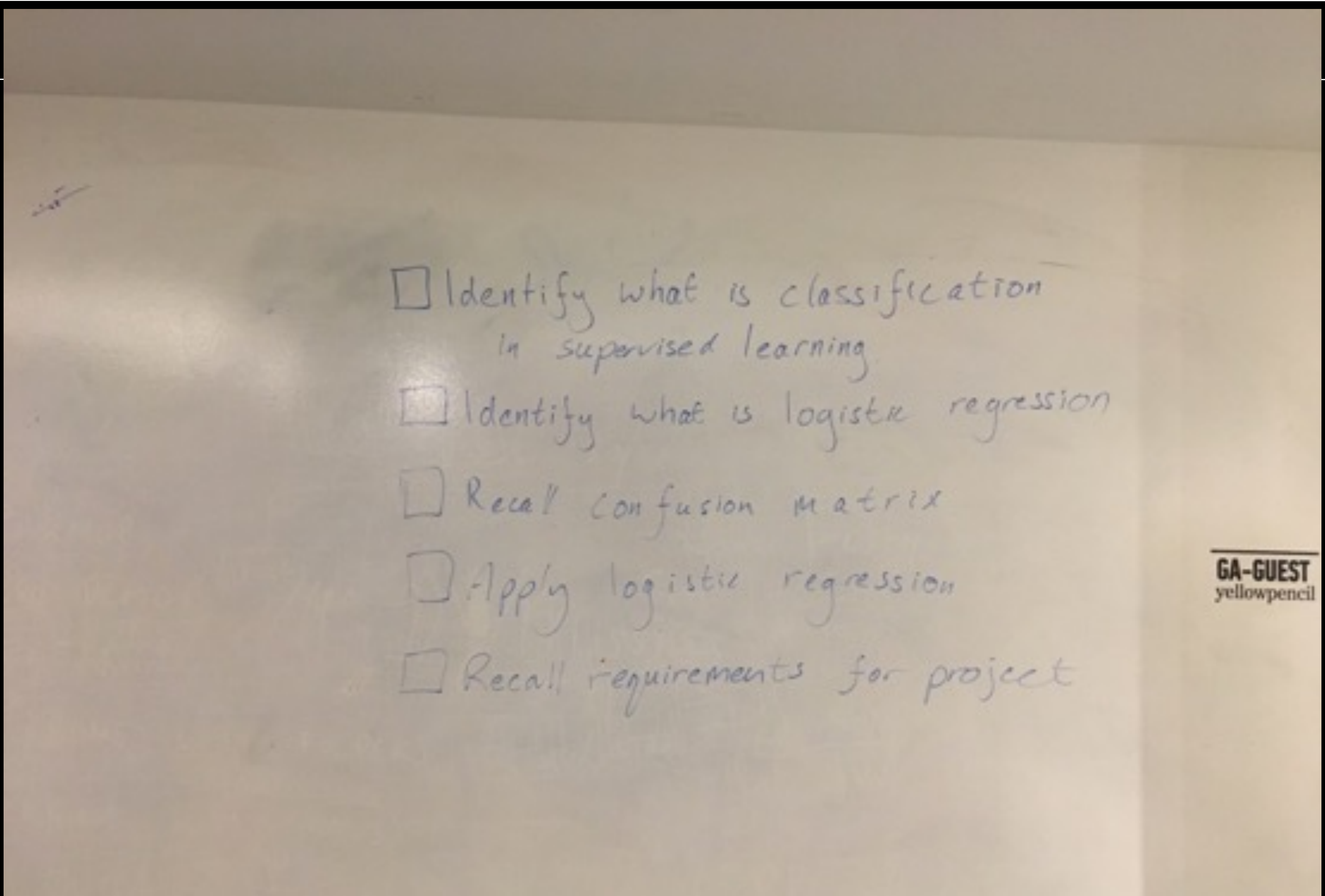
ROC CURVE


True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0

LAB - evaluation metrics

DISCUSSION TIME

- Questions from previous lesson?
- What are we trying to do when we use Logistic Regression?
- How would you evaluate a regression problem?



□ Identify what is classification in supervised learning

□ Identify what is logistic regression

□ Recall confusion matrix

□ Apply logistic regression

□ Recall requirements for project

GA-GUEST
yellowpencil

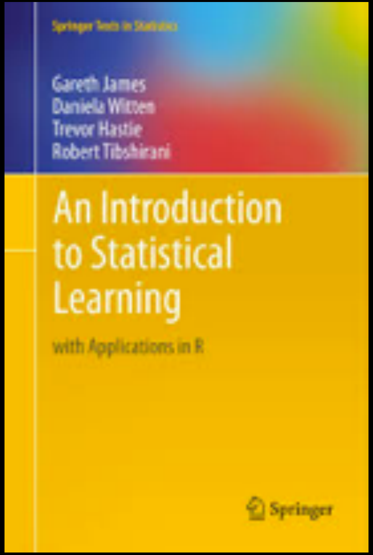
QUESTIONS

- What are we trying to do when we use Logistic Regression?
- Why use it instead of Linear Regression for classification?
- Evaluating a logistic Regression model

HOMEWORK

An Introduction to Statistical Learning

- Chapter 5 - Resampling Techniques
- Caltech's Learning From Data course visualising bias and variance (15 mins)
    - http://work.caltech.edu/library/081.html
- Rahul Patwari has a great video on ROC Curves (12 minutes)
    - https://www.youtube.com/watch?v=21Igj5Pr6u4
- Have a look at scikit-learn’s documentation on model evaluation
    - http://scikit-learn.org/stable/modules/model_evaluation.html



Springer Texts in Statistics

Gareth James
Daniela Witten
Trevor Hastie
Robert Tibshirani

An Introduction to Statistical Learning

with Applications in R

Springer