How to validate model?

資料科學 Data Science
張家銘 Jia-Ming Chang
政治大學資訊科學系

Copyright declaration 版權說明

- Some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani.
- The web site of the book
- The credit of individual is indicated in the bottom part of the slide.
  - ie.,

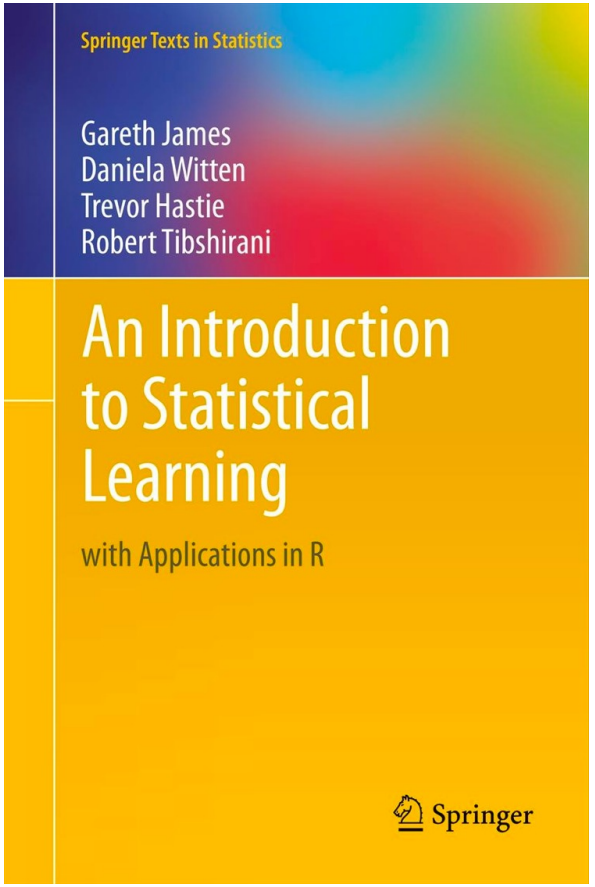


Figure 3.18, *An Introduction to Statistical Learning with Applications in R,* by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani

Copyright declaration 版權說明

- Some of the figures in this presentation are taken from "Practical Data Science with R (Manning, 2019)"
- The web site of the book
- The credit of individual is indicated in the bottom part of the slide.
  - ie.,

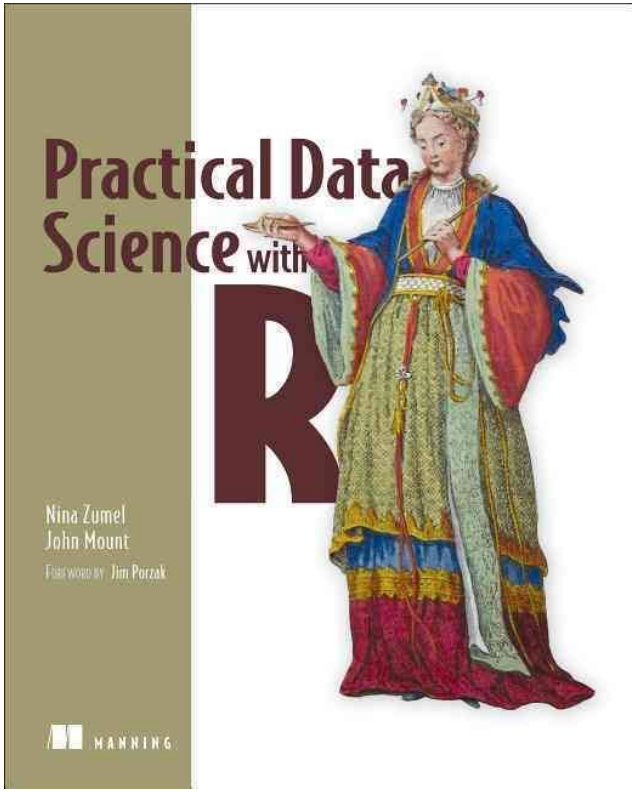
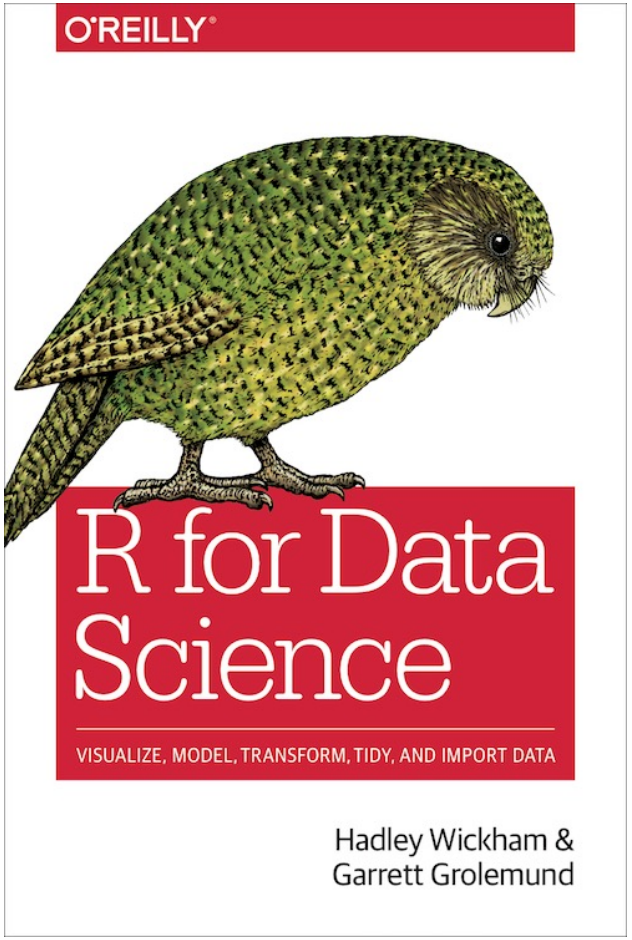


Figure 7.6, *Practical Data Science with R* by Nina Zumel and John Mount

Copyright declaration 版權說明

- Some of the figures in this presentation are taken from "R for Data Science" under the Creative Commons Attribution-NonCommercial-NoDerivs 3.0 License.
- The web site of the book
- The credit of individual is indicated in the bottom part.
  - ie.,

R for Data Science by Garrett Grolemund, Hadley Wickham



Recap for the last week

Confusion Matrix

- Definition
  - True positive
  - False positive
  - True negative
  - False negative

	pred	
Good.Loan	BadLoan	GoodLoan
BadLoan	41	259
GoodLoan	13	687

Worked example [edit]

Suppose the fecal occult blood (FOB) screen test is used in 2030 people to look for bowel cancer:

expected

pred

		Patients with bowel cancer (as confirmed on endoscopy)		
		Condition positive	Condition negative	
Fecal occult blood screen test outcome	Test outcome positive	True positive (TP) = 20	False positive (FP) = 180	Positive predictive value = TP / (TP + FP) = 20 / (20 + 180) = **10%**
	Test outcome negative	False negative (FN) = 10	True negative (TN) = 1820	Negative predictive value = TN / (FN + TN) = 1820 / (10 + 1820) ≈ **99.5%**
		Sensitivity = TP / (TP + FN) = 20 / (20 + 10) ≈ **67%**	Specificity = TN / (FP + TN) = 1820 / (180 + 1820) = **91%**	

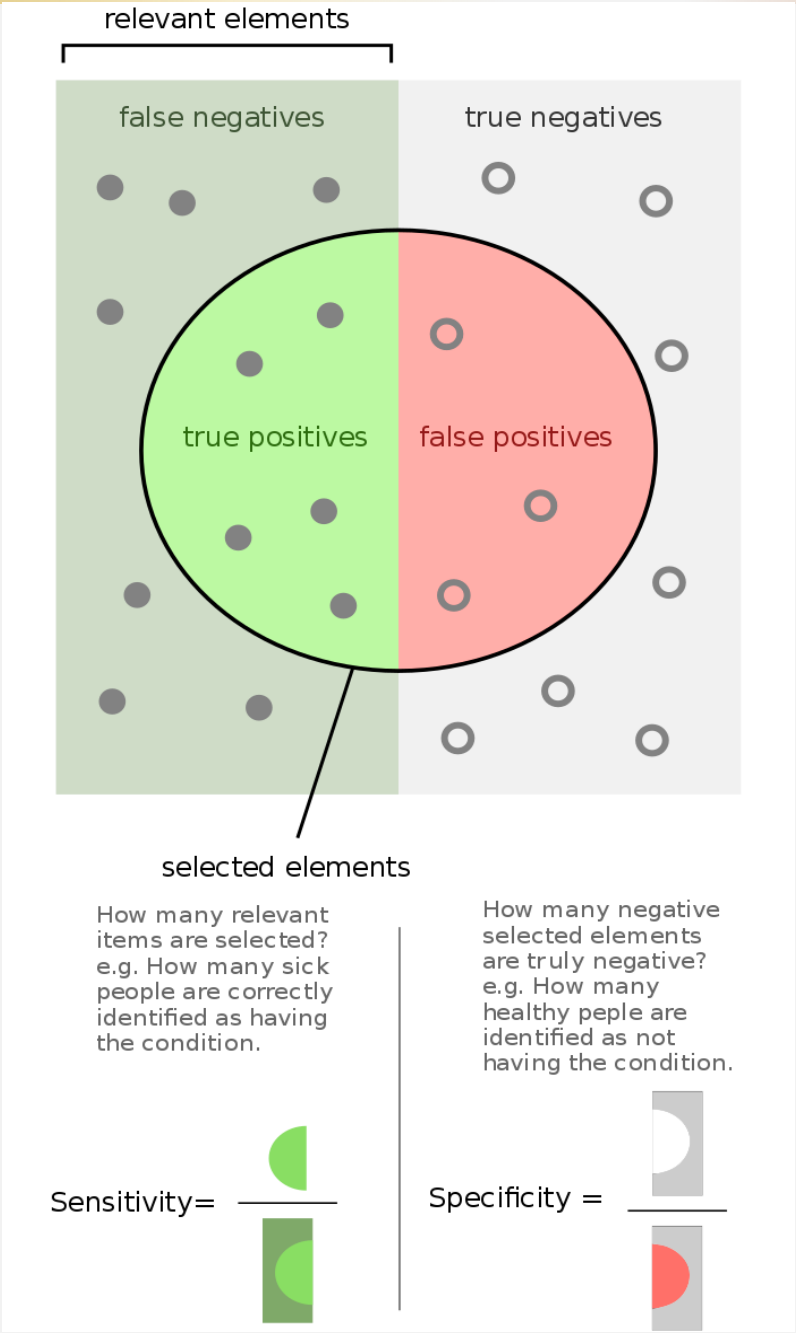
		predicted condition			
	total population	prediction positive	prediction negative	Prevalence = $\frac{\Sigma \text{ condition positive}}{\Sigma \text{ total population}}$	
true condition	condition positive	**True Positive (TP)**	**False Negative (FN)** (type II error)	True Positive Rate (TPR), Sensitivity, Recall, Probability of Detection = $\frac{\Sigma \text{ TP}}{\Sigma \text{ condition positive}}$	False Negative Rate (FNR), Miss Rate = $\frac{\Sigma \text{ FN}}{\Sigma \text{ condition positive}}$
	condition negative	**False Positive (FP)** (Type I error)	**True Negative (TN)**	False Positive Rate (FPR), Fall-out, Probability of False Alarm = $\frac{\Sigma \text{ FP}}{\Sigma \text{ condition negative}}$	True Negative Rate (TNR), Specificity (SPC) = $\frac{\Sigma \text{ TN}}{\Sigma \text{ condition negative}}$
	Accuracy = $\frac{\Sigma \text{ TP} + \Sigma \text{ TN}}{\Sigma \text{ total population}}$	Positive Predictive Value (PPV), Precision = $\frac{\Sigma \text{ TP}}{\Sigma \text{ prediction positive}}$	False Omission Rate (FOR) = $\frac{\Sigma \text{ FN}}{\Sigma \text{ prediction negative}}$	Positive Likelihood Ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$	Diagnostic Odds Ratio (DOR) = $\frac{\text{LR+}}{\text{LR−}}$
		False Discovery Rate (FDR) = $\frac{\Sigma \text{ FP}}{\Sigma \text{ prediction positive}}$	Negative Predictive Value (NPV) = $\frac{\Sigma \text{ TN}}{\Sigma \text{ prediction negative}}$	Negative Likelihood Ratio (LR−) = $\frac{\text{FNR}}{\text{TNR}}$	

Credit by https://en.wikipedia.org/wiki/Positive_and_negative_predictive_values

Common Classification Performance Measures

Table 5.5 Example classifier performance measures

Measure	Formula	Email spam example	Akismet spam example
Accuracy	`(TP+TN)/(TP+FP+TN+FN)`	0.9214	0.9987
Precision	`TP/(TP+FP)`	0.9187	0.9999
Recall	`TP/(TP+FN)`	0.8778	0.9988
Sensitivity	`TP/(TP+FN)`	0.8778	0.9988
Specificity	`TN/(TN+FP)`	0.9496	0.9965



Credit by https://en.wikipedia.org/wiki/Sensitivity_and_specificity

Common Classification Performance Measure

Sensitivity TPR	"We have to cut a lot of spam, otherwise the user won't see a benefit."	"If we cut spam down to 1% of what it is now, would that be a good user experience?"
Specificity TNR	"We must be at least *three nines* on legitimate email; the user must see at least 99.9% of their non-spam email."	"Will the user tolerate missing 0.1% of their legitimate email, and should we keep a spam folder the user can look at?"

Recommend writing the business goals as maximizing sensitivity while maintaining a specificity of at least 0.999

com end writing the business goals as maximizing specificity while maintaining a o en ivity at least 0.999

Evaluating scoring models

- RMSE
  - sqrt(mean((*prediction-actualValues*)^2))
- R-SQUARED, R^2
  - 1 - sum((*pred-actVal*)^2)/sum((mean(*actVal*)-*actVal*)^2)
  - *R-squared = correlation*^{2}

R-squared = correlation²

- Under "general conditions", as Wikipedia says, R^2 is also the square of the correlation between the actual and predicted outcomes.
  1. *f* is the model that minimizes squared-error loss
  2. Because it is the optimum (in the sense of item 1), there is no shift of *f* that will improve the fit.
  3. Because it is the optimum (in the sense of item 1), there is no scaling of *f* that will improve the fit.

Evaluating probability models

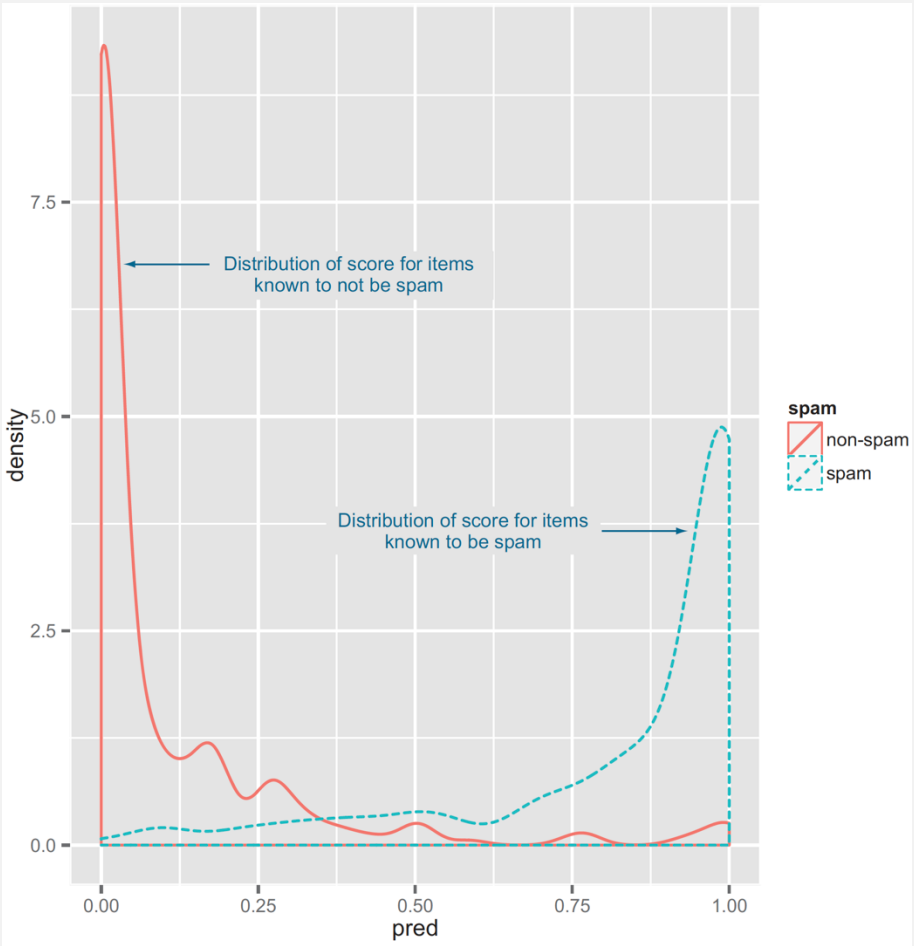


Figure 5.8 **Distribution of score broken up by known classes**

The Receiver Operating Characteristic Curve

- ROC curve
- AUC or area under the curve

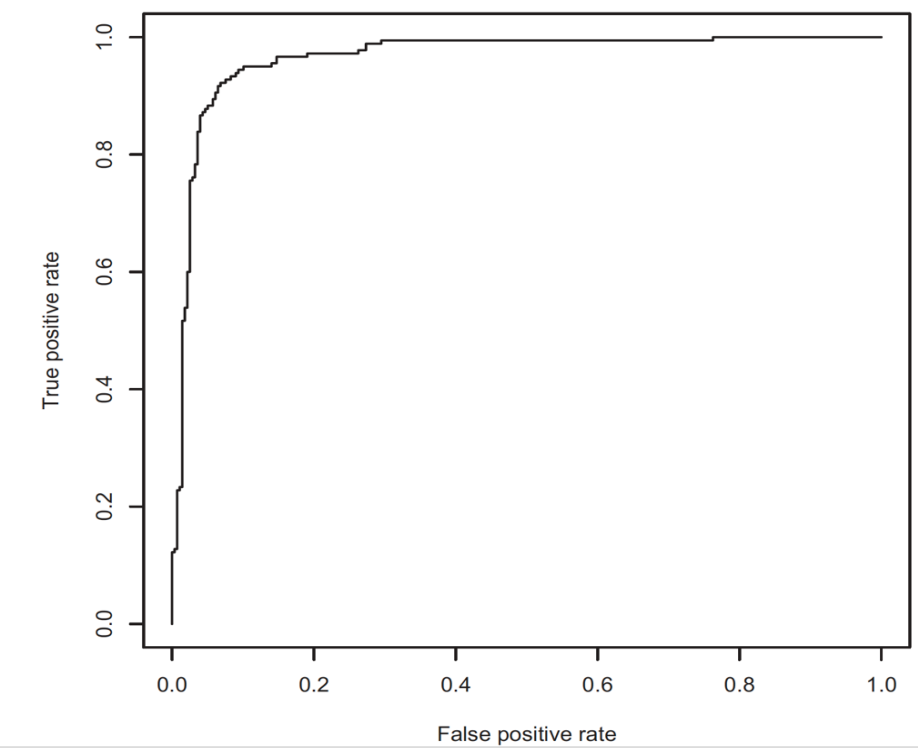
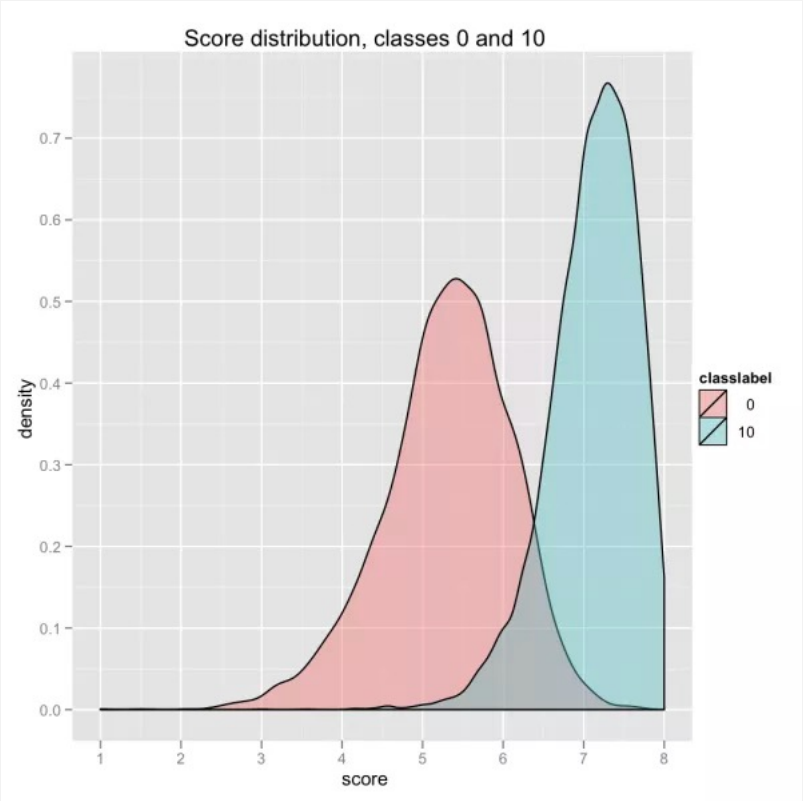


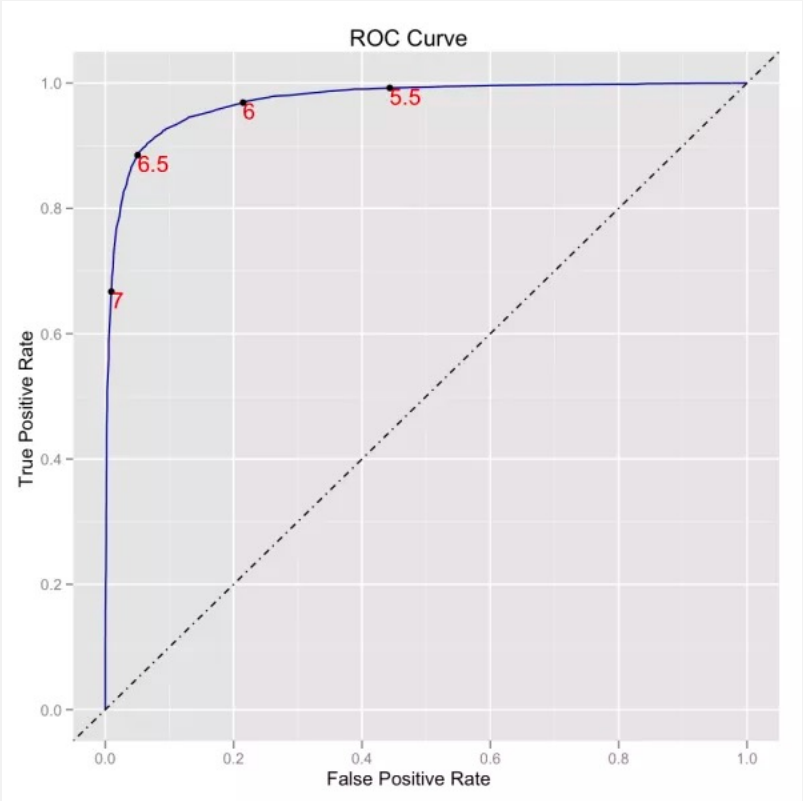
Figure5.9, *Practical Data Science with R* by Nina Zumel, John Mount

More on ROC/AUC

- The ROC curve is a useful tool, but you have to use it for appropriate tasks.
- The ROC curve is useful tool designing a classifier from a scoring function (though I prefer the “double hump graph”).

More on ROC/AUC





Credit by Win-Vector Blog, http://www.win-vector.com/blog/2013/01/more-on-rocauc/

More on ROC/AUC

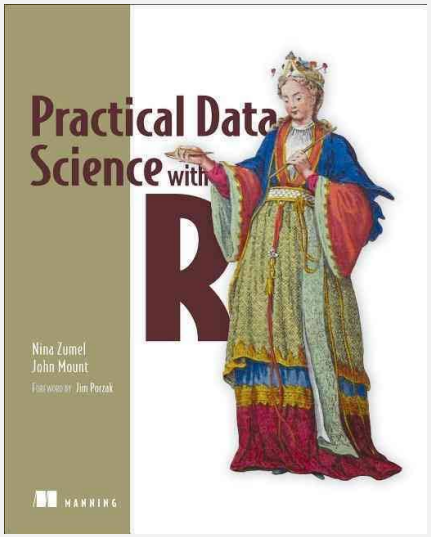
- the speed the score parameter is moving along the curve and what density of data is associate with each score.
- you get the first 1/2 units of area for free (no credit to you there) it is only the second 1/2 that is at all meaningful.

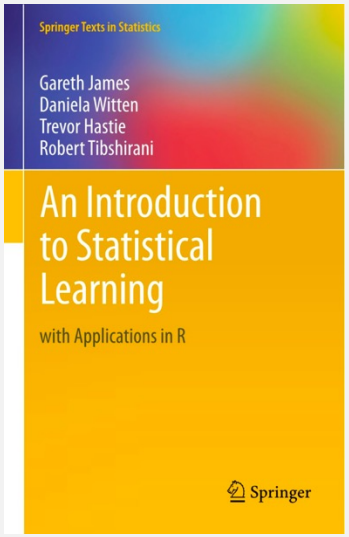
報告中 PCR 及快篩的精密性

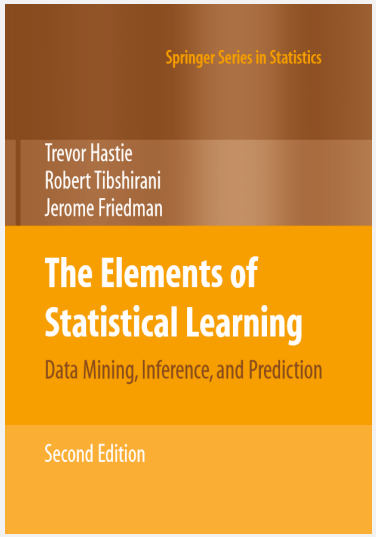
檢測工具＼普篩對象	呼吸道症狀就醫人口（4800000）		無症狀人口（18000000）	
	盛行率極大值：（π=0.0018）	盛行率合理值：（π=0.000016）	盛行率極大值：（π=0.0018）	盛行率合理值：（π=0.00000056）
PCR 之精密性（真陽性/採檢陽性）	0.9448（8208/8687）	0.1319（71/551）	0.9448（30780/32577）	0.0050（9/1809）
快篩之精密性（真陽性/採檢陽性）	0.1191（6480/54394）	0.0012（56/48056）	0.1191（24300/203976）	0.0000（8/180008）

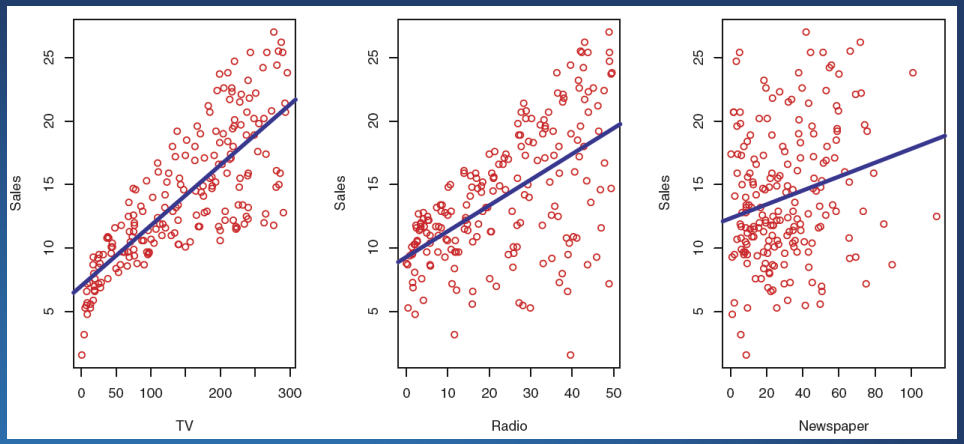
Today

1. appendix B: Important statistical concepts
2. Cha 02. Statistical Learning
3. Cha 07. Model Assessment and Selection









$$\hat{Y} = \hat{f}(X)$$

Given $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ where $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$

Figure 2.1, *An Introduction to Statistical Learning with Applications in R* by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

How Do We Estimate *f*?

- find a function $\hat{f}$ such that $\hat{Y} \approx \hat{f}(X)$ for any observation (X, Y)
  - parametric
  - non-parametric

Parametric Methods

- the problem of estimating f
  - down to one of estimating a set of parameters
- make an simple assumption about the linear functional form
  - $f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$

Parametric Methods

- income $= \beta_0 + \beta_1$education $+ \beta_2$seniority
- the functional form used to estimate f is very different from the true f, in which case the resulting model will not fit the data well.

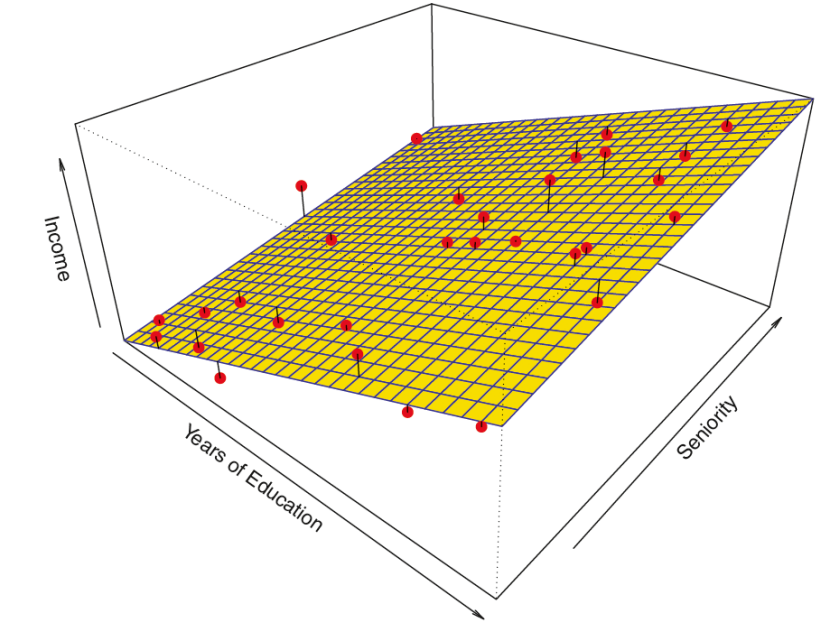


Figure 2.4, *An Introduction to Statistical Learning with Applications in R* by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

Non-parametric Methods

- do not make explicit assumptions about the functional form of *f*
  - they have the potential to accurately fit a wider range of possible shapes for *f*
  - do not reduce the problem of estimating f to a small number of parameters, a very large number of observations (far more than is typically needed for a parametric approach) is required in order to obtain an accurate estimate for *f*

A smooth thin-plate spline fit

https://en.wikipedia.org/wiki/Spline_(mathematics)

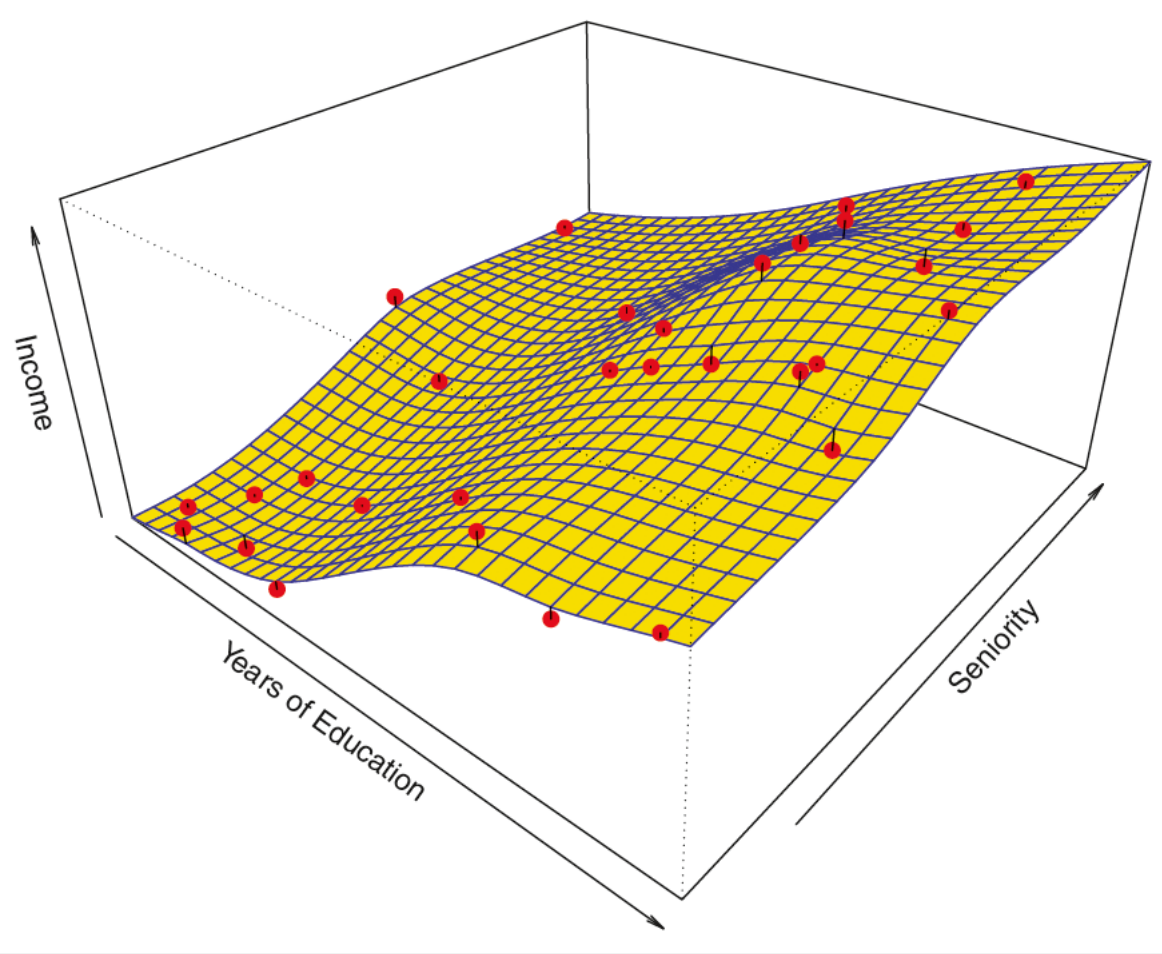
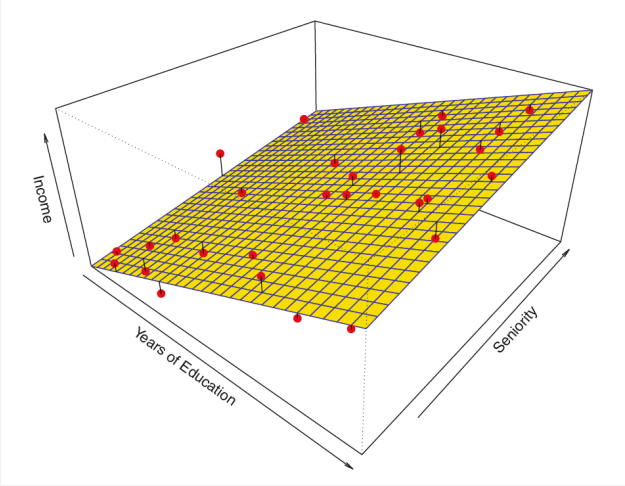
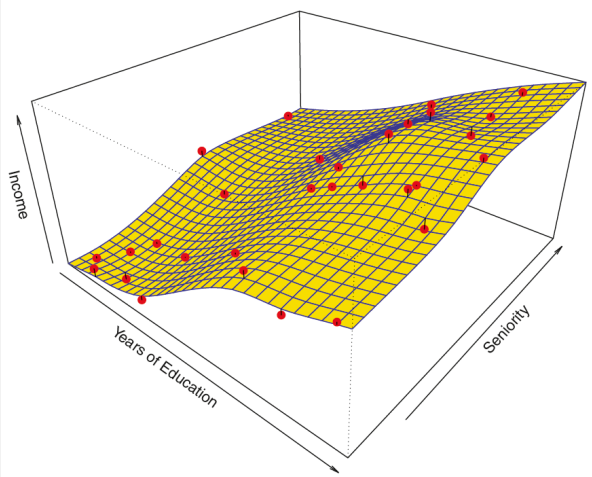


Figure 2.5, *An Introduction to Statistical Learning with Applications in R* by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

Which model fits well?





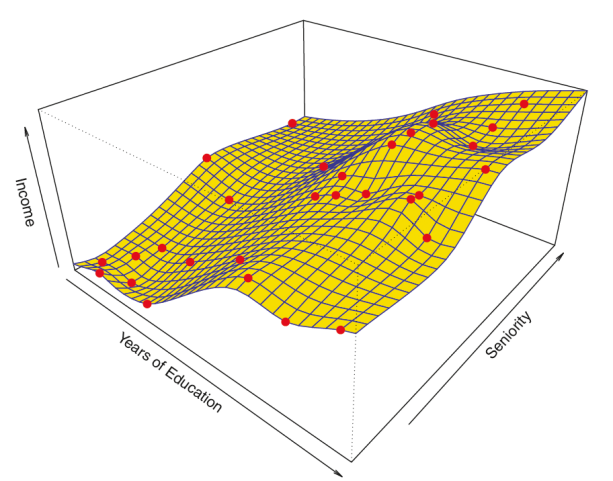


Figure 2.4,2.5,2.6, *An Introduction to Statistical Learning with Applications in R* by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

Prediction Accuracy v.s. Model Interpretability (複雜模型較難解釋)

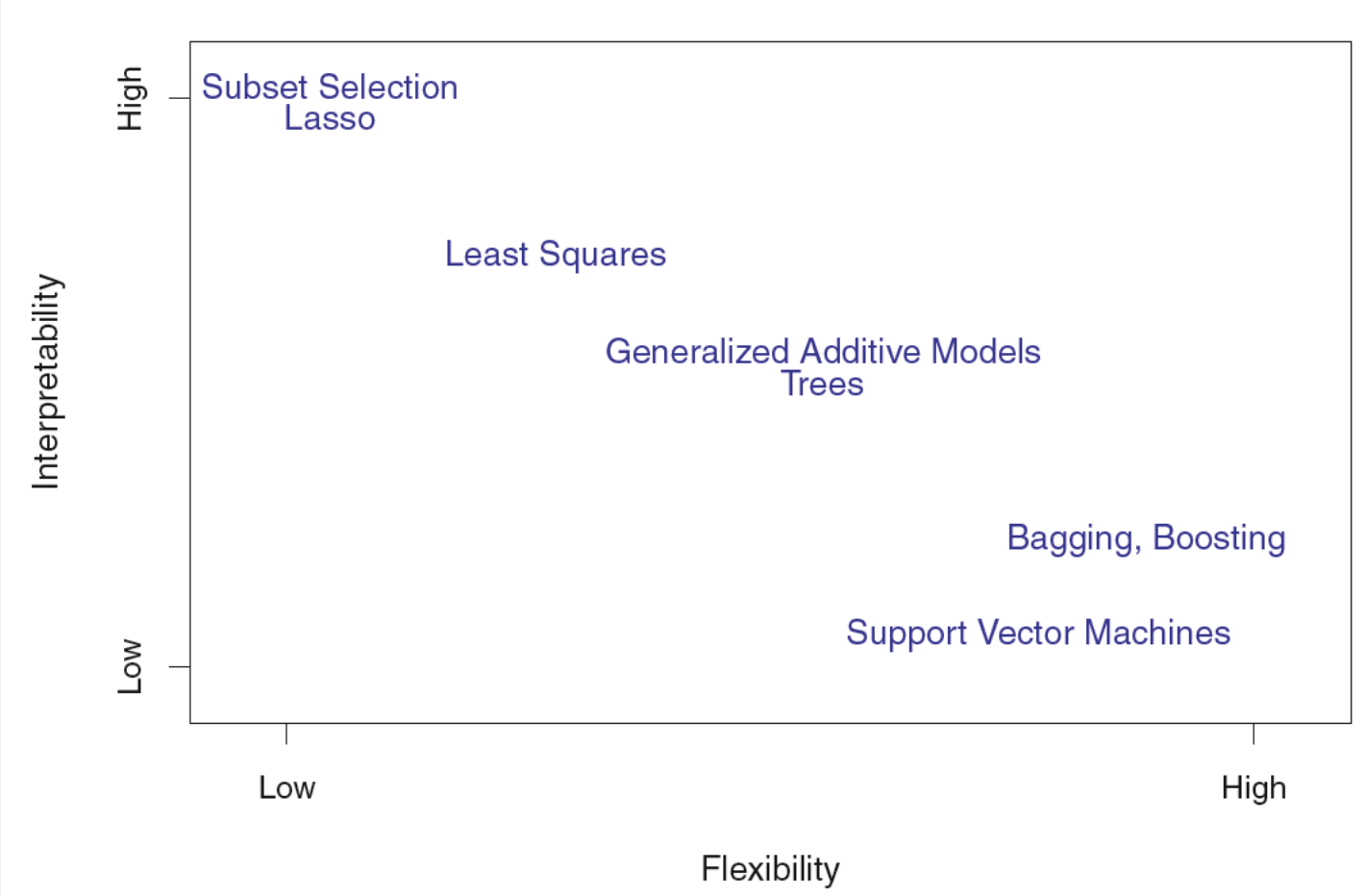


Figure 2.7, *An Introduction to Statistical Learning with Applications in R* by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

Which one is more challenging?

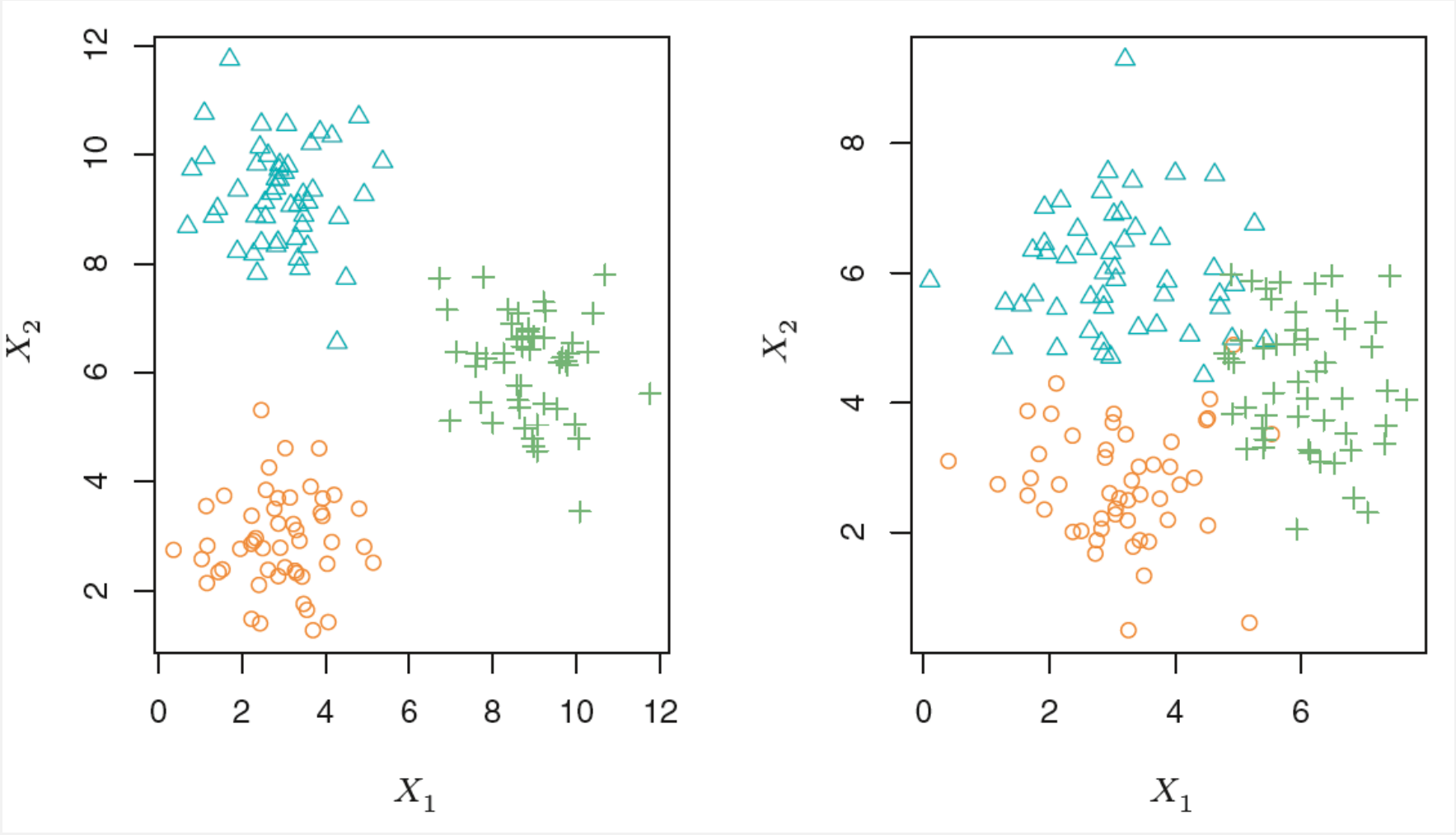


Figure 2.8, *An Introduction to Statistical Learning with Applications in R* by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

Model quality

Bias-Variance Decomposition

Goal

- Model selection
  - estimating the performance of different models in order to choose the best one.
- Model assessment
  - having chosen a final model, estimating its prediction error (generalization error) on new data.

Statistical theory

- EXCHANGEABILITY: for any permutation $j_1,\ldots j_m$ of $1,\cdots,m$, the joint probability of seeing $x[i,],y[i]$ is equal to the joint probability of seeing $x[j_i,],y[j_i]$.
- so, we can make train/test splits => even train/calibrate/test splits ?
- once you look at your test data, it's less exchangeable with what will be seen in production in the future (discuss in cross-validation)

{ training set
calibration / validation set
test set

The slide is about a concept in statistical theory known as exchangeability. Exchangeability is a property of a sequence of random variables, where the joint probability distribution is invariant to permutations of the indices. That means, no matter how you reorder the sequence, the joint probability remains the same.

The implication of this for model training is significant. If we assume that our data points are exchangeable, we can randomly split our dataset into training and test sets, and expect that these splits are representative of the whole data. This is a common practice in machine learning to evaluate the performance of a model.

However, when you use your test data to make decisions about the model, such as calibrating it or choosing between models, the test data becomes less exchangeable with future unseen data because it has influenced the model. This is why sometimes a third split, a calibration or validation set, is used: to make decisions about the model without compromising the test set.

Finally, it's noted that once you look at your test data, it's less exchangeable with what will be seen in production in the future. This is a warning against overfitting to the test set, which can happen when you tune your model too closely to the specificities of your test data rather than to the underlying data generating process. The mention of "discuss in cross-validation" suggests that a more detailed discussion on this topic will follow, likely addressing how cross-validation can be used to mitigate some of these issues.

Training v.s. Testing error

mean squared error

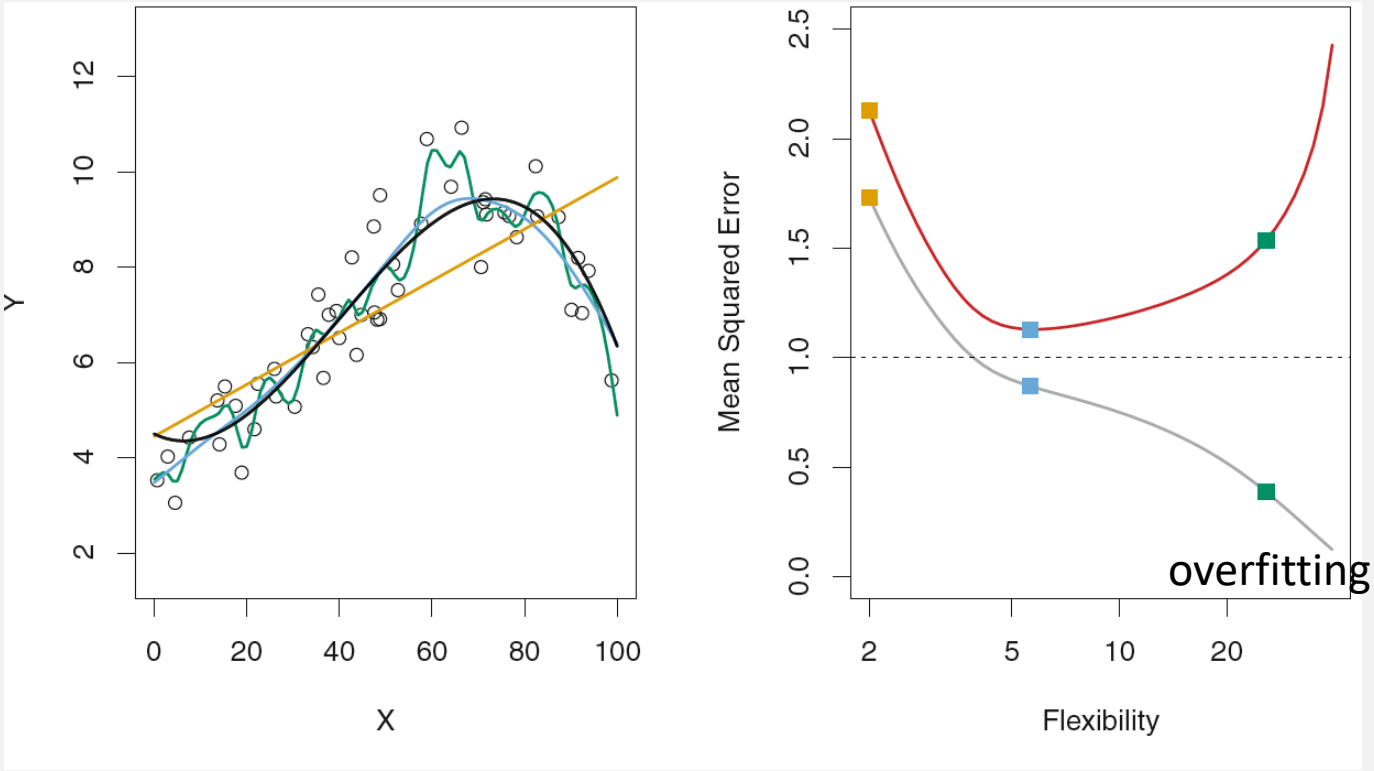
- Test MSE : (x_0, y_0) is a previously unseen test observation not used to train the statistical learning method
- $AVE\left(y_0 - \hat{f}(x_0)\right)^2$

$\hat{f}(x_0)$: model prediction

mean squared

Three estimates of *f (black curve)*

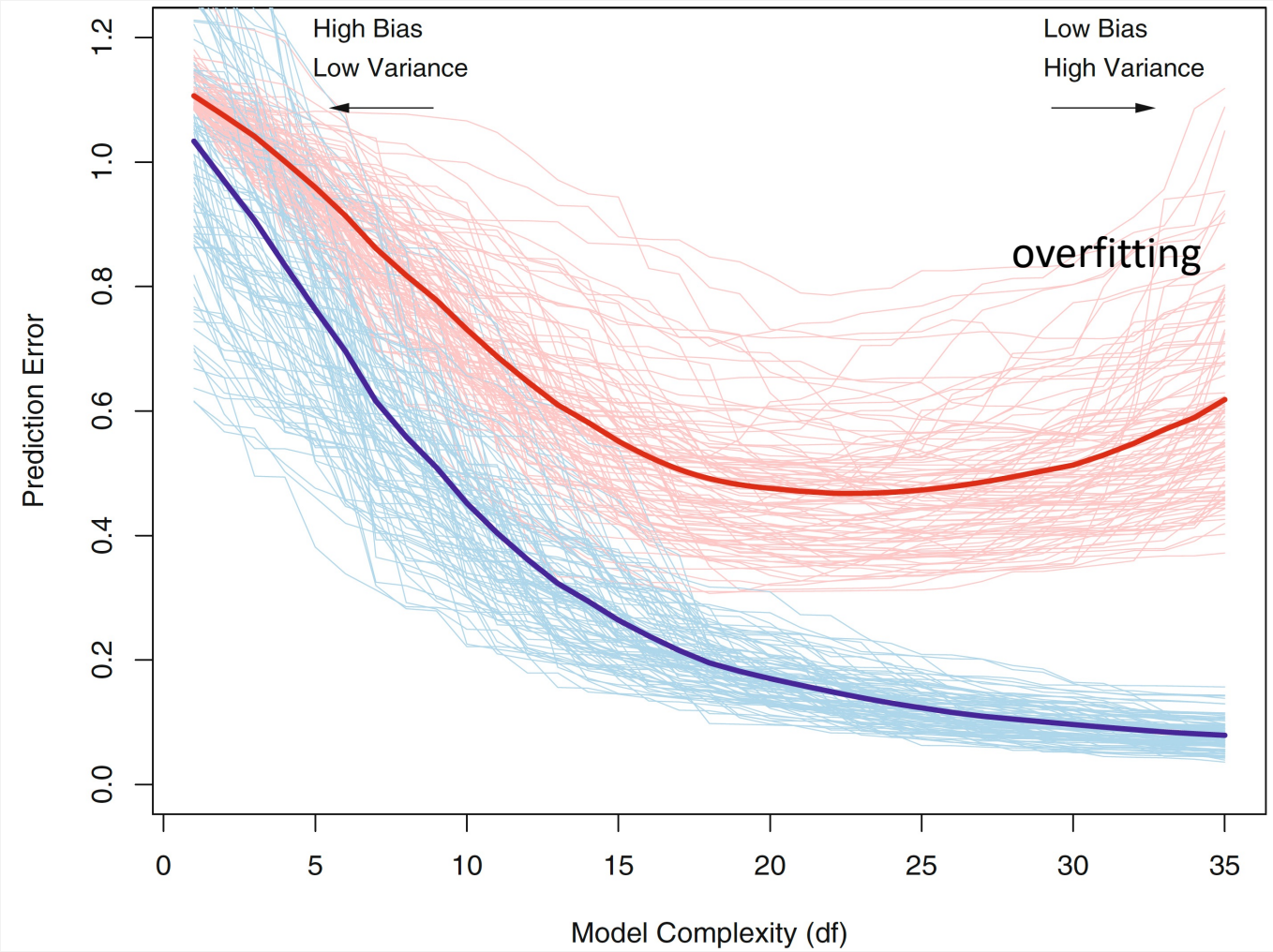
- Which line is Training Error? Testing Error?



There is no guarantee that the method with the lowest training MSE will also have the lowest test MSE

As the model complexity is increased

- The light blue curves = the training error err
- The light red curves = the conditional test error for 100 training sets of size 50 each
- The solid curves = the expected test error and the expected training error



Bias-Variance Decomposition for expected test MSE

- An expression for the expected prediction error of a regression fit $\hat{f}(X)$ at an input point $X = x_0$, using squared-error loss:

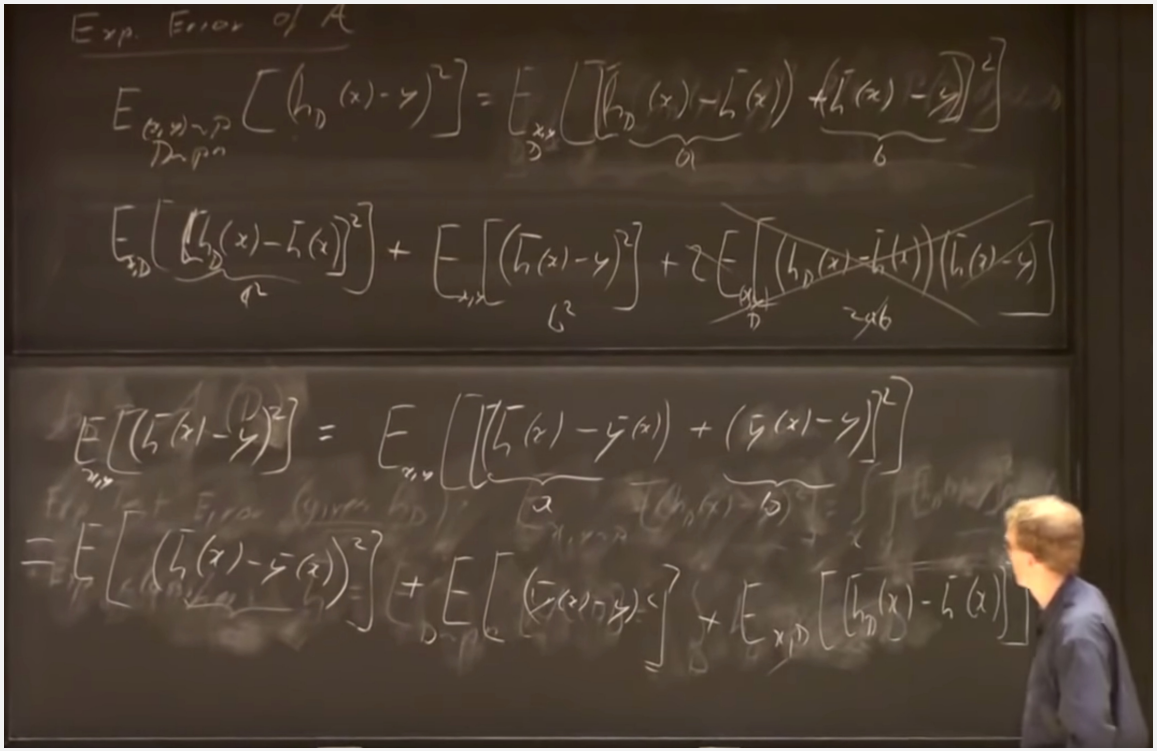
- $$E(x_0) = E\left[\left(Y - \hat{f}(x_0)\right)^2 | X = x_0\right]$$
$$= \sigma_\varepsilon^2 + \left[E\hat{f}(x_0) - f(x_0)\right]^2 + E\left[\hat{f}(x_0) - E\hat{f}(x_0)\right]^2$$
$$= \sigma_\varepsilon^2 + \text{Bias}^2\left(\hat{f}(x_0)\right) + \text{Var}\left(\hat{f}(x_0)\right)$$
$$= \text{Irreducible Error} + \text{Bias}^2 + \text{Variance}.$$

Bias-Variance Decomposition

- $E(x_0) = \sigma_\varepsilon^2 + \text{Bias}^2\left(\hat{f}(x_0)\right) + \text{Var}\left(\hat{f}(x_0)\right)$
  - the variance of the error ϵ (the noise inherent in the data)
    - the variance of the target around its true mean $f(x_0)$
    - cannot be avoided no matter how well we estimate $f(x_0)$, unless $\sigma_\varepsilon^2 = 0$
  - the squared bias of $\hat{f}(x_0)$
    - the amount by which the average of our estimate differs from the true mean
  - the variance of $\hat{f}(x_0)$, nonnegative
    - The expected squared deviation of $\hat{f}(x_0)$ around its mean

Derivation1 of Bias-Variance Decomposition

- Credit by Dr. Kilian Weinberger, Machine Learning, Cornell CS4780 SP17
  - https://youtu.be/zUJbRO0Wavo?t=2441
  - http://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote12.html



Derivation2 of Bias-Variance Decomposition

- https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff

$$
\begin{aligned}
\mathrm{E}\left[(y-\hat{f})^2\right] &= \mathrm{E}\left[(f+\varepsilon-\hat{f})^2\right] \\
&= \mathrm{E}\left[(f+\varepsilon-\hat{f}+\mathrm{E}[\hat{f}]-\mathrm{E}[\hat{f}])^2\right] \\
&= \mathrm{E}\left[(f-\mathrm{E}[\hat{f}])^2\right]+\mathrm{E}[\varepsilon^2]+\mathrm{E}\left[(\mathrm{E}[\hat{f}]-\hat{f})^2\right]+2\,\mathrm{E}\left[(f-\mathrm{E}[\hat{f}])\varepsilon\right]+2\,\mathrm{E}\left[\varepsilon(\mathrm{E}[\hat{f}]-\hat{f})\right]+2\,\mathrm{E}\left[(\mathrm{E}[\hat{f}]-\hat{f})(f-\mathrm{E}[\hat{f}])\right] \\
&= (f-\mathrm{E}[\hat{f}])^2+\mathrm{E}[\varepsilon^2]+\mathrm{E}\left[(\mathrm{E}[\hat{f}]-\hat{f})^2\right]+2(f-\mathrm{E}[\hat{f}])\,\mathrm{E}[\varepsilon]+2\,\mathrm{E}[\varepsilon]\,\mathrm{E}\left[\mathrm{E}[\hat{f}]-\hat{f}\right]+2\,\mathrm{E}\left[\mathrm{E}[\hat{f}]-\hat{f}\right](f-\mathrm{E}[\hat{f}]) \\
&= (f-\mathrm{E}[\hat{f}])^2+\mathrm{E}[\varepsilon^2]+\mathrm{E}\left[(\mathrm{E}[\hat{f}]-\hat{f})^2\right] \\
&= (f-\mathrm{E}[\hat{f}])^2+\mathrm{Var}[\varepsilon]+\mathrm{Var}\left[\hat{f}\right] \\
&= \mathrm{Bias}[\hat{f}]^2+\mathrm{Var}[\varepsilon]+\mathrm{Var}\left[\hat{f}\right] \\
&= \mathrm{Bias}[\hat{f}]^2+\sigma^2+\mathrm{Var}\left[\hat{f}\right]
\end{aligned}
$$

model validation

- the testing of a model on new data => will it show similar quality on new data in production?

Problem	Description
Bias	Systematic error in the model, such as always underpredicting.
Variance	Undesirable (but non-systematic) distance between predictions and actual values. Often due to oversensitivity of the model training procedure to small variations in the distribution of training data.

Bias-Variance Decomposition

- $E[(y[i] - f(x[i,]))^2]$ = bias^2 + variance + irreducibleError
  - Model bias : your chosen modeling technique will never get right => increase model complexity
  - Model variance : your modeling technique gets wrong due to incidental relations in the data. That is, a retraining of the model on new data might make different errors.
  - Irreducible error : truly unmodelable portion of the problem given the current variables. ie, x[i,]=x[j,], then (y[i]-y[j])^2 contributes to the irreducible error. noise

Irreducible error

- What is dash line? $Var(\epsilon)$

  => test MSE can never lie below the dash line

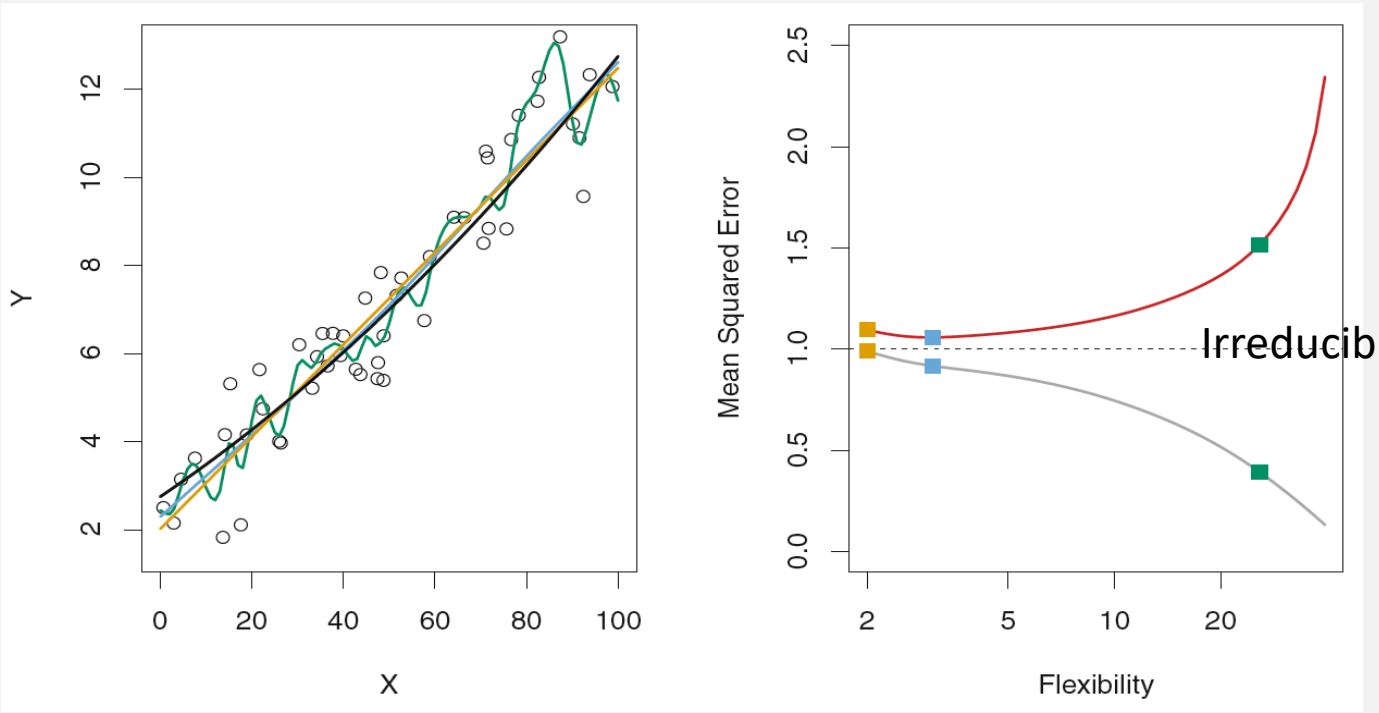


Figure2.10, *An Introduction to Statistical Learning with Applications in R* by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

Bias

- the error that is introduced by approximating a real-life problem
- the true f is substantially non-linear => no matter how many training observations we are given, it will not be possible to produce an accurate estimate using linear regression
- => linear regression results in high bias in this example.

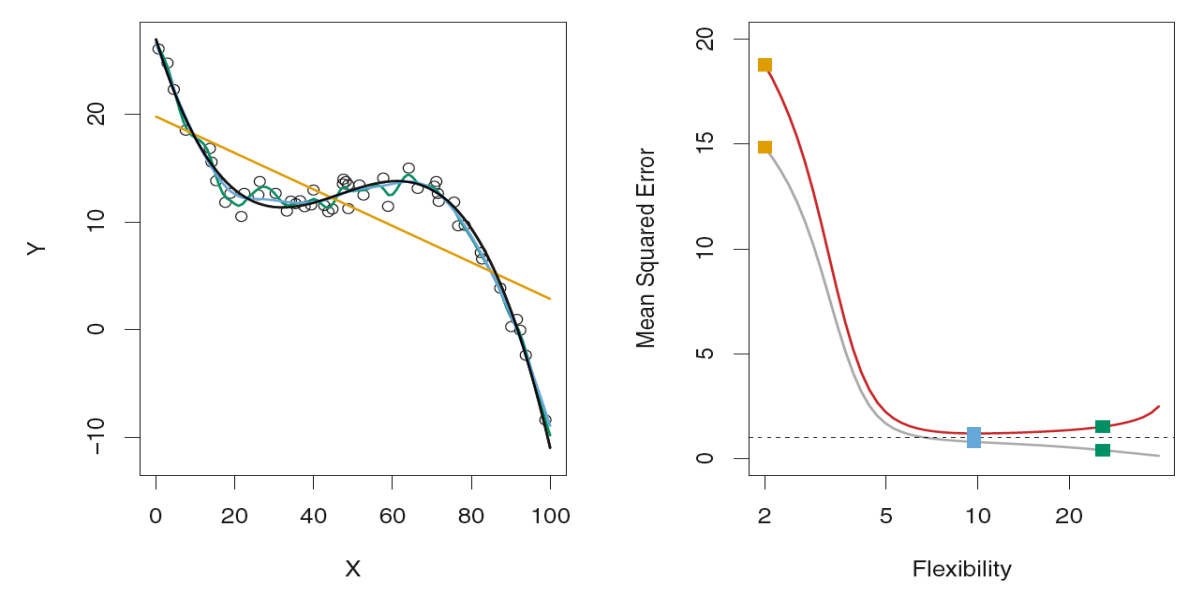


Figure 2.11, *An Introduction to Statistical Learning with Applications in R* by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

Variance

- the amount by which $\hat{f}$ would change if we estimated it using a different training data set.
- if a method has high variance
  - small changes in the training data can result in large changes in $\hat{f}$
  - more flexible statistical methods have higher variance (picks up more noise)
- Averaging is a powerful tool => Under fairly mild assumptions, averaging reduces variance
- Null model

↓

0 variance, high bias

↳ Taking multiple estimates and averaging then, the effects of variance can be mitigated

Which method is high variance? green

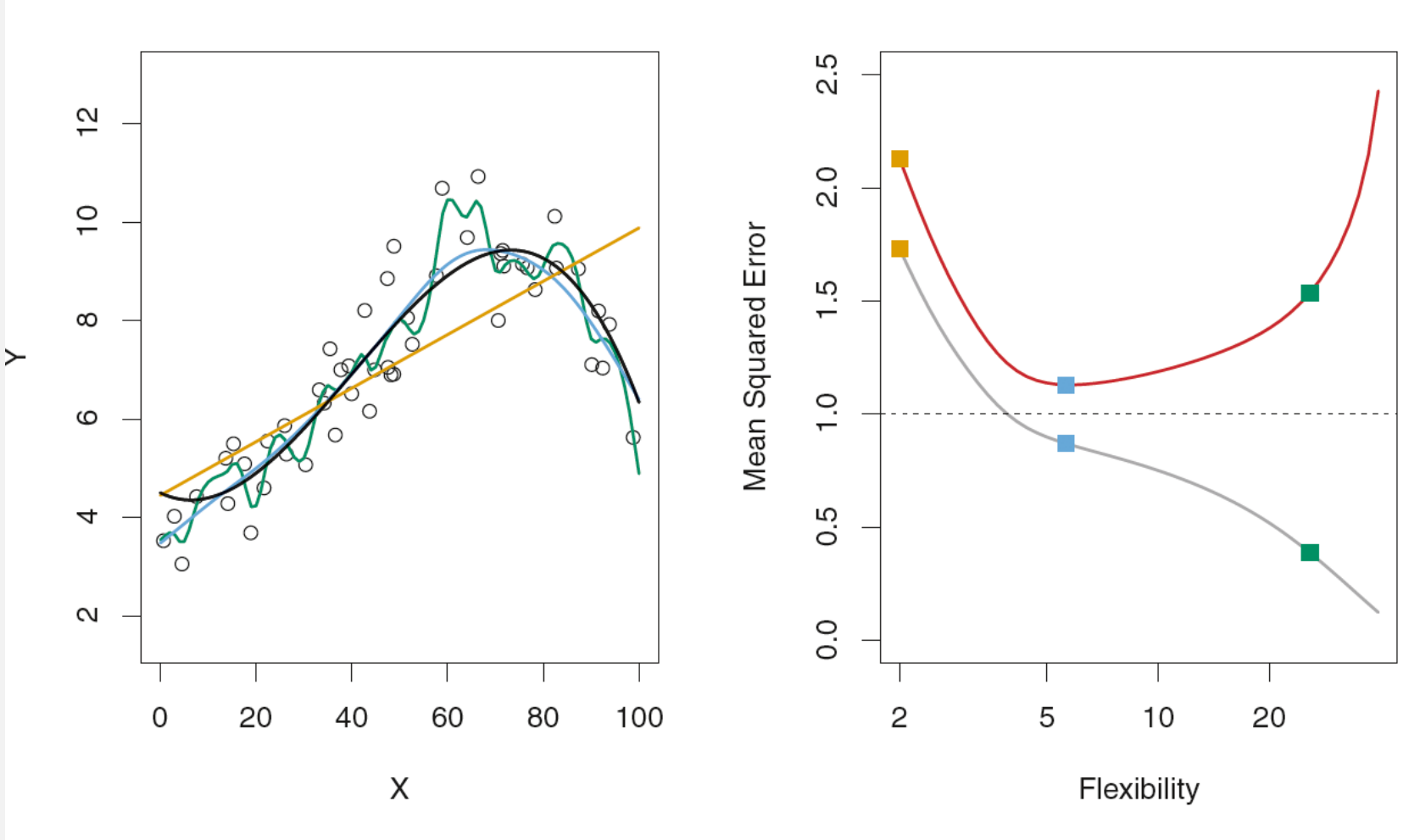


Figure 2.9, *An Introduction to Statistical Learning with Applications in R* by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

Bias-Variance trade off

- a method extremely low bias but high variance
  - drawing a curve that passes through every single training observation
- a method with very low variance but high bias
  - fitting a horizontal line to the data

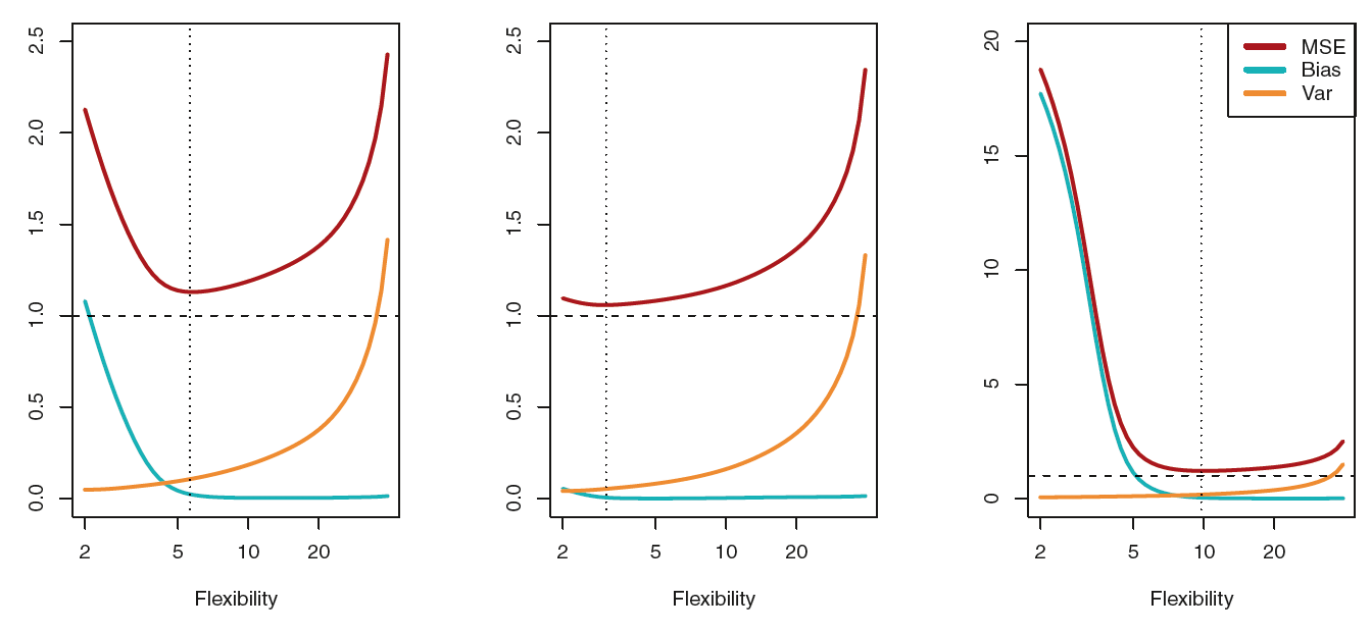


Figure 2.12, *An Introduction to Statistical Learning with Applications in R* by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

Bias v.s. Variance

- more flexible methods
  - the variance will increase
  - the bias will decrease
- increase the flexibility of a class of methods
  - the bias tends to initially decrease faster than the variance increases
  - Consequently, the expected test MSE declines
  - $E\left(y_0 - \hat{f}(x_0)\right)^2 = \mathrm{Var}\left(\hat{f}(x_0)\right) + \left[\mathrm{Bias}\left(\hat{f}(x_0)\right)\right]^2 + \mathrm{Var}(\epsilon)$

Bias-Variance trade off

- increase the flexibility of a class of methods
  - the bias tends to initially decrease faster than the variance increases
- at some point increasing flexibility has little impact on the bias but starts to significantly increase the variance.

⇒ the model starts picking up noise

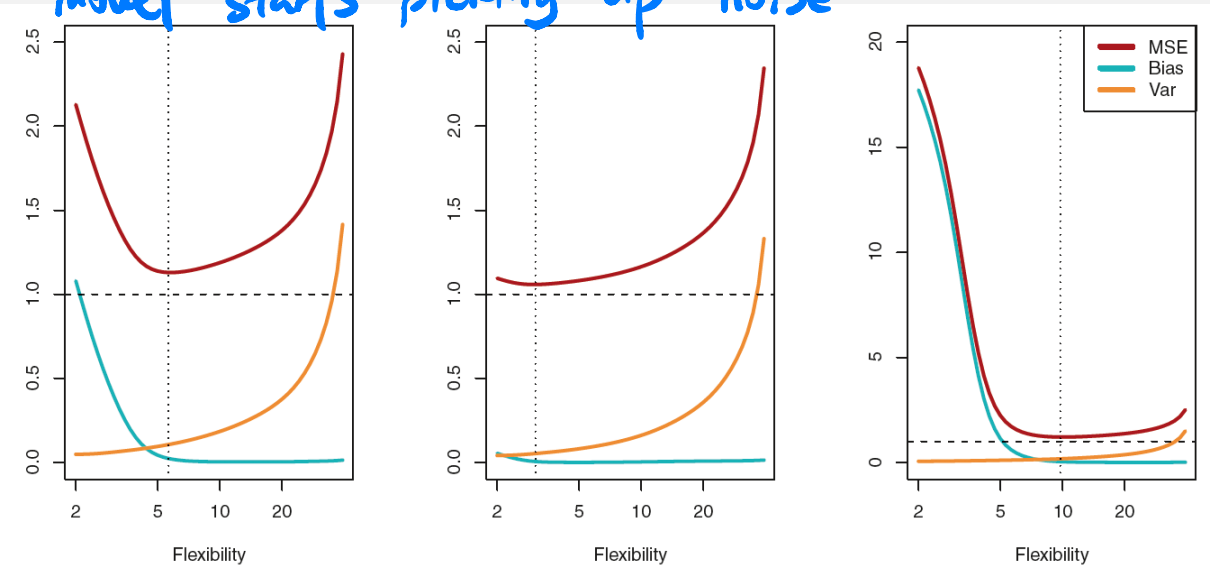


Figure 2.12, *An Introduction to Statistical Learning with Applications in R* by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

U-shape (testing set : red line)

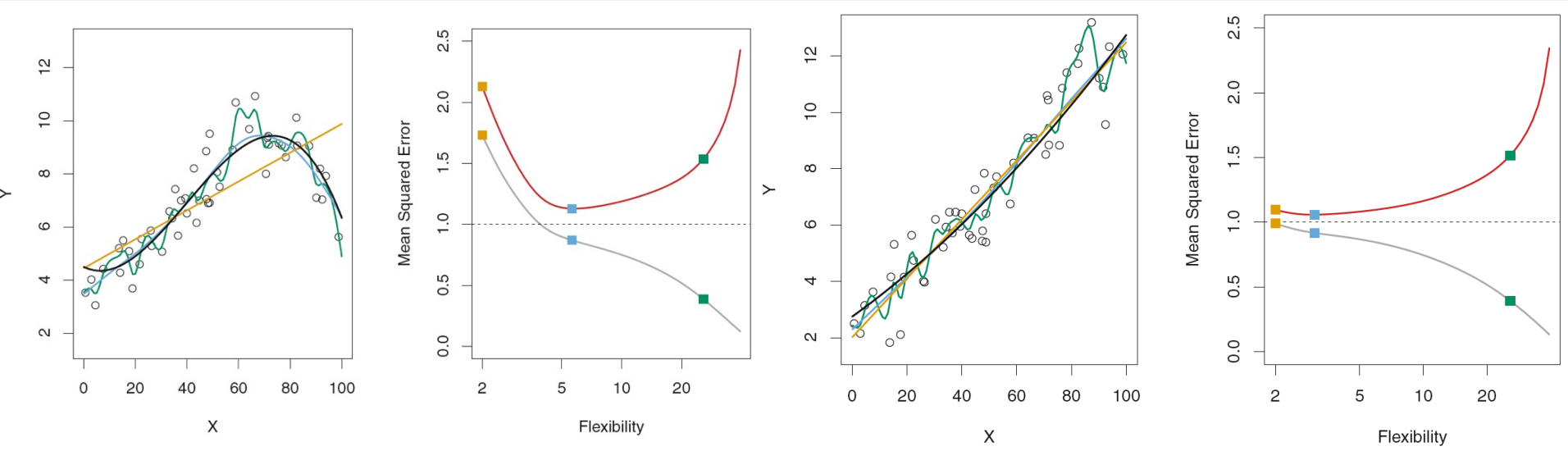


Figure 2.9、2.10, *An Introduction to Statistical Learning with Applications in R* by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

Bias-Variance trade off

- The vertical dotted line indicates the flexibility level corresponding to the smallest test MSE.

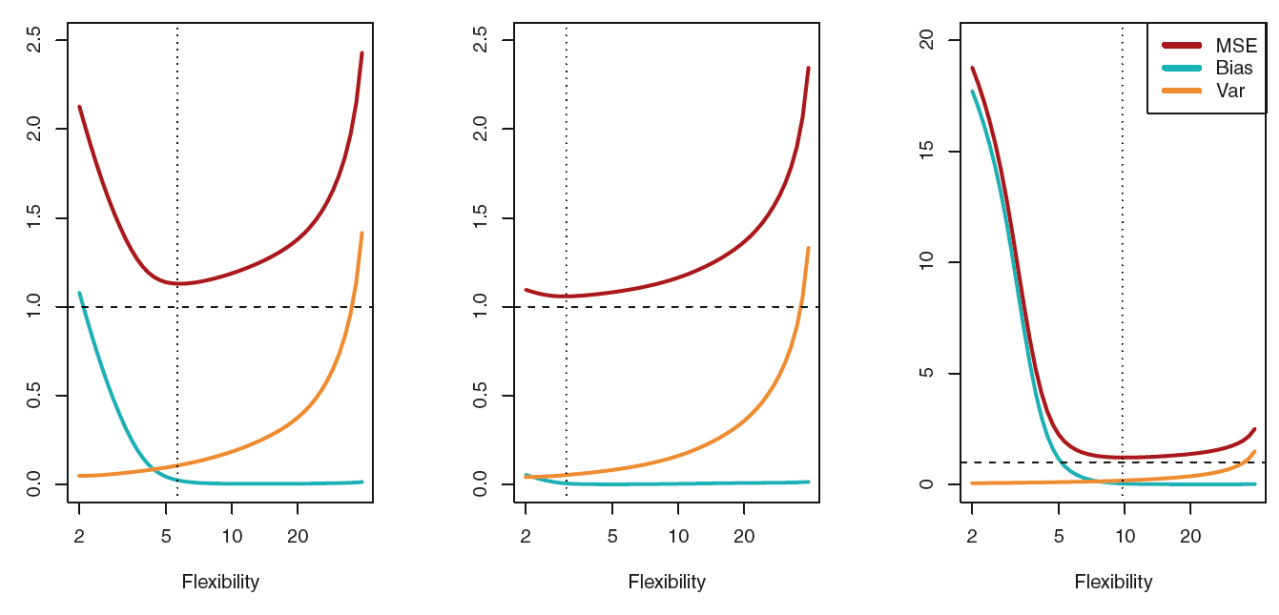


Figure 2.12, *An Introduction to Statistical Learning with Applications in R* by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

Overfitting

Overfitting

- training error : a model's prediction error on the data that it trained
- generalization error : a model's prediction error on new data
- overfit : if generalization error is large => prefer simpler models

Overfit	Features of the model that arise from relations that are in the training data, but not representative of the general population. Overfit can usually be reduced by acquiring more training data and by techniques like regularization and bagging.

Overfitting

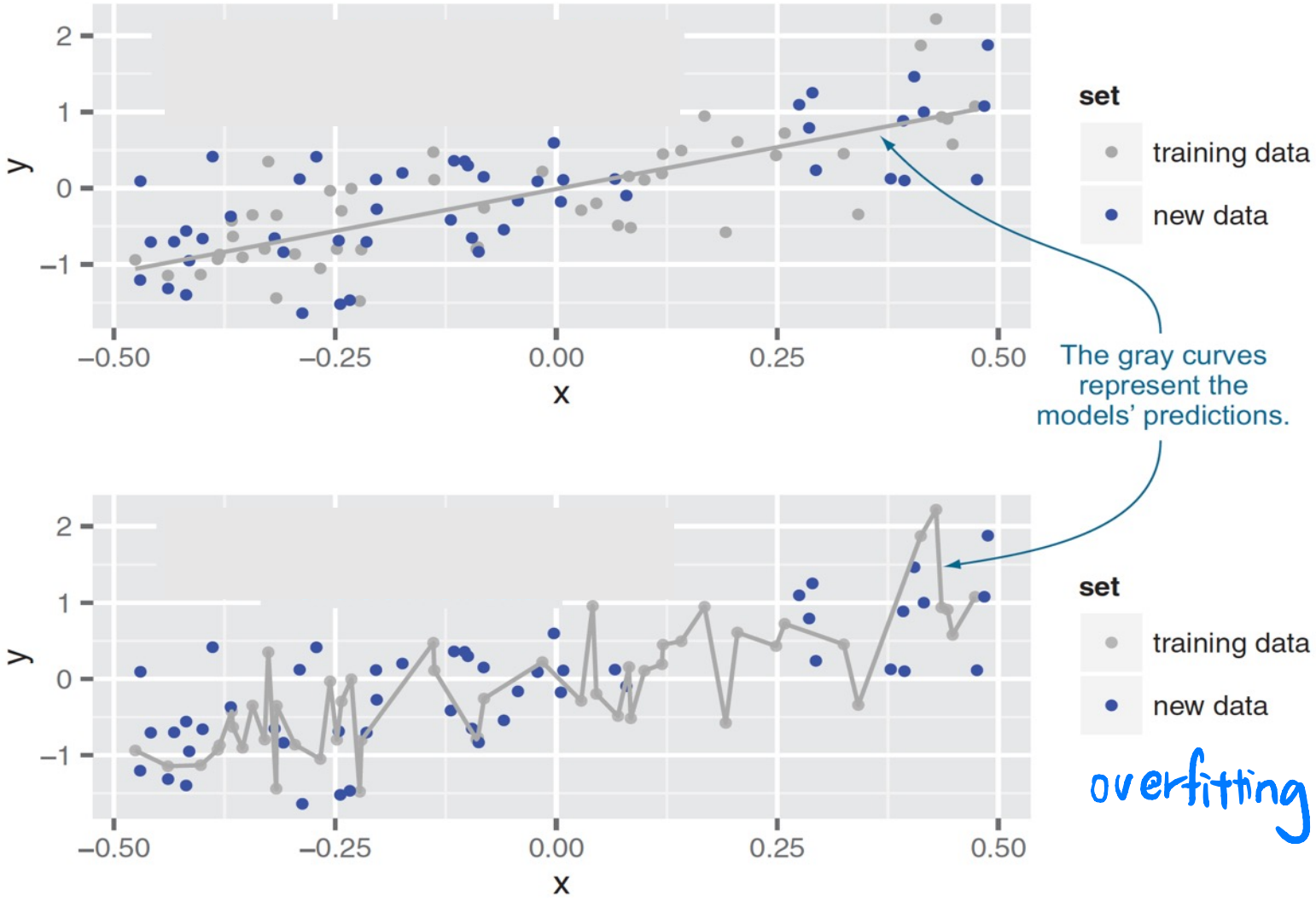


Figure5.11, *Practical Data Science with R* by Nina Zumel, John Mount

How to check if a model fit is good?

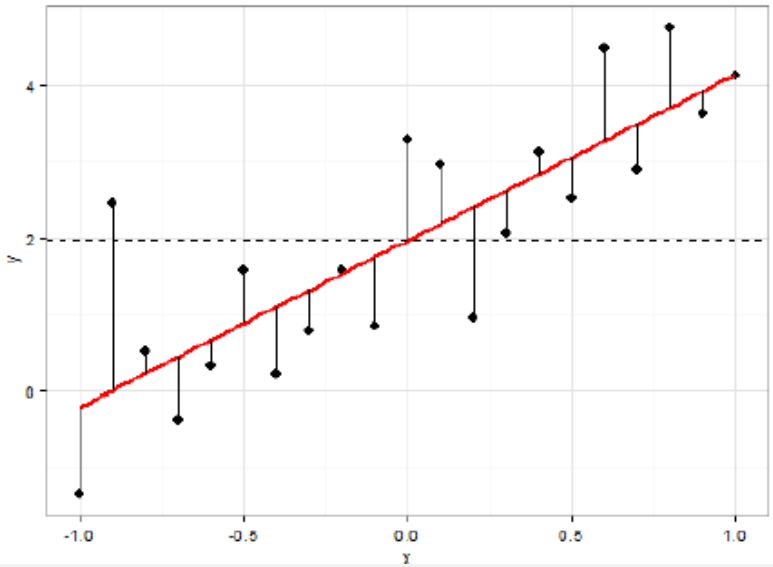
$$\sum(y_i - f_i)^2 = 18.568$$

$$\sum(y_i - \bar{y})^2 = 55.001$$

$$R^2 = 1 - \frac{18.568}{55.001}$$

$$R^2 = 0.6624$$

A decent model fit!



How to check if a model fit is good?

$$\sum(y_i - f_i)^2 = 15.276$$

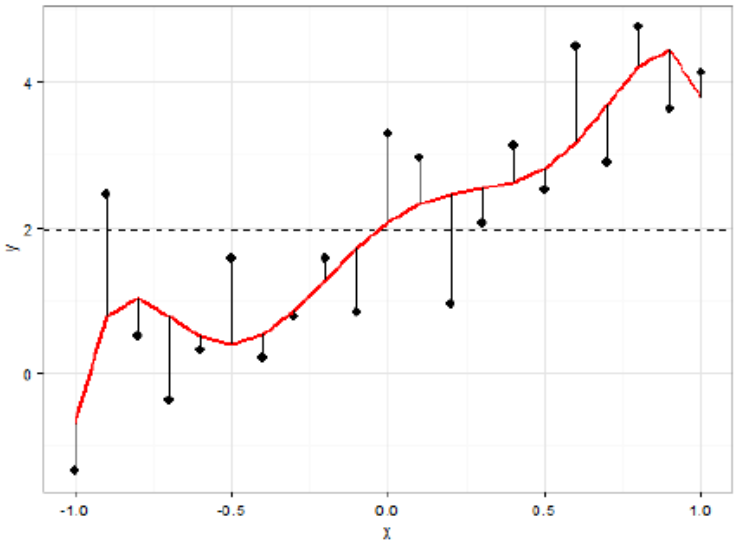
$$\sum(y_i - \bar{y})^2 = 55.001$$

$$R^2 = 1 - \frac{15.276}{55.001}$$

$$R^2 = 0.72$$

Is this a better model?

No, overfitting!



Overfitting

- Modeling techniques tend to overfit the data.
- Multiple regression
  - Every time you add a variable to the regression, the model's R^2 goes up
  - Naïve interpretation: *every* additional predictive variable helps to explain yet more of the target's variance
  - Left to its own devices, Multiple Regression will fit *too many* patterns

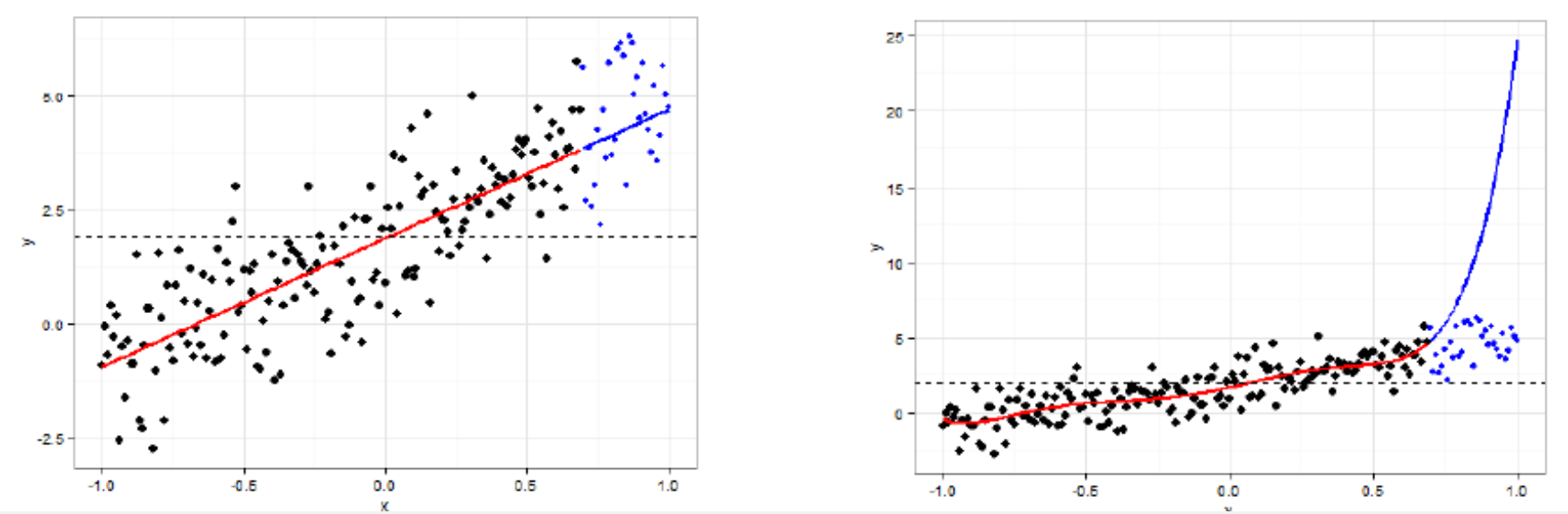
Overfitting

- Error on the dataset used to *fit* the model can be misleading
  - Doesn't predict future performance.
- Too much complexity can diminish model's accuracy on future data.
  - Sometimes called the Bias-Variance Tradeoff.

Overfitting

- What are the consequences of overfitting?

"*Overfitted models will have high R^2 values, but will perform poorly in predicting out-of-sample cases*"



Three-way cross-validation

Ensuring model quality

Hold-out data (testing data)

- Testing On ~~Held-out Data~~
- *K*-fold cross-validation
- the related ideas of empirical resampling and bootstrapping

Empirical resampling and bootstrapping are statistical procedures used to estimate the sampling distribution of an estimator by resampling with replacement from the original data, and to assess the stability of said estimator.

Empirical Resampling:

- This is a broader term that encompasses any method of repeatedly drawing samples from the original data. It allows for the estimation of the variability of a statistic without relying on strong assumptions about the form of the population distribution from which the data are sampled.

Bootstrapping:

- Bootstrapping is a specific type of empirical resampling.
- In bootstrapping, numerous subsamples of the observed dataset are automatically and repeatedly drawn, with replacement. Each subsample is the same size as the original dataset.
- For each bootstrap sample, a statistic of interest (like the mean, median, or regression coefficients) is calculated.
- After many bootstrap samples (commonly thousands), you'll have a bootstrap distribution of the statistic, which can be used to compute standard errors, confidence intervals, or test hypotheses.
- Bootstrapping is especially useful with small datasets or when the theoretical distribution of a statistic is complex or unknown.

The underlying principle of bootstrapping is that the empirical distribution of the observed data is a good approximation of the population distribution that we're trying to infer about. By resampling from our dataset, we're essentially simulating the process of taking more samples from the population.

Bootstrapping has two key advantages:

1. **Non-parametric**: It does not assume a parametric distribution of the data, making it flexible and applicable in various situations.
2. **Simple yet powerful**: Despite its simplicity, it can yield robust estimates of the distribution of the estimator, allowing for accurate inference even in complex situations.

In the context of machine learning, bootstrapping can also be used in algorithms like Bagging (Bootstrap Aggregating) to reduce variance and avoid overfitting. Each model in an ensemble is trained on a different bootstrap sample, and their predictions are combined (e.g., averaged) to produce a final prediction. This technique leverages the power of bootstrapping to create a more stable and accurate predictive model.

Ensuring model quality

- Split
  - a subset of the training data is used to build a model
  - a complementary subset of the training data is used to score the model
  - repeated cross-validation, replicated cross-validation

Cross-validation

- In cross-validation the original sample is split into two parts.
  - One part is called the training (or *derivation)* sample
  - the other part is called the *validation* (*testing*) sample.

How should the sample be split?

The most common approach is to divide the sample randomly, thus theoretically eliminating any systematic differences.

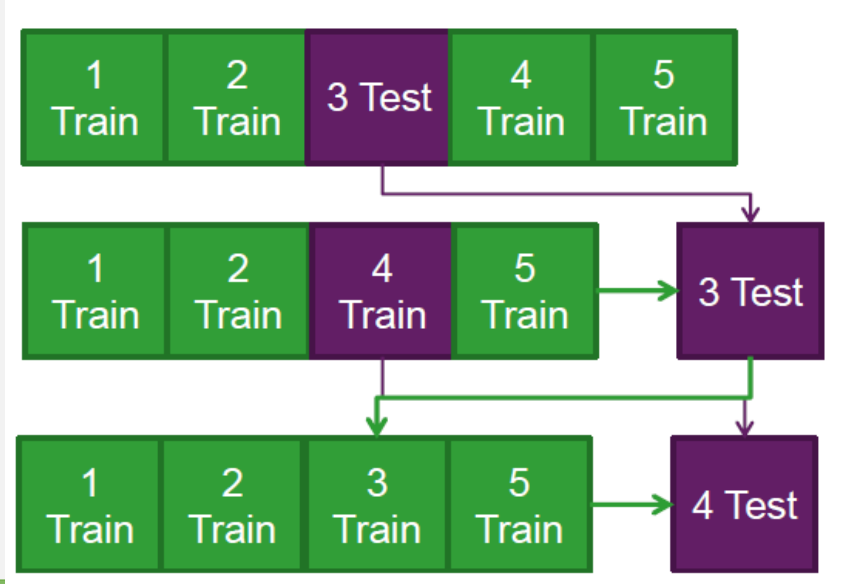
- Modeling of the data uses one part only. The model selected for this part is then used to predict the values in the other part of the data. A valid model should show good predictive accuracy.
- R^2 offers no protection against is overfitting. On the other hand, cross validation, by allowing us to have cases in our testing set that are different from the cases in our training set, inherently offers protection against overfitting.

K-fold Cross Validation

- Since data are often scarce, there might not be enough to set aside for a validation sample
- To work around this issue *k*-fold CV works as follows:
  1. Split the sample into *k* subsets of equal size
  2. For each fold estimate a model on all the subsets except one
  3. Use the left out subset to test the model, by calculating a CV metric of choice
  4. Average the CV metric across subsets to get the CV error
- This has the advantage of using all data for estimating the model, however finding a good value for *k* can be tricky.

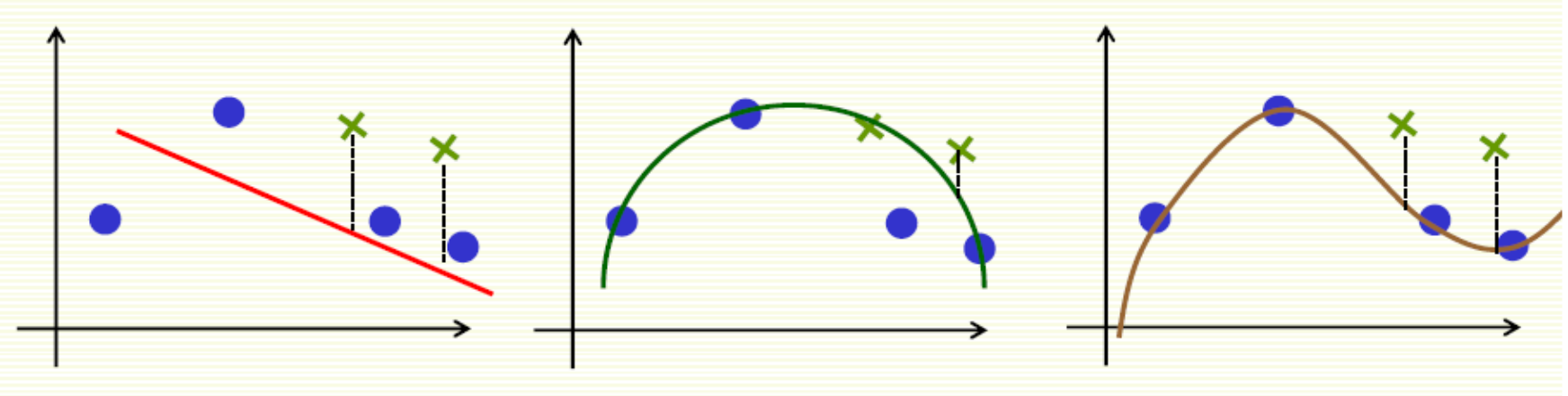
5-fold Cross Validation Example

1. Split the data into 5 samples
2. Fit a model to the training samples and use the test sample to calculate a CV metric.
3. Repeat the process for the next sample, until all samples have been used to either train or test the model



Training/test Data Split

- What about test error? Seems appropriate
  - degree 2 is the best model according to the test error
- Except what do we report as the test error now?
  - Test error should be computed on data that was **not used for training at all**
  - Here used "test" data for training, i.e. choosing model



Splitting data into training, calibration, and test sets

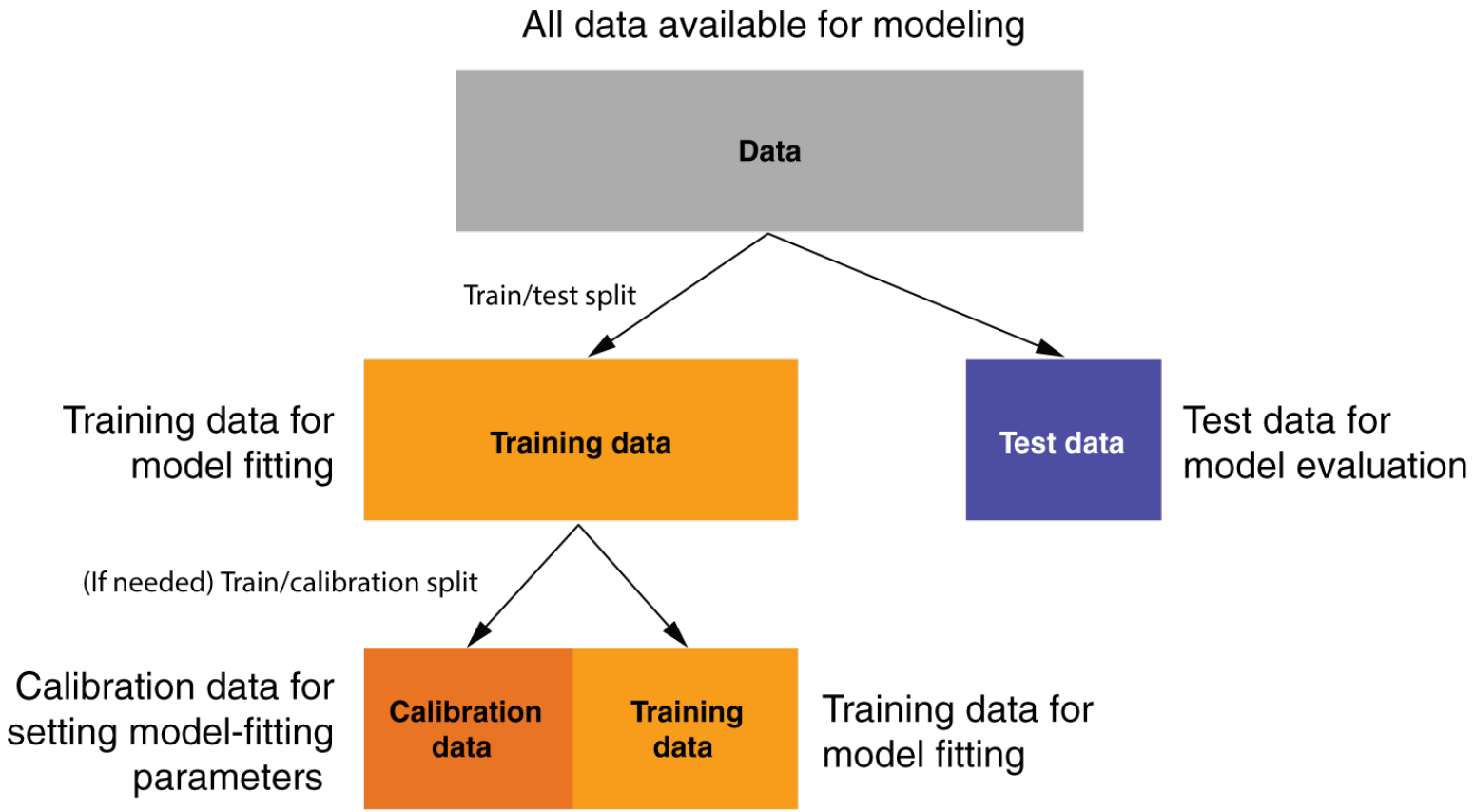


Figure 4.12, Practical Data Science with R (sec.) by Nina Zumel, John Mount

Setting Hyperparameters

Idea #1: Choose hyperparameters that work best on the data

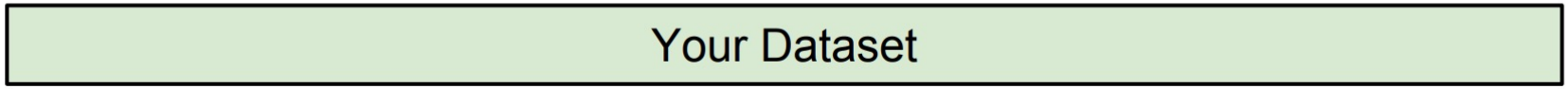
BAD: K = 1 always works perfectly on training data

Your Dataset

Setting Hyperparameters

Idea #1: Choose hyperparameters that work best on the data

BAD: K = 1 always works perfectly on training data

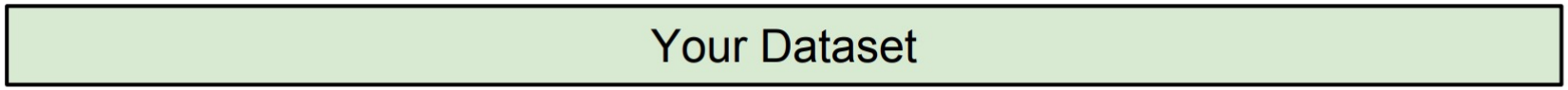


Idea #2: Split data into **train** and **test**, choose hyperparameters that work best on test data

Setting Hyperparameters

Idea #1: Choose hyperparameters that work best on the data

BAD: K = 1 always works perfectly on training data



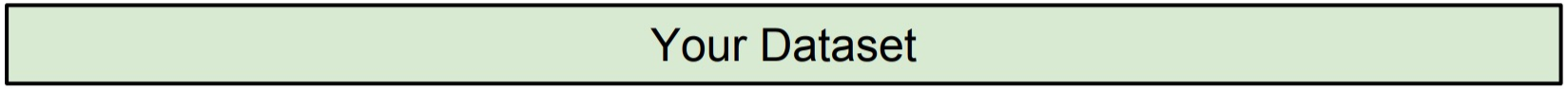
Idea #2: Split data into **train** and **test**, choose hyperparameters that work best on test data

BAD: No idea how algorithm will perform on new data

Setting Hyperparameters

Idea #1: Choose hyperparameters that work best on the data

BAD: K = 1 always works perfectly on training data



Idea #2: Split data into **train** and **test**, choose hyperparameters that work best on test data

BAD: No idea how algorithm will perform on new data

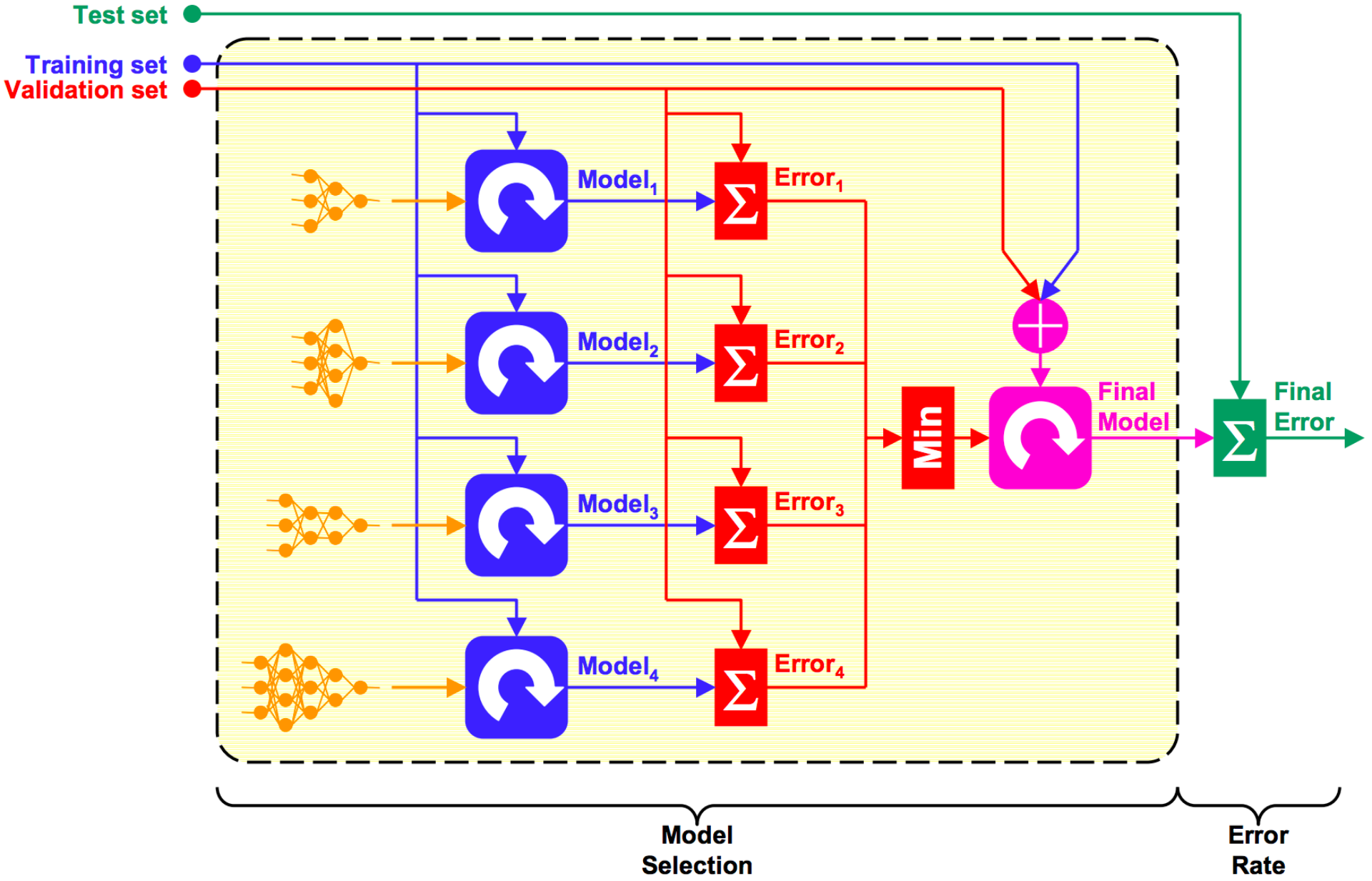
Idea #3: Split data into **train**, **val**, and **test**; choose hyperparameters on val and evaluate on test

Better!

Why three-way data split?

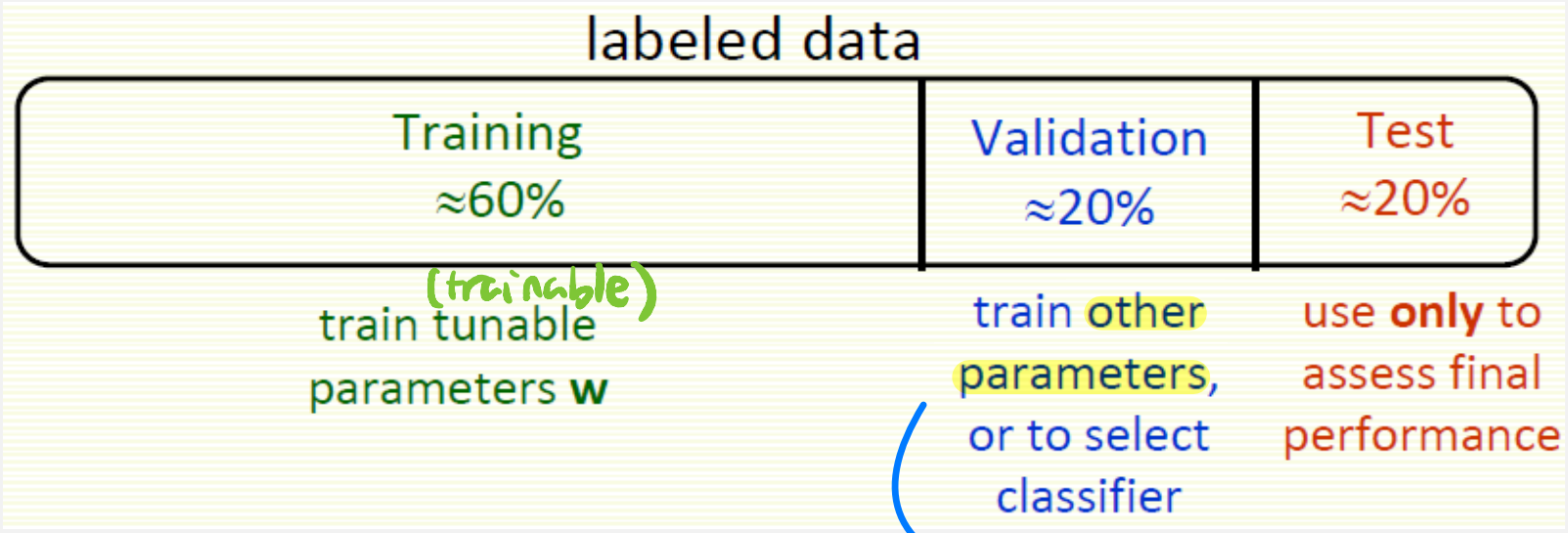
- data leakage
  Importantly, any data preparation prior to fitting the model or tuning of the hyperparameter of the model must occur within the for-loop on the data sample. This is to avoid data leakage where knowledge of the test dataset is used to improve the model. This, in turn, can result in an optimistic estimate of the model skill.
- Ideally, the test set should be kept in a “vault,” and be brought out only at the end of the data analysis. Suppose instead that we use the test-set repeatedly, choosing the model with smallest test-set error. Then the test set error of the final chosen model will underestimate the true test error, sometimes substantially.

Three-way data splits



Validation Data

- Same question when choosing among several classifiers
  - our polynomial degree example can be looked at as choosing among 3 classifiers (degree 1, 2, or 3)
  - Solution: split the labeled data into three parts



Three-way data split

- In cross-validation the original sample is split into two parts.
  - One part is called the training (or *derivation)* sample
  - the other part is called the *validation (or validation + testing)* sample.
- The training set : to fit the models
- The validation set : to estimate prediction error for model selection
- The test set : for assessment of the generalization error of the final chosen model

Setting Hyperparameters

Your Dataset

Idea #4: **Cross-Validation**: Split data into **folds**,
try each fold as validation and average the results

fold 1	fold 2	fold 3	fold 4	fold 5	test
fold 1	fold 2	fold 3	fold 4	fold 5	test
fold 1	fold 2	fold 3	fold 4	fold 5	test

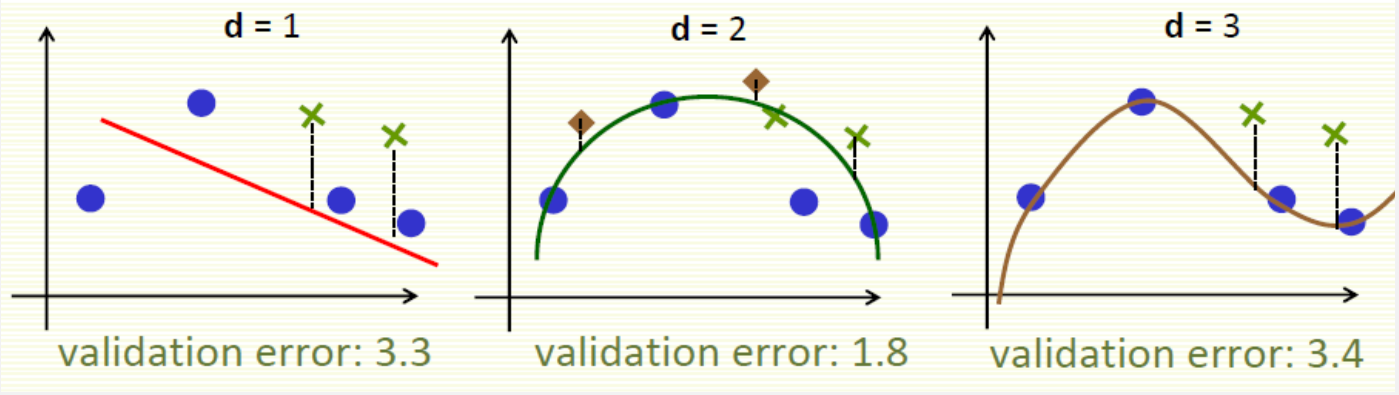
Useful for small datasets, but not used too frequently in deep learning

Training/Validation/Test Data

- Training Data
- Validation Data

  d = 2 is chosen
- Test Data

  1.3 test error computed for **d** = 2

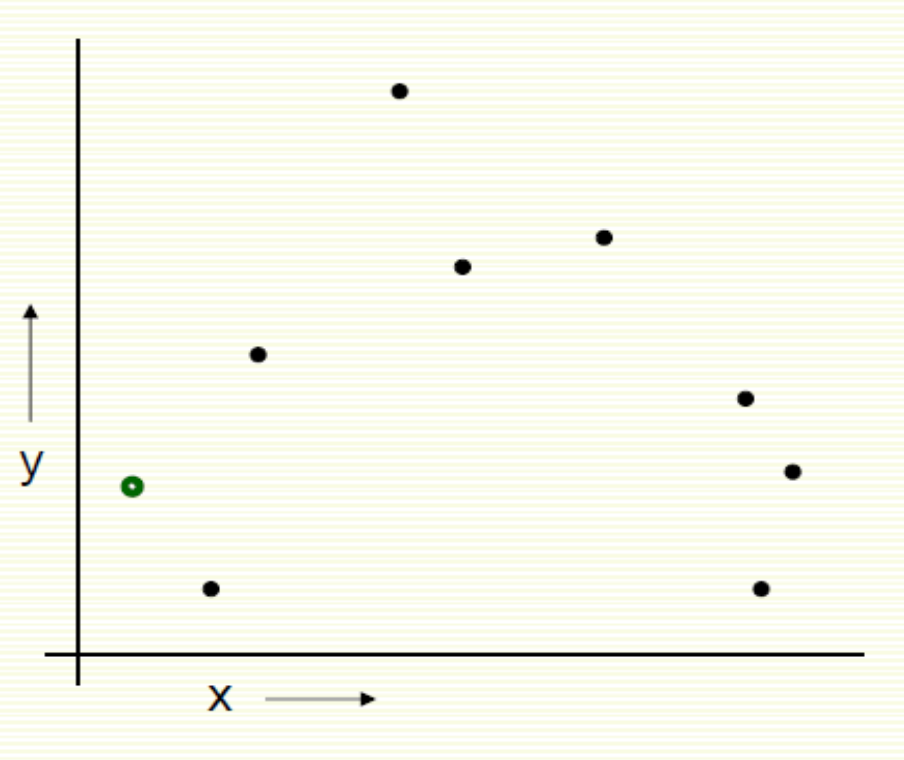


What portion of the sample should be in each part?

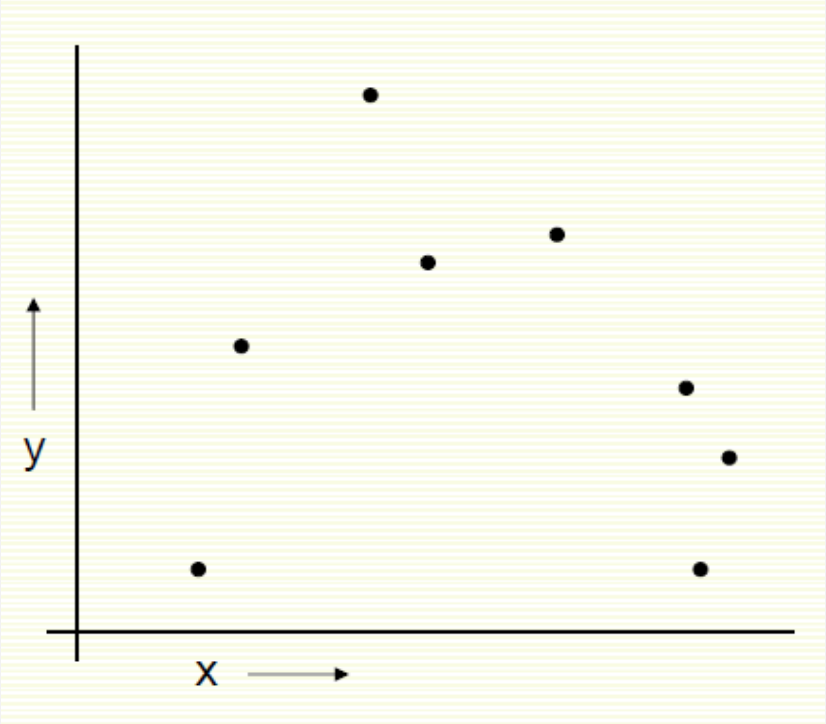
- If sample size is very large, it is often best to split the sample in half.
- For smaller samples, it is more conventional to split the sample such that 2/3 of the observations are in the derivation sample and 1/3 are in the validation sample.

LOOCV (Leave-One-Out Cross Validation)

- For k=1 to n
  - Let ($\mathbf{x}^k$,$\mathbf{y}^k$) be the ***k*** example



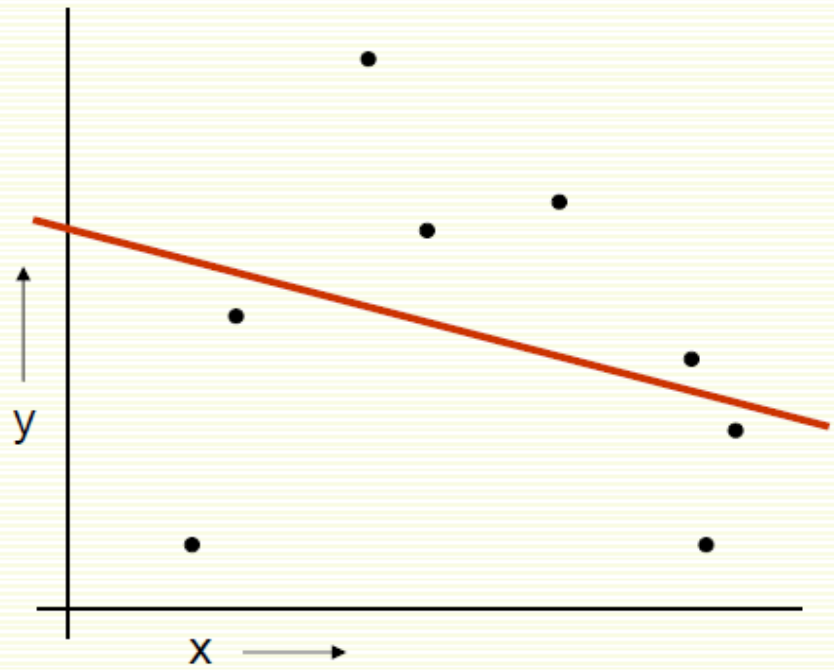
LOOCV



For **k**=1 to **n**

1. Let ($\mathbf{x}^k$,$\mathbf{y}^k$) be the **k**th example
2. Temporarily remove ($\mathbf{x}^k$,$\mathbf{y}^k$) from the dataset

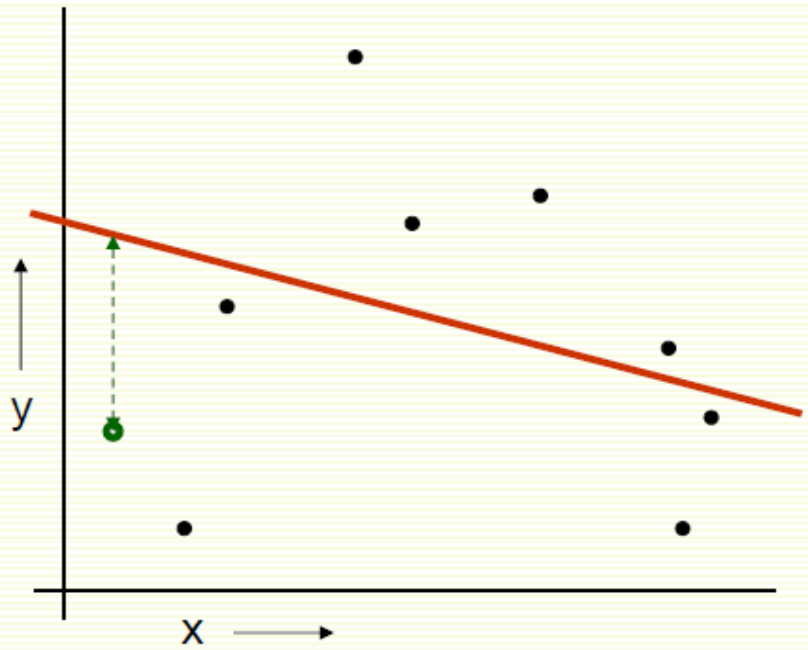
LOOCV



For **k**=1 to **n**

1. Let ($\mathbf{x}^k$,$\mathbf{y}^k$) be the **k**th example
2. Temporarily remove ($\mathbf{x}^k$,$\mathbf{y}^k$) from the dataset
3. Train on the remaining **n**-1 examples

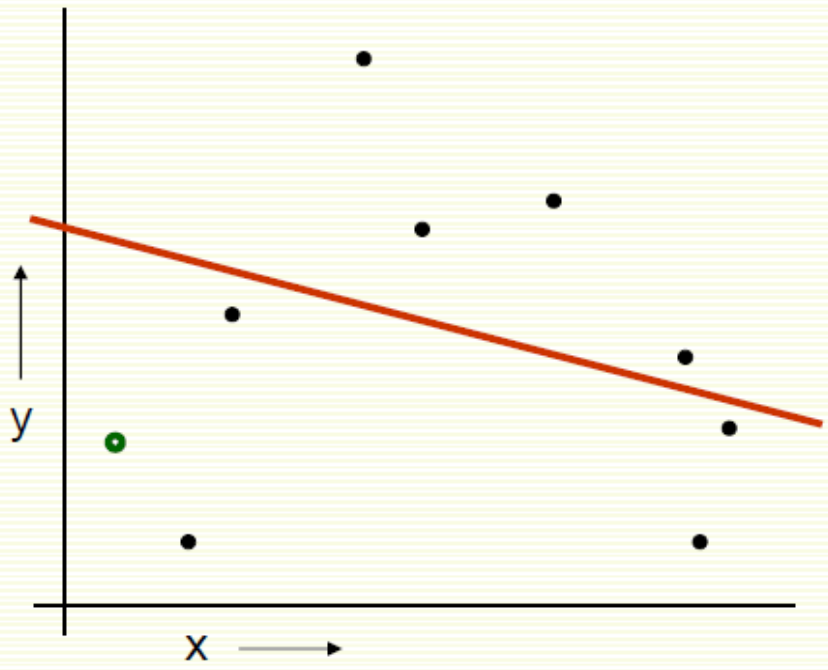
LOOCV



For **k**=1 to **n**

1. Let ($\mathbf{x}^k$,$\mathbf{y}^k$) be the **k**th example
2. Temporarily remove ($\mathbf{x}^k$,$\mathbf{y}^k$) from the dataset
3. Train on the remaining **n**-1 examples
4. Note your error on ($\mathbf{x}^k$,$\mathbf{y}^k$)

LOOCV

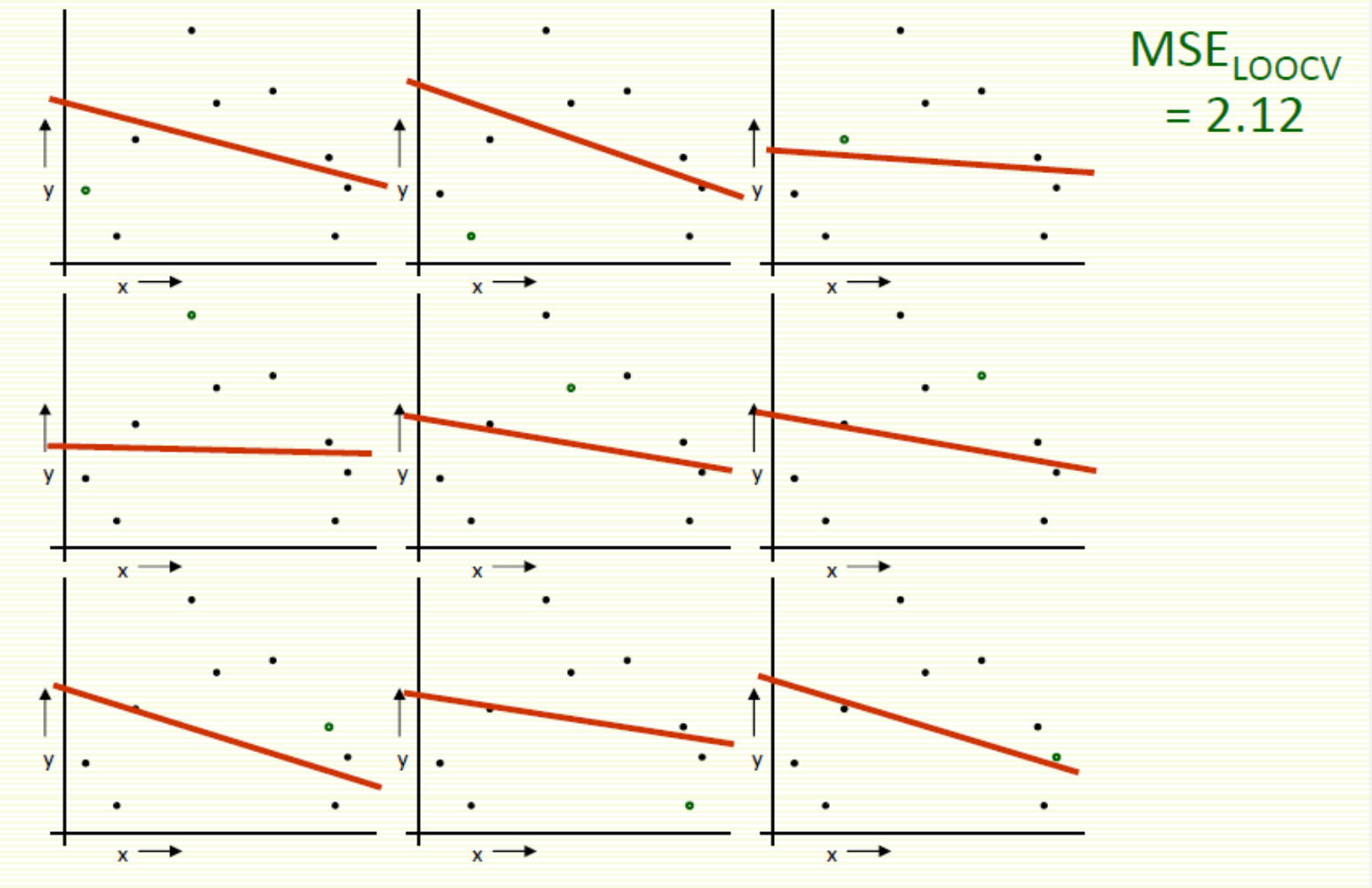


For **k**=1 to **n**

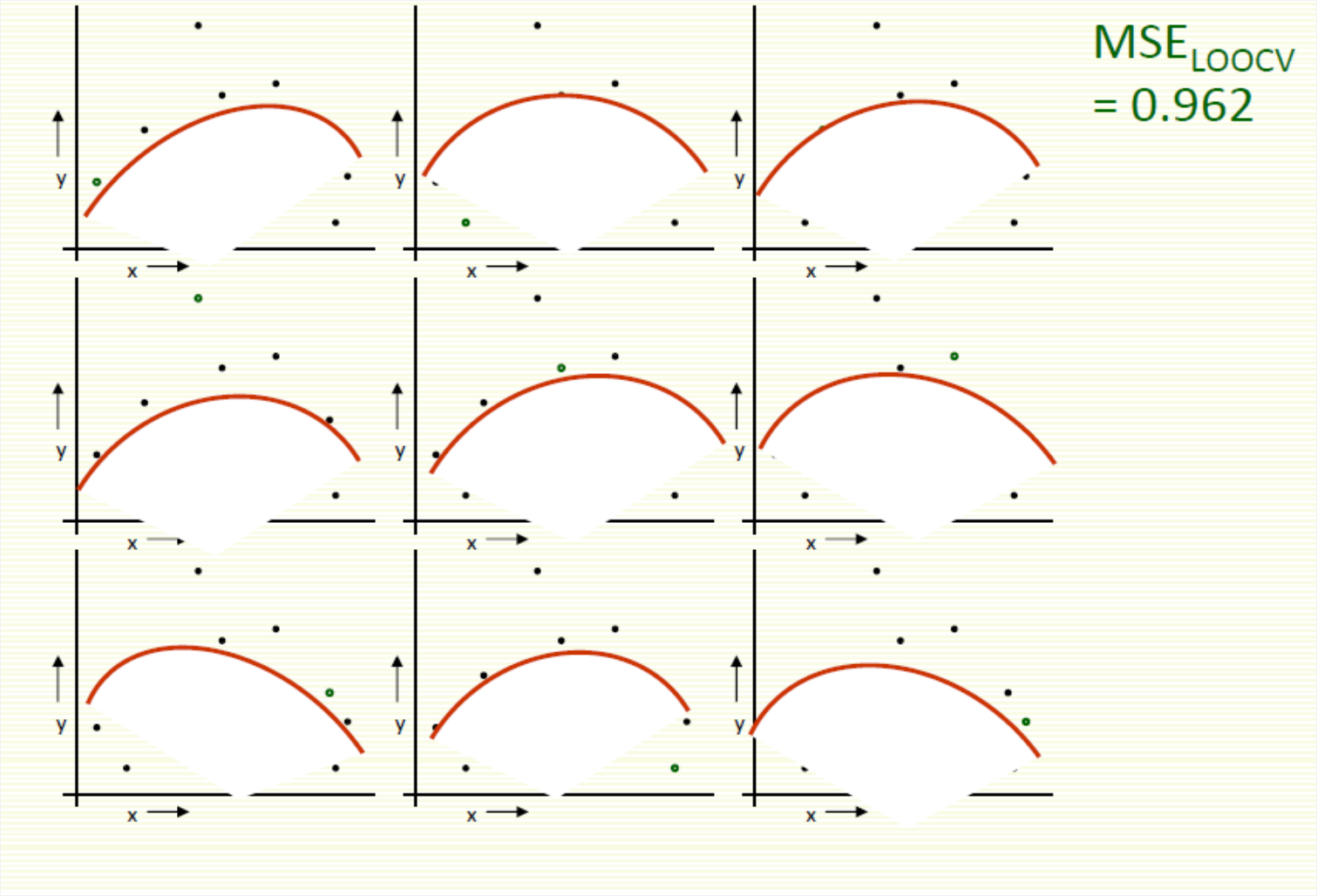
1. Let ($\mathbf{x}^k$,$\mathbf{y}^k$) be the **k**th example
2. Temporarily remove ($\mathbf{x}^k$,$\mathbf{y}^k$) from the dataset
3. Train on the remaining **n**-1 examples
4. Note your error on ($\mathbf{x}^k$,$\mathbf{y}^k$)

When you've done all points, report the mean error

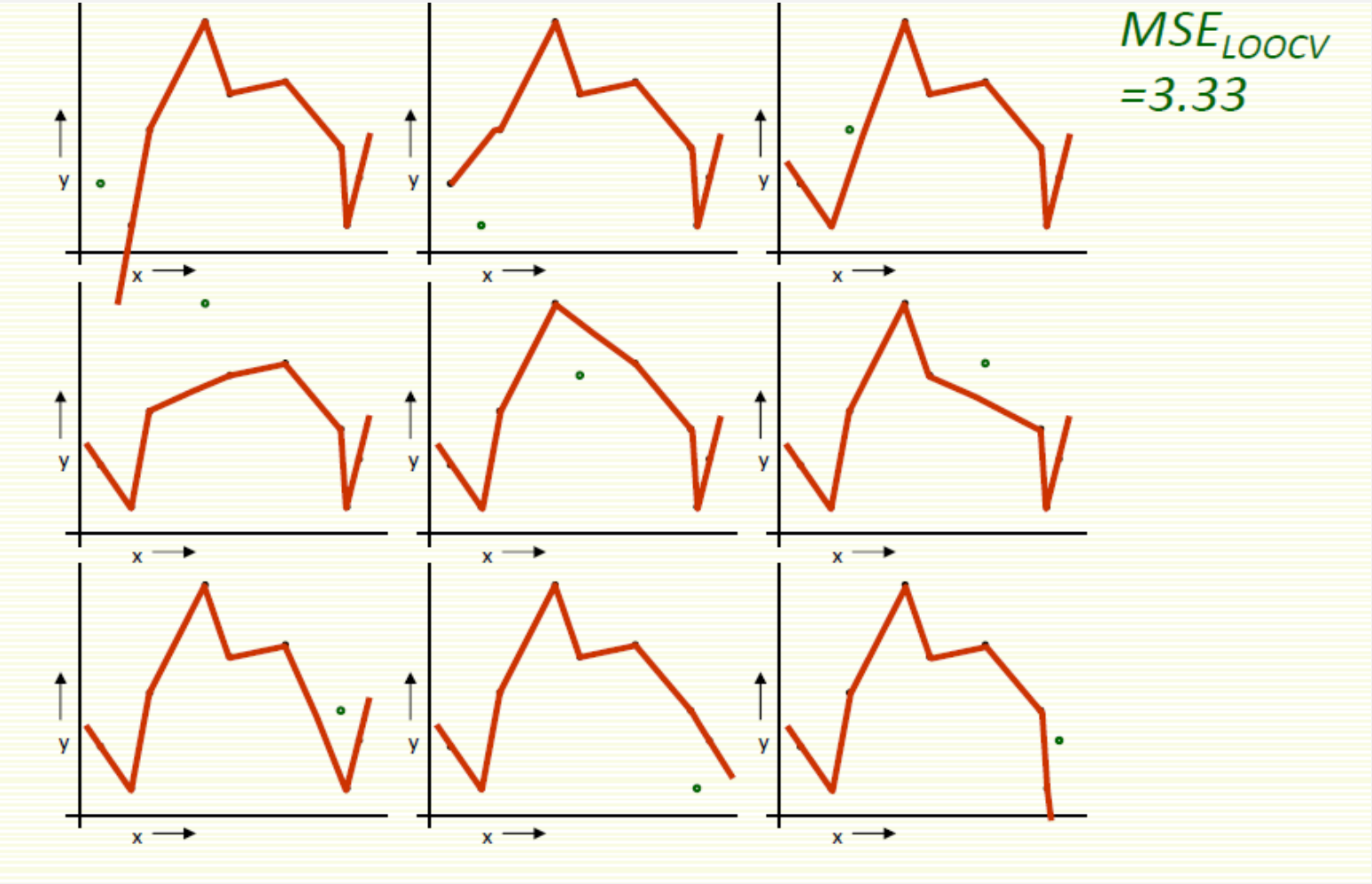
LOOCV



LOOCV for Quadratic Regression



LOOCV for Join The Dots



Which kind of Cross Validation?

	Downside	Upside
Test-set	may give unreliable estimate of future performance	cheap
Leave-one-out	expensive	doesn't waste data
10-fold	wastes 10% of the data,10 times more expensive than test set	only wastes 10%, only 10 times more expensive instead of **n** times
3-fold	wastes more data than 10-fold, more expensive than test set	slightly better than test-set
N-fold	Identical to Leave-one-out	

Improve cross-validation

- Even better: *repeated* cross-validation
- Example:
  - 10-fold cross-validation is repeated 10 times and results are averaged (reduce the variance)

Best Practice for Reporting Model Fit

- Use Cross Validation to find the best model
- Report the RMSE and MAPE statistics from the cross validation procedure
- Report the R^2 from the model as you normally would.
- The added cross-validation information will allow one to evaluate not how much variance can be explained by the model, but also the predictive accuracy of the model. Good models should have a high predictive AND explanatory power!

→ Mean Absolute Percentage Error

common metrics for evaluating the performance of regression models

Sampling for modeling and validation

- train/calibration, test
- Splitting into test and training using a random group mark

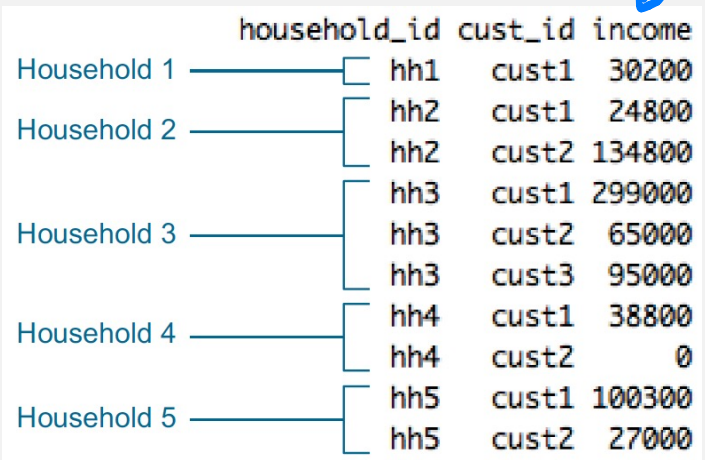
```
custdata$gp <- runif(dim(custdata)[1])
testSet <- subset(custdata, custdata$gp <= 0.1)
trainingSet <- subset(custdata, custdata$gp > 0.1)
testSet <- subset(custdata, custdata$gp <= 0.2)
trainingSet <- subset(custdata, custdata$gp > 0.2)
```

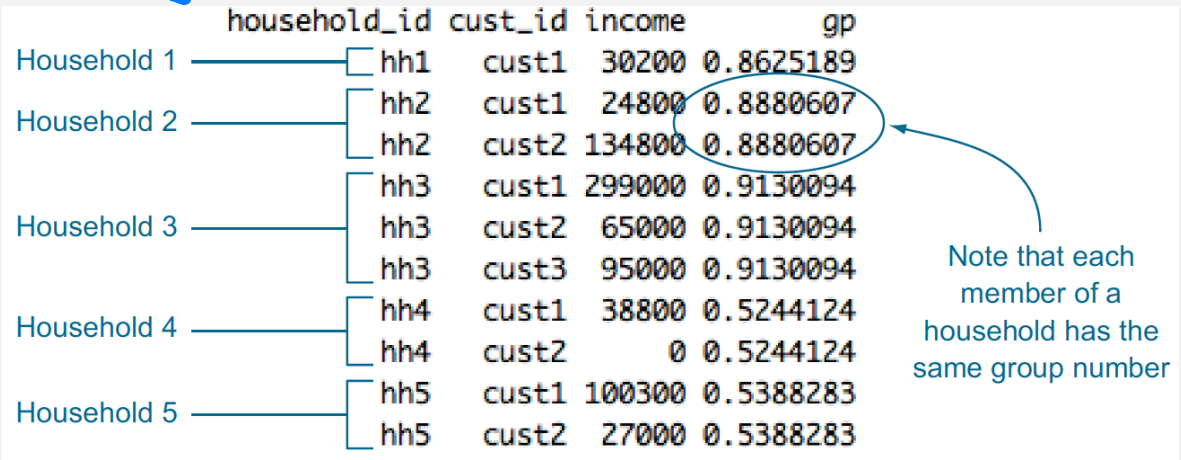
Record grouping

```
hh <- unique(hhdata$household_id)

households <- data.frame(household_id = hh, gp =
runif(length(hh)))

hhdata <- merge(hhdata, households, by="household_id")
```





Lower bound
v.s.
Upper bound

Model evaluation and critique

- To decide if a given score is high or low
  - a null model : tells us what low performance looks like
  - best single-variable model : tells us what a simple model can achieve
  - a Bayes rate model : tells us what high performance looks like

Determining lower bounds on model performance

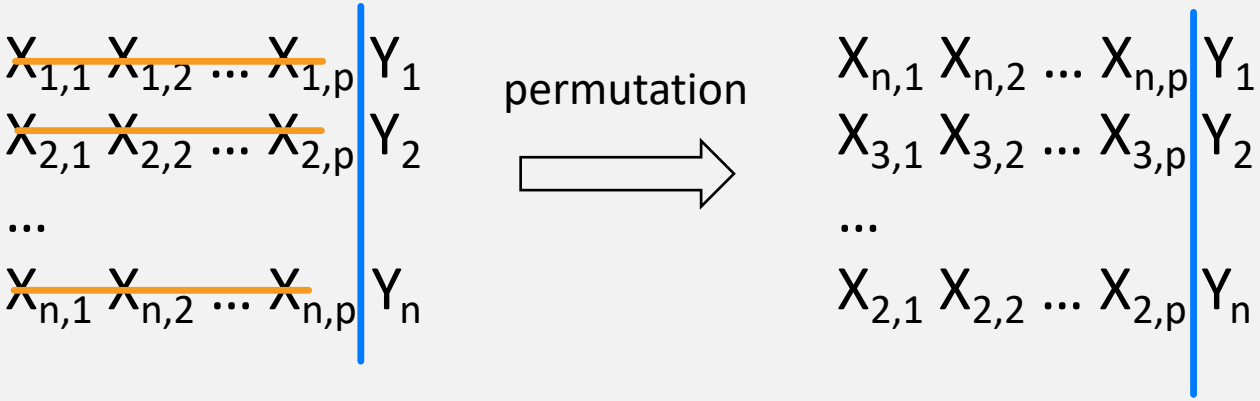
- NULL MODEL - as being “the obvious guess”
  - single constant (returns the same answer for all situations)
  - independent (doesn’t record any important relation or interaction between inputs and outputs)

How to build a null model?

- Common way
  - a categorical problem => always return the most popular Category
    - A model that labels all loans as GoodLoan => 70% accurate
  - a score model => often the average of all the outcomes

Where to find null models?

- Independent : there's no relationship between features & target
- Permutation test
  - permute the input (or independent) variables among examples
  - there's no real relation between the modeling features (which we have permuted among examples) and the quantity to be predicted



Permutation Test:

- A permutation test is a non-parametric method used to test the hypothesis of no effect or no difference. You can create a null distribution of a statistic (like a correlation coefficient or mean difference) by calculating the statistic many times, each time permuting the labels or values of the input data.

How it Works:

- You permute the input (or independent) variables among examples. This means you randomly shuffle the values of each feature column in your dataset, breaking any real association between the features and the target.
- After permutation, you re-calculate the statistic of interest (for example, the coefficient in a regression model) for this permuted dataset.
- By comparing the statistic calculated on the original data to the distribution of the statistic under the permuted datasets, you can assess the likelihood of observing your data's statistic under the null hypothesis.

The last point of the slide emphasizes that because you've broken the real relationship by permutation, the resulting model features are independent of the target and can serve as a null model against which the actual model can be benchmarked.

Single-variable models

- We also suggest comparing any complicated model against the best single-variable model you have available

Determining upper bounds on model performance

- Bayes Rate Model (also saturated model)
  - only makes mistakes when there are multiple examples with the exact same set of known facts (same *x*s) but different outcomes (different *y*s)
  - *unexplainable variance*: how much of the variation in your output can't be explained by your input variables.
    - i.e., loans that equal more than 15% of the borrower's disposable income will default; otherwise, loans are good.

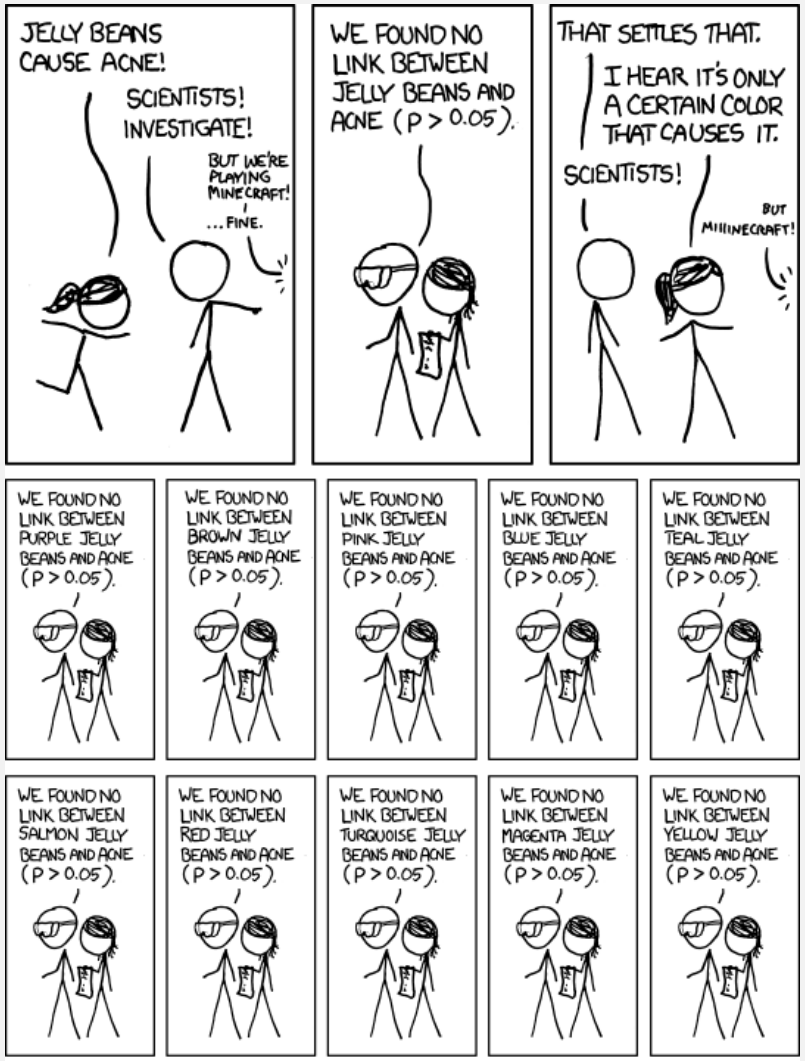
Determining upper bounds on model performance

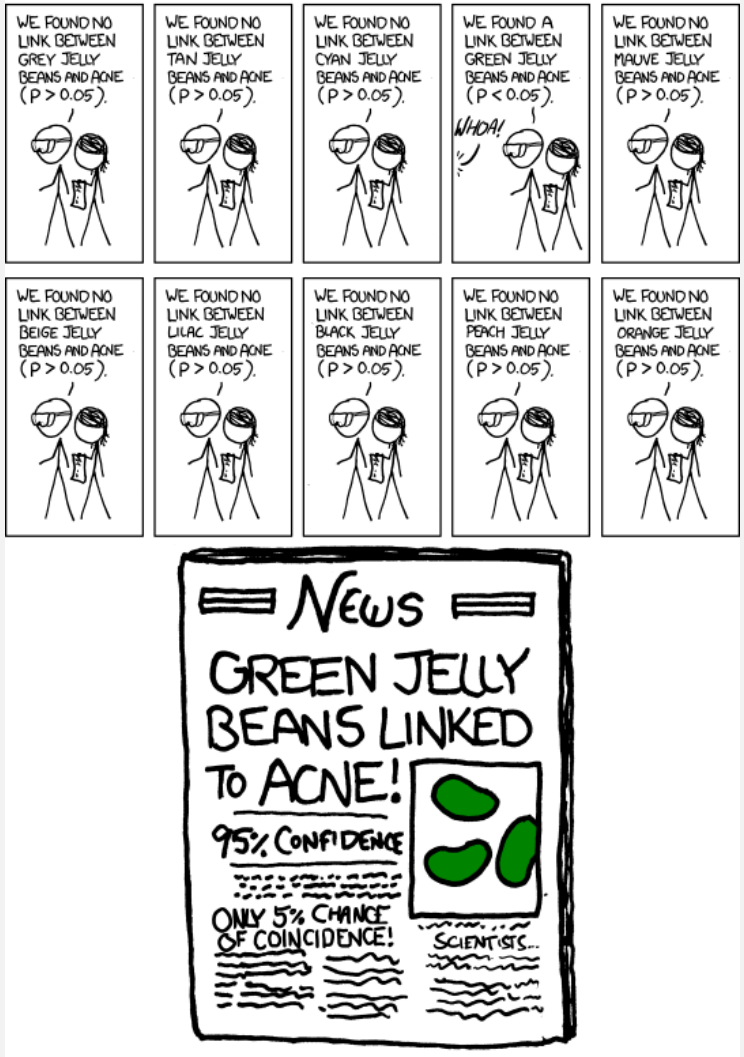
- Bayes rate : The limit on prediction accuracy due to unexplainable variance.
- => You can think of the Bayes rate as describing the best accuracy you can achieve given your data.

The Bayes error rate (or Bayes rate) gets its name because it reflects the lowest possible error that can be achieved by a classifier that has perfect knowledge of the underlying probability distributions that govern the data. In other words, it's the error rate of an optimal Bayes classifier —a theoretical model that can probabilistically predict outcomes with the highest accuracy possible given the distribution of the data. This optimal classifier is derived using Bayes' Theorem, hence the name "Bayes rate."

The Bayes classifier, which the Bayes rate refers to, makes the most probable prediction for each case based on the known distributions and Bayes' Theorem. The "rate" part of the term "Bayes rate" refers to the frequency of errors this optimal classifier would make in the long run.

Null model, *p*-value





Credit by Permanent link to this comic: https://xkcd.com/882/

Your

~~You~~ model v.s. Null model

- your model should out-perform the null model
  - 73% > 70% significantly better?

Nonsignificance	A model that appears to show an important relation when in fact the relation may not hold in the general population, or equally good predictions can be made without the relation.

Significance testing

- if it's very unlikely that a naive model could score as well as our model!!!
- Example:
  - you've trained a model to predict how much a house will sell for, based on certain variables
  - *null model* : the average selling price of a house in the neighborhood
  - Mispredict given house's selling price
    - err.model from you
    - err.null from null model

Significance testing

- *null hypothesis* D = (err.null - err.model) == 0
- *p-value* is the probability of null hypothesis
  - ***p* < 0.05 =>** reject the null hypothesis
- Student's t-test, an f-test, fisher.test() (for the confusion matrix)

Type I & Type II error

- https://en.wikipedia.org/wiki/Type_I_and_type_II_errors
- α, type I error = the rejection of a true null hypothesis (false alarm)
  - a “false positive” finding or conclusion
  - Specificity = 1- α
- β, type II error = the failure to reject a false null hypothesis (miss)
  - a "false negative" finding or conclusion
  - Sensitivity = 1- β

A/B test

歐巴馬競選網站主視覺實驗

價值6000萬美元的AB測試

Original trial VS. Family trial

https://goo.gl/77ZwXz

歐巴馬競選網站主視覺實驗

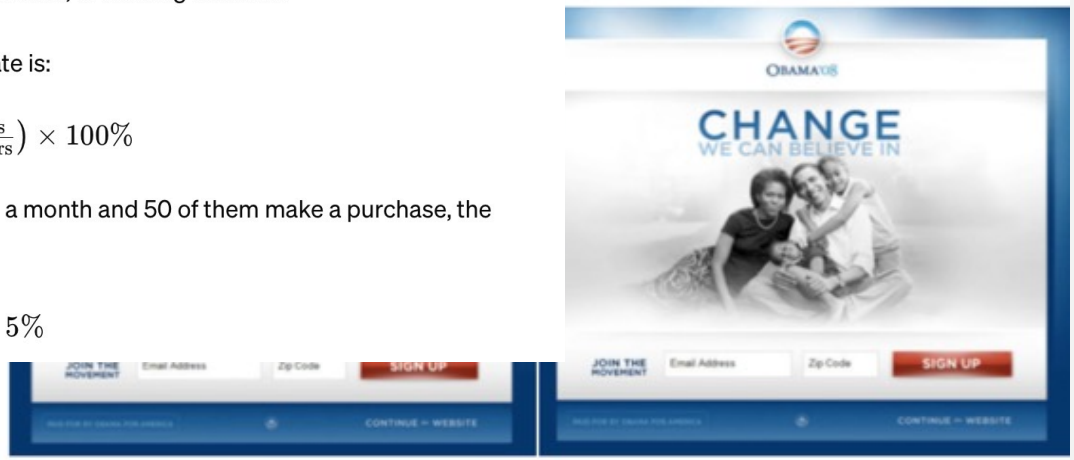
The conversion rate is a metric commonly used in the context of websites, marketing, and sales, referring to the percentage of visitors who take a desired action. The specific action considered a "conversion" can vary depending on the context but often includes activities such as making a purchase, signing up for a service, filling out a form, or clicking on a link.

The formula for calculating the conversion rate is:

$$\text{Conversion Rate} = \left(\frac{\text{Number of Conversions}}{\text{Total Number of Visitors}}\right) \times 100\%$$

For instance, if a website has 1,000 visitors in a month and 50 of them make a purchase, the conversion rate would be:

$$\text{Conversion Rate} = \left(\frac{50}{1{,}000}\right) \times 100\% = 5\%$$



Summary	Original trait	Family trait
Visitors	51,794	51,696
Sign-up	4,425	4,996
Conv. Rate	8.54%	9.66%
		✔

A/B tests

- Hard statistical problems usually arise from poor experimental design.
- A (control) / B(treatment) testing:
  - Each group is big enough that you get a reliable measurement (this drives significance).
  - Each group is (up to a single factor) distributed exactly like populations you expect in the future (this drives relevance). In particular, both samples are run in parallel at the same time.
  - The two groups differ only with respect to the single factor you're trying to test.

Evaluating A/B tests

- evalABtest.R
- Building simulated A/B test data

```
set.seed(123515)
d <- rbind(
  data.frame(group='A',converted=rbinom(100000,size=1,p=0.05)),
  data.frame(group='B',converted=rbinom(10000,size=1,p=0.055))
)
```

- Summarizing the A/B test into a contingency table

```
tab <- table(d)
print(tab)
```

The contingency table

- AKA
  - a cross tabulation
  - a crosstab
- a table showing the distribution of one variable in rows and another in columns, used to study the association between the two variable.

	converted	
group	0	1
A	94979	5021
B	9398	602

Chi-squared test

Chi-squared/Fisher’s exact test

- assess for independence between two variables when the comparing groups are independent and not correlated
  - chi-squared test : apply an approximation assuming the sample is large
  - Fisher's exact test : runs an exact procedure especially for small-sized samples

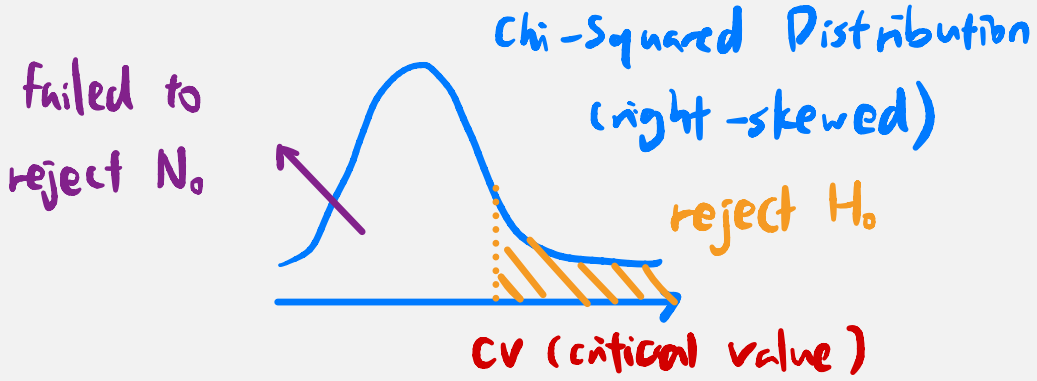
Credit by Hae-Young Kim, Statistical notes for clinical researchers: Chi-squared test and Fisher's exact test

Chi-squared test

- used to compare the distribution of a categorical variable in a sample or a group with the distribution in another one.
- If the distribution of the categorical variable is not much different over different groups, we can conclude the distribution of the categorical variable is not related to the variable of groups.

$< \alpha$

null hypothesis (H_0)



Chi-squared test example 1&2

- condition (A and B) v.s. gender (male and female)
  - there is equal chance of having the condition among men and women, we will find the chance of observing the condition is the same regardless of gender and can conclude their relationship as independent.
- Contingency tables and procedure for Chi-squared test

	Observed frequency (O)			**Expected frequency (E)**			**O – E**			**$(O - E)^2 / E$**
		A	B	Total	A	B	Total	A	B	Total	A
Example 1	Male	50	50	100	50	50	100	0	0	0	0
	Female	50	50	100	50	50	100	0	0	0	0
	Total	100	100	200	100	100	200	0	0	0	0
		A	B	Total	A	B	Total	A	B	Total	A
Example 2	Male	30	70	100	30	70	100	0	0	0	0
	Female	30	70	100	30	70	100	0	0	0	0
	Total	60	140	200	60	140	200	0	0	0	0

Credit by Hae-Young Kim, Statistical notes for clinical researchers: Chi-squared test and Fisher's exact test

Chi-squared test example 3

- condition (A and B) v.s. gender (male and female)
  - women had a greater chance to have the condition A (p = 0.7) compared to men (p = 0.3)
  - men have a specific condition more than women, there is bigger chance to find a person with the condition among men than among women.
- gender is NOT independent from the condition

		Observed frequency (O)			Expected frequency (E)			O − E			$(O-E)^2/E$
		A	B	Total	A	B	Total	A	B	Total	A
Example 3	Male	30	70	100	50	50	100	−20	20	0	8
	Female	70	30	100	50	50	100	20	−20	0	8
	Total	100	100	200	100	100	200	0	0	0	16

Credit by Hae-Young Kim, Statistical notes for clinical researchers: Chi-squared test and Fisher's exact test

The test statistic of chi-squared test

- $\chi^2 = \sum \frac{(O-E)^2}{E} \sim \chi^2$ with degrees of freedom $(r - 1)(c - 1)$, where
  - O : observed frequency
  - E : expected frequency
  - r : the number of rows of the contingency table
  - c : the number of columns of the contingency table

the Degrees of Freedom (DF)

- the maximum number of logically independent values, which are values that have the freedom to vary, in the data sample.
- if the variance is to be estimated from a random sample of *N* independent scores, then the degrees of freedom =
  - the number of independent scores (*N*) minus the number of parameters estimated as intermediate steps (one, namely, the sample mean)
  - $N - 1$.
- Example
  - The values of the five integers must have an average of six.
  - If four of the items {3, 8, 5, and 4}, the fifth number must be 10.
  - Because the first four numbers can be chosen at random, the degrees of freedom is four.

Credit by https://www.investopedia.com/terms/d/degrees-of-freedom.asp

The chi-squared test of examples

- For the 'male and A' cell in example 3
  - $\frac{(O-E)^2}{E} = \frac{(30-50)^2}{50} = 8$
- Chi-squared statistic calculate
  - Example 1 & 2 : 0
  - Example 3 : $\sum \frac{(O-E)^2}{E} = 8 + 8 + 8 + 8 = 32$

		Observed frequency (O)			Expected frequency (E)			O – E			$(O-E)^2 / E$
		A	B	Total	A	B	Total	A	B	Total	A
Example 1	Male	50	50	100	50	50	100	0	0	0	0
	Female	50	50	100	50	50	100	0	0	0	0
	Total	100	100	200	100	100	200	0	0	0	0
		A	B	Total	A	B	Total	A	B	Total	A
Example 2	Male	30	70	100	30	70	100	0	0	0	0
	Female	30	70	100	30	70	100	0	0	0	0
	Total	60	140	200	60	140	200	0	0	0	0
		A	B	Total	A	B	Total	A	B	Total	A
Example 3	Male	30	70	100	50	50	100	−20	20	0	8
	Female	70	30	100	50	50	100	20	−20	0	8
	Total	100	100	200	100	100	200	0	0	0	16

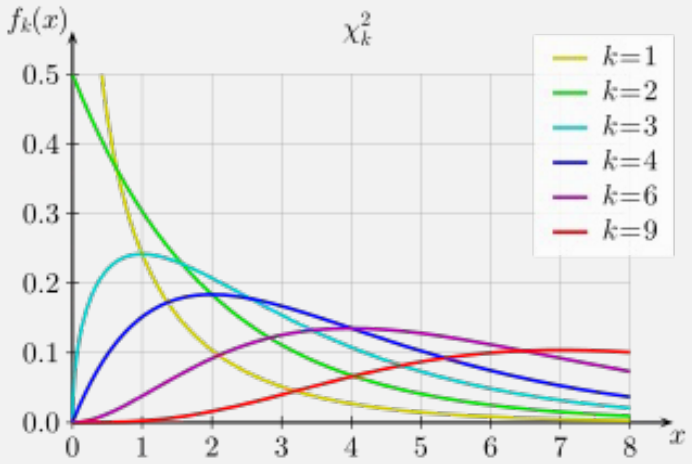
The chi-squared distribution

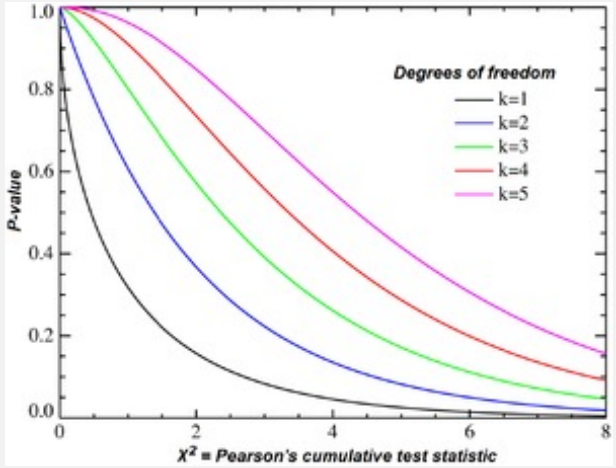
- If Z_1, ..., Z_k are independent, standard normal random variables, then the sum of their squares is distributed according to the chi-squared distribution with k degrees of freedom

$$Q = \sum_{i=1}^{k} Z_i^2$$

- Usually denoted as $Q \sim \chi^2(k)$ or $Q \sim \chi_k^2$

PDF





Credit by https://en.wikipedia.org/wiki/Chi-squared_distribution

The chi-squared distribution

- At an alpha level of 0.05 (Type I error, FP)
  - DF = 1, critical value = ?

```
qchisq(0.95, df = 1)
```

// 3.841459

Make conclusion referring to the chi-squared distribution

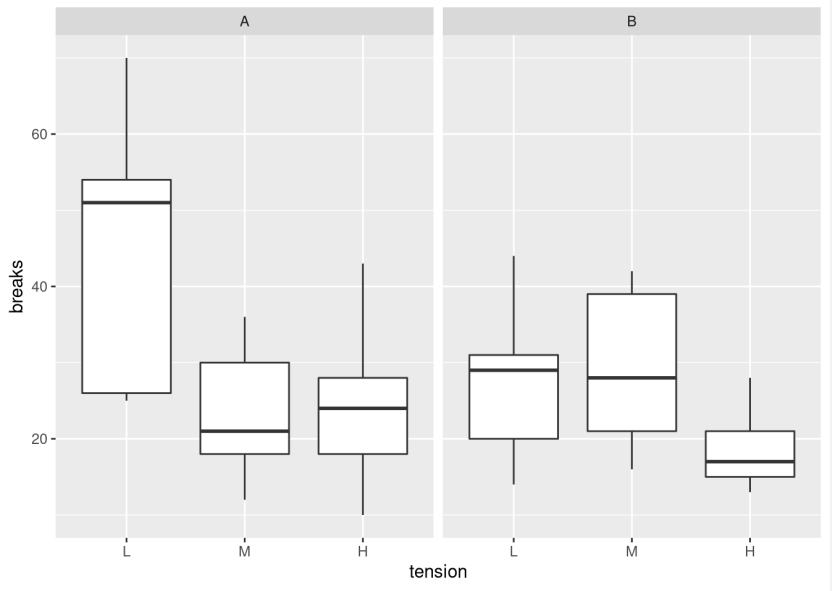
- The null hypothesis, H0: independent (no association)
- Example 3
  - The degrees of freedom (DF) = 1
    - the data has two rows and two columns: (r - 1) * (c - 1) = (2 - 1) * (2 - 1) = 1
  - a large chi-squared statistic of 32 > 3.84
  - large enough to reject the null hypothesis of independence
    => conclude a significant association between two variables.

chi-squared table

	P										
DF	0.995	0.975	0.2	0.1	0.05	0.025	0.02	0.01	0.005	0.002	0.001
1	.0004	.00016	1.642	2.706	3.841	5.024	5.412	6.635	7.879	9.55	10.828
2	0.01	0.0506	3.219	4.605	5.991	7.378	7.824	9.21	10.597	12.429	13.816
3	0.0717	0.216	4.642	6.251	7.815	9.348	9.837	11.345	12.838	14.796	16.266
4	0.207	0.484	5.989	7.779	9.488	11.143	11.668	13.277	14.86	16.924	18.467
5	0.412	0.831	7.289	9.236	11.07	12.833	13.388	15.086	16.75	18.907	20.515
6	0.676	1.237	8.558	10.645	12.592	14.449	15.033	16.812	18.548	20.791	22.458
7	0.989	1.69	9.803	12.017	14.067	16.013	16.622	18.475	20.278	22.601	24.322
8	1.344	2.18	11.03	13.362	15.507	17.535	18.168	20.09	21.955	24.352	26.124
9	1.735	2.7	12.242	14.684	16.919	19.023	19.679	21.666	23.589	26.056	27.877
10	2.156	3.247	13.442	15.987	18.307	20.483	21.161	23.209	25.188	27.722	29.588
11	2.603	3.816	14.631	17.275	19.675	21.92	22.618	24.725	26.757	29.354	31.264
12	3.074	4.404	15.812	18.549	21.026	23.337	24.054	26.217	28.3	30.957	32.909
13	3.565	5.009	16.985	19.812	22.362	24.736	25.472	27.688	29.819	32.535	34.528
14	4.075	5.629	18.151	21.064	23.685	26.119	26.873	29.141	31.319	34.091	36.123
15	4.601	6.262	19.311	22.307	24.996	27.488	28.259	30.578	32.801	35.628	37.697
16	5.142	6.908	20.465	23.542	26.296	28.845	29.633	32	34.267	37.146	39.252
17	5.697	7.564	21.615	24.769	27.587	30.191	30.995	33.409	35.718	38.648	40.79
18	6.265	8.231	22.76	25.989	28.869	31.526	32.346	34.805	37.156	40.136	42.312
19	6.844	8.907	23.9	27.204	30.144	32.852	33.687	36.191	38.582	41.61	43.82
20	7.434	9.591	25.038	28.412	31.41	34.17	35.02	37.566	39.997	43.072	45.315

Another example for chi-square test

- chiSquare_example.R
- warpbreaks data set from the textile industry
  - breaks : the number of times there was a break in a warp thread
  - wool $\in \{A,B\}$: the type of wool that was tested
  - tension $\in \{L,M,H\}$: the tension that was applied to the thread (either low, medium, or high)
- We would like to identify whether one type of wool outperforms the other for different levels of tensions.



Fisher’s exact test

Fisher's test for independence

- the null hypothesis : conversion is independent of group => A=B contingency table

```
fisher.test(tab)
```

- How to interpreter?
  - *odds ratio* or *clinical significance*
  - 1.2 => 20% relative improvement in conversion rate between the A and B groups

- Even if a result is statistically significant, it may not be clinically or practically significant. Clinical significance refers to the real-world relevance or importance of the finding.
- For example, a statistically significant result might show a 5% improvement in conversion rates, but if this improvement does not justify the cost or effort involved in implementing the change, it might not be considered clinically significant.

```
> fisher.test(tab)

        Fisher's Exact Test for Count Data

data:  tab
p-value = 2.469e-05
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 1.108716 1.322464
sample estimates:
odds ratio
  1.211706
```

In statistical hypothesis testing, the p-value is a measure used to evaluate the strength of the evidence against the null hypothesis provided by the data. Technically, the p-value is defined as the probability, under the null hypothesis H_0, of obtaining a result equal to or more extreme than what was actually observed.

When you perform a test of significance on a set of data, you calculate a test statistic (e.g., z-score, t-score) that measures the degree of deviation of the sample result from the hypothesis stated in H_0. The p-value is then found by referring this test statistic to its corresponding probability distribution under the null hypothesis.

The calculation of the p-value depends on whether the test is one-tailed or two-tailed:

- **One-tailed test**: You calculate the probability of observing a result as extreme as the test statistic in one direction (either greater than or less than).
- **Two-tailed test**: You calculate the probability of observing a result as extreme as the test statistic in both directions (both greater than and less than).

If the p-value is low (commonly set at a threshold of 0.05), it suggests that the observed data are unlikely under the null hypothesis, and thus, the null hypothesis is rejected. This low p-value indicates a statistically significant difference from what the null hypothesis would predict. Conversely, a high p-value indicates that the observed data are consistent with the null hypothesis, and no evidence exists to reject it.

The odds ratio (OR) is a measure of association between two variables, often used in the context of a 2×2 contingency table. It represents how the odds of the outcome change with the presence or absence of a particular factor when comparing two groups.

Here's how you calculate and interpret it:

Calculation:

In a 2×2 table with the following setup:

	Outcome Positive	**Outcome Negative**
Group A	a	b
Group B	c	d

The odds of the outcome in Group A are $\frac{a}{b}$, and the odds in Group B are $\frac{c}{d}$.

The odds ratio is the ratio of these two odds:

$$\text{OR} = \frac{\frac{a}{b}}{\frac{c}{d}} = \frac{a \times d}{b \times c}$$

Interpretation:

- **OR = 1**: The odds of the outcome are the same in both groups. There's no association between the factor and the outcome.
- **OR > 1**: The odds of the outcome are higher in Group A than in Group B. There's a positive association between the factor and the outcome.
- **OR < 1**: The odds of the outcome are lower in Group A than in Group B. There's a negative association between the factor and the outcome.

Confidence Interval:

- The 95% confidence interval (CI) for the OR gives a range of values that is likely to contain the true OR. If the CI does not include 1, it suggests that the OR is statistically significant.

Contextual Interpretation:

- Even if an OR is statistically significant, it's important to consider the size and direction of the association and whether it's meaningful in practical terms. For instance, an OR of 2.0 suggests that the outcome is twice as likely in Group A than in Group B, which might be highly relevant depending on the context.
- In clinical studies, a large OR might indicate a strong effect of a treatment or risk factor. However, for ORs close to 1, even if statistically significant, the actual difference in odds might not be clinically important.

Limitations:

- The OR does not directly give you the difference in risk or probability between two groups.
- It is not symmetrical; the OR for the risk of an outcome associated with a factor is not the reciprocal of the OR for the risk of the outcome associated with the absence of that factor.
- The interpretation of ORs can be counterintuitive in studies with low incidence rates, as ORs can overestimate the risk.

In summary, the odds ratio is a valuable measure of the strength and direction of the association between exposure and outcome, but it should be interpreted with an understanding of the broader context and potential limitations.

Evaluating A/B tests

- observed A and B rates?

```
(aConversionRate <- tab['A','1']/sum(tab['A',]))
(bConversionRate <- tab['B','1']/sum(tab['B',]))
(commonRate <- sum(tab[,'1'])/sum(tab))
```

- Could such a difference be likely for this sample size due to mere chance and measurement noise?

The conversion rate is a metric commonly used in the context of websites, marketing, and sales, referring to the percentage of visitors who take a desired action. The specific action considered a "conversion" can vary depending on the context but often includes activities such as making a purchase, signing up for a service, filling out a form, or clicking on a link.

The formula for calculating the conversion rate is:

$$\text{Conversion Rate} = \left(\frac{\text{Number of Conversions}}{\text{Total Number of Visitors}}\right) \times 100\%$$

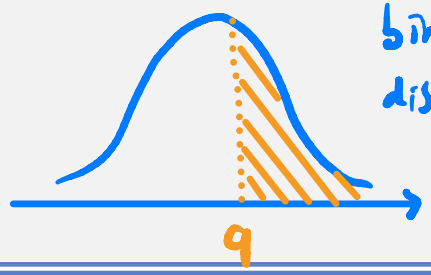
For instance, if a website has 1,000 visitors in a month and 50 of them make a purchase, the conversion rate would be:

$$\text{Conversion Rate} = \left(\frac{50}{1{,}000}\right) \times 100\% = 5\%$$

Frequentist significance test

- assume that *A* and *B* come from an identical distribution with a common conversion rate
- => how likely it would be that the B group scores as high as it did by mere chance!

```
print(pbinom( # cumulative binomial
    lower.tail=F, # upper tail (higher score)
    q=tab['B','1']-1, # (number of success observed in B) -1
    size=sum(tab['B',]), # number of trials
    prob=commonRate # assumption
    ))
```



binomial distribution (note that b.d. is actually discrete!)

The term "frequentist significance test" doesn't refer to a single specific test but rather a category of hypothesis tests within the frequentist statistical framework. In frequentist statistics, significance testing is used to determine whether the observed data are inconsistent with a specified null hypothesis.

The "frequentist" label distinguishes these tests from Bayesian methods, which incorporate prior probabilities in addition to the observed data. Frequentist tests rely on the frequency or proportion of data to make inferences about probabilities.

Common frequentist significance tests include:

- **t-tests**: For comparing means between two groups.
- **ANOVA (Analysis of Variance)**: For comparing means across three or more groups.
- **Chi-squared tests**: For assessing relationships between categorical variables.
- **F-tests**: For comparing variances or for tests in regression analysis.
- **Z-tests**: For comparing sample and population means when the population variance is known.
- **Binomial tests**: For testing outcomes with two possible states in a binary variable, like success/failure.

Each of these tests calculates a test statistic which, under the null hypothesis, follows a known probability distribution. The significance of the test is determined by how far the test statistic falls into the tail of this distribution, often resulting in a p-value, which is the probability of observing a test statistic as extreme as or more extreme than the one calculated from your data, assuming the null hypothesis is true. If the p-value is less than a predetermined threshold (often 0.05), the result is declared statistically significant, leading ↓ e rejection of the null hypothesis.

Statistical test power

(get a small p-value)

power: probability that the test will correctly reject H_0

- probability of rejecting the null hypothesis when the null hypothesis is false
- = 1 - *p*-value ???
  - a travel site that has 6,000 unique visitors per day and a 4% conversion rate from page views to purchase enquiries
  - test a new design B for the site
- depends upon
  - significance level (α)
  - sample size
  - effect size
  - population variance

Statistical test power

- seeing a B conversion rate in the range of 4.1 ~ 4.9% if the true B conversion rate were in fact 4.5% => how many customers to the B treatment? estimate sample size

Parameter	Meaning	Value for our example
`confidence` (or `power`)	This is how likely you want it to be that the test result is correct. We'll write `confidence = 1 - errorProb`.	0.95 (or 95% confident), or `errorProb=0.05`.
`targetRate`	This is the conversion rate you hope the B treatment achieves: the further away from the A rate, the better.	We hope the B treatment is at least 0.045 or a 4.5% conversion rate.
`difference`	This is how big an error in conversion rate estimate we can tolerate.	We'll try to estimate the conversion rate to within plus/minus 0.4%, or 0.004, which is greater than the distance from our `targetRate` and our historical A conversion rate.

Sample size estimate

Power Analysis

- Size $= \left\lceil \frac{-\log(errorProb) * targetRate}{difference^2} \right\rceil$
- sampleSize.R

```
estimate <- function(targetRate,difference,errorProb)
{
    ceiling(-log(errorProb)*targetRate/(difference^2))
}
(est <- estimate(0.045,0.004,0.05))
```

8426

- We need about 8,426 visitors to have a 95% ~~chance of~~ → 1-0.05 observing a B conversion rate of at least 0.041 if the true unknown B conversion rate is at least 0.045.

0.045 - 0.004
= 0.041

Design experiments

```
ceiling(-log(errorProb)*targetRate/(difference^2))
```

- Sample size
  - Confidence
  - Difference

Design experiments

- Performance difference vs confidence
  - measure large performance differences with high confidence
  - measure small performance differences with even moderate confidence.

Parameter	Meaning	Value for our example
`confidence` (or `power`)	This is how likely you want it to be that the test result is correct. We'll write `confidence = 1 - errorProb`.	0.95 (or 95% confident), or `errorProb=0.05`.
`targetRate`	This is the conversion rate you hope the B treatment achieves: the further away from the A rate, the better.	We hope the B treatment is at least 0.045 or a 4.5% conversion rate.
`difference`	This is how big an error in conversion rate estimate we can tolerate.	We'll try to estimate the conversion rate to within plus/minus 0.4%, or 0.004, which is greater than the distance from our `targetRate` and our historical A conversion rate.

Design experiments

```
estimate <-
function(targetRate,difference,errorProb)
estimate(0.045,0.004,0.05)
```

- Performance difference vs confidence
  - measure large performance differences with high confidence

    `estimate(0.045,0.005,0.04)` #5794
  - measure small performance differences with even moderate confidence.

    `estimate(0.045,0.003,0.06)` #14068

Exact binomial sample size calculation

- We need about 8,426 visitors to have a 95% chance of observing a B conversion rate of at least 0.041 if the true unknown B conversion rate is at least 0.045. #0.04536946

```
pbinom(ceiling(0.041* 8426),8426,0.045)
```

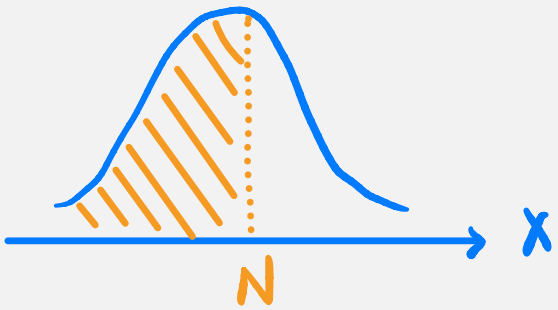
The failure odds are around 4% (under the 5% we're designing for), which means the estimate size was slightly high.

Binary search that finds a non-positive value of a function

```
errorProb <- function(targetRate,difference,size)
{
   pbinom(ceiling((targetRate-difference)*size),
      size=size,prob=targetRate)
}
```

N = # of success that would occur if the true rate were decreased by the specified difference

binomial distribution



Binary search that finds a non-positive value of a function

```
binSearchNonPositive <- function(fEventuallyNegative) {   # Note: 3
  low <- 1
  high <- low+1
  while(fEventuallyNegative(high)>0) {
    high <- 2*high
  }
  while(high>low+1) {
    m <- low + (high-low) %/% 2
    if(fEventuallyNegative(m)>0) {
      low <- m
    } else {
      high <- m
    }
  }
  high
}

actualSize <- function(targetRate,difference,errorProb) {
  binSearchNonPositive(function(n) {
      errorProb(targetRate,difference,n) - errorProb })
}
```

integer division $a \%/\% b \equiv \lfloor \frac{a}{b} \rfloor$

↗ (as opposed to $a/b \equiv \frac{a}{b}$)

search upper half

search lower half

errorProb(targetRate, difference, n) − errorProb > 0

⇒ errorProb(targetRate, difference, n) > errorProb

errorProb(targetRate, difference, n): the calculated error probability at sample size n

errorProb: the acceptable error probability threshold

anonymous function

1. **Initialization**:
   - `low` starts at 1, and `high` starts just above `low` at 2. These two variables define the bounds of the search range.
2. **Exponential Search to Establish Upper Bound**:
   - The first `while` loop increases `high` exponentially (`high = 2 * high`) until the function `fEventuallyNegative(high)` returns a value that is no longer positive. This step quickly expands the search range to include a value where the condition might be satisfied (non-positive result).
3. **Binary Search for Precise Finding**:
   - The second `while` loop then employs a binary search between `low` and `high`. The midpoint `m` is calculated, and the condition `fEventuallyNegative(m)` is tested:
     - If the result is positive, the lower bound `low` is moved up to `m`, because the target must be higher than `m`.
     - If the result is non-positive, the upper bound `high` is reduced to `m`, narrowing down to a more precise range where the condition first becomes non-positive.
4. **Convergence**:
   - The loop continues until `high` is just one more than `low`, ensuring that `high` is the smallest integer for which the function `fEventuallyNegative` returns a non-positive value.

Binary search that finds a non-positive value of a function

```
size <- actualSize(0.045,0.004,0.05)
print(errorProb(0.045,0.004,size))
```

#7623

#0.04983659

- So it's enough to route 7,623 visitors to the B treatment to expect a successful measurement.

Multiple Testing

False Discovery Rate

Venue shopping reduces test power

$\frac{1}{0.05}$

- *multiple tests* : if you run 20 treatments, each with a *p*-value goal of 0.05, you would expect one test to appear to show significant improvement, even if all 20 treatments are useless.
- Statistical hypothesis testing is based on rejecting the null hypothesis if the likelihood of the observed data under the null hypotheses is low. If multiple hypotheses are tested, the chance of observing at least a rare event increases, and therefore, the likelihood of incorrectly rejecting a null hypothesis (i.e., making a Type I error) increases.

false alarm, false positive

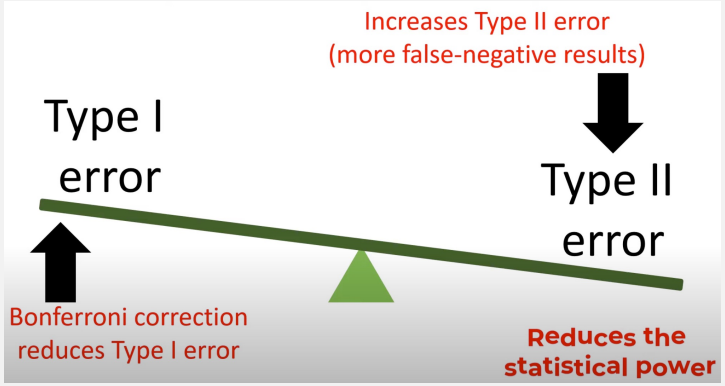
Bonferroni correction

(An example of multiple comparisons correction method)

- The Bonferroni correction compensates for that increase by testing each individual hypothesis at a significance level of

$$\alpha/m$$

  - α is the desired overall alpha level
  - m is the number of hypotheses.
- if a trial is testing m=20 hypotheses with a desired α = 0.05, then the Bonferroni correction would test each individual hypothesis at 0.05/20=0.0025.
- p cutoff
  - p / the # of tests you intend to run
  - 5%/20 = 0.25%



Number of errors committed when testing *m* null hypotheses

$\frac{m_0}{m} = \pi$

- Testing simultaneously *m* (null) hypotheses, of which m_0 is true.
- *R* is the number of hypotheses rejected.
  - Significant => then reject null hypotheses

	Declared non-significant	Declared significant	Total
True null hypotheses (H_0)	U (TN)	V (FP)	m_0
Non-true null hypotheses (H_a)	T (FN)	S (TP)	$m - m_0$
	$m - R$	R	m

Benjamini, Yoav; Hochberg, Yosef (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B*. 57 (1): 289–300. MR 1325392.

PCER & FWER

$FWER = 1-(1-\alpha)^t$

(α: significance level
t: # of tests)

- A per comparison error rate (PCER) = $E(V/m)$
- The familywise error rate (FWER) = $P(V \geq 1)$

V: type I error (false alarm)

- Testing individually each hypothesis at level α => $E(V/m) \leq \alpha$
- Testing individually each hypothesis at level α/m => $P(V \geq 1) \leq \alpha$

Bonferroni Correction

	Declared non-significant	*Declared significant*	*Total*
True null hypotheses	**U** TN	**V** FP	m_0
Non-true null hypotheses	**T** FN	**S** TP	$m - m_0$
	$m - \mathbf{R}$	**R**	m

Benjamini, Yoav; Hochberg, Yosef (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B*. 57 (1): 289–300. MR 1325392.

False Discovery Rate (FDR)

- The proportion of errors committed by falsely rejecting null hypotheses

$$Q = V/(V + S)$$

- $Q_e = E(Q) = E\{V/(V + S)\} = E\{V/R\}$

$$\frac{FP}{FP + TP}$$

	Declared non-significant	*Declared significant*	*Total*
True null hypotheses	U TN	V FP	m_0
Non-true null hypotheses	T FN	S TP	$m - m_0$
	$m - R$	R	m

Benjamini, Yoav; Hochberg, Yosef (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B*. 57 (1): 289–300. MR 1325392.

Specialized statistical tests

- Building a synthetic uncorrelated income example

```
set.seed(235236)
d <- data.frame(EarnedIncome=100000*rlnorm(100),
CapitalGains=100000*rlnorm(100)) # 100 records
print(with(d,cor(EarnedIncome,CapitalGains)))
```

lognormal

Pearson's Correlation coefficient R

Specialized statistical tests

Spearman Rank Coefficient Test

- Pearson coefficient : for normally distributed data is a Student t-test
- Spearman's rho or Kendall's tau: compare the data by rank (instead of by value)

```
with(d,cor(EarnedIncome,CapitalGains,method='spearman'))
with(d,cor.test(EarnedIncome,CapitalGains,method='spearman'
))
```

  - Unfortunately, neither **cor( )** or **cov( )** produce tests of significance, although you can use the **cor.test( )** function to test a single correlation coefficient.
  - truly uncorrelated data would show a coefficient this large about 76% of the time

Sigr package to wrap up test results

```
ctest <-
with(d,cor.test(EarnedIncome,CapitalGains,method='spearm
an'))
sigr::wrapCorTest(ctest) # use Sigr to format output

d <- data.frame(x=c(1,2,3,4,5,6,7,7),
                y=c(1,1,2,2,3,3,4,4))
ct <- cor.test(d$x,d$y)
wrapCorTest(ct)
```

Practice time @ KDD data

Preparing the KDD data for analysis *

- https://github.com/WinVector/zmPDSwR/tree/master/KDD2009
- KDDexam.R

```
d <- read.table('orange_small_train.data.gz',
   header=T,
   sep='\t',
   na.strings=c('NA',''))
churn <- read.table('orange_small_train_churn.labels.txt',
   header=F,sep='\t')
d$churn <- churn$V1
appetency <-
read.table('orange_small_train_appetency.labels.txt',
   header=F,sep='\t')
d$appetency <- appetency$V1
upselling <-
read.table('orange_small_train_upselling.labels.txt',
   header=F,sep='\t')
```

Preparing the KDD data for analysis

```
d$upselling <- upselling$V1
set.seed(729375)
d$rgroup <- runif(dim(d)[[1]])
dTrainAll <- subset(d,rgroup<=0.9)
```

Function to build single-variable models for categorical variables

```
mkPredC <- function(outCol,varCol,appCol) {
   pPos <- sum(outCol==pos)/length(outCol)
   naTab <- table(as.factor(outCol[is.na(varCol)]))
   pPosWna <- (naTab/sum(naTab))[pos]
   vTab <- table(as.factor(outCol),varCol)
   pPosWv <- (vTab[pos,]+1.0e-3*pPos)/(colSums(vTab)+1.0e-3)
   pred <- pPosWv[appCol]
   pred[is.na(appCol)] <- pPosWna
   pred[is.na(pred)] <- pPos
   pred
}
```

Running a repeated cross-validation experiment

```
var <- 'Var217'
aucs <- rep(0,100)
for(rep in 1:length(aucs)) {
   useForCalRep <- rbinom(n=dim(dTrainAll)[[1]],size=1,prob=0.1)>0
   predRep <- mkPredC(dTrainAll[!useForCalRep,outcome],
      dTrainAll[!useForCalRep,var],
      dTrainAll[useForCalRep,var])
   aucs[rep] <- calcAUC(predRep,dTrainAll[useForCalRep,outcome])
 }
mean(aucs)
sd(aucs)
```

Empirically cross-validating performance

```
fCross <- function() {
   useForCalRep <-
rbinom(n=dim(dTrainAll)[[1]],size=1,prob=0.1)>0
   predRep <- mkPredC(dTrainAll[!useForCalRep,outcome],
      dTrainAll[!useForCalRep,var],
      dTrainAll[useForCalRep,var])
   calcAUC(predRep,dTrainAll[useForCalRep,outcome])
}
aucs <- replicate(100,fCross())
```

Take home message

- Always first explore your data, but don’t start modeling before designing some measurable goals.
- Divide you model testing into establishing the model’s effect (performance on various metrics) and soundness (likelihood of being a correct model versus arising from overfitting).

Take home message

- Keep a portion of your data out of your modeling work for final testing. You may also want to subdivide your training data into training and calibration and to estimate best values for various modeling parameters.
- Keep many different model metrics in mind, and for a given project try to pick the metrics that best model your intended business goal.

References

- More on ROC/AUC
  - http://www.win-vector.com/blog/2013/01/more-on-rocauc/
- Bayesian and Frequentist Approaches: Ask the Right Question
  - http://www.win-vector.com/blog/2013/05/bayesian-and-frequentist-approaches-ask-the-right-question/

AITC
教育部人工智慧技術及應用人才培育計畫
Artificial Intelligence Talent Cultivation Program