Evaluation Metrics

Machine Learning II

Master in Business Analytics and Big Data

acastellanos@faculty.ie.edu

Definitions

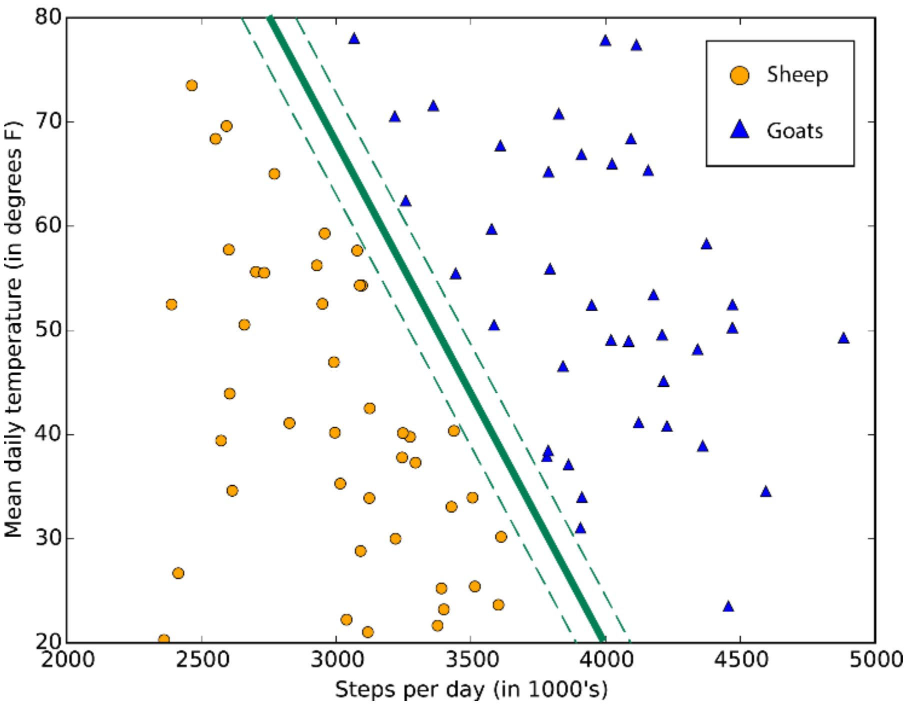
- **Model Evaluation:** performance of a model, from a **data science point of view**, and being able to translate that into the **business goals** aimed at with its construction.
  - Different problems, different models, different performance measures.

- **Model Validation**: Measure how sure we're that the model **will work in production** (new, unseen data) as well as when it was trained.
  - Namely, do we have enough training data? Is it representative?

Problem-to-ML Methods (short review)

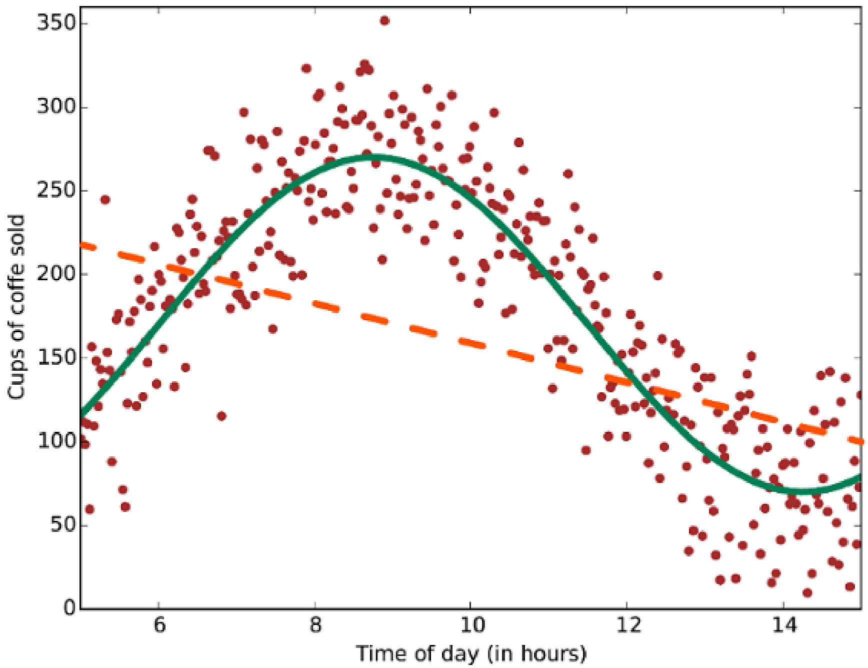
- **Classification**
- **Scoring**
- **No-target methods**

Classification problems



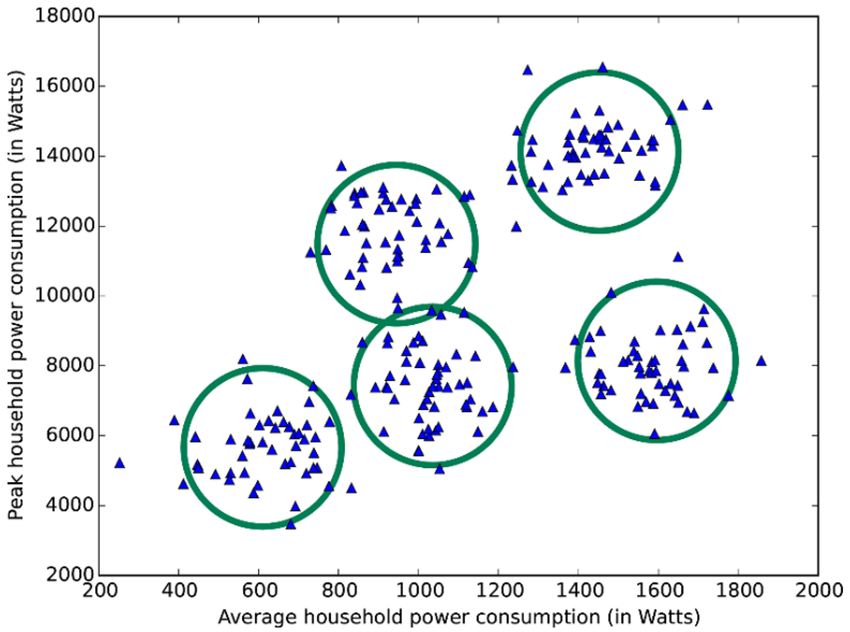
- Assign ***labels*** (***categories***) to untrained ***observations*** (***objects***).
- Can be multicategory (multinomial) or two-category (binomial).
  - We can always turn a binary classifier into a multicategory classifier.

Scoring Problems (Regression)



- Predict the output for a new set of values at the input, or estimate the probability of an event.
  - Fraud Detection
  - Predict increase in sales for a particular marketing campaign.
  - Predict a value, given a known set of past observations.
- Methods:
  - Linear Regression
  - Logistic Regression

No-target Problems



- There's no outcome we want to predict
- Identify patterns or relationships in the data.
- Methods:
  - Clustering
    - Useful when we don't know what we're looking for.
    - Ambiguous.
  - Apriori algorithms
    - Recommendation systems, association rules (market basket analysis).
  - Nearest neighbor
    - Supervised classification method

Model Evaluation Metrics

- **Classification**
- **Scoring**
- **Probability Estimation**
- **Clustering**

Evaluating Classification models

- Most common
  - Confusion Matrix
  - Accuracy
- In general terms, we build a table of the counts at each combination of factors:

		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

```
> table(data$label, data$testPrediction > threshold)
          TestPrediction

label       TRUE    FALSE
  TRUE       TP       FN
  FALSE      FP       TN
```

We're turning a score into a prediction

Accuracy

- Most **common measure of performance** for classifiers
- **Definition:** # of items categorized correctly divided by the total nr. of items.

		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

$$Acc = \frac{TP + TN}{TP + FP + TN + FN}$$

- Accuracy tell us how **'accurate' is our model predicting categories for unseen data**, and will also tell us what will be the expected error rate.
- Caveat: **DO NOT use accuracy for unbalanced classes** (i.e.: predict *rare* events).

Precision and Recall

- **Precision**: fraction of predictions that actually are in the class
  - **Measure of confirmation:** how often my model predictions are correct

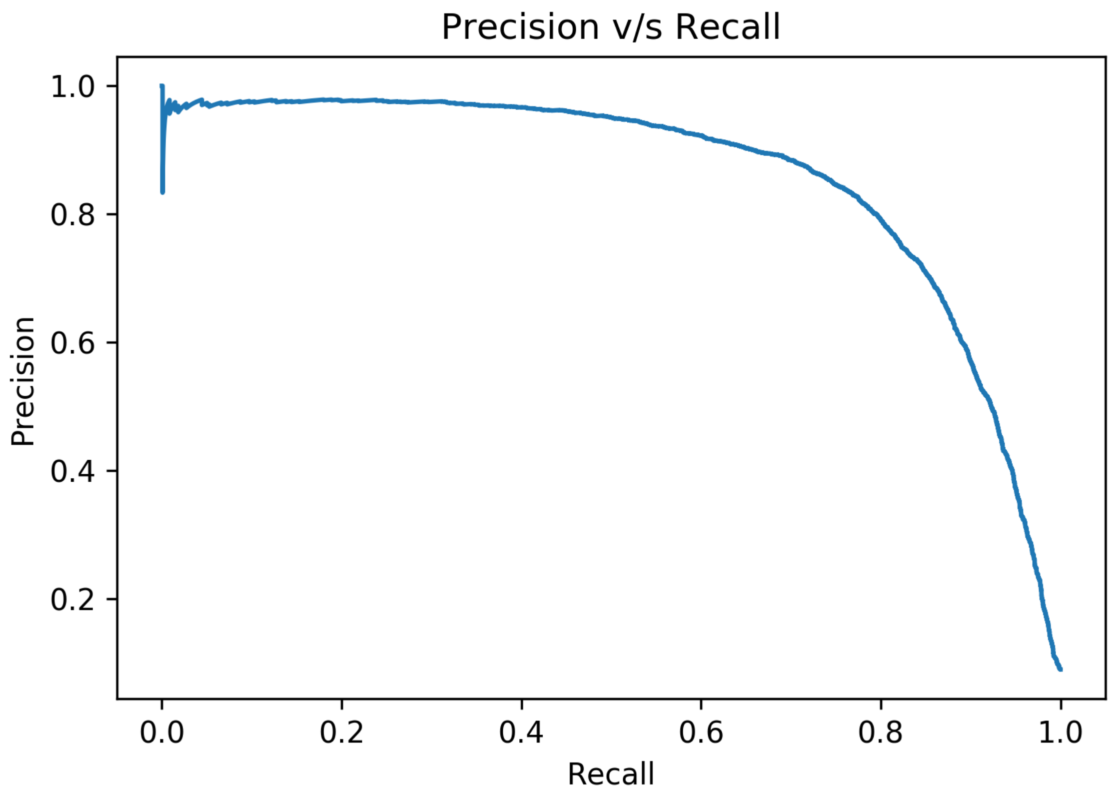
$$Precision = \frac{TP}{TP + FP}$$

		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

- **Recall**: fraction of observations in the class that actually are detected.
  - **Measure of utility:** how much my model finds what it has to find

$$Recall = \frac{TP}{TP + FN}$$

Precision-Recall Tradeoff



F1 score, or *F Score*, or *F Measure*

- **F1 is a combination of precision and recall.**
- Any model sacrifying any of them will lower its F1 score.

$$F1 = \frac{2 \cdot precission \cdot recall}{precission + recall}$$

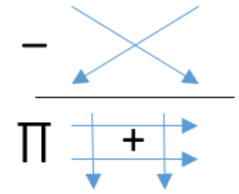
- Useful to evaluate models where we want to select **find a balance** between precision and recall
- Again, F1 metric **is not a suitable method** of combining precision and recall if there is a **class imbalance**.

Matthew's Correlation Coefficient

- MCC measures the quality of a binary classifier.

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(FN + TN)(FP + TN)(TP + FN)}}$$

		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN



- Range of -1 to 1:
  - -1 indicates a completely wrong binary classifier
  - +1 indicates a completely correct binary classifier

- It is a fair measure that **can be used with unbalanced classes**

Compare Accuracy, Recall and MCC

		Prediction	
		Positive	Negative
Actual	Positive	0	24
	Negative	0	327

Accuracy: 0.932
MCC: 0.0

		Prediction	
		Positive	Negative
Actual	Positive	24	0
	Negative	327	0

Recall: 1.0
MCC: 0.0

		Prediction	
		Positive	Negative
Actual	Positive	24	0
	Negative	0	327

MCC: 1.0

Cohen's Kappa

- Is your classifier is performing better than simply guessing at random according to the frequency of each class

		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

$$K = \frac{P(o) - P(e)}{1 - P(e)}$$

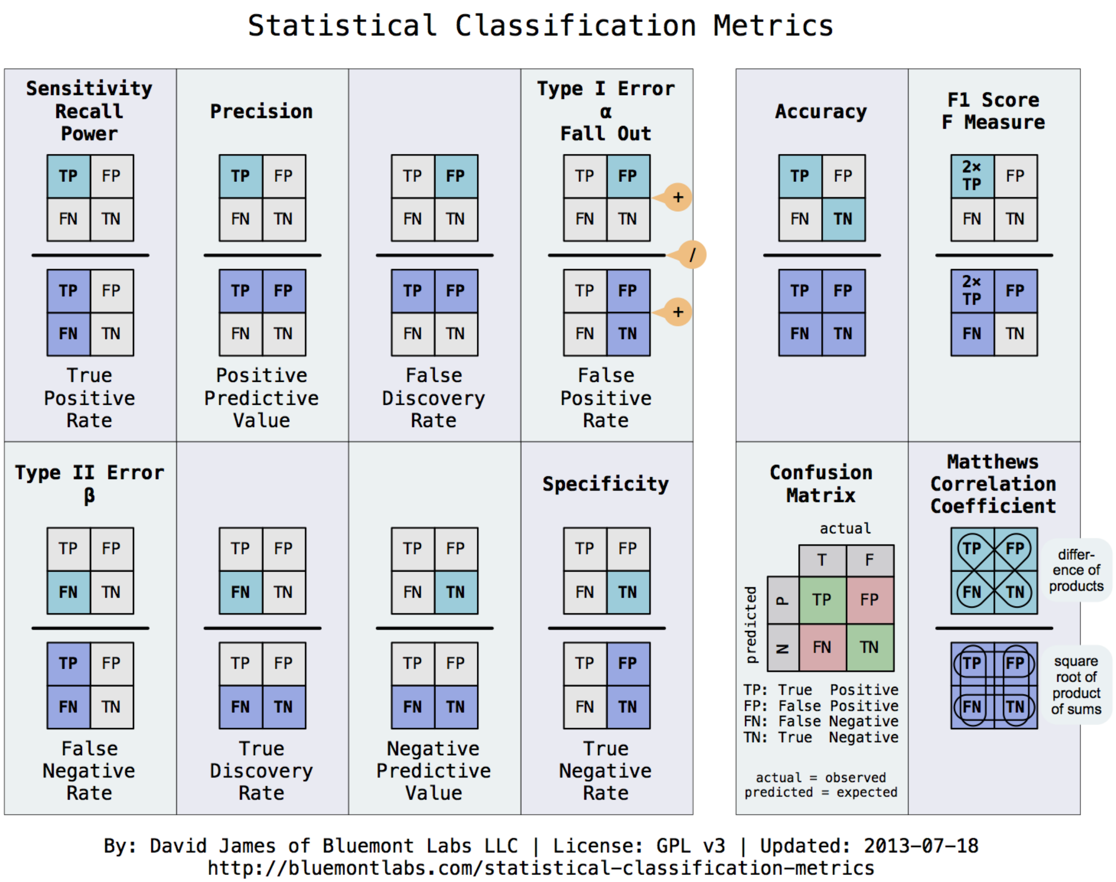
- Range: 0 to 1
- It **can be used with unbalanced classes**

What to do with imbalanced datasets

1. Collect more data
2. Change Metric: use MCC, the confusion matrix directly, F1 or precision/recall.
3. Resample with bootstrapping
4. Generate synthetic samples
5. Change the algorithm (try with trees)
6. Change approach: instead of classifying, maybe you should try anomaly detection.

https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/

Classification Metrics Summary



Application

Measure	Typical business need	Follow-up question
Accuracy	"We need most of our decisions to be correct."	"Can we tolerate being wrong 5% of the time? And do users see mistakes like spam marked as non-spam or non-spam marked as spam as being equivalent?"
Precision	"Most of what we marked as spam had darn well better be spam."	"That would guarantee that most of what is in the spam folder is in fact spam, but it isn't the best way to measure what fraction of the user's legitimate email is lost. We could cheat on this goal by sending all our users a bunch of easy-to-identify spam that we correctly identify. Maybe we really want good specificity."
Recall	"We want to cut down on the amount of spam a user sees by a factor of 10 (eliminate 90% of the spam)."	"If 10% of the spam gets through, will the user see mostly non-spam mail or mostly spam? Will this result in a good user experience?"
Sensitivity	"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	"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?"

From: Practical Data Science with R. pg. 98.

Intuition

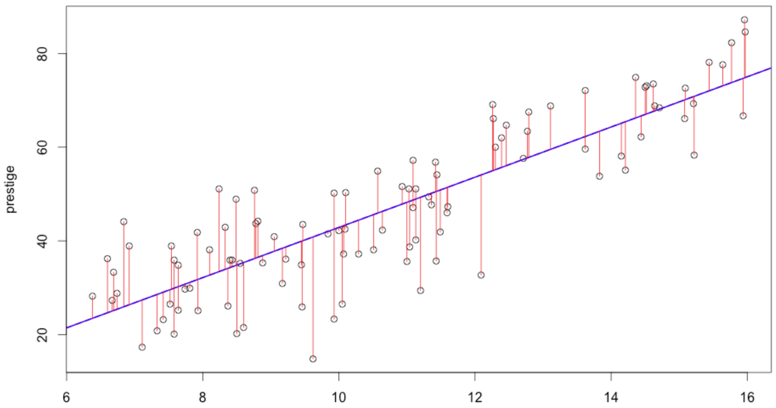
Measure	Formula	Example	Intuition
Accuracy	$\frac{(TP+TN)}{(TP+FP+TN+FN)}$	0.9214	Overall, my model is predicting the correct class in 92,14% of the cases, or missing in 7.86% of the cases
Precision	$\frac{TP}{(TP+FP)}$	0.9187	In 8.13% of the cases I'm including **false** predictions of the positive class.
Recall	$\frac{TP}{(TP+FN)}$	0.8778	I'm **missing** 12.22% of the positive cases when predicting it with my model.
Specificity	$\frac{TN}{(TN+FP)}$	0.9496	I'm **missing** 5.04% of the negative cases when predicting it with my model.

Evaluating Scoring Models

- Measure the difference between our predictions and the actual outcomes (*residuals*).

Dataset(Prestige):

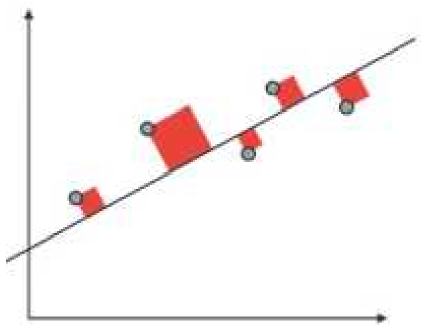
- Education: Average education of occupational incumbents, years, in 1971.
- Prestige: Pineo-Porter prestige score for occupation, from a social survey conducted in the mid-1960s.




```
attach(Prestige)
fit = lm(prestige ~ education, data=Prestige)
plot(education, prestige)
abline(fit, col="blue", lwd=2)
segments(education, prestige, education, fit$fitted.values, col="red")
residuals <- (prestige - fit$fitted.values)
```

Root Mean Square Error

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$



- The most common goodness of fit.
- Interpreted as a *standard deviation* of the prediction
- In the same units as y.

R-Squared (Coefficient of determination)

$$TSS = SS_{tot} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

$$RSS = SS_{res} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} {e_i}^2$$

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS} = 1 - \frac{SS_{res}}{SS_{tot}}$$

- What fraction of the y variation is explained (can be predicted) by the model.
- Best possible value = 1. Near 0, bad.
- Difficult to explain to the business.

Correlation (Pearson)

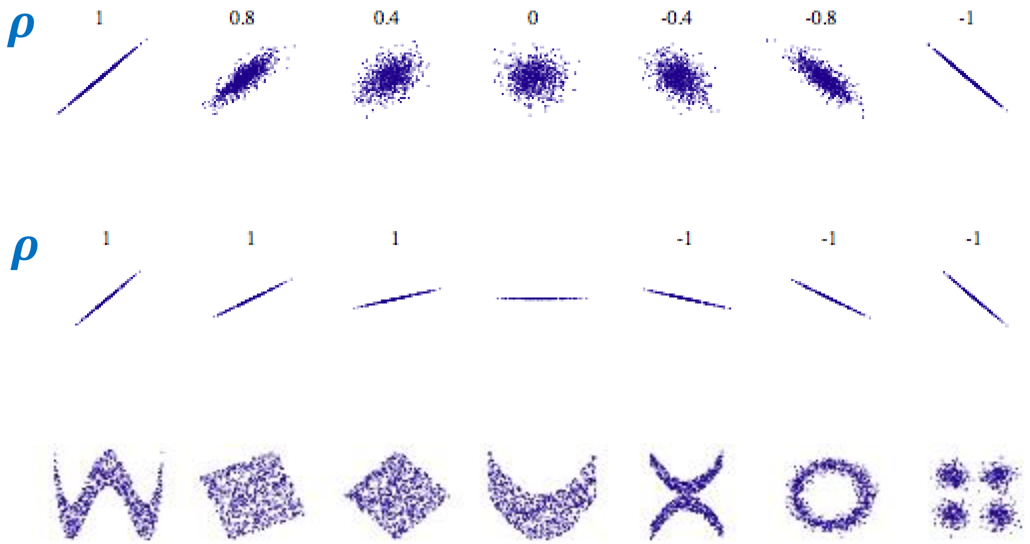
Correlation is very helpful in checking if variables are potentially useful in a model. **Do not use it to evaluate model quality**.

$$p_i = \frac{(x_i - \mu_X)}{\sigma_X} \frac{(y_i - \mu_Y)}{\sigma_Y}$$

Pearson's Correlation, $\rho = \frac{1}{n} \sum p_i$

- Pearson's correlation only measures **linear relationships.**
- Scale independent, ranges between -1 and +1
- Dimensionless
- Variables must be normally distributed and homoscedastic (have the same standard deviation in different groups).

Interpretation of Pearsons’ Correlation

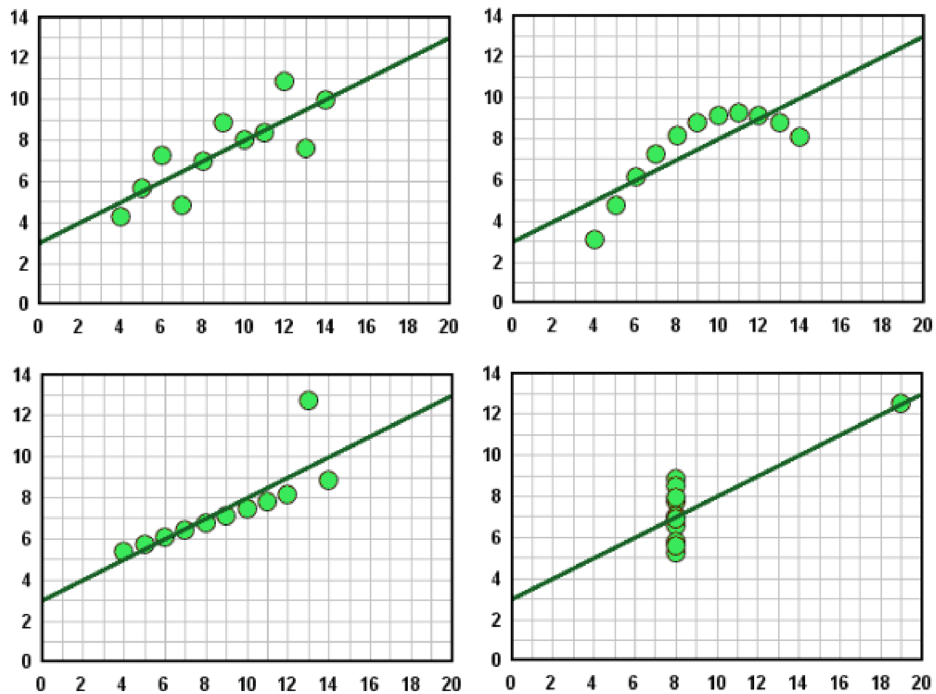


Value of ρ?

Pearson’s correlation **only** measures **linear relationships.**
Scatterplot before deciding based on Correlation Coefficient.

Correlation when non-linear

Anscombe's quartet



Property	Value
Mean(x)	9 (exact)
Var(x)	11 (exact)
Mean(y)	7.50
Var(y)	4.122 or 4.127
Cor(x,y)	0.816
Linear Reg.	y = 3.00 + 0.500x

Pearson's Correlation of Anscombe's quartet:

```
cor(anscombe$x1, anscombe$y1, method="pearson")
## [1] 0.8164205
cor(anscombe$x2, anscombe$y2, method="pearson")
## [1] 0.8162365
cor(anscombe$x3, anscombe$y3, method="pearson")
## [1] 0.8162867
cor(anscombe$x4, anscombe$y4, method="pearson")
## [1] 0.8165214
```

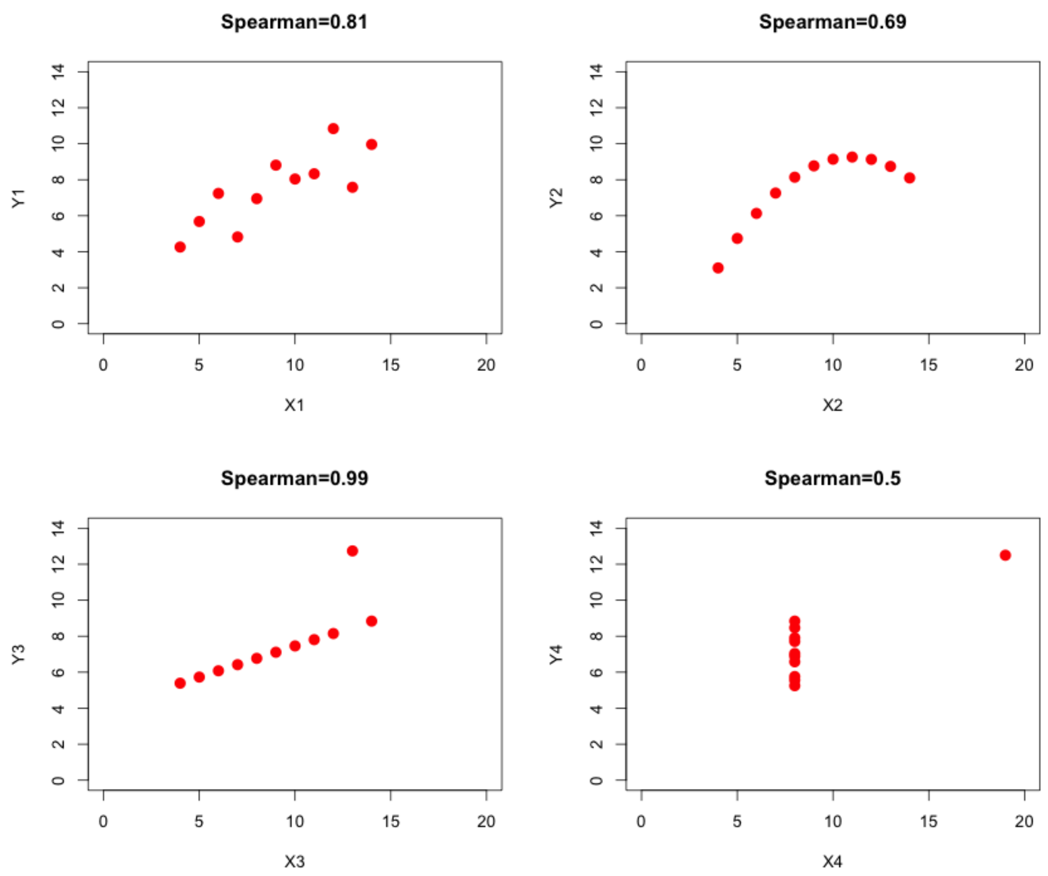
Correlation (Spearman)

$$1 - \frac{6\sum d_i^2}{n(n^2-1)}$$

(d = distance between the rank of the two variables).

- Spearman benchmarks monotonic relationship (**rank or ordered relations**).
- Mitigates the effect of outliers and skewed distributions.
- Monotonic = as X gets larger, Y keeps getting larger, or keeps getting smaller.

Spearman (Anscombe)



Absolute Error

$$Abs.Error = \sum_{i=1}^{n} |y_i - \hat{y}_i|$$

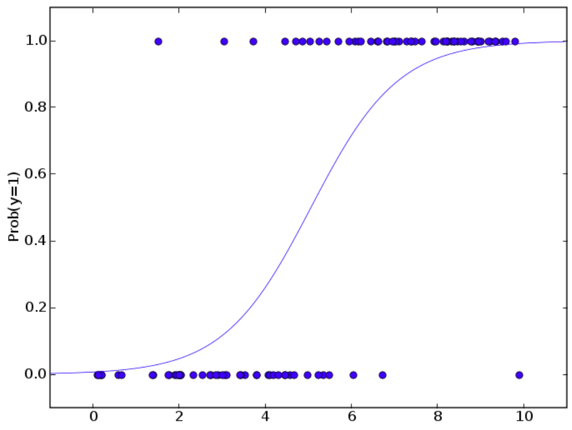
$$Mean\ Abs.Error = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$

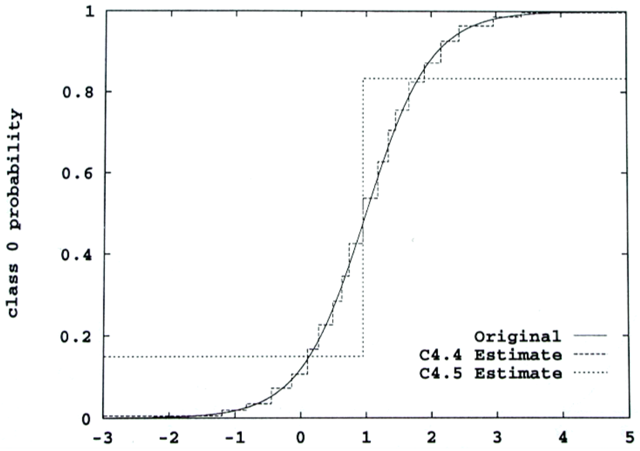
$$Abs.Error = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{\sum_{i=1}^{n} |y_i|}$$

- These measures are OK (financial models) to be reported, but...
- **…don’t make them the project goal** or to attempt to optimize them.
- Absolute error tend not to “get aggregates right” or “roll up reasonably” as most of the squared errors do.

Evaluating Probability Models

- **Probability Models** return a class where each observation belongs, together with an estimated probability (confidence) of the item being in that class.
  - Examples: Logistic Regression or Decision Trees.



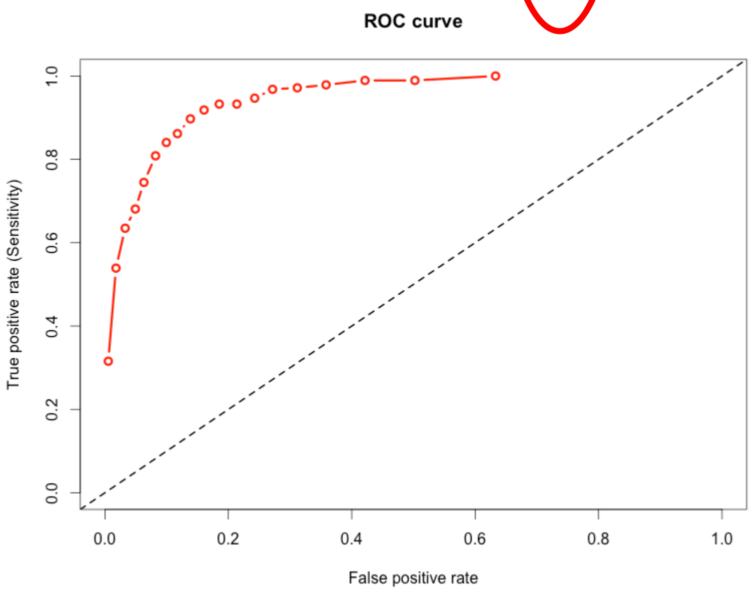


ROC Curve

- a.k.a. The True Operating Characteristic Curve.

- **How to:**
  - Compute **TRUE positive rate**, and **FALSE positive rate** for a RANGE of score thresholds.
  - Compute the **Area Under the Curve (AUC)** for the previous values

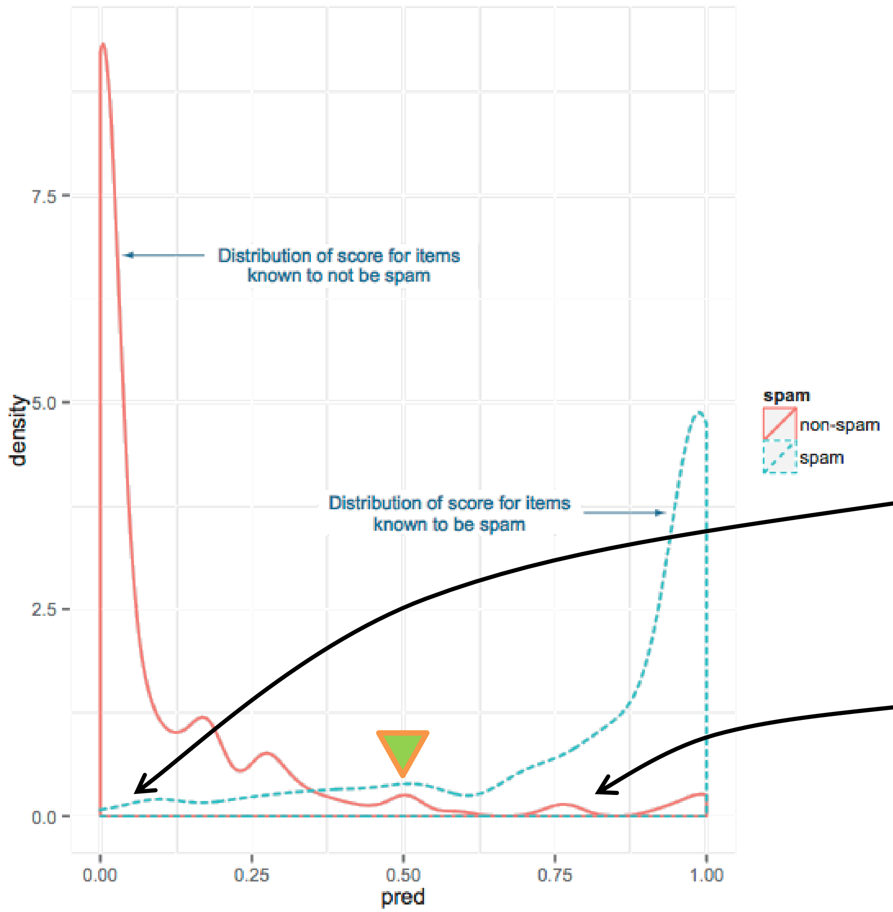
		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN



ROC Curve (remarks)

- The AUC **does not have as straightforward a business intuition** as we would hope. **Difficult to interpret.**
- It's a **single value summary expectation** of the classifier performance.
  - If you're about to use a single value (plot), use Precision and Recall.
  - F1 is also a good single-value measure of the classifier quality.
  - In case of doubt, combine them to take decisions.
- Your decision, from the threshold chosen in the ROC curve, are irrelevant to the design of other classifiers on the same problem.
- **Double density plots** are easier to explain.

Double Density Plots



If I set the **threshold** here (▼), I will **misclassify**:

- All the items known to be spam (blue) before that point.
- All the items known NOT to be spam (red) beyond that point.

Evaluating Clustering Models (1/2)

- Clustering implies trying with different values of ‘k’. The problem is to decide **which one is the best.**
- Useful **checks**:
  - Clusters with very few samples (individual samples)
  - Clusters with too many samples (nothing in common among the samples)
- **Compactness**: compare the distance between items in the same cluster to the distance between items from different clusters.
  - Mean **intra-cluster** must be smaller than **inter-cluster**.

Evaluating Clustering Models (2/2)

- As classification
  - **Do not use the labels for clustering**, and measure the overall performance assigning observations with the same label to the same cluster.
- Silhouette
  - Silhouette measures how similar an object is to its own cluster (cohesion) compared to other clusters (separation).
  - Ranges from -1 to 1
- Try different algorithms
  - K-Medians, K-Medoids, Mixture of Gaussians, Density based clustering, etc.

Model Validation

- **Model Problems**
- **Train-Test Splitting**
- **Cross Validation**
- **Significance Testing**

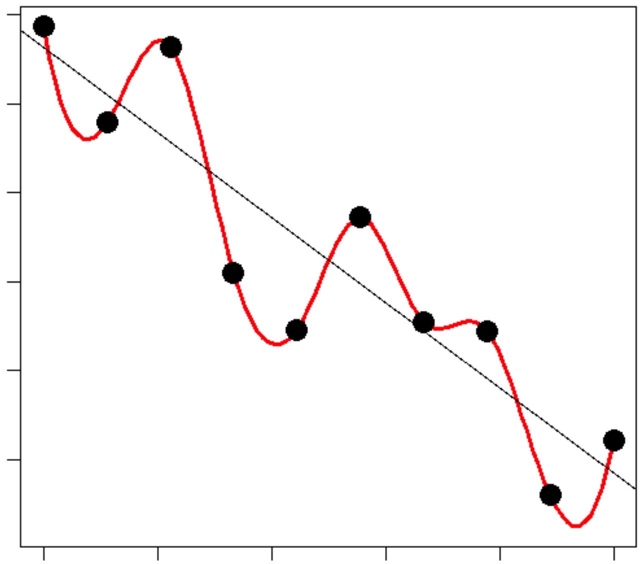
Model Problems

- Bias
  - Systematic error in the model
- Variance
  - Oversensitivity of the model to small variations in the data
- Overfit
  - Features of the model that arise from relations that are in the training data, but not representative of the general population.
- Nonsignificance
  - A model built on the assumption of an important relation when in fact the relation may not hold in the general population.

The challenge

High Variance

Drawing a curve through every training observation

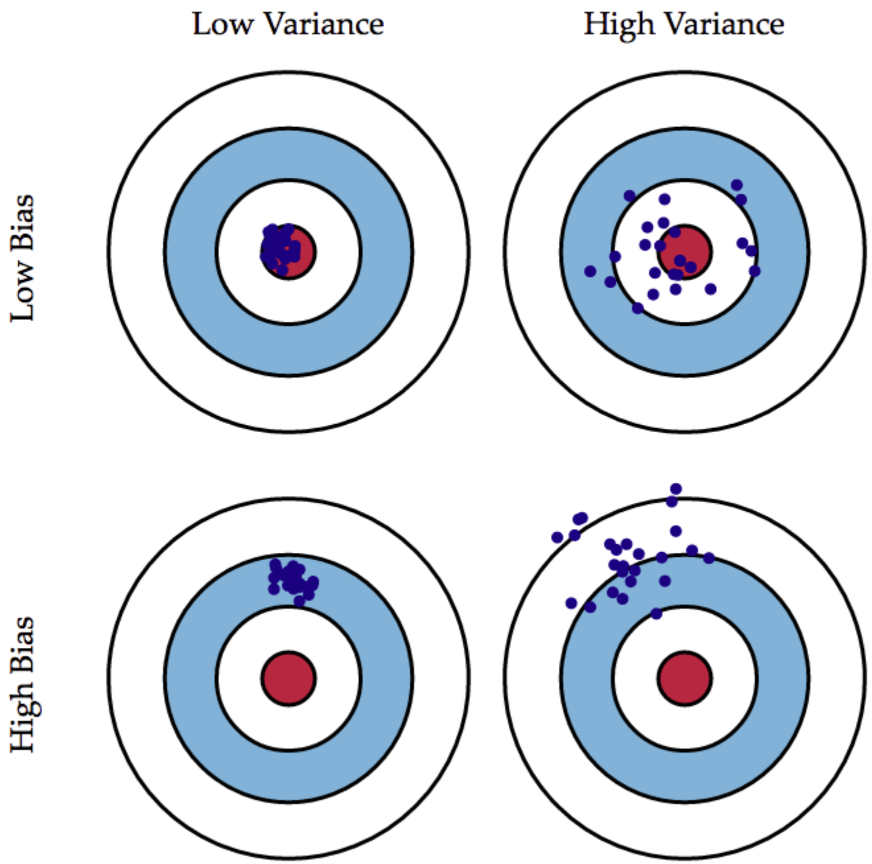


High Bias

Fitting a straight horizontal line to the data

The challenge is finding a model with **low variance and bias**

Bias-Variance Trade-Off

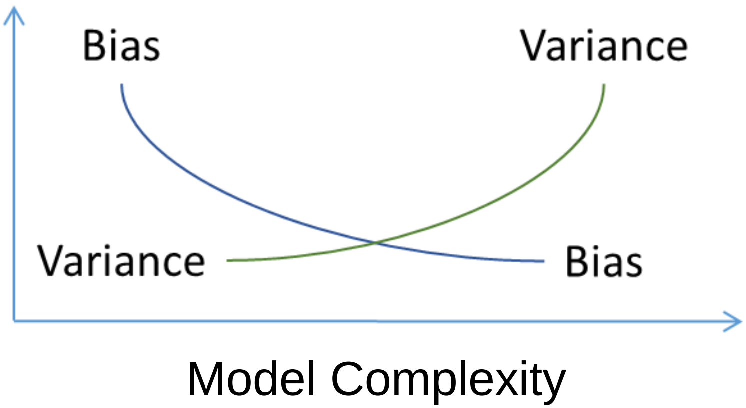


Bias-Variance Trade-Off

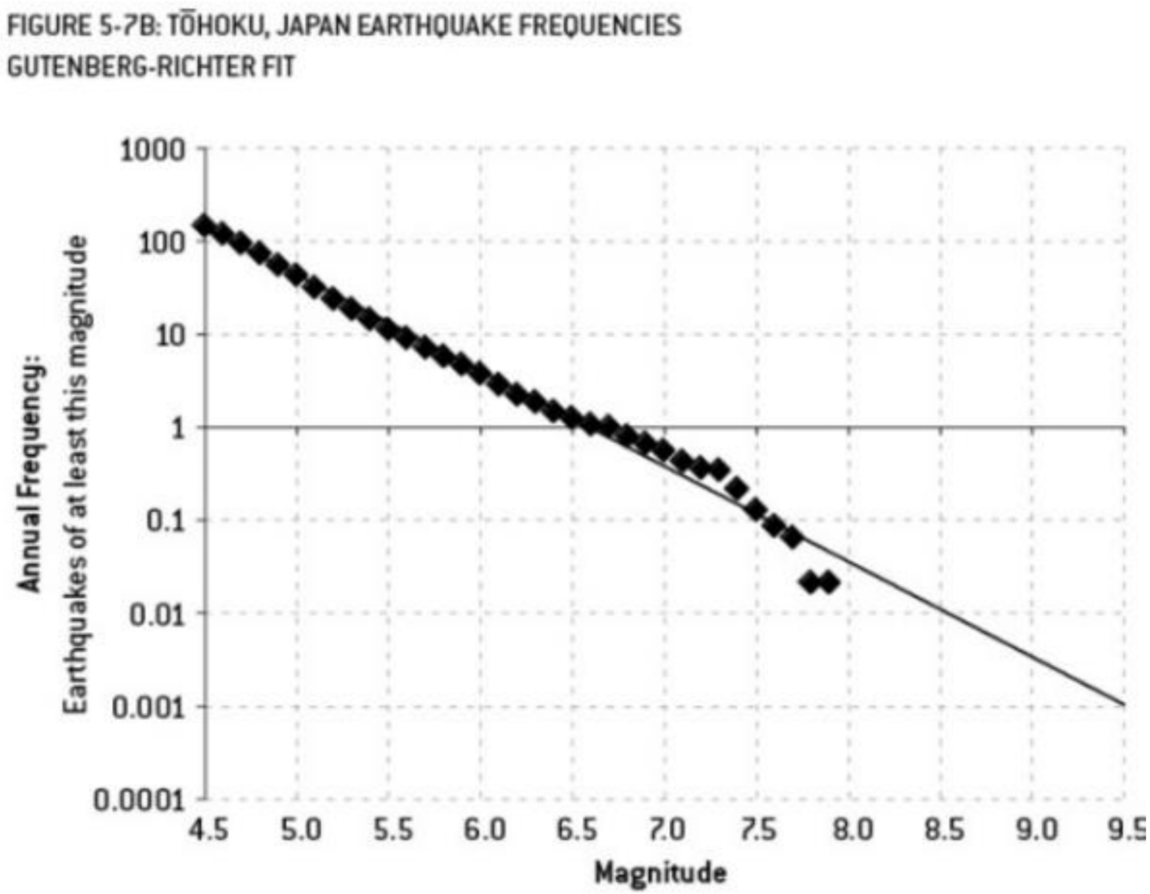
- The expected test MSE for a given value is:

$$E(y_0 - \hat{f}(x_0))^2 = \underbrace{Var(\hat{f}(x_0))}_{\text{Variance}} + \underbrace{\left[\text{Bias}(\hat{f}(x_0))\right]^2}_{\text{Bias}} + Var(\varepsilon)$$

- **Variance**: How much our estimation changes, when using different data sets. How much the predictions for a given point vary between different realizations of the model.
- **Bias**: The error introduced by approximating a complicated relationship by a simpler model. Bias measures how far off in general your models' predictions are from the correct value.



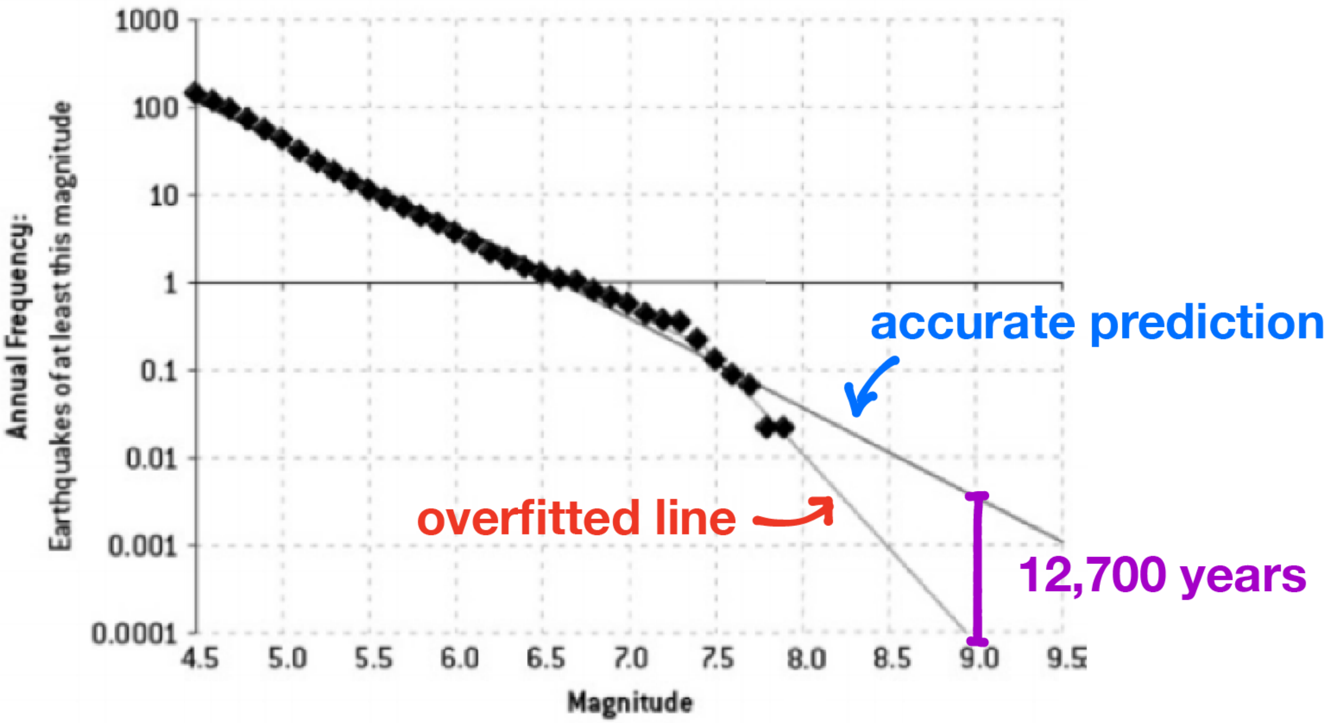
Bias-Variance Trade-Off



https://ml.berkeley.edu/blog/2017/07/13/tutorial-4/
https://mpra.ub.uni-muenchen.de/69383/1/MPRA_paper_69383.pdf

Bias-Variance Trade-Off

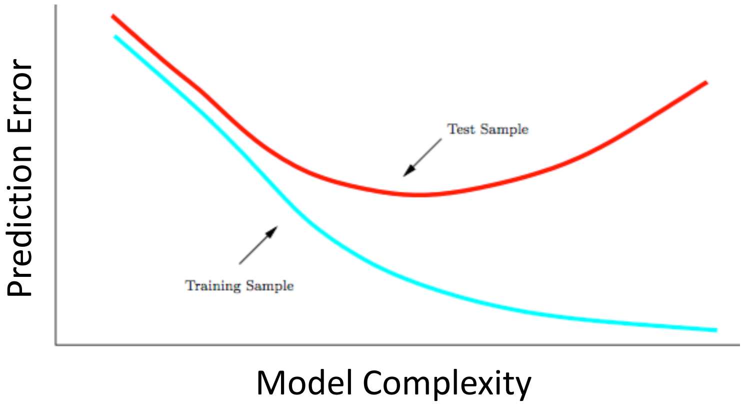
FIGURE 5-7C: TŌHOKU, JAPAN EARTHQUAKE FREQUENCIES
CHARACTERISTIC FIT



https://ml.berkeley.edu/blog/2017/07/13/tutorial-4/
https://mpra.ub.uni-muenchen.de/69383/1/MPRA_paper_69383.pdf

Errors

- Test error and Training error:
  - The **test error** is the average error that results from predicting the response on a new observation
  - The **training error** can be calculated by applying the machine learning method to the observations used in its training.



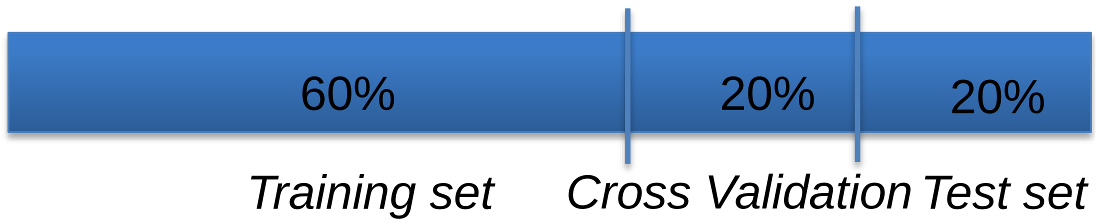
- But the training error rate often is quite different from the test error rate, and in particular the former can **dramatically underestimate** the latter.

Cross Validation

- **Splitting datasets into training and test is NOT enough!**
  - When different models can be fit on different datasets, the problem persists:
    1. Fit a model to the training set for each polynomial degree (d).
    2. Find the polynomial degree (d) with the least error using the test set.
    3. Estimate the generalization error also using the test set
- In this case, we minimized our MSE and selected a model which best behaves on the test set:
  - **The MSE will be greater for any other dataset.**

Cross Validation (cont.)

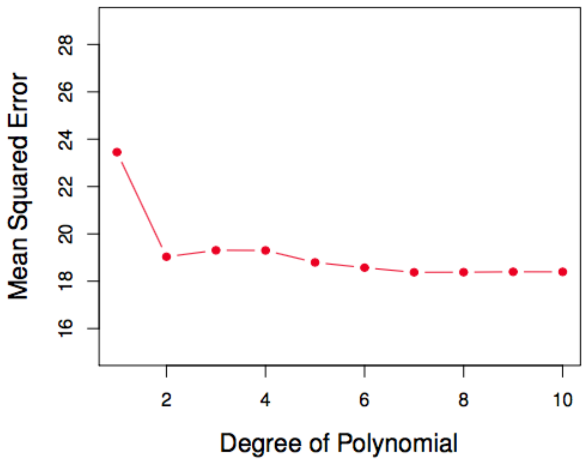
- Cross validation is considered a re-sampling method, to **avoid optimistic error estimates**. Typical distribution is:



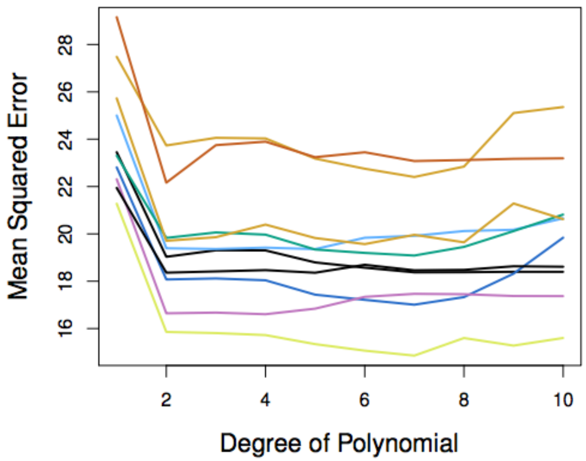
- **Now, the approach is:**
  1. Fit a model to the training set for each polynomial degree (d).
  2. Find the polynomial degree (d) with the least error using the Cross Validation set.
  3. Estimate the generalization error using the test set

Single vs. Multiple splits

Validation set error may tend to overestimate the test error for the model fit on the entire data set.



Single Split



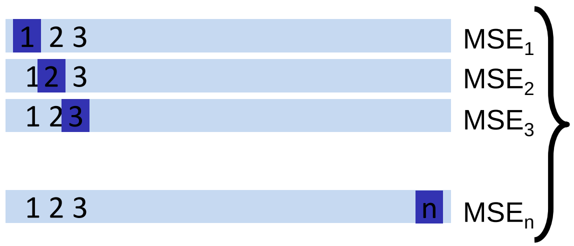
Multiple Splits

Taken from Introduction to Statistical Learning

Cross Validation resampling

- **Variants of the static 60-20-20**
  - **Leave-One-Out Cross Validation (LOOCV)**

LOOCV ***doesn't shake up the data enough***. *The estimates from each fold are highly correlated and hence their average can have high variance.*

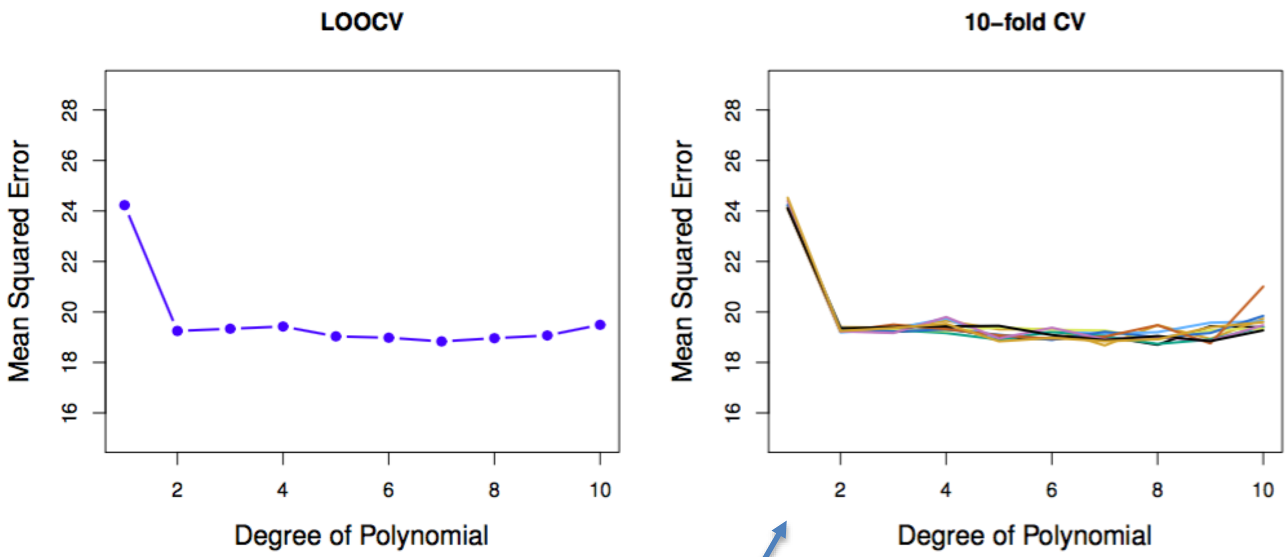


$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} MSE_i$$

  - ***K-fold*** **cross validation:** randomly divide the data into *K* equal-sized parts. Leave out part *k*, fit the model to the other *K – 1* parts (combined), and predict (measure MSE) for the left-out k^{th} part. Do this for each part k=1,…,K and average results.

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MSE_i$$

LOOCV & K-Fold



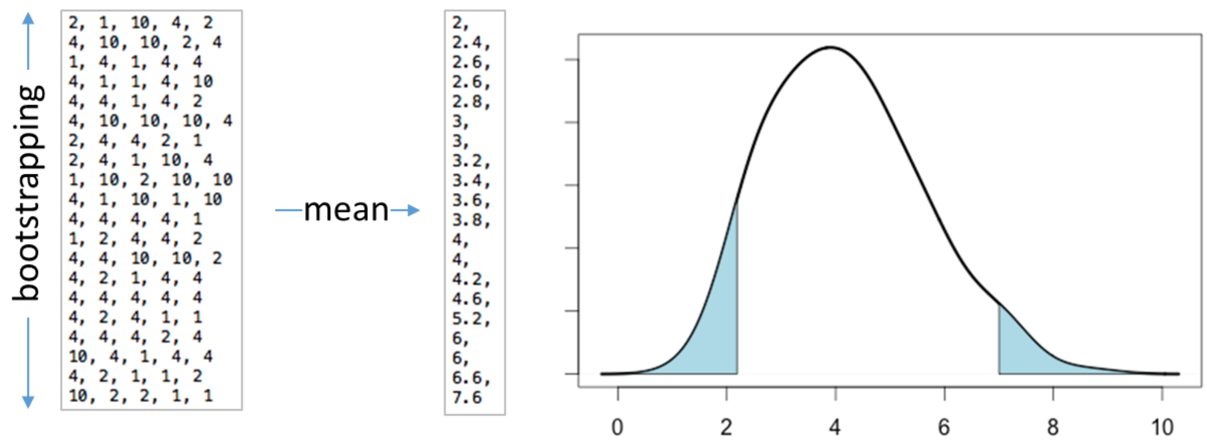
Each 10-fold is run nine times, each with a different random split of the data into 10 parts.

Bootstrap

- Bootstrapping is a powerful technique to **estimate a population parameter, using brute force**. Applied when
  - **Sample size is very small** (usually < 40-50)
  - **Estimate of a summary indicator within a confidence interval** (the mean, with a 90% confidence interval)
- **Some examples**
  - Measure the **mean** weight of a product from an entire product line, sampling just a few of them.
  - Maximize on portfolio investment (minimize **variance**), based on just a few samples of existing investments

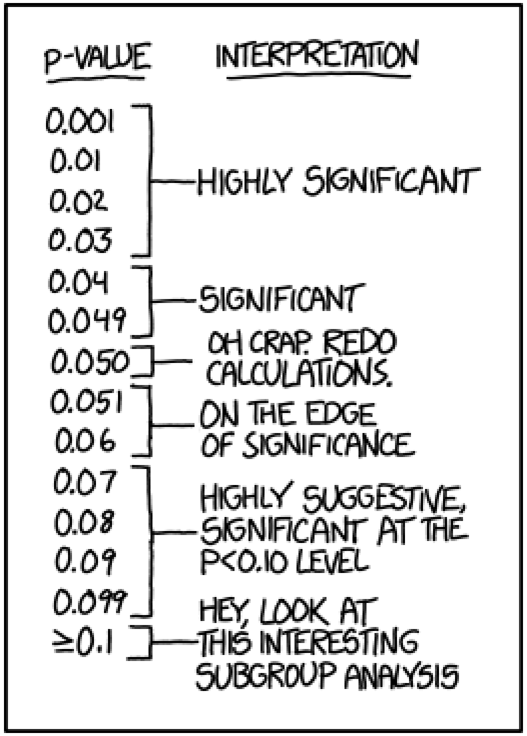
Bootstrapping

1. We begin with a sample from a population we know nothing about.
2. We want a 90% confidence interval about the mean of the value sampled.
3. The values we got in the sample are: 1, 2, 4, 4, 10
4. We generate different samples by taking sets of 5 elements with repetition, from our tiny sample:
5. We compute the mean of each sample
6. Since we want a 90% confident interval, we select the 5% and 95% percentiles of the distribution of means, as the endpoints of the interval.
7. **The 90% confidence is given by the interval [2.4, 6.6]**



Significance Testing

- **p-value**
  - Significance also goes under the name of p-value
  - We can accept our model's, if it's very unlikely that a naive model (a ***null hypothesis***) could score as well as our model (reject **null hypothesis**).
  - The traditional statistical method of computing significance or p-values is through a Student's t-test or an f-test.



Significance Testing