

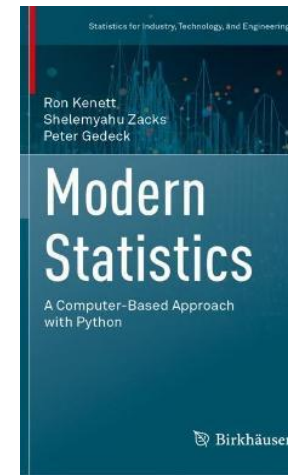
A Biomed Data Analyst Training Program

Model performance

Professor Ron S. Kenett

Chapter 7

Modern Analytic Methods: Part I



7.4 Determining Model Performance

The performance of a model can be measured in various ways. The Python package `scikit-learn` contains a wide variety of different metrics. A few of them are listed in Table 7.1.

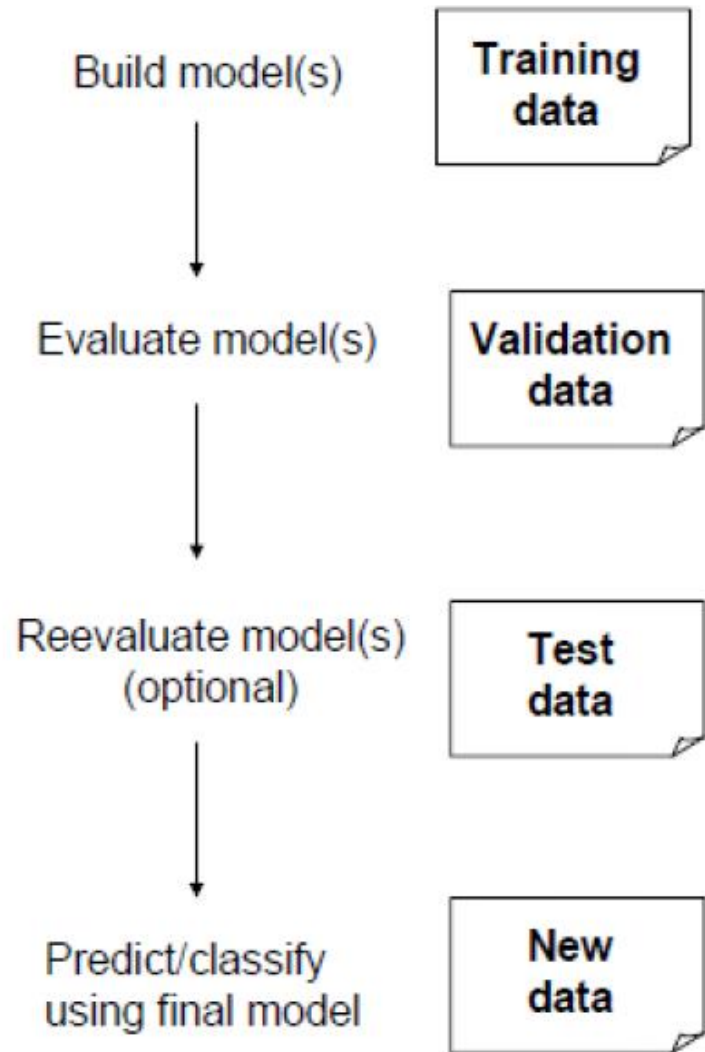
In order to avoid overfitting, one needs to compare results derived from fitting the model with a training set to results with a validation set not involved in fitting the model. There are basically two approaches to achieve this.

A first approach is applicable with large data sets. In this context one can randomly select a subset, through uniform or stratified sampling. This results in

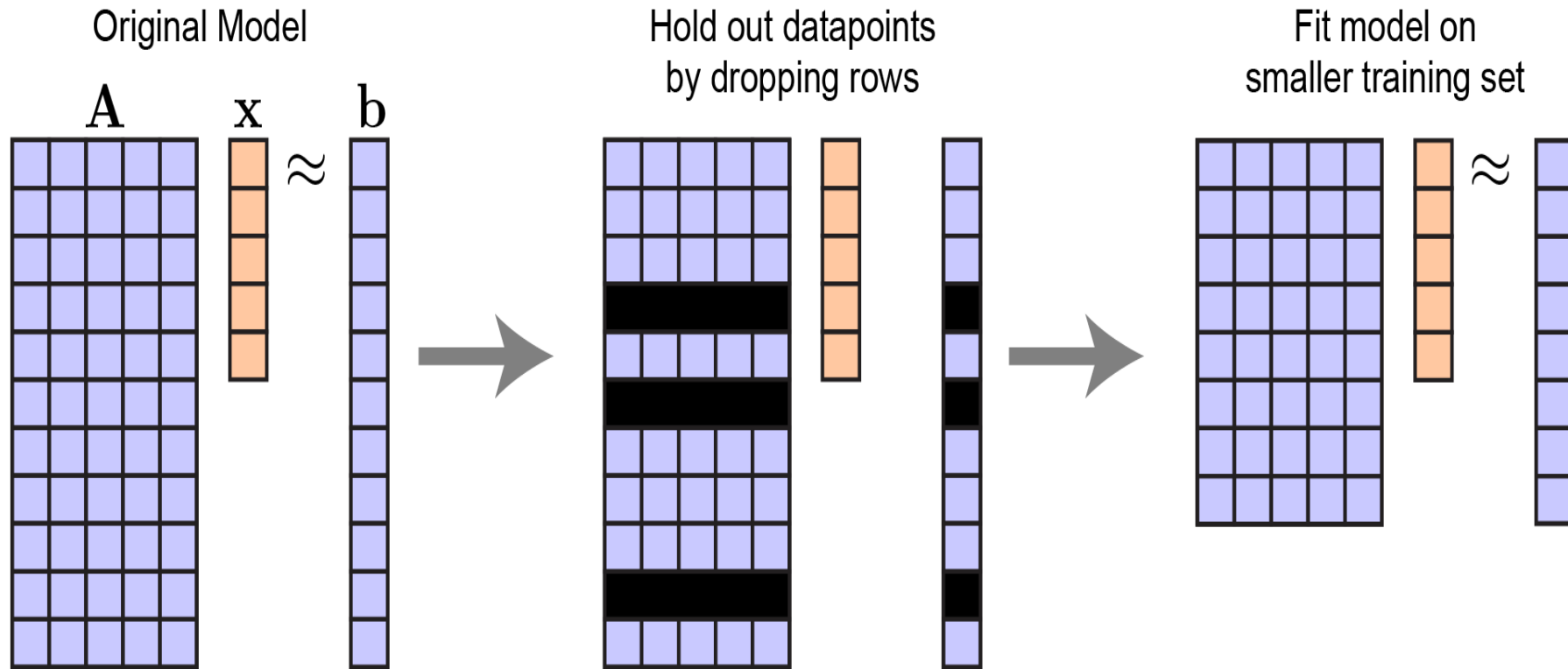
Table 7.1 Model performance metrics

<i>Classification</i>	
Accuracy	Accuracy is defined as the number of correct predictions made by the model on a data set
Balanced accuracy	Modification of accuracy suitable for imbalanced data sets
ROC	Numerical measures of model performance
<i>Regression</i>	
R^2	Coefficient of determination (see Sect. 4.3.2.1)
R^2_{adj}	Adjusted coefficient of determination (see Sect. 4.3.2.1)
MSE	Mean squared error is defined as the mean squared difference between actual and predicted y
MAE	Mean absolute error is defined as the mean absolute difference between actual and predicted y
AIC	Akaike information criterion
BIC	Bayesian information criterion

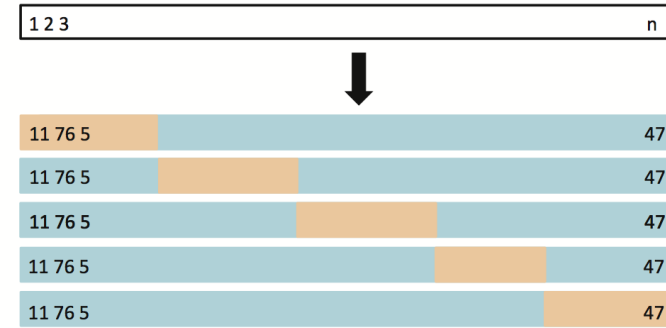
Data Splitting



Holdout Sets



K-Fold Cross-Validation



Randomly divide the data set of into K folds (typically $K = 5$ or 10).

The first fold is treated as a validation set, and the method is fit on the remaining $K - 1$ folds. The MSE is computed on the observations in the *held-out* fold. The process is repeated K times, taking out a different part each time.

By averaging the K estimates of the test error, we get an estimated validation (test) error rate for new observations.

K-Fold Cross-Validation

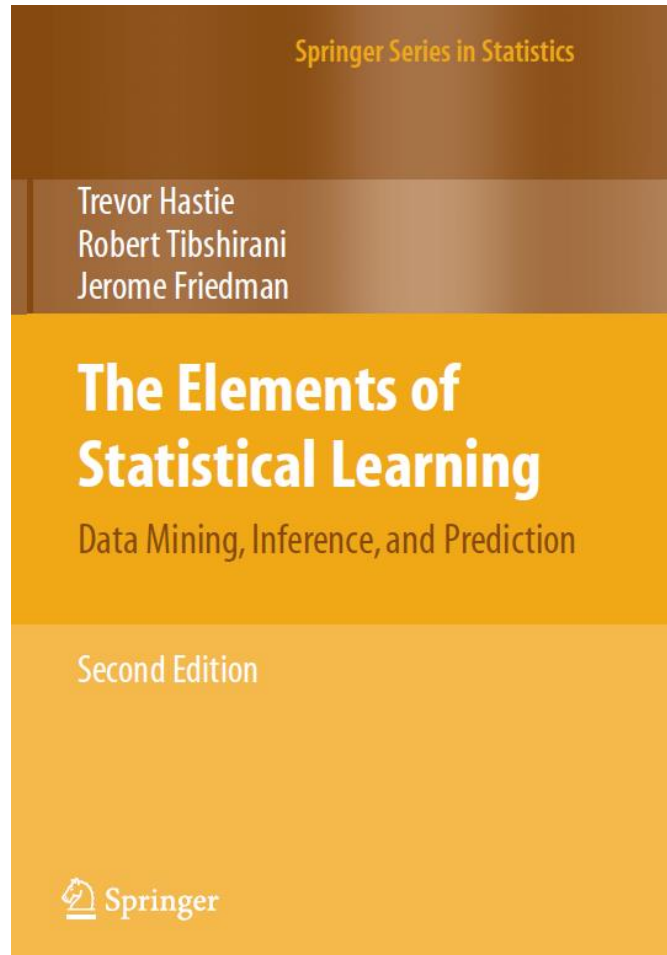
- Let the K folds be C_1, \dots, C_K , where C_k denotes the indices of the observations in fold k . There are n_k observations in fold k : if N is a multiple of K , then $n_k = n / K$.

- Compute: $CV_{(K)} = \sum_{k=1}^K \frac{n_k}{n} \text{MSE}_k$

where $\text{MSE}_k = \frac{1}{n_k} \sum_{i \in C_k} (Y_i - \hat{Y}_i)^2$ and \hat{Y}_i is the fitted value for observation i , obtained from the data with fold k removed.

1	2	3	4	5
Train	Train	Validation	Train	Train

Cross Validation





What value should we choose for K ? With $K = N$, the cross-validation estimator is approximately unbiased for the true (expected) prediction error but can have high variance because the N “training sets” are so similar to one another. The computational burden is also considerable, requiring N applications of the learning method. In certain special problems, this computation can be done quickly. On the other hand, with $K = 5$ say, cross-validation has lower variance. But bias could be a problem. The performance of the classifier improves as the training set size increases to 100 observations; increasing the number further to 200 brings only a small benefit. If our training set had 200 observations, 5 fold cross-validation would estimate the performance of our classifier over training sets of size 160, which is virtually the same as the performance for training set size 200. Thus cross-validation would not suffer from much bias. However, if the training set had 50 observations, 5 fold cross-validation would estimate the performance of our classifier over training sets of size 40, and it would be an underestimate of $1 - \text{Err}$. Hence as an estimate of Err , cross-validation would be biased upward.

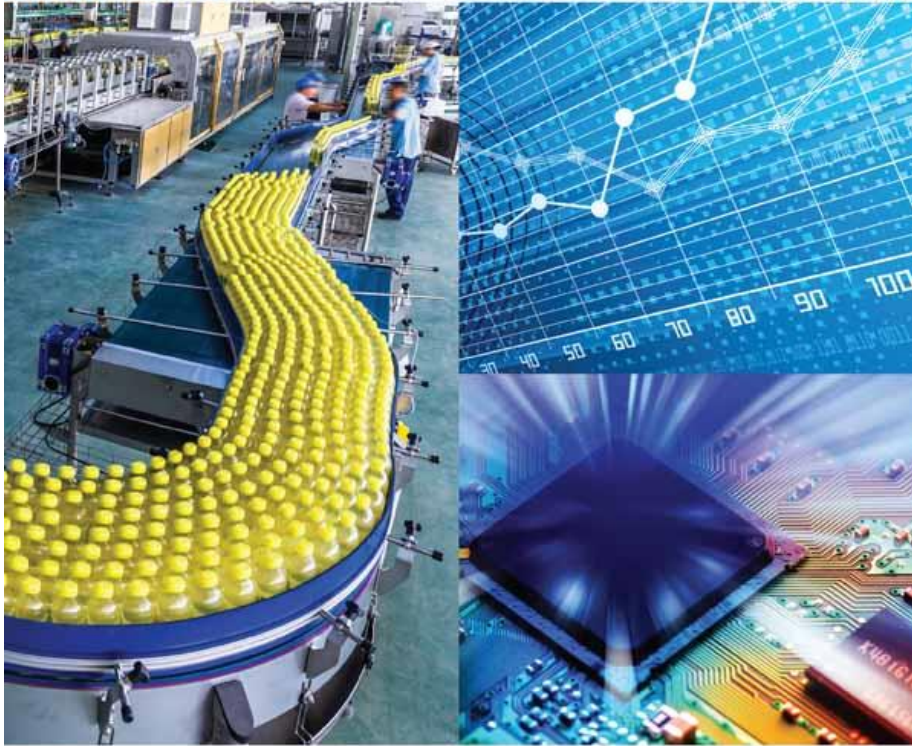
Self-supervised cross validation using data generation structure

Ron S. Kenett✉, Chris Gotwalt, Laura Freeman, Xinwei Deng✉

First published: 27 June 2022 | <https://doi.org/10.1002/asmb.2701>

[Read the full text >](#)

 PDF  TOOLS  SHARE



EDITOR-IN-CHIEF
Fabrizio Ruggeri
CNR-IMATI, Milan, Italy

EDITORS
Rituparna Sen
Indian Statistical Institute, India
Emmanuel Yashchin
IBM, USA

Special Issue: Data Science in Process Industries
Guest Editors: Marco S. Reis and Nikolaus Haselgruber

WILEY

<https://onlinelibrary.wiley.com/doi/abs/10.1002/asmb.2701>

Abstract

Modern statistics and machine learning typically involve large amounts of data coupled with computationally intensive methods. In a predictive modeling context, one seeks models that achieve high predictive accuracy on new datasets. This is typically implemented by partitioning the data into training and hold-out data sets. The allocation is often conducted randomly, at the row level of the data matrix. In this work, we discuss an overlooked gap in machine learning and predictive modeling, the role of data structure and data generation process in the partitioning of observational data into training and hold-out datasets. Ignoring such structures can lead to deficiencies in model generalizability and operationalization. We highlight that explicitly embracing the data generation structure to partition the data for validating predictive model is essential to the success of data science projects. The proposed approach is called **befitting cross validation (BCV)**. It relies on an information quality perspective of analytics. This requires an assessment with inputs from domain experts, in contrast to automated approaches that are purely data driven. BCV is motivated by the objective of generating information

Befitting Cross Validation (BCV) Principles

- **BCV Principle 1:** The formation of training and hold-out datasets should reflect the goal of the study
- **BCV Principle 2:** The training dataset and the hold-out dataset should have the same data generation structure as the whole dataset
- **BCV Principle 3:** The construction of the hold-out dataset should reflect the data generation structure needed for the predictive model



patient

visit

test



	<u>RESULT</u>	<u>UNITS</u>	<u>REFERENCE</u>	<u>VALUES</u>	<u>REMARKS</u>
EOSINOPHILS	0.27	K/UL	(0.00 - 0.60)	(...*...)
EOSINOPHILS %	2.90	%	(0.00 - 6.00)	(...*...)
BASOPHILS	0.06	K/UL	(0.00 - 0.15)	(...*...)
BASOPHILS %	0.60	%	(0.00 - 1.50)	(...*...)
RBC	5.10	M/UL	(4.30 - 6.00)	(...*...)
HEMOGLOBIN	14.70	G/DL	(13.50 - 17.50)	(...*...)
HEMATOCRIT	44.50	%	(38.00 - 50.00)	(...*...)
MCV	87.30	FL	(80.00 - 98.00)	(...*...)
MCH	28.80	PG	(27.00 - 33.00)	(...*...)
MCHC	33.00	G/DL	(32.00 - 35.50)	(...*...)
RDW	14.30	%	(11.00 - 16.00)	(...*...)
NRBC	0.00	K/UL	(0.00 - 0.01)	(*...)
NRBC%	0.00	%	(0.00 - 0.10)	(*...)
PLATELETS	268.00	K/UL	(150.00 - 400.00)	(...*...)
PLATELET-LARGE CELL RATIO	35.40	%			
N.R. 19.4-43.7%					
MPV	11.40	FL			
N.R. 6.5-11.5 FL					

Training



Date of visit

Patient ID

*	*	*							
			*	*					
					*				*
	*				*				
							*	*	*
[Blue shaded area]									

Validation



Date of visit

Patient ID

[Blue shaded area]									
	*		*						
		*	*						
					*	*			

BCV

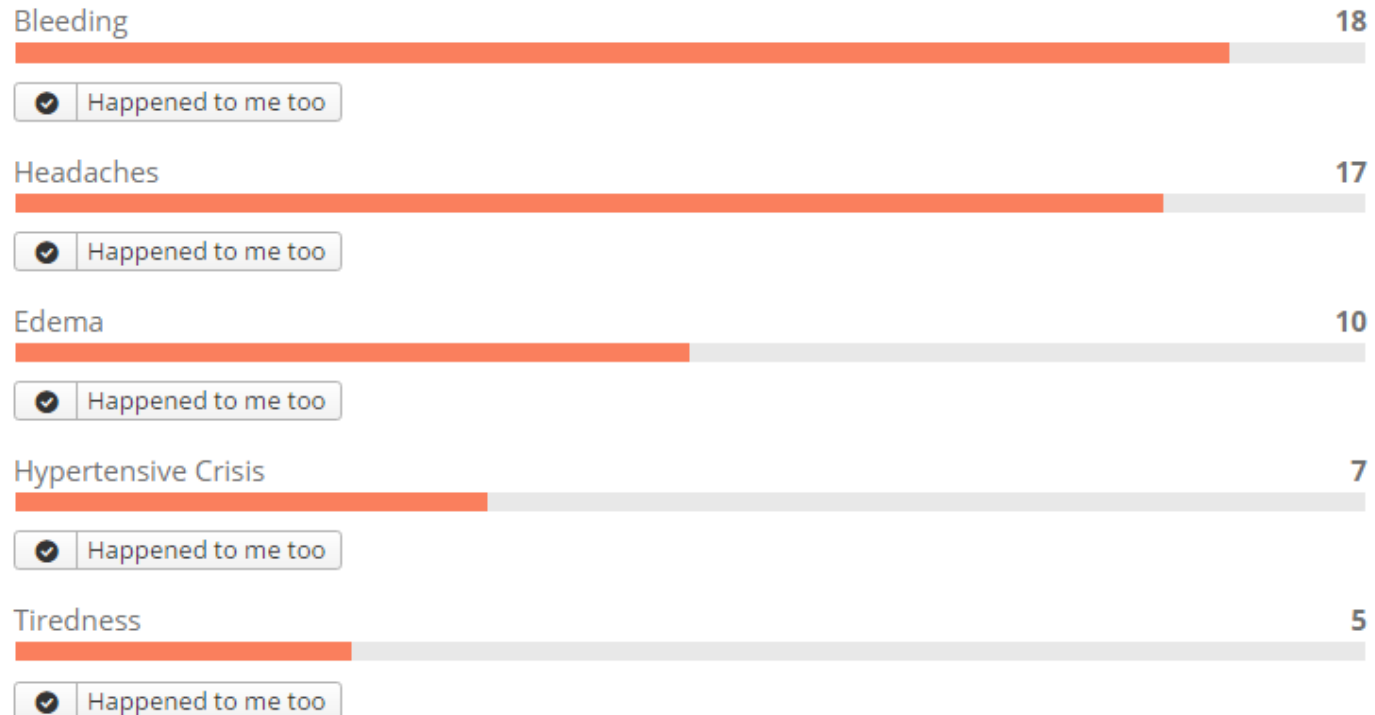
Association Rules

Overview Side Effects & Concerns Alternatives Drug Facts Reviews

Nicardipine side effects and concerns

116 different concerns were found, out of 310 posts about Nicardipine.

Click on the topics below to find out what patients said about each issue.



By report

Unique Subject Identifier	Reported Term for the Adverse Event
11001	Vasoconstriction
11001	Hypertension
11001	Intracranial pressure increased
11002	Pulmonary oedema
11002	Pyrexia
11002	Hepatic function abnormal
11002	Intracranial pressure increased
11002	Enanthema
11002	Hydrocephalus
11002	Subarachnoid haemorrhage
11002	Hyperglycaemia
11002	Isosthenuria
11002	Anaemia
11002	Pericardial effusion
11002	Cardiac failure congestive
11002	Hypertension
11002	Alveolitis
11002	Sepsis neonatal
11002	Vasoconstriction
11002	Atrial fibrillation
11002	Electrocardiogram T wave inversion
11002	Cerebral infarction
11002	Brain oedema
11002	Coagulopathy

By patient

Transaction Listing	
Transaction ID	Item Set
101001	Hydrocephalus, Pyrexia, Vasoconstriction, Vomiting
101002	Alveolitis, Hydrocephalus, Hyperglycaemia, Pulmonary oedema, Urinary tract infection, Vasoconstriction, Ventricular extrasystoles
101004	Brain oedema, Coma, Hydrocephalus, Hyperglycaemia, Hypotension, Intracranial pressure increased, Subarachnoid haemorrhage, ...
101005	Alveolitis, Anaemia, Heart rate increased, Hydrocephalus, Hyperglycaemia, Hypertension, Hypokalaemia, Infection, Intestinal perfo...
101006	Cardiac failure congestive, Pulmonary oedema, Vasoconstriction
101007	Alveolitis, Cerebral infarction, Hyperglycaemia, Hypotension, Intracranial pressure increased, Isosthenuria, Phlebitis, Pyrexia, Respir...
101008	Cerebral hypoperfusion, Ependymitis, Hydrocephalus, Pulmonary oedema, Supraventricular tachycardia, Vasoconstriction
101009	Vomiting
101010	Anaemia, Hepatic function abnormal, Hypertension, Infection, Vasoconstriction, Ventricular extrasystoles
101011	Brain oedema, Hydrocephalus, Hypotension, Infection, Urinary tract infection, Vasoconstriction
101012	Hypotension
101013	Apnoea, Atelectasis, Brain oedema, Cerebral haemorrhage, Convulsion, Hyponatraemia, Hypotension, Intracranial pressure increas...
101014	Hyponatraemia, Hypotension, Intracranial pressure increased, Respiratory disorder, Subarachnoid haemorrhage, Vasoconstriction, ...
101015	Cardiovascular disorder, Pyrexia, Vasoconstriction, Vomiting
101016	Phlebitis, Pulmonary oedema, Pyrexia
101017	Vasoconstriction
11001	Hypertension, Intracranial pressure increased, Vasoconstriction
11002	Alveolitis, Anaemia, Atrial fibrillation, Brain oedema, Cardiac failure congestive, Cerebral infarction, Coagulopathy, Electrocardiogr...
11003	Alveolitis, Anoxia, Atelectasis, Brain oedema, Convulsion, Hypertension, Hypokalaemia, Hypotension, Intracranial pressure increas...
11004	Alveolitis, Anaemia, Atelectasis, Brain oedema, Heart rate increased, Hydrocephalus, Hyperglycaemia, Hypokalaemia, Intracranial p...
11005	Alveolitis, Atrial fibrillation, Brain oedema, Coagulopathy, Convulsion, Hepatic function abnormal, Hepatitis, Hydrocephalus, Hyper...
11006	Hypertension, Pulmonary oedema, Urinary tract infection, Vasoconstriction
11007	Diabetes insipidus, Enanthema, Hepatic function abnormal, Hyperglycaemia, Isosthenuria, Leukocytosis, Phlebitis, Polyuria, Pulmo...
11008	Alveolitis, Anaemia, Atelectasis, Atrial fibrillation, Blood lactate dehydrogenase increased, Cardiac failure congestive, Gastrointesti...
11009	Atelectasis, Brain oedema, Cerebral infarction, Hyperglycaemia, Infection, Intracranial pressure increased, Isosthenuria, Pyrexia, Uri...
11010	Brain oedema, Hydrocephalus, Intracranial pressure increased, Vasoconstriction
11011	Blood lactate dehydrogenase increased, Hypokalaemia, Hypotension, Lethargy, Overdose, Urinary tract infection, Vasoconstriction
11012	Blindness, Hyperglycaemia, Hypotension, Sinus headache, Urinary tract infection, Vasodilatation
11013	Brain oedema, Heart rate increased, Hydrocephalus, Hypertension, Hypokalaemia, Pulmonary oedema, Pyrexia, Sinus bradycardia, ...
11014	Brain oedema, Cardiac arrest, Cerebral ischaemia, Coma, Hyperglycaemia, Hypertension, Hypotension, Intracranial pressure increa...

Holdout Validation Options in JMP

Validation Column utility available on *Predictive Modeling* Menu

Stratified Random is a first choice for most data.

Grouped Random will keep rows within a group in the same portion of the data.

Cutpoint is for time sequenced data.

Make Validation Column

A validation column divides the rows of the data table into a training set to estimate the model; a validation set to help choose a model that predicts well; and sometimes a test set to check prediction after the model is chosen.

Specify how to allocate rows to Training, Validation and Test sets.
Enter either rates or counts.

Total Rows	442
Training Set	<input type="text" value="0.667"/>
Validation Set	<input type="text" value="0.333"/>
Test Set	<input type="text" value="0"/>

New Column Name

Choose a method to create the holdback sets:

- Formula Random Formula column with a random function.
- Fixed Random Column with no formula. Seed: .
- Stratified Random Column with sets that are balanced across levels of selected columns.
- Grouped Random Column with each level of the grouping column assigned entirely to one set.
- Cutpoint Column with holdback sets based on time series cutpoints.

Evaluating Predictive Performance

Measuring Predictive Error

Not the same as “goodness-of-fit”

We want to know how well the model predicts **new data**, not how well it fits the data it was trained with

Key component of most measures is difference between actual y and predicted y (“error”)

Measures of Error

MAE, MAD, or AAE: Mean absolute error/deviation or average absolute error

Gives an idea of the magnitude of errors

Average error

Gives an idea of systematic over- or under-prediction

MAPE: Mean absolute percentage error

RMSE (root-mean-squared-error) or **RASE** (root average squared error):
Square the errors, find their average, take the square root

Total SSE: Total sum of squared errors

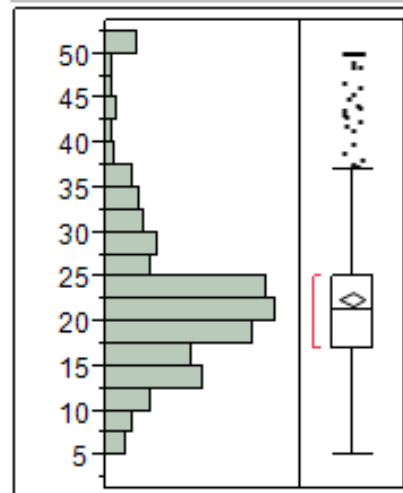
Validation

Method	N	RSquare	RASE
K Nearest Neighbors	126	0.8450	3.7430
Neural Boosted	126	0.8437	3.7582
Support Vector Machines	126	0.8057	4.1901
Boosted Tree	126	0.7698	4.5611
Bootstrap Forest	126	0.7350	4.8942
Fit Stepwise	126	0.6714	5.4496
Fit Least Squares	126	0.6696	5.4643
Generalized Regression Lasso	126	0.6696	5.4643
Decision Tree	126	0.6359	5.7362

Training

Method	N	RSquare	RASE
Boosted Tree	380	0.9760	1.4026
Bootstrap Forest	380	0.9400	2.2177
Support Vector Machines	380	0.8941	2.9473
Neural Boosted	380	0.8810	3.1237
Decision Tree	380	0.8548	3.4505
K Nearest Neighbors	380	0.8069	3.9799
Fit Least Squares	380	0.7555	4.4784
Generalized Regression Lasso	380	0.7555	4.4784
Fit Stepwise	380	0.7551	4.4817

mvalue



Quantiles

100.0%	maximum	50
99.5%		50
97.5%		50
90.0%		34.9
75.0%	quartile	25
50.0%	median	21.2
25.0%	quartile	16.95
10.0%		12.7
2.5%		8.235
0.5%		5.321
0.0%	minimum	5

Summary Statistics

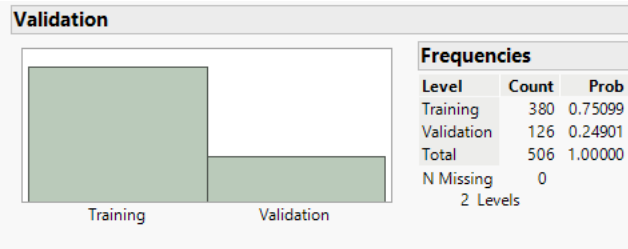
Mean	22.532806
Std Dev	9.1971041
Std Err Mean	0.4088611
Upper 95% Mean	23.336085
Lower 95% Mean	21.729528
N	506



Candidates

Term	Candidate SS	LogWorth
crim	8266.17273	32.6638216
zn	6669.06251	24.9773486
indus	11083.22547	48.7519537
chas	1312.07927	4.1110954
nox	9536.22405	39.5670978
rooms	19339.55503 *	118.7473483
age	5573.64765	19.6751451
distance	4994.54054	17.1453361
radial	6708.64333	24.6205659
tax	8618.08428	34.5266980
pt	10438.69478	44.8775094
b	5259.31980	18.2910466
lstat	18896.19401	113.7427626

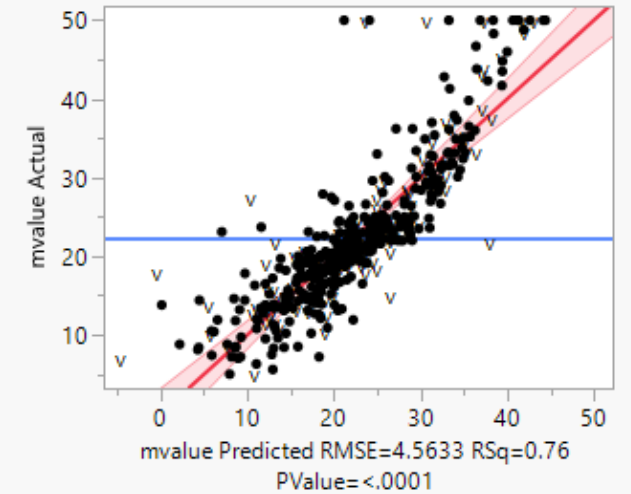
Least Squares Regression



Crossvalidation

Source	RSquare	RASE	Freq
Training Set	0.7555	4.4784	380
Validation Set	0.6696	5.4643	126

Actual by Predicted Plot



Validation

Method	N	RSquare	RASE
Neural Boosted	126	0.8704	3.4229
K Nearest Neighbors	126	0.8450	3.7430
Support Vector Machines	126	0.8057	4.1901
Bootstrap Forest	126	0.7742	4.5169
Boosted Tree	126	0.7698	4.5611
Fit Stepwise	126	0.6714	5.4496
Fit Least Squares	126	0.6696	5.4643
Generalized Regression Lasso	126	0.6696	5.4643
Decision Tree	126	0.6359	5.7362

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob> t
Intercept	26.179042	5.776661	4.53	<.0001*
crim	-0.055294	0.050779	-1.09	0.2769
zn	0.0322328	0.016177	1.99	0.0471*
indus	0.0412445	0.065292	0.63	0.5280
chas[0]	-0.883567	0.452631	-1.95	0.0517
nox	-16.45433	4.087227	-4.03	<.0001*
rooms	4.82453	0.487871	9.89	<.0001*
age	-0.00545	0.014688	-0.37	0.7108
distance	-1.235934	0.21679	-5.70	<.0001*
radial	0.2203161	0.072558	3.04	0.0026*
tax	-0.010085	0.003957	-2.55	0.0112*
pt	-0.903422	0.144214	-6.26	<.0001*
b	0.0138322	0.003077	4.49	<.0001*
lstat	-0.467271	0.057654	-8.10	<.0001*

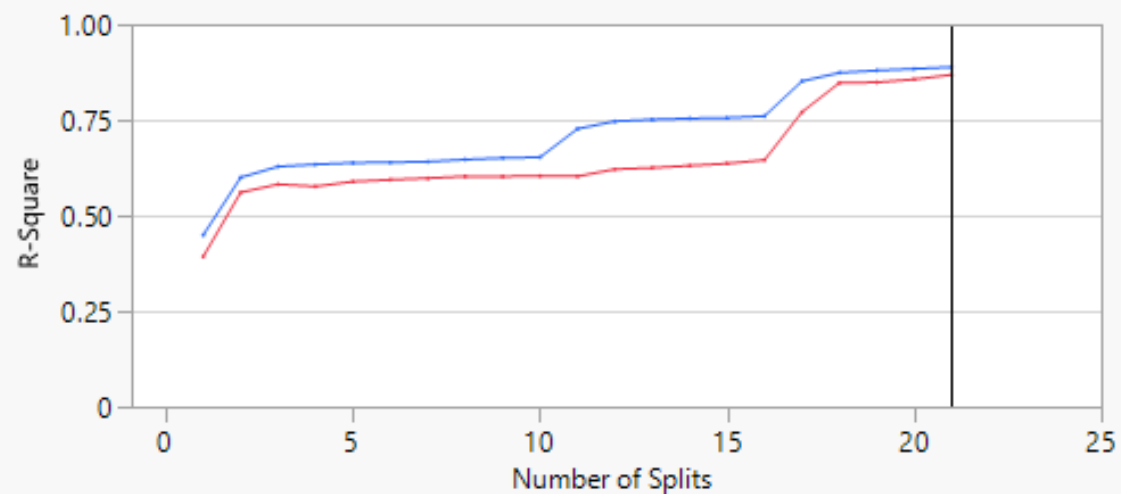
Summary Across the Folds

Method	N Trials Folds	Sum Freq	Validation Set Folds		
			RSquare	Mean RASE	StdDev RASE
Neural Boosted	5	101.20	0.8954	2.9218	0.26628
Boosted Tree	5	101.20	0.8860	3.0666	0.36667
Bootstrap Forest	5	101.20	0.8650	3.3349	0.56637
K Nearest Neighbors	5	101.20	0.8515	3.5052	0.54104
Support Vector Machines	5	101.20	0.8355	3.6512	0.57647
Decision Tree	5	101.20	0.7802	4.2195	0.69817
Fit Stepwise	5	101.20	0.7203	4.8105	0.46968
Generalized Regression Lasso	5	101.20	0.7197	4.8150	0.45423
Fit Least Squares	5	101.20	0.7174	4.8339	0.46423

Partition for mvalue

	RSquare	RASE	N	Number of Splits	AICc
Training	0.889	3.1087289	405	21	2116.95
Validation	0.868	3.119241	101		

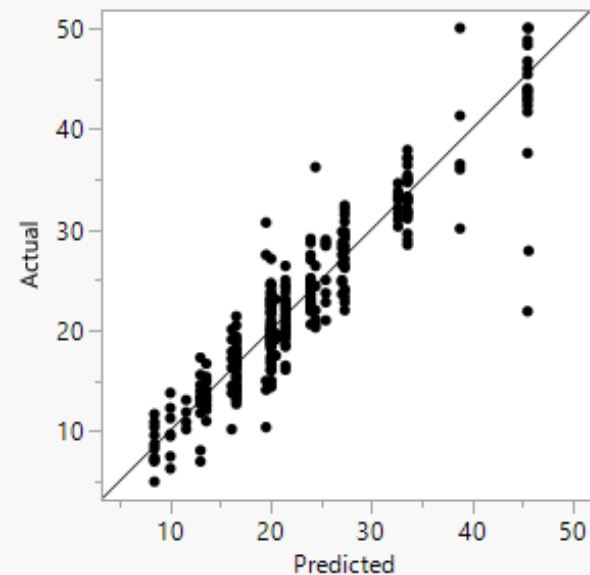
Split History



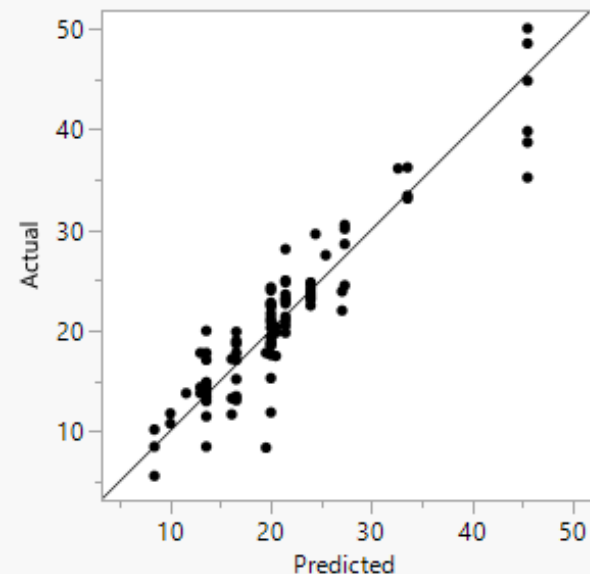
Validation Data in Red

Actual by Predicted Plot

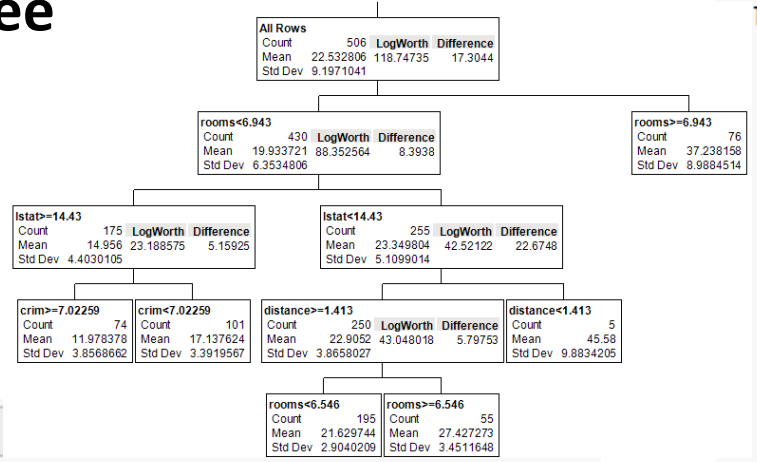
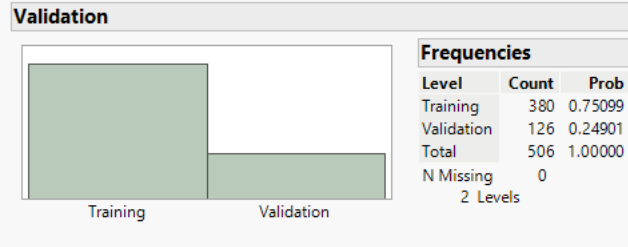
Training Set



Validation Set



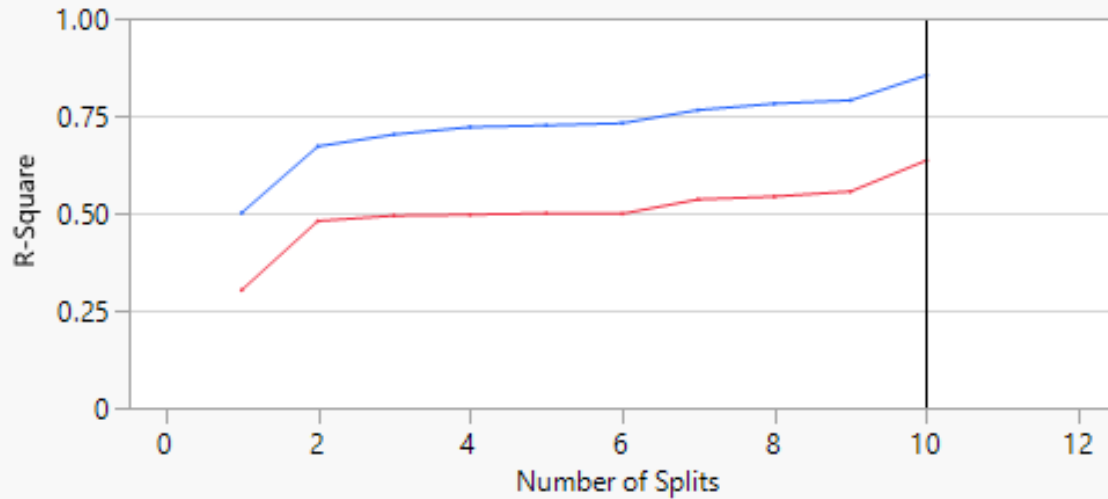
Regression (Decision) Tree



Partition for mvalue

	RSquare	RASE	N	Number of Splits	AICc
Training	0.855	3.4504892	380	10	2044.52
Validation	0.636	5.7362427	126		

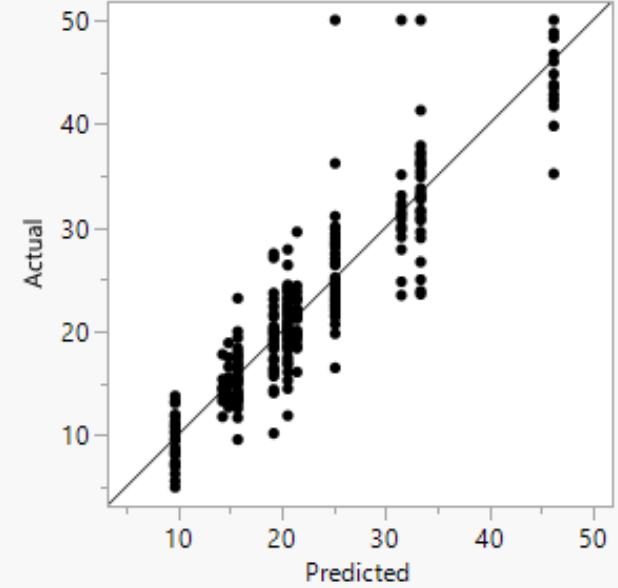
Split History



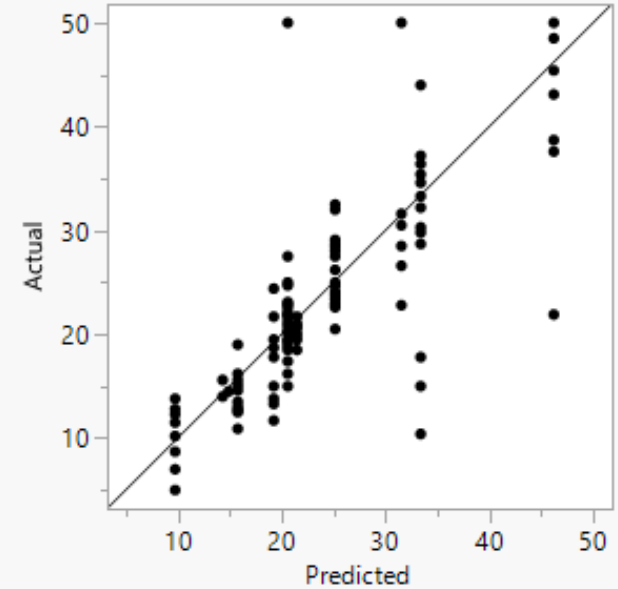
Validation Data in Red

Actual by Predicted Plot

Training Set



Validation Set

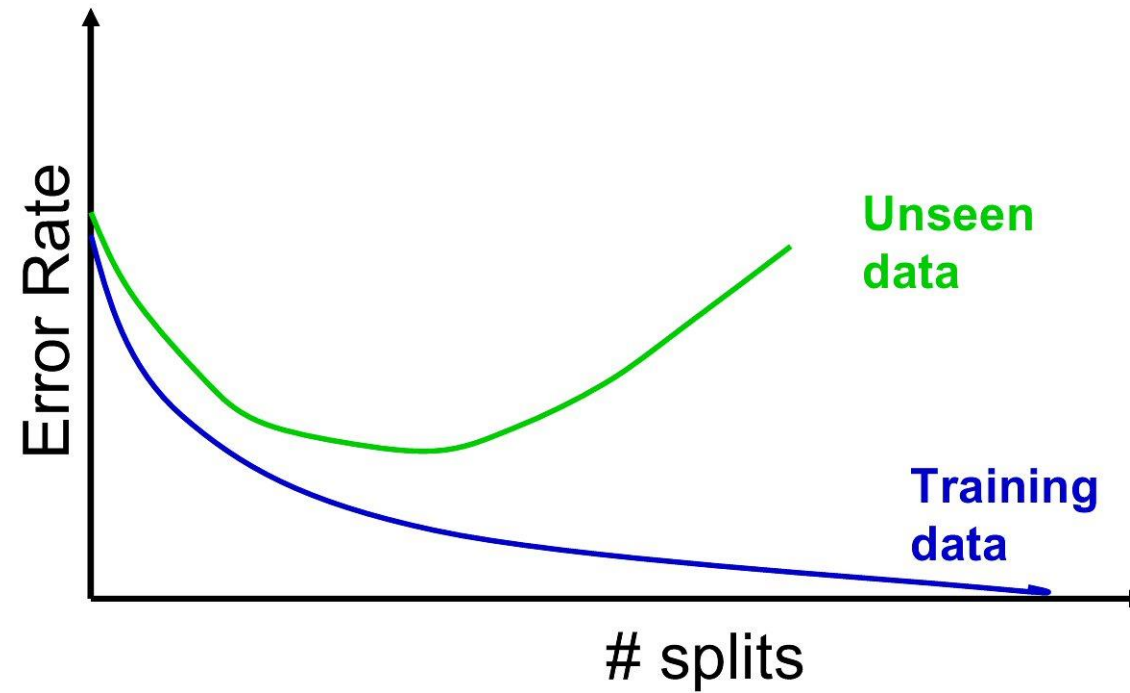


Prediction Error

Crossvalidation

Source	RSquare	RASE	Freq
Training Set	0.7621	11.109	18
Validation Set	0.3381	16.771	22

Decision Tree Error Rate



Accuracy Measures (Classification)

Decision Tree



Fit Details

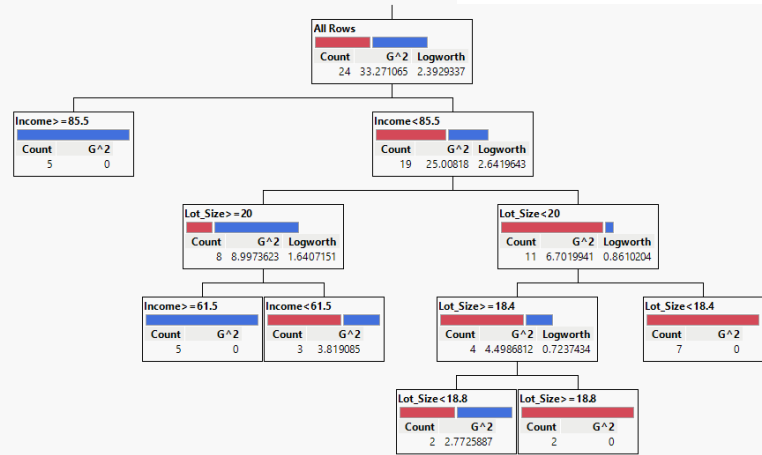
Measure	Training	Definition
Entropy RSquare	0.7060	$1 - \text{Loglike}(\text{model}) / \text{Loglike}(0)$
Generalized RSquare	0.8323	$(1 - (L(0) / L(\text{model}))^{(2/n)}) / (1 - L(0)^{(2/n)})$
Mean -Log p	0.2038	$\sum -\text{Log}(p[j]) / n$
RASE	0.2333	$\sqrt{\sum (y[j] - p[j])^2 / n}$
Mean Abs Dev	0.1620	$\sum y[j] - p[j] / n$
Misclassification Rate	0.0833	$\sum (p[j] \neq p\text{Max}) / n$
N	24	n

Confusion Matrix

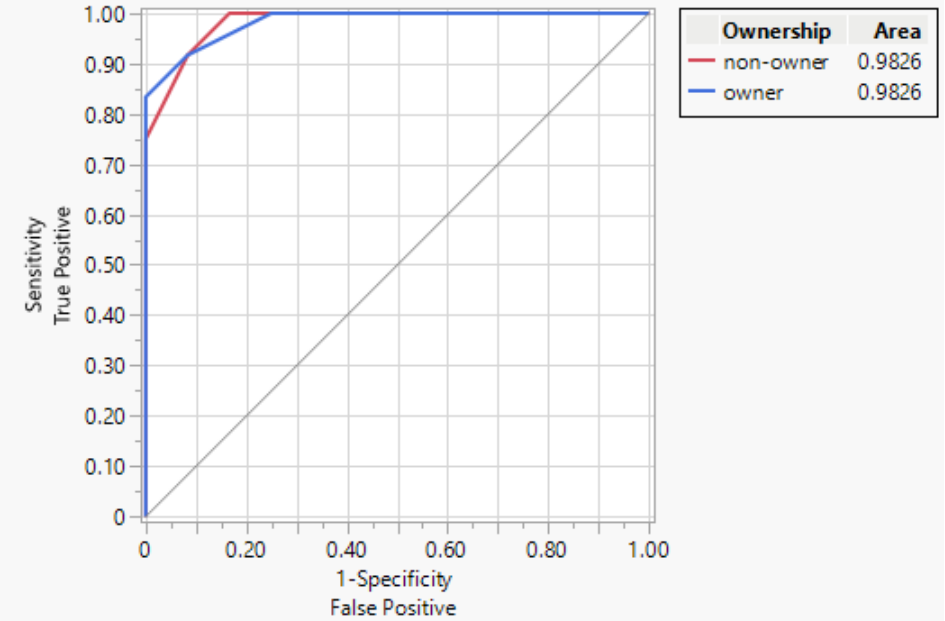
Training

Actual Ownership	Predicted Count	
	non-owner	owner
non-owner	12	0
owner	2	10

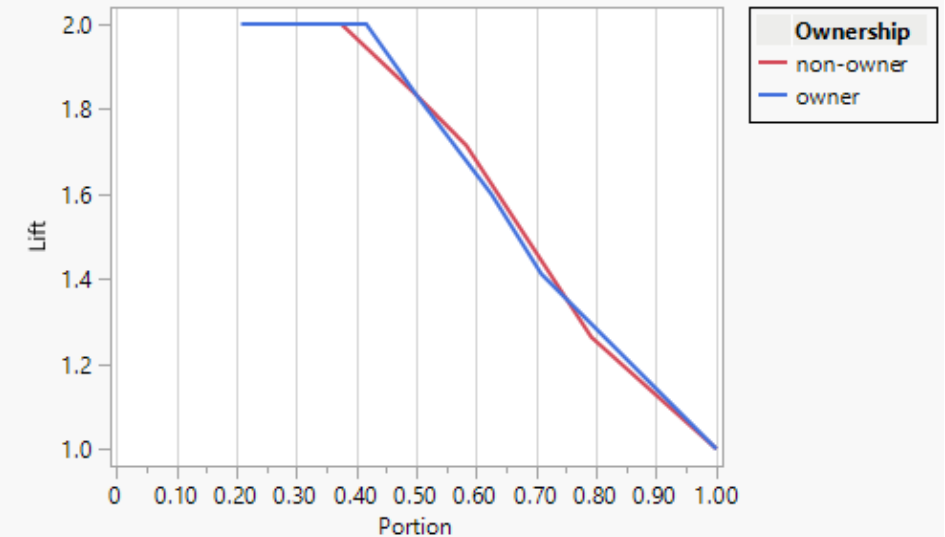
Actual Ownership	Predicted Rate	
	non-owner	owner
non-owner	1.000	0.000
owner	0.167	0.833



Receiver Operating Characteristic on Training Data



Lift Curve on Training Data



Misclassification error

Error = classifying a record as belonging to one class when it belongs to another class.

Error rate = percent of misclassified records out of the total records in the validation data.

Naïve Rule

Naïve rule: classify all records as belonging to the most prevalent class

- Often used as benchmark: we hope to do better than that
- Exception: when goal is to identify high-value but rare outcomes, we may do well by doing worse than the naïve rule (see “lift” – later)

Separation of Records

“High separation of records” means that using predictor variables attains low error

“Low separation of records” means that using predictor variables does not improve much on naïve rule

Confusion Matrix

		Predicted Class		
		Count	0	1
Actual Class	0	2689	25	2714
	1	85	201	286
	Total	2774	226	3000

201 1's correctly classified as "1"

85 1's incorrectly classified as "0"

25 0's incorrectly classified as "1"

2689 0's correctly classified as "0"

Error Rate

		Predicted Class		
		Count	0	1
Actual Class	0	2689	25	2714
	1	85	201	286
	Total	2774	226	3000

Overall error rate = $(25+85)/3000 = 3.67\%$

Accuracy = $1 - \text{err} = (201+2689)/3000 = 96.33\%$

If multiple classes, error rate is:

(sum of misclassified records)/(total records)

Cutoff for classification

Most data mining algorithms classify via a 2-step process:

For each record,

1. Compute **probability of belonging to class “1”**
 2. Compare to cutoff value, and classify accordingly
- Default cutoff value is 0.50
 - If ≥ 0.50 , classify as “1”
 - If < 0.50 , classify as “0”
 - Can use different cutoff values
 - Typically, error rate is lowest for cutoff = 0.50

Cutoff Table (Riding Mowers)

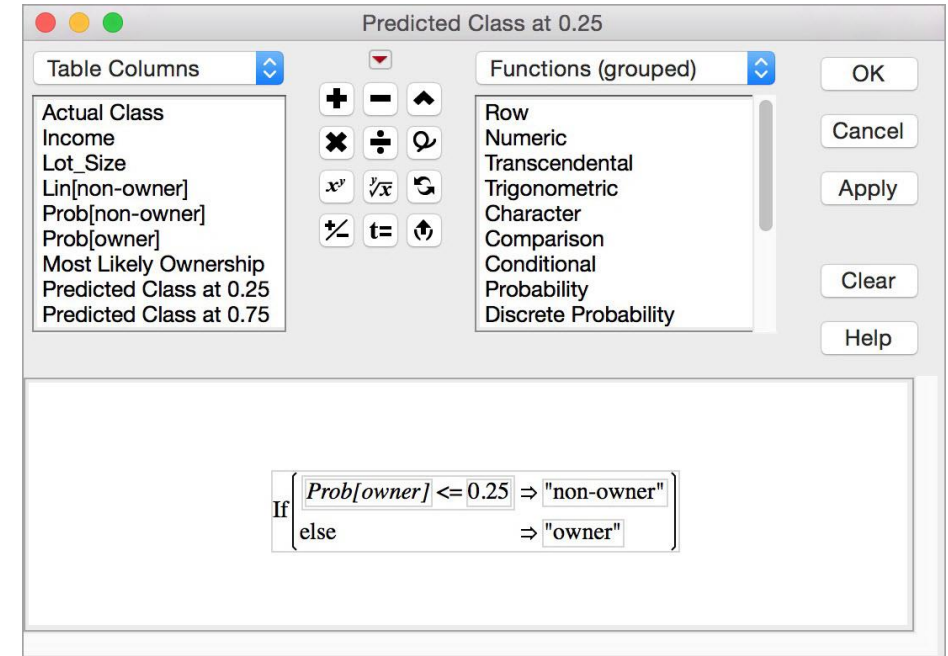
If cutoff is 0.50: 12 records are classified as “owner”

If cutoff is 0.75: 8 records are classified as “owner”

	Actual Class	Prob[owner]	Most Likely Ownership	Predicted Class at 0.25	Predicted Class at 0.75
1	owner	0.102697297	non-owner	non-owner	non-owner
2	owner	0.561213154	owner	owner	non-owner
3	owner	0.941747935	owner	owner	owner
4	owner	0.7323180727	owner	owner	non-owner
5	owner	0.9998892877	owner	owner	owner
6	owner	0.9803565717	owner	owner	owner
7	owner	0.9448657753	owner	owner	owner
8	owner	0.9978033093	owner	owner	owner
9	owner	0.6683372877	owner	owner	non-owner
10	owner	0.9859447076	owner	owner	owner
11	owner	0.8794488943	owner	owner	owner
12	owner	0.8830365999	owner	owner	owner
13	non-owner	0.7088413043	owner	owner	non-owner
14	non-owner	0.4863742324	non-owner	owner	non-owner
15	non-owner	0.0917943278	non-owner	non-owner	non-owner
16	non-owner	0.1213584078	non-owner	non-owner	non-owner
17	non-owner	0.5788381	owner	owner	non-owner
18	non-owner	0.0133127658	non-owner	non-owner	non-owner
19	non-owner	0.0331493122	non-owner	non-owner	non-owner
20	non-owner	0.2010408317	non-owner	non-owner	non-owner
21	non-owner	0.0057710408	non-owner	non-owner	non-owner
22	non-owner	0.0033442195	non-owner	non-owner	non-owner
23	non-owner	0.0138203993	non-owner	non-owner	non-owner
24	non-owner	0.0646960932	non-owner	non-owner	non-owner

Confusion Matrix for Different Cutoffs

- The cutoff can be changed using a formula in a new column.
- Different cutoffs can be compared to see which provides the lowest misclassification error on the validation set



▼ **Tabulate**

	Most Likely Ownership		Predicted Class at 0.25		Predicted Class at 0.75							
	non-owner	owner	non-owner	owner	non-owner	owner						
Actual Class	Row %	N	Row %	N	Row %	N	Row %	N	Row %	N	Row %	N
non-owner	83.33%	10	16.67%	2	75.00%	9	25.00%	3	100.00%	12	0.00%	0
owner	8.33%	1	91.67%	11	8.33%	1	91.67%	11	33.33%	4	66.67%	8

ROC and Lift

When One Class is More Important

In many cases it is more important to identify members of one class

- Tax fraud
- Credit default
- Response to promotional offer
- Detecting electronic network intrusion
- Predicting delayed flights

In such cases, we are willing to tolerate greater overall error, in return for better identifying the important class for further attention

Alternate Accuracy Measures

If “ C_1 ” is the important class:

Sensitivity = % (or proportion) of “ C_1 ” class correctly classified

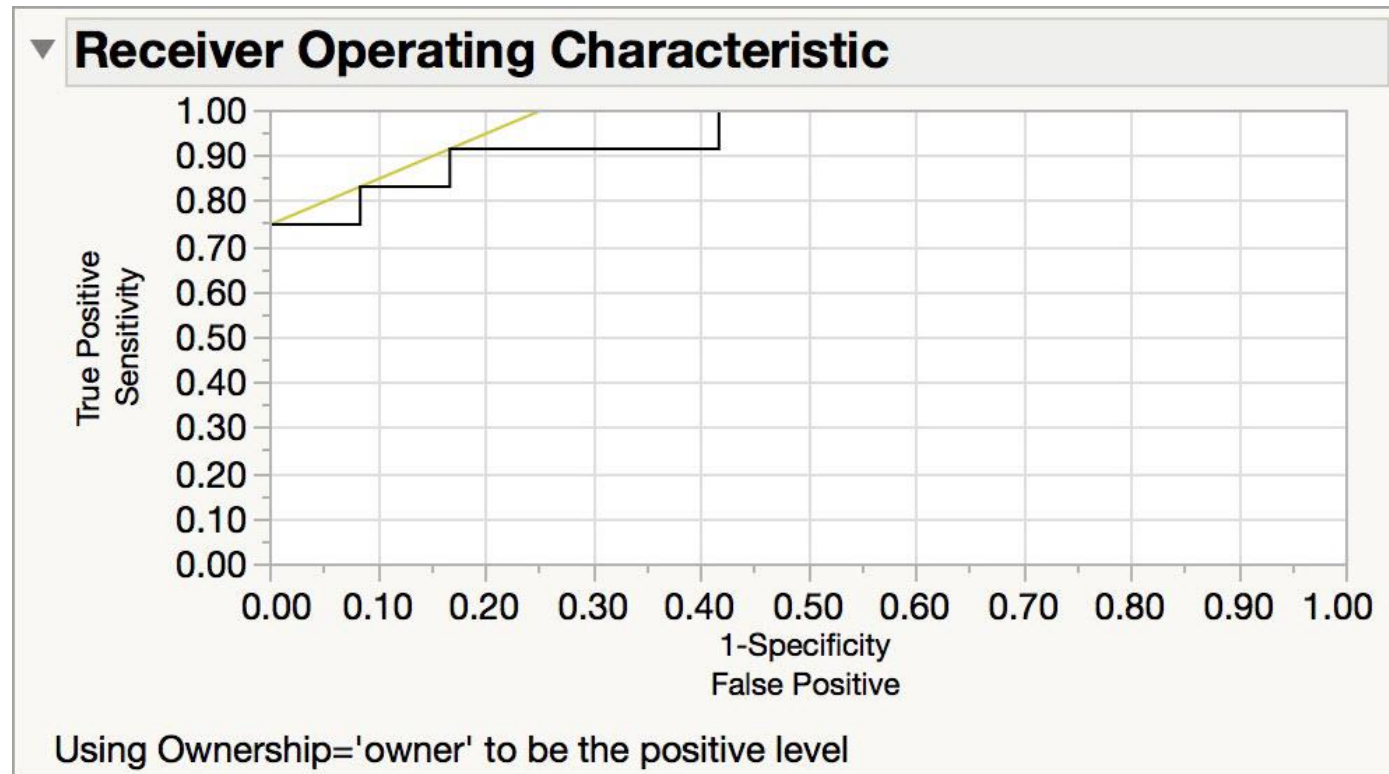
Specificity = % of “ C_0 ” class correctly classified

False positive rate = proportion of predicted “ C_1 ’s” that were not “ C_1 ’s”

False negative rate = proportion of predicted “ C_0 ’s” that were not “ C_0 ’s”

ROC Curve

Sensitivity and 1-Specificity are plotted on an ROC Curve.



ROC curves

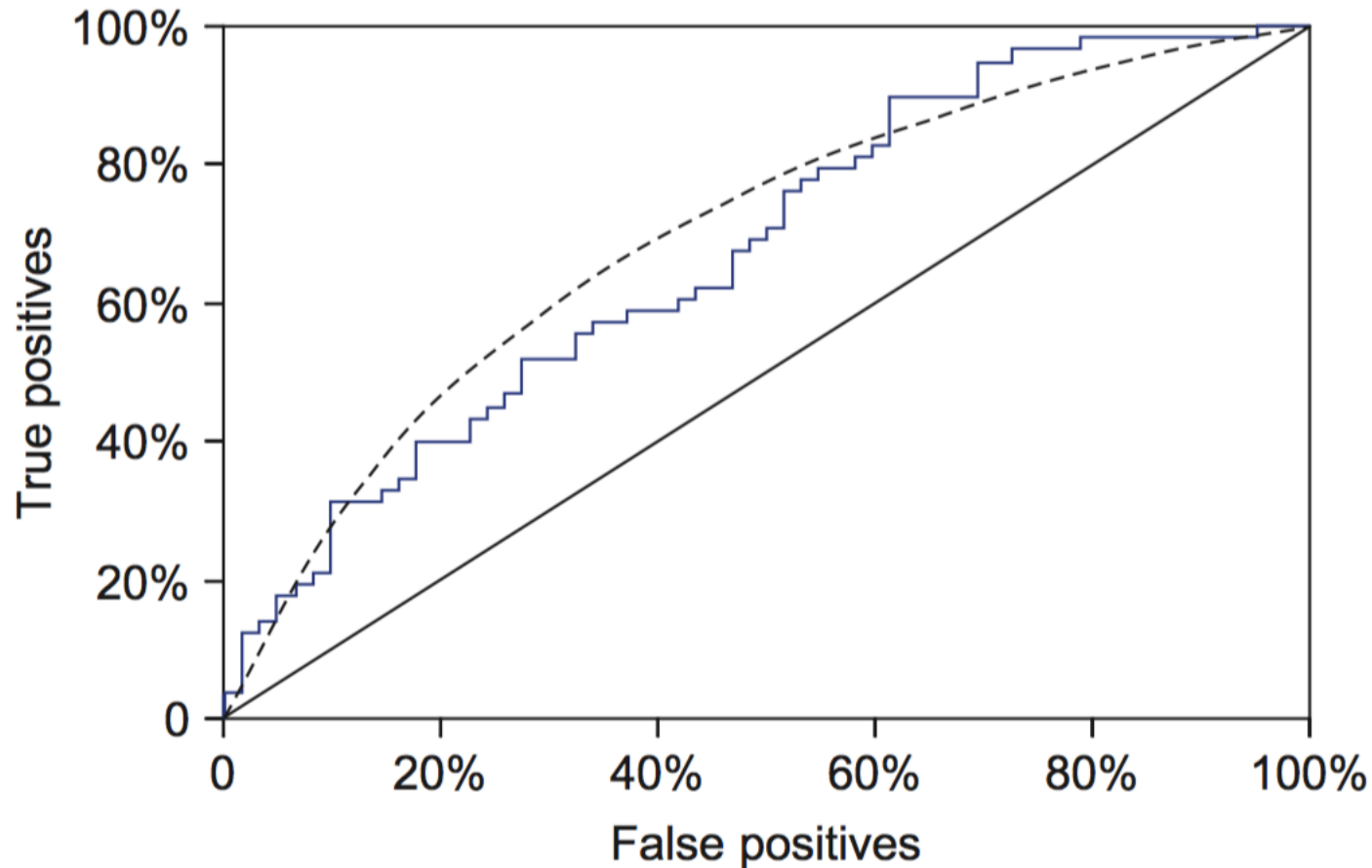
ROC curves are similar to lift charts

- Stands for “receiver operating characteristic”
- Used in signal detection to show tradeoff between hit rate and false alarm rate over noisy channel

Differences to lift chart:

- y axis shows percentage of true positives in sample *rather than absolute number*
- x axis shows percentage of false positives in sample *rather than sample size*

A sample ROC curve



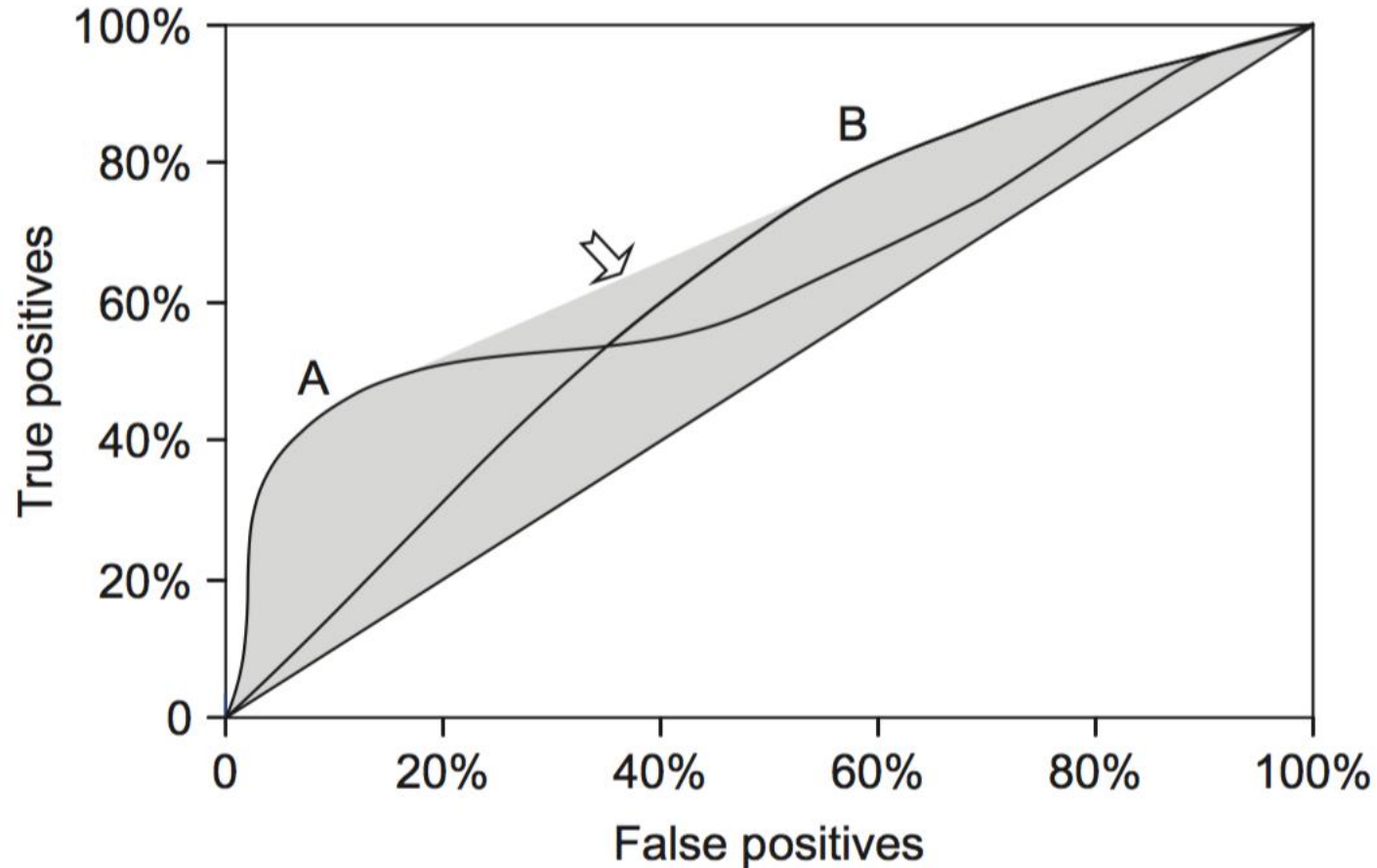
- Jagged curve—one set of test data
- Smoother curve—use cross-validation

Cross-validation and ROC curves

Simple method of getting a ROC curve using cross-validation:

- Collect probabilities for instances in test folds
- Sort instances according to probabilities
- Another possibility is to generate an ROC curve for each fold and average them

ROC curves for two schemes



- For a small, focused sample, use method A
- For a larger one, use method B
- Inbetween, choose between A and B with appropriate probabilities

Lift Curves

Useful for assessing performance in terms of identifying the most important class

Helps evaluate, e.g.,

- How many tax records to examine
- How many loans to grant
- How many customers to mail offer to

Lift Curves

Compare performance of data mining (DM) model to “no model, pick randomly”

Measures ability of DM model to identify the important class, relative to its average prevalence

Lift curves give explicit assessment of results over a large number of cutoffs

Lift Curves

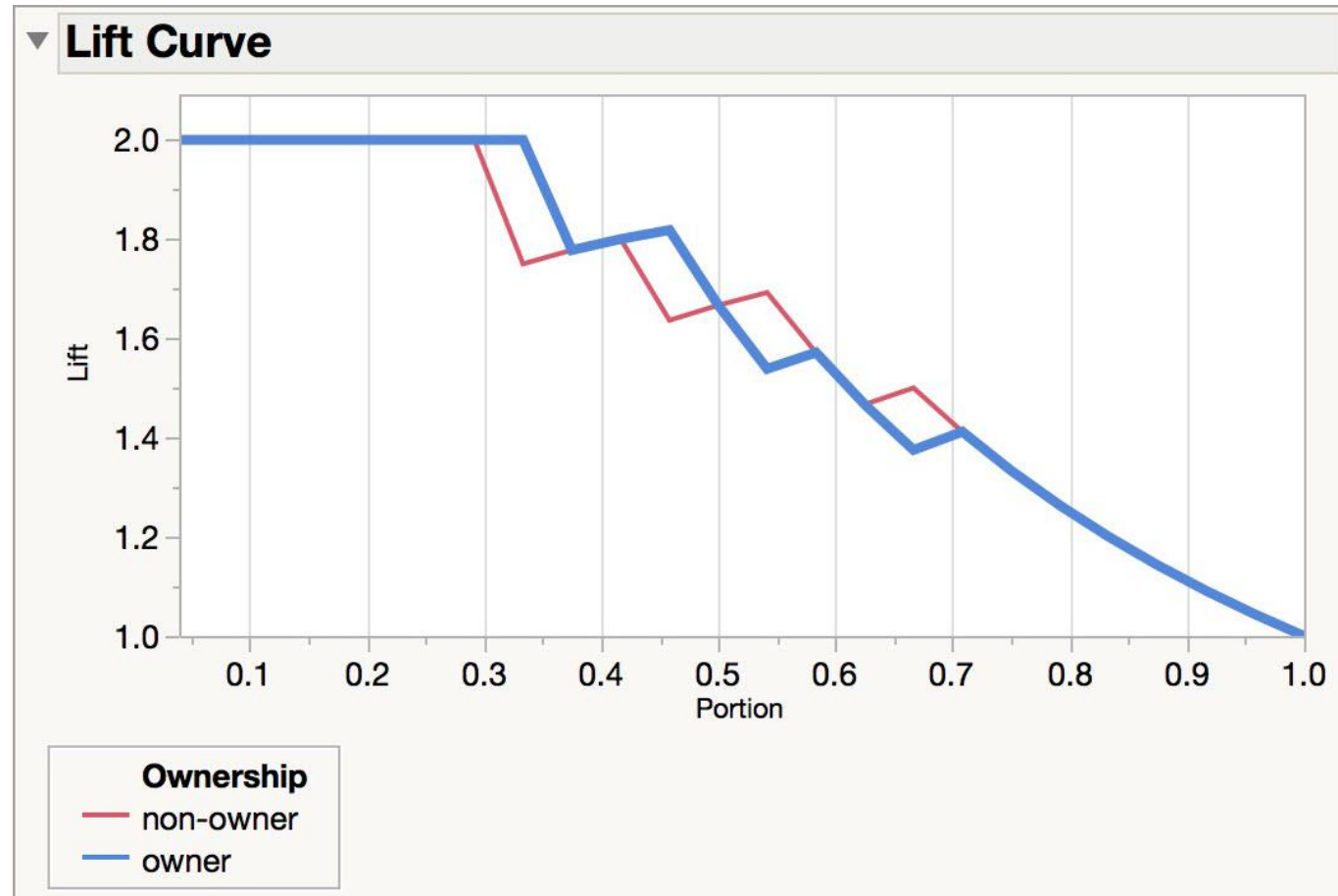
Compare lift to “no model” baseline:

- For a given proportion of our data, in terms of propensity, how much better does our model predict than the “no model” baseline.

What is lift:

- How much better the model classifies than random assignment.

Lift Curves



For the top 30% of our data (top propensities), the lift is 2.0. The model correctly identifies 2x the number of owners than the “no model” baseline.

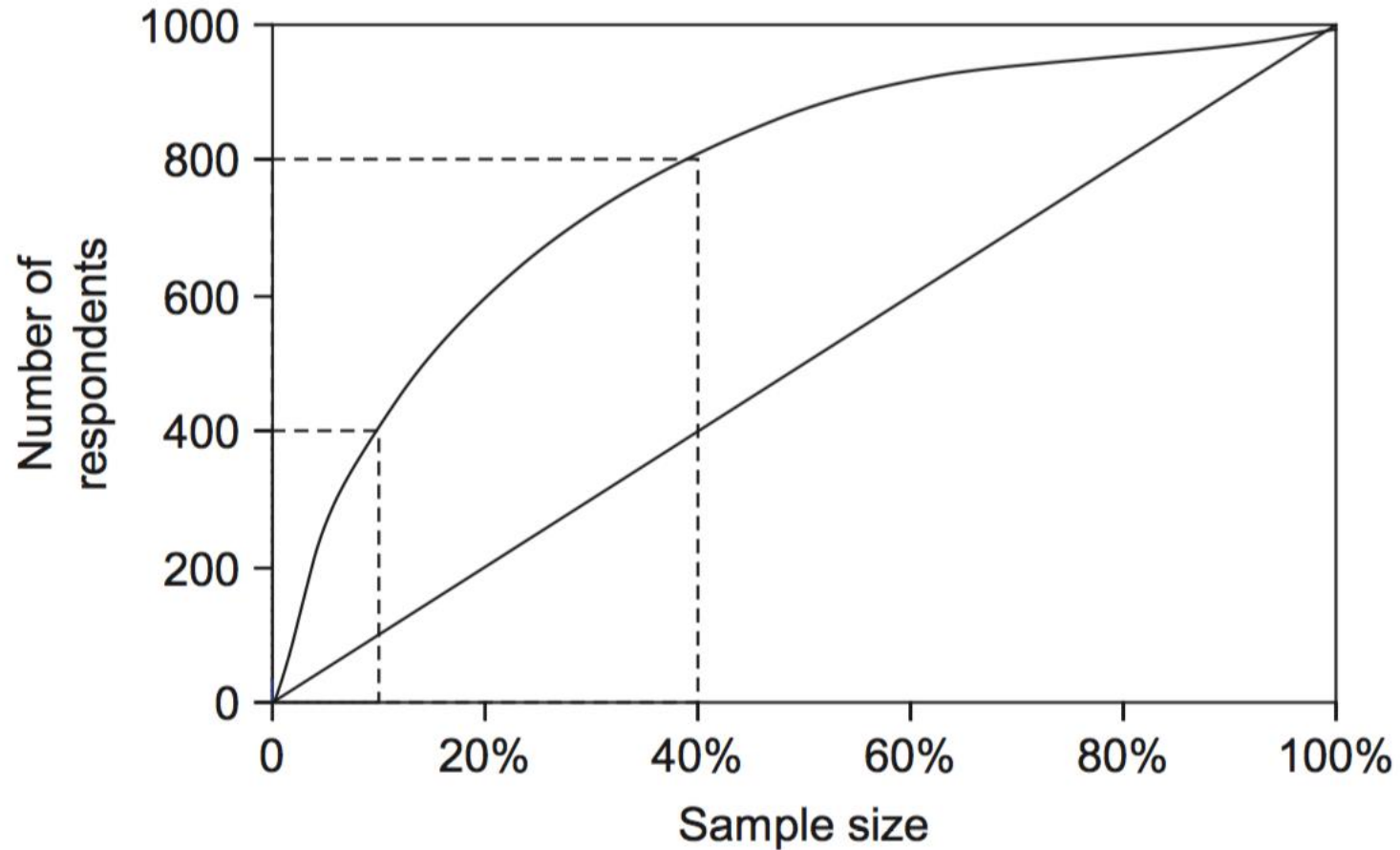
Generating a lift chart

- Sort instances based on predicted probability of being positive:

Rank	Predicted	Actual	Rank	Predicted	Actual Class
1	0.95	Yes	11	0.77	No
2	0.93	Yes	12	0.76	Yes
3	0.93	No	13	0.73	Yes
4	0.88	Yes	14	0.65	No
5	0.86	Yes	15	0.63	Yes
6	0.85	Yes	16	0.58	No
7	0.82	Yes	17	0.56	Yes
8	0.80	Yes	18	0.49	No
9	0.80	No	19	0.48	Yes
10	0.79	Yes

- x axis in lift chart is sample size for each probability threshold
- y axis is number of true positives above threshold

A hypothetical lift chart



40% of responses
for 10% of cost

80% of responses
for 40% of cost

Asymmetric Costs

Misclassification Costs May Differ

The cost of making a misclassification error may be higher for one class than the other(s)

Looked at another way, the benefit of making a correct classification may be higher for one class than the other(s)

Example – Response to Promotional Offer

Suppose we send an offer to 1000 people, with 1% average response rate (“1” = response, “0” = nonresponse):

- “Naïve rule” (classify everyone as “0”) has error rate of 1% (seems good)
- Using DM we can correctly classify eight 1’s as 1’s.
It comes at the cost of misclassifying twenty 0’s as 1’s and two 0’s as 1’s.

The Confusion Matrix

	Predict Class 0	Predict Class 1
Actual 0	970	20
Actual 1	2	8

Error rate = $(2+20) = 2.2\%$ (higher than naïve rate)

Introducing Costs & Benefits

Suppose:

- Profit from a “1” is \$10
- Cost of sending offer is \$1

Then:

- Under naïve rule, all are classified as “0”, so no offers are sent: no cost, no profit

Introducing Costs & Benefits

- Under DM predictions, 28 offers are sent.
 - 8 respond with profit of \$10 each
 - 20 fail to respond, cost \$1 each
 - 972 receive nothing (no cost, no profit)
- Net profit = \$60

Profit	Predict Class 0	Predict Class 1
Actual 0	0	– \$20
Actual 1	0	\$80

Generalize to Cost Ratio

Sometimes actual costs and benefits are hard to estimate

- Need to express everything in terms of costs (i.e., cost of misclassification per record)
- Goal is to minimize the average cost per record

A good practical substitute for individual costs is the **ratio** of misclassification costs (e.g., “misclassifying fraudulent firms is 5 times worse than misclassifying solvent firms”)

Minimizing Cost Ratio

q_1 = cost of misclassifying an actual “1”,

q_0 = cost of misclassifying an actual “0”

Minimizing the **cost ratio** q_1/q_0 is identical to minimizing the average cost per record

Note: Opportunity costs

- As we see, best to convert everything to costs, as opposed to a mix of costs and benefits
- E.g., instead of “benefit from sale” refer to “opportunity cost of lost sale”
- Leads to same decisions, but referring only to costs allows greater applicability

Multiple Classes

For m classes, confusion matrix has m rows and m column:

- Theoretically, there are $m(m-1)$ misclassification costs, since any case could be misclassified in $m-1$ ways
- Practically too many to work with
- In decision-making context, though, such complexity rarely arises – one class is usually of primary interest

Oversampling and Asymmetric Costs

Rare Cases

Asymmetric costs/benefits typically go hand in hand with presence of rare but important class

- Responder to mailing
- Someone who commits fraud
- Debt defaulter

Often we oversample rare cases to give model more information to work with

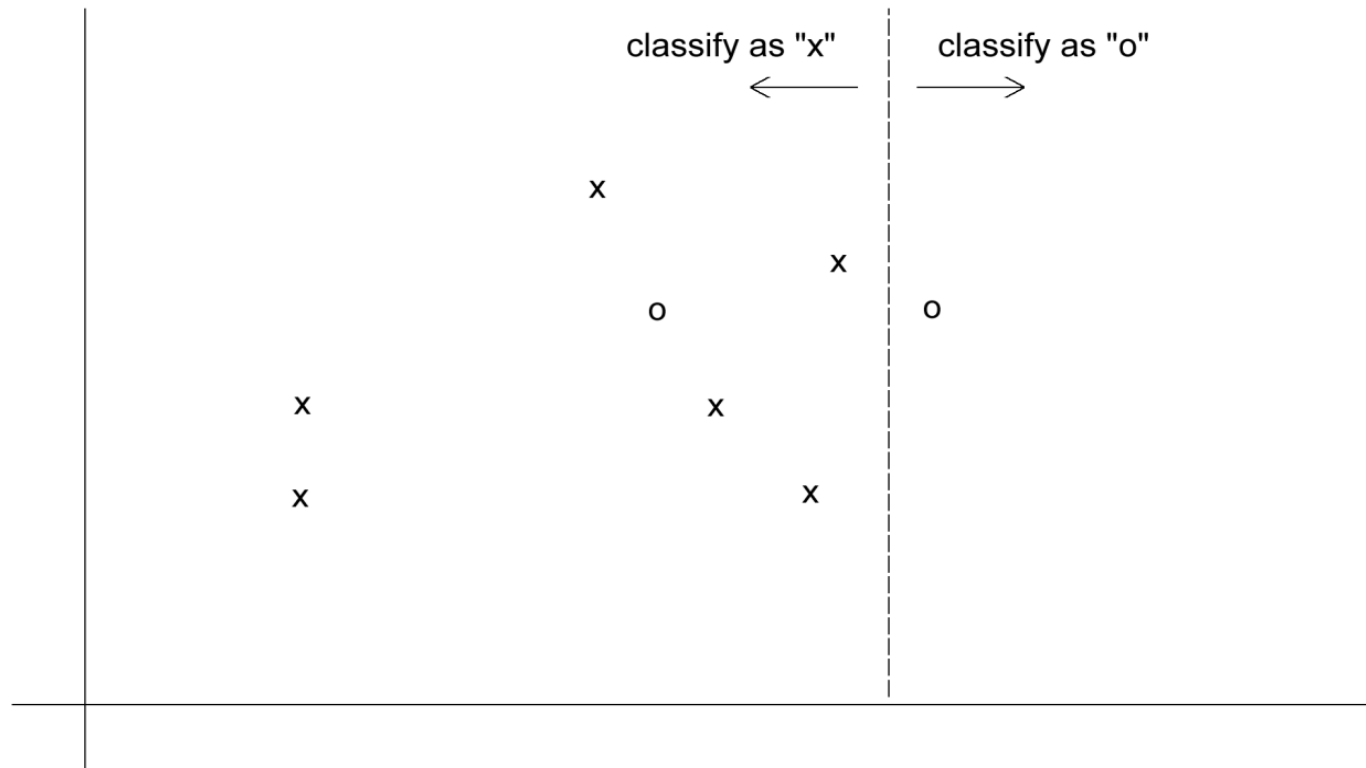
Typically use 50% “1” and 50% “0” for training

Example

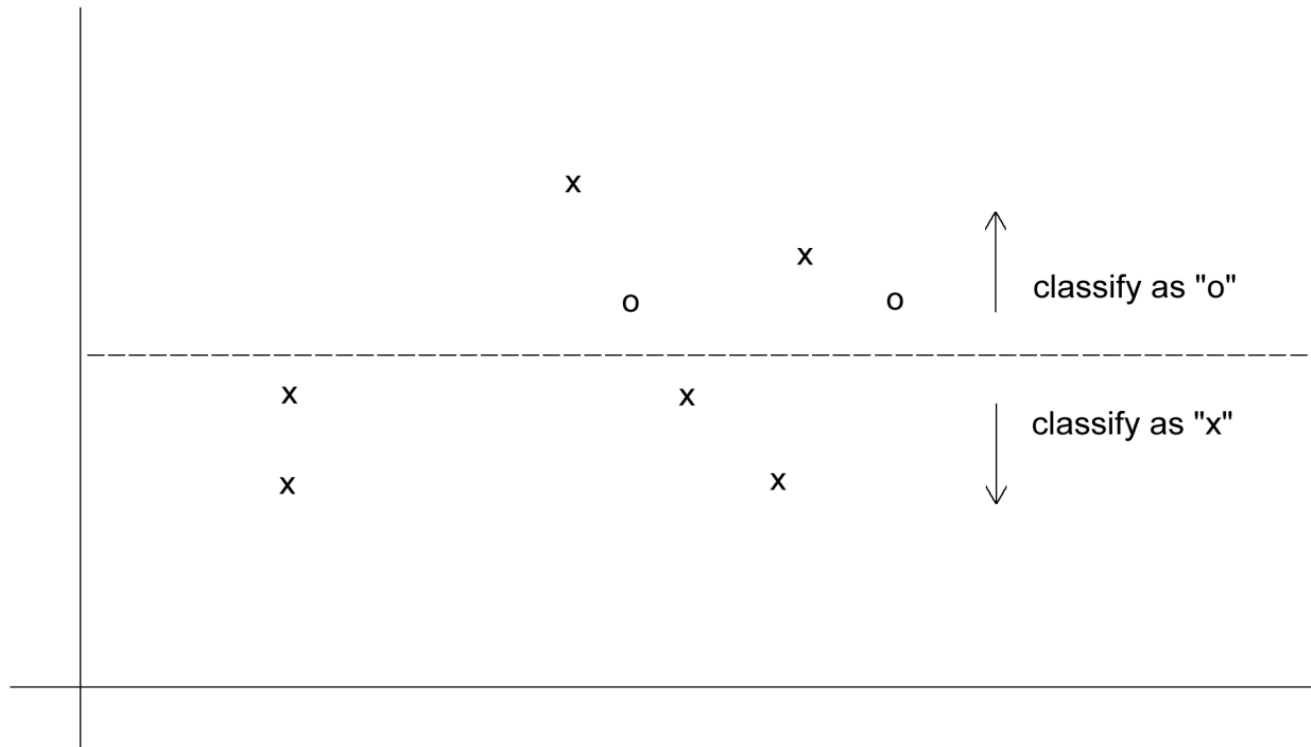
The graphs on the following slides show optimal classification under three scenarios:

- assuming equal costs of misclassification
- assuming that misclassifying “o” is five times the cost of misclassifying “x”
- Oversampling scheme allowing DM methods to incorporate asymmetric costs

Classification: equal costs

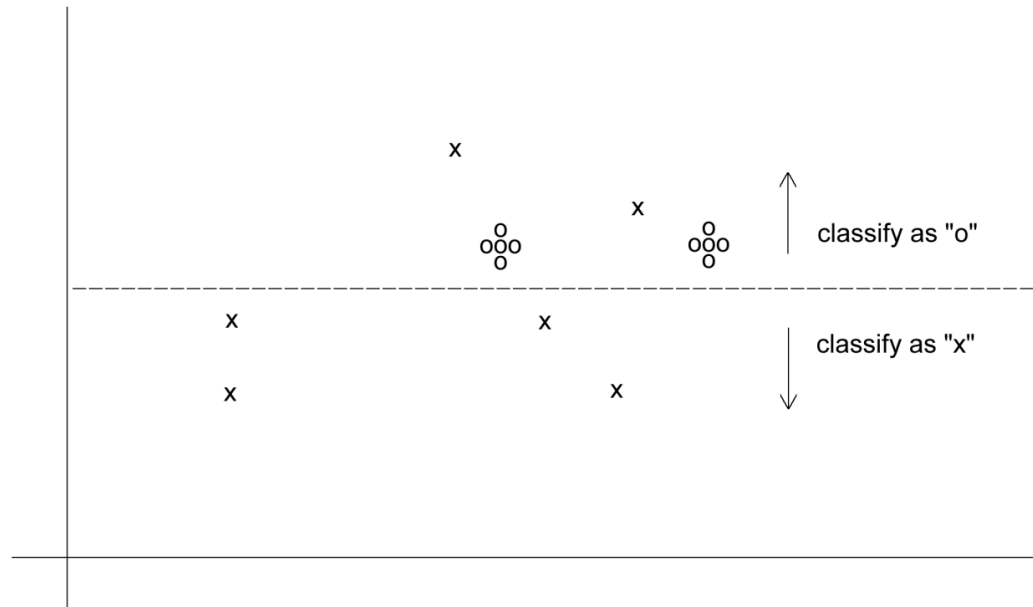


Classification: Unequal costs



Oversampling Scheme

Oversample "o" to appropriately weight/
misclassification costs



An Oversampling Procedure

1. Separate the responders (rare) from non-responders
2. Randomly assign half the responders to the training sample, plus equal number of non-responders
3. Remaining responders go to validation sample
4. Add non-responders to validation data, to maintain original ratio of responders to non-responders
5. Randomly take test set (if needed) from validation

Classification Using Triage

- Instead of classifying as C_1 or C_0 , we classify as:
 - C_1
 - C_0
 - Can't say Take into account a gray area in making classification decisions

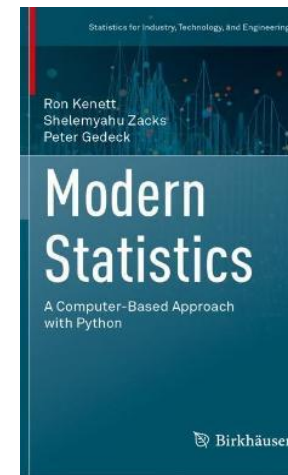
The third category might receive special human review

Summary

- Evaluation metrics are important for comparing across DM models, for choosing the right configuration of a specific DM model, and for comparing to the baseline
- Major metrics: confusion matrix, error rate, predictive error
- Other metrics when
 - one class is more important
 - asymmetric costs
- When important class is rare, use oversampling
- In all cases, metrics are computed from validation data

Chapter 3

Statistical Inference and Bootstrapping



Preview In this chapter we introduce basic concepts and methods of statistical inference. The focus is on estimating the parameters of statistical distributions and testing hypotheses about them. Problems of testing if certain distributions fit observed data are also considered.

Bootstrapping background

A computer intensive method, proposed in 1979 by Brad Efron from Stanford University in order to conclude something about a statistic T computed from data, **using only the data.**

Bootstrapping can be considered a **Principle of Statistical Inference** that can handle non-standard situations in an **intuitive** and **robust** way



von Minchausen, KFH (1783) *The Guide to Happy People*, Berlin

Efron, B (1979) Bootstrap Methods: Another Look at the Jackknife, *Annals of Statistics* 7(1): 1-26

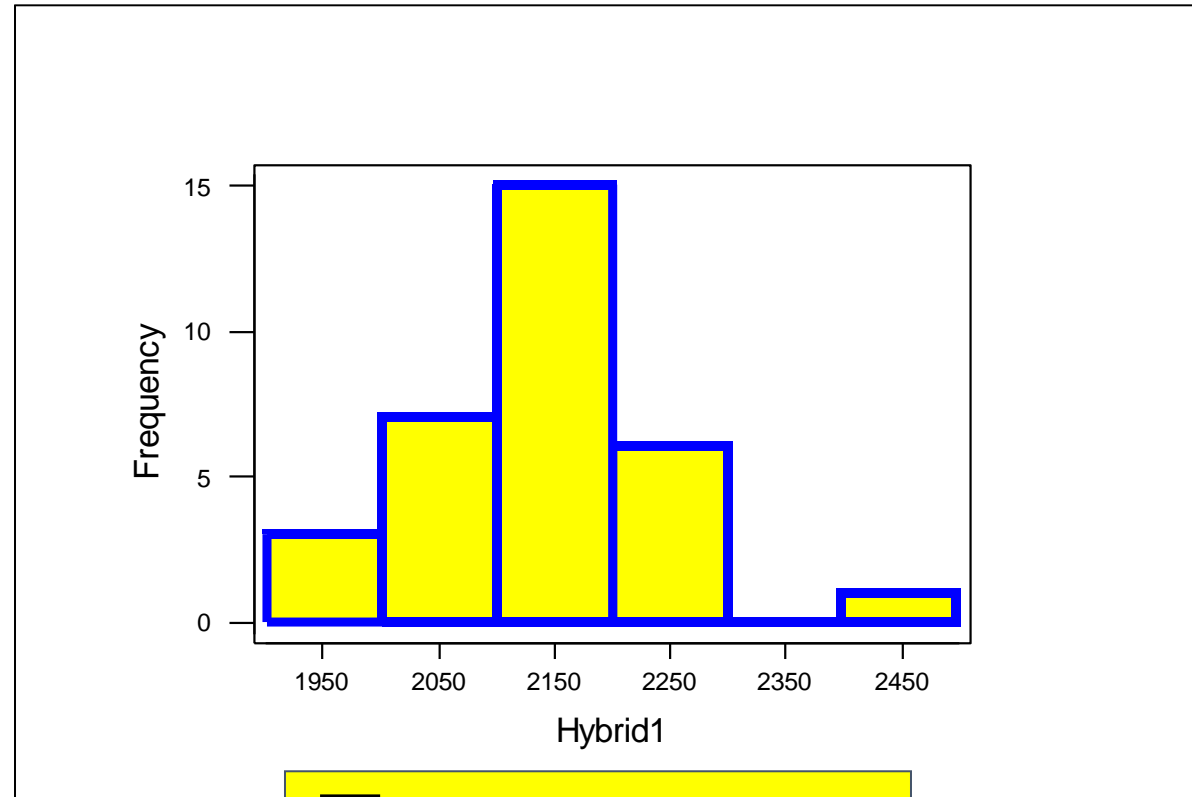
Efron, B and Tibshirani, R (1993) *An Introduction to the Bootstrap*, Chapman and Hall, New York

How does it work?

- Take a Random Sample With Replacement (RSWR) and compute the statistic T
- Resample M times and re-compute statistic T
- Derive Empirical Bootstrap Distribution (EBD) and Bootstrap Confidence Interval (BI) for population parameter

Testing for the mean

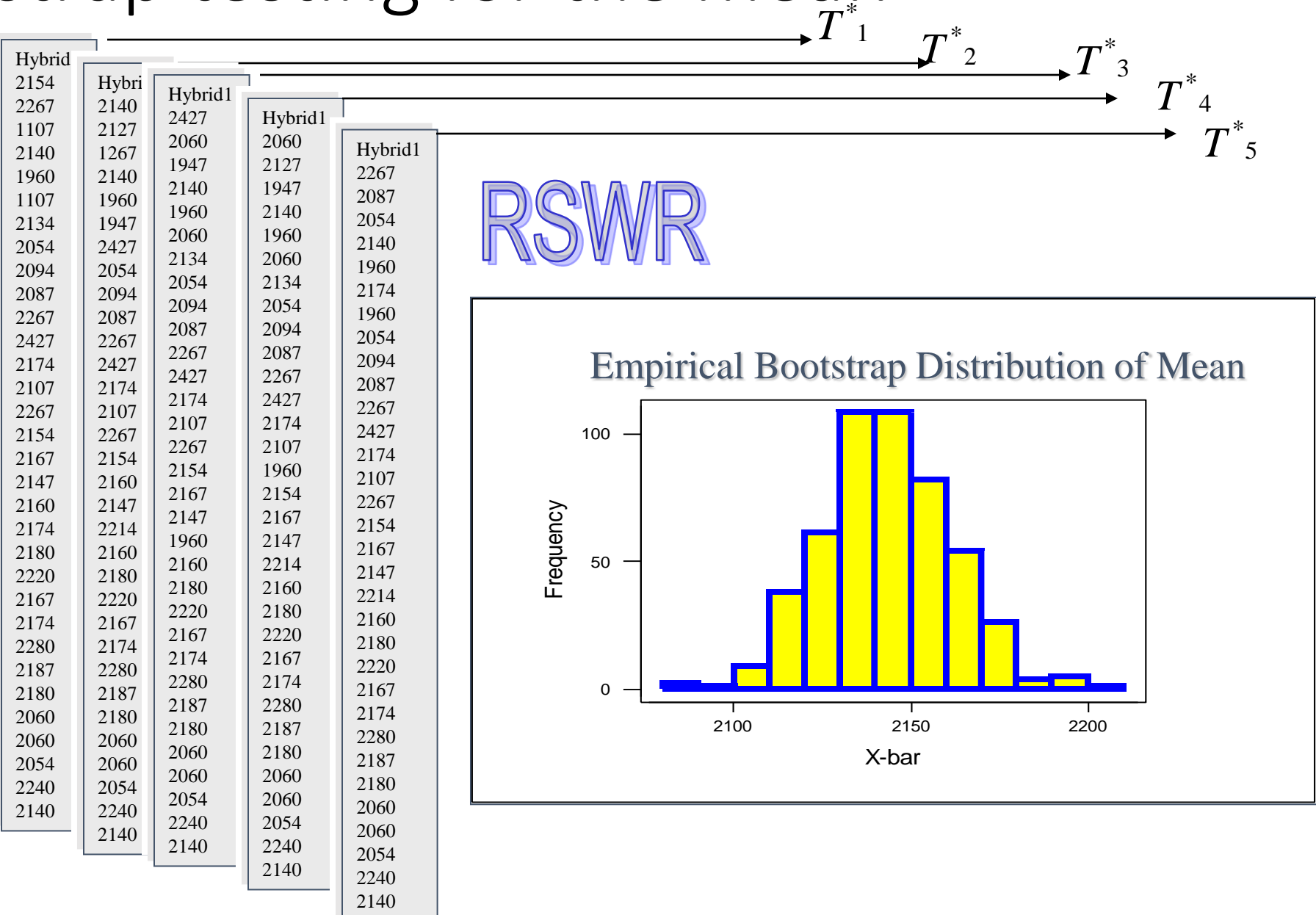
Hybrid1
2060
2127
1947
2140
1960
1960
2134
2054
2094
2087
2267
2427
2174
2107
2267
2154
2167
2147
2214
2160
2180
2220
2167
2174
2280
2187
2180
2060
2060
2054
2240
2140



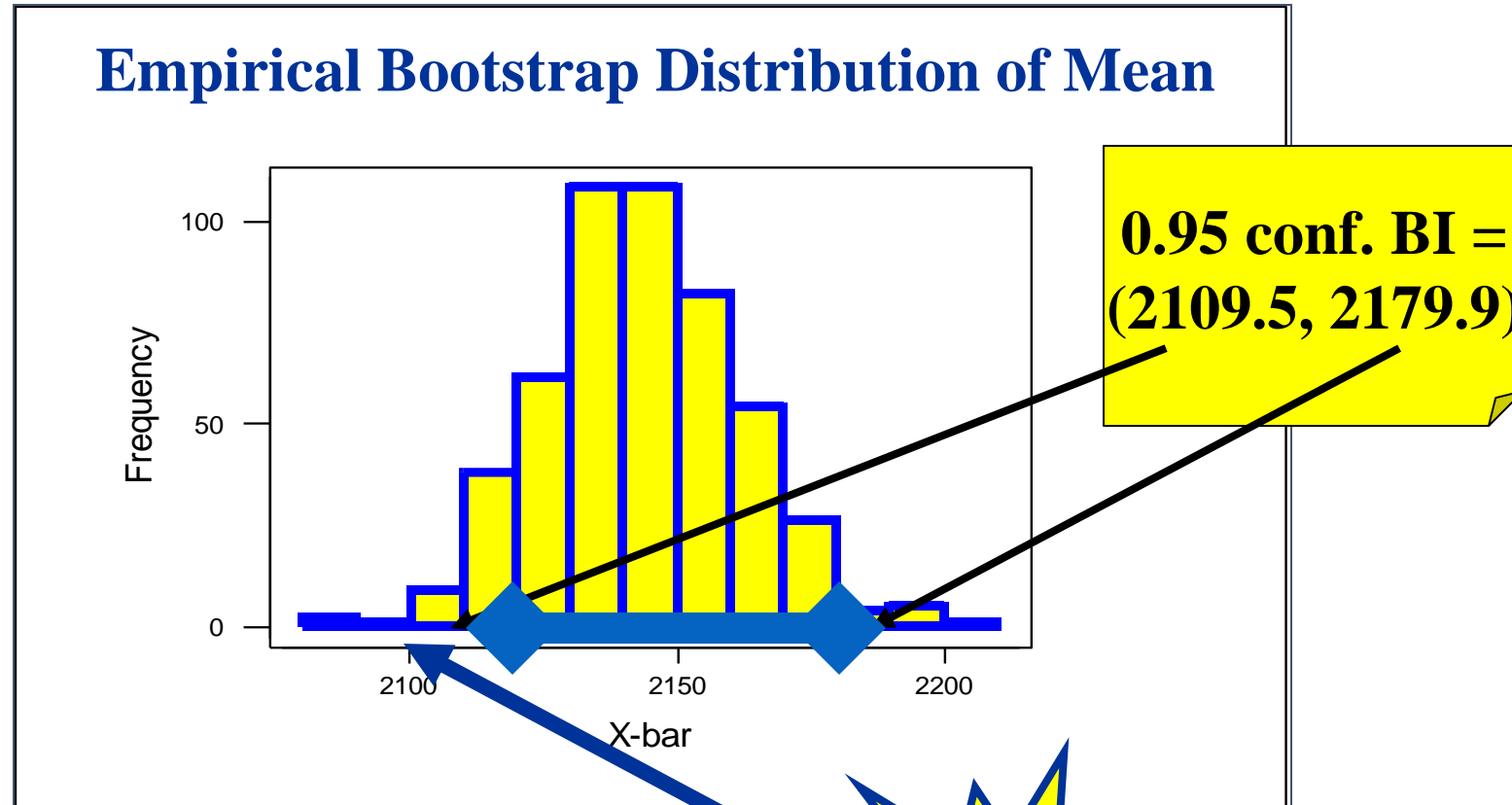
$$\bar{X}_{32} = 2143.4\Omega$$

Is this significantly different from 2100ohm ?

Bootstrap testing for the mean



Empirical Bootstrap Distribution of the mean

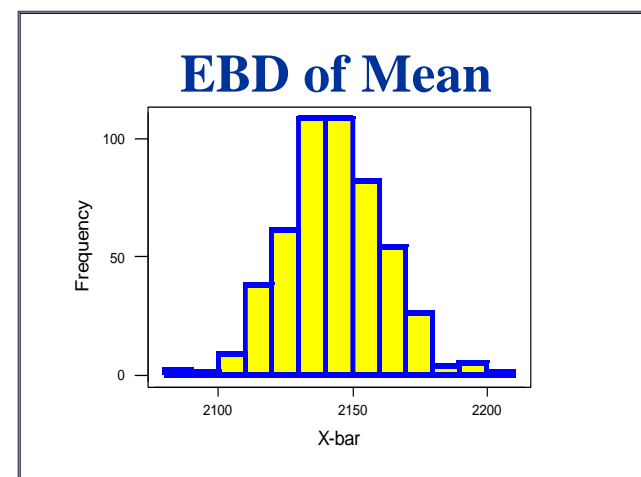
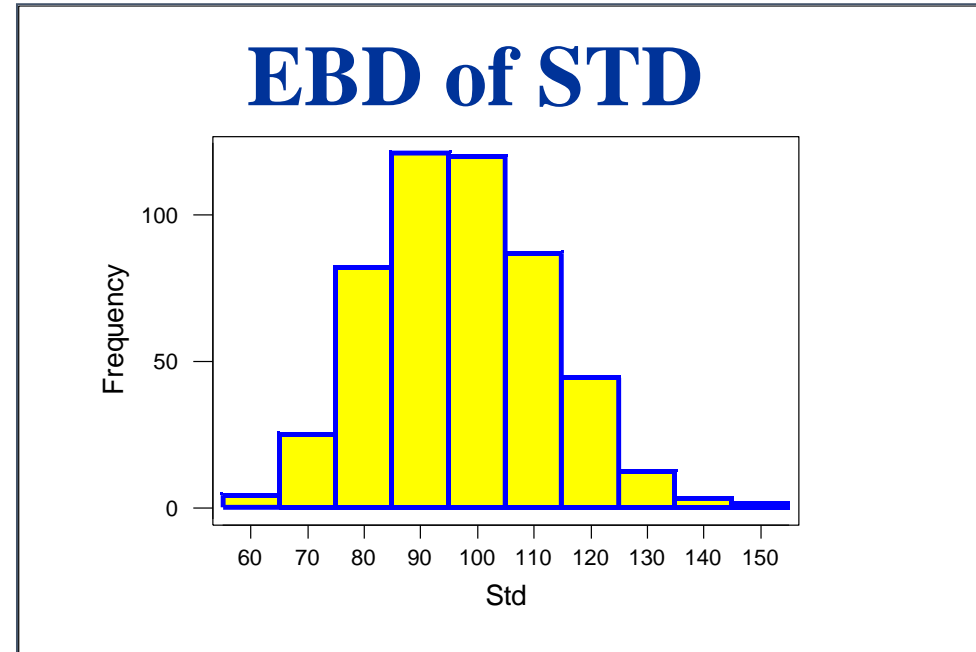


$$\bar{X}_{32} = 2143.4\Omega \neq$$

2100

Bootstrapping the standard deviation

X-bar	Std
2135.03	87.850
2149.84	121.631
2141.19	109.258
2149.09	78.084
2134.00	103.856
2122.13	73.843
2119.66	86.625
2113.59	107.136
2138.97	101.693
2163.00	67.725



Bootstrapping the ANOVA table

Hybrid1	Hybrid2	Hybrid3
2060	1907	1887
2127	1940	1834
1947	1700	1587
2140	1934	1814
1960	1707	1614
1960	1680	1680
2134	1940	1747
2054	1794	1660
2094	1707	1600

$$\bar{X}_1 = 2143.406$$

$$\bar{X}_2 = 1902.813$$

$$\bar{X}_3 = 1850.344$$

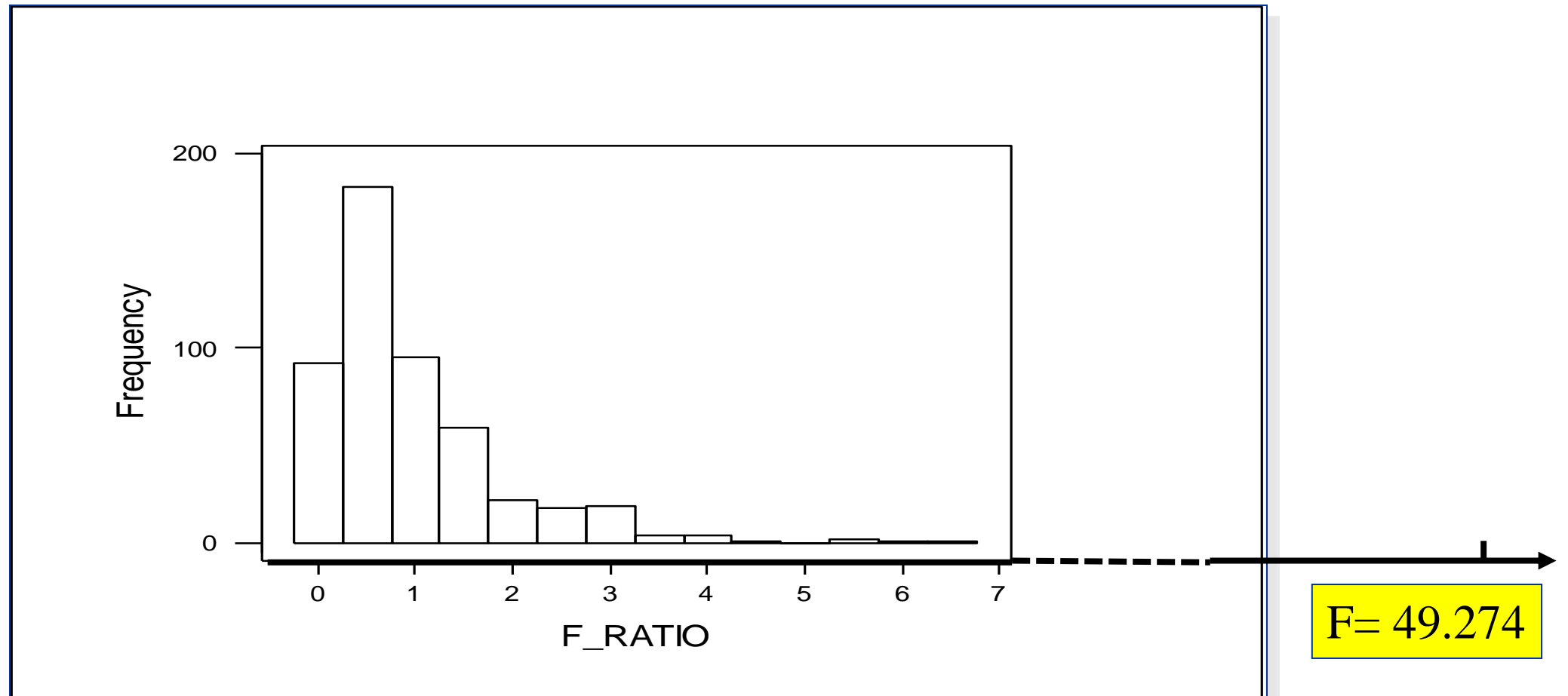
$$S^2_1 = 9929.54$$

$$S^2_2 = 16648.35$$

$$S^2_3 = 21001.01$$

$$F = \text{MSBetween} / \text{MSWithin} = 49.274$$

EBD of F values under H0



Kenett, R.S. and Zacks, S. (2021) Modern Industrial Statistics: With Applications in R, MINITAB, and JMP, 3rd Edition, Wiley.

Kenett, R.S., Zacks, S. and Gedeck, P (2022) Modern Statistics: A Computer-Based Approach with Python, Springer.

Befitting Bootstrap Analysis (BBA) Principles

- **BBA Principle 1:** The generation of bootstrapped datasets should reflect the goal of the study.
- **BBA Principle 2:** The bootstrapped dataset should inherit the same data generation structure as the original dataset.

<https://onlinelibrary.wiley.com/doi/abs/10.1002/qre.802>

QUALITY AND RELIABILITY ENGINEERING INTERNATIONAL

Qual. Reliab. Engng. Int. 2006; **22**:659–667

Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/qre.802

Special Issue

Bootstrap Analysis of Designed Experiments

Ron S. Kenett^{1,2,*},†, Effi Rahav³ and David M. Steinberg^{1,3}

¹*KPA Ltd, 25 Hataasia Street, P.O. Box 2525, Raanana 43100, Israel*

²*Department of Statistics and Applied Mathematics, 'Diego de Castro' University of Torino, Piazza Arbarello 8, 10121 Torino, Italy*

³*Tel Aviv University, Tel Aviv 69978, Israel*

Wave soldering

Objective: Decrease number of solder defects following wave soldering



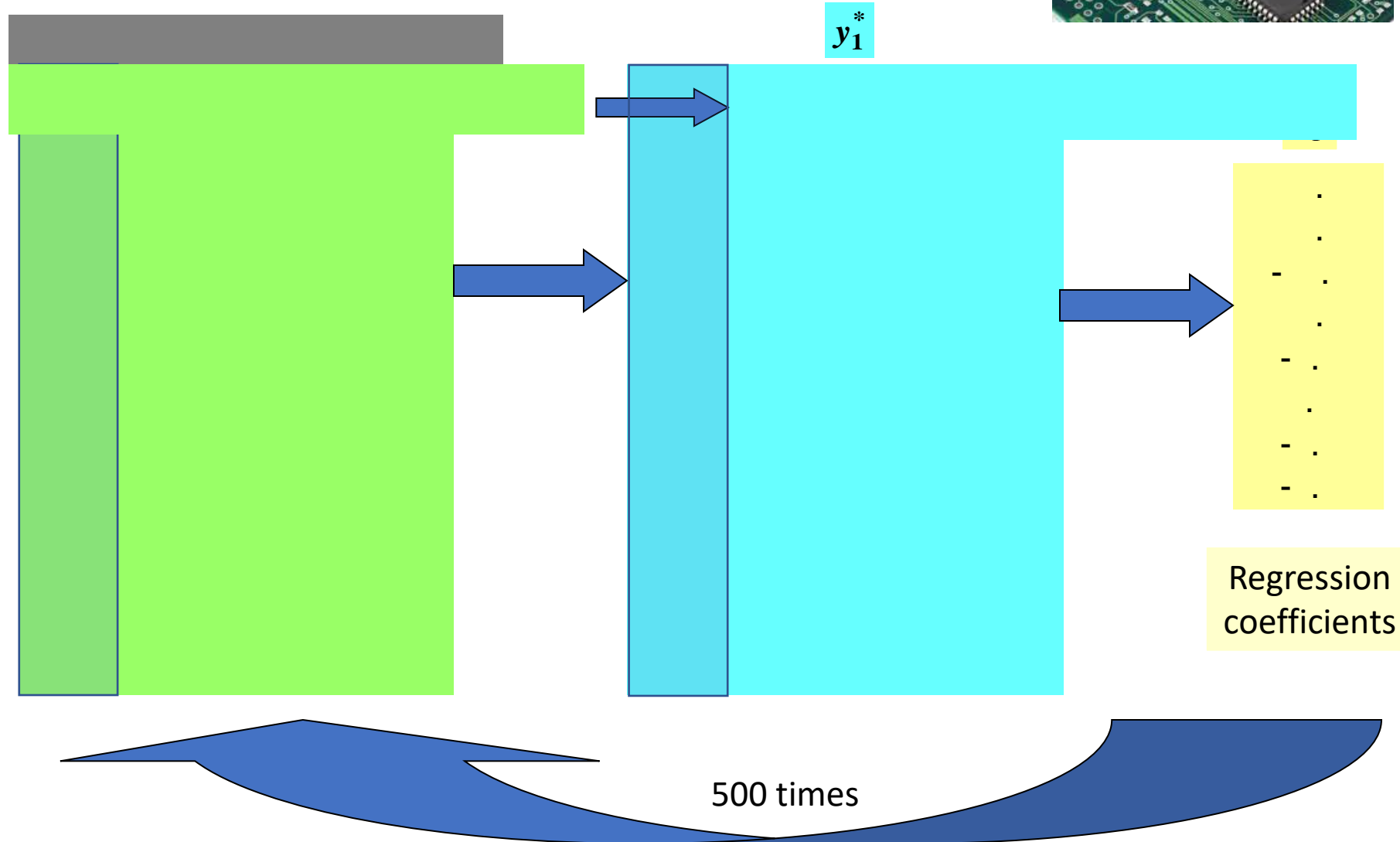
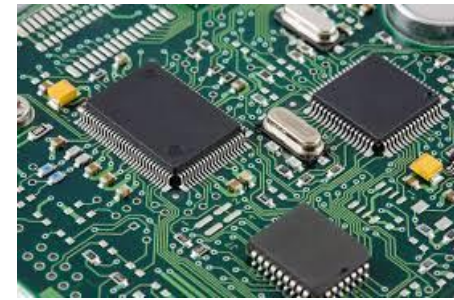
Factors

Levels

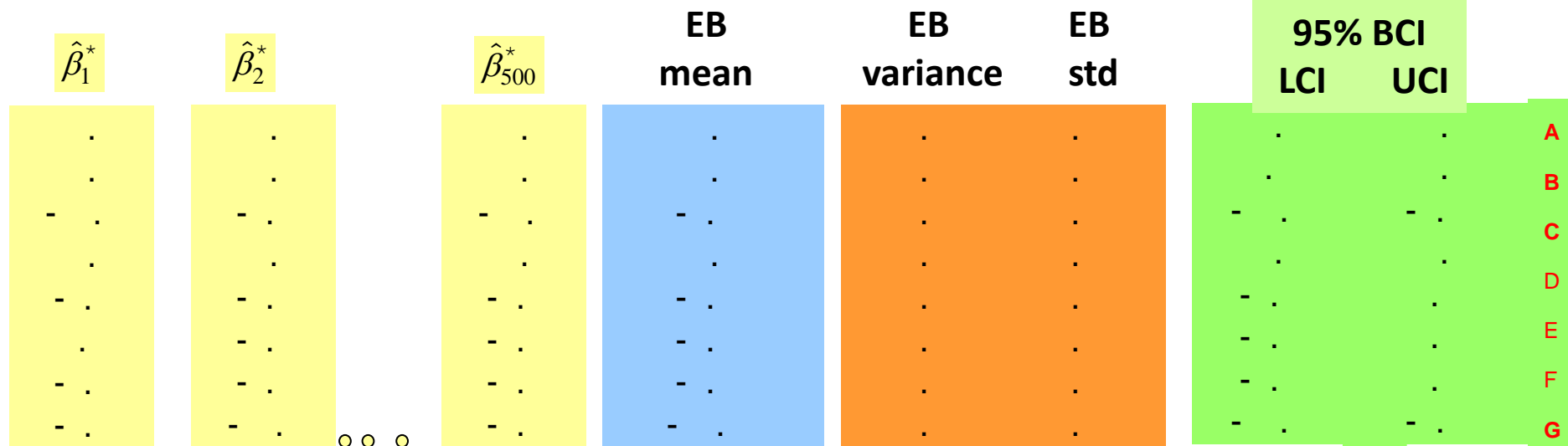
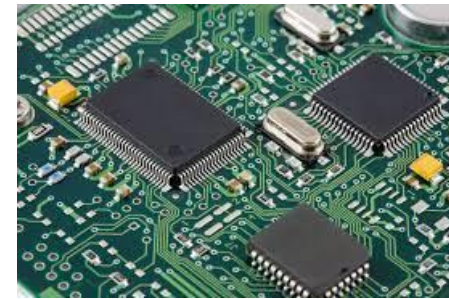
A: prebake temperature	-1	+1
B: flux density	-1	+1
C: conveyor speed	-1	+1
D: preheat condition	-1	+1
E: cooling time	-1	+1
F: solder agitator	-1	+1
G: solder temperature	-1	+1

Pattern	A	B	C	D	E	F	G
1	-----	-1	-1	-1	-1	-1	-1
2	----++++	-1	-1	-1	1	1	1
3	---+---+	-1	-1	1	-1	1	-1
4	----+---+	-1	-1	1	1	-1	1
5	--+---++	-1	1	-1	-1	1	1
6	-+---+--	-1	1	-1	1	-1	-1
7	---+---++	-1	1	1	-1	-1	1
8	-++++--	-1	1	1	1	1	-1
9	+-----+	1	-1	-1	-1	-1	1
10	+---+---	1	-1	-1	1	1	-1
11	+--+-++	1	-1	1	-1	1	-1
12	+---+---	1	-1	1	1	-1	1
13	++-+---	1	1	-1	-1	1	-1
14	+++----+	1	1	-1	1	-1	-1
15	+++-----	1	1	1	-1	-1	-1
16	+++++++	1	1	1	1	1	1

Wave soldering



Applying the bootstrap



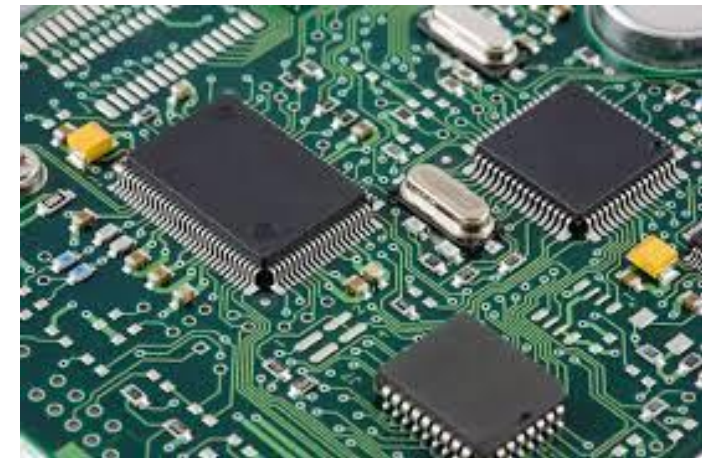
- A: prebake temperature**
- B: flux density**
- C: conveyor speed**
- D: preheat condition
- E: cooling time
- F: solder agitator
- G: solder temperature**



EBD versus Regression estimates*

	Empirical Bootstrap			95% BCI	
	mean	variance	std	LCI	UCI
A	0.5476	0.0596	0.2441	0.1929	0.9477
B	-0.5462	0.0597	0.2443	-0.935	-0.186
C	1.4333	0.0604	0.2458	1.0754	1.8512
D	-0.1468	0.0577	0.2402	-0.5253	0.2114
E	-0.2822	0.0577	0.2402	-0.6391	0.1014
F	-0.166	0.0611	0.2472	-0.5817	0.1939
G	-0.8878	0.062	0.249	-1.2623	-0.4692

	Least Squares			
	Value	Std.	t value	Pr(> t)
A	0.5404	0.2986	1.8095	0.0779
B	-0.5393	0.2986	-1.806	0.0784
C	1.4238	0.2986	4.7678	0
D	-0.153	0.2986	-0.5123	0.6113
E	-0.282	0.2986	-0.9442	0.3507
F	-0.1563	0.2986	-0.5235	0.6035
G	-0.8932	0.2986	-2.991	0.0047

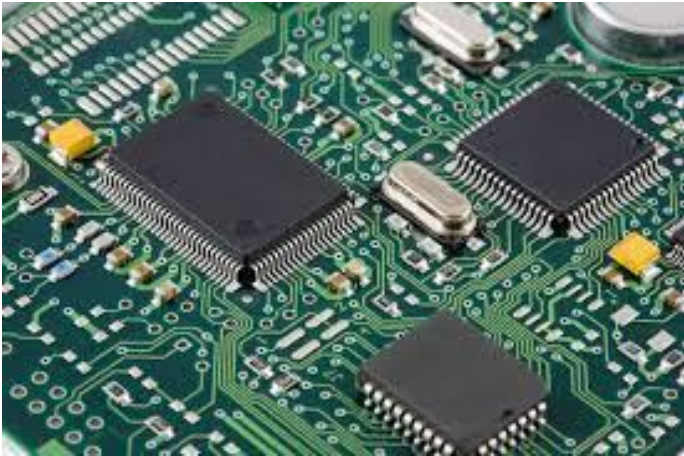


*After applying a square root transformation

EBD versus Regression estimates

	No interactions			With interactions		
	Regr.	Bootstrap	Delta	Regr.	Bootstrap	Delta
A	0.299	0.244	18%	0.239	0.249	-4%
B	0.299	0.244	18%	0.239	0.237	1%
C	0.299	0.246	18%	0.239	0.243	-2%
D	0.299	0.240	20%	[Redacted]		
E	0.299	0.240	20%			
F	0.299	0.247	17%			
G	0.299	0.249	17%	0.239	0.246	-3%
A*B				0.239	0.240	-1%
A*C				0.239	0.244	-2%
B*C				0.239	0.235	2%
A*G				0.239	0.240	-1%
B*G				0.239	0.239	0%
C*G				0.239	0.245	-3%
A*B*C				0.239	0.245	-3%
A*B*G				0.239	0.239	0%
A*C*G				0.239	0.245	-3%
B*C*G				0.239	0.239	0%
A*B*C*G				0.239	0.235	2%

Better model



<https://www.youtube.com/watch?v=Yi-e4sMK5tA&t=701s>



Bootstrapping,
cross validation
and SVEM:
Differences and
similarities with
applications to
industrial
processes

Ron S. Kenett and Chris Gotwalt

ENBIS Spring Meeting on Data Science in Process Industries

17 - 18th May 2021

Play (k)

