VALIDATION & STABILIZATION

Methodology Training Document (Module 5)

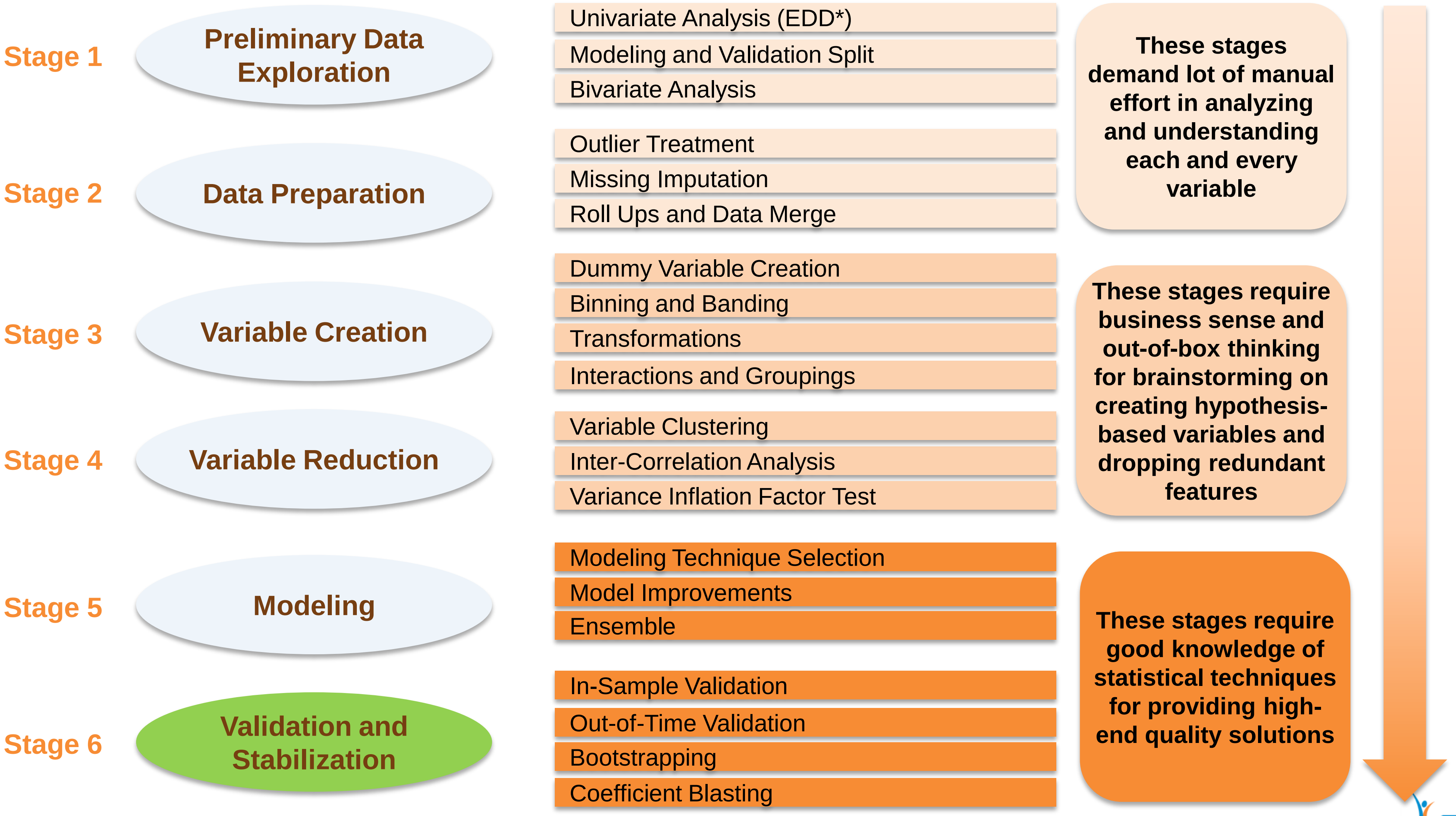
YEAR 2015

DART

EXL
look deeper.

EXL Decision Analytics Methodology Snapshot

We apply a set of highly effective tools, techniques and best practices for the end-to-end model development cycle



Stage 1 – **Preliminary Data Exploration**
- Univariate Analysis (EDD*)
- Modeling and Validation Split
- Bivariate Analysis

Stage 2 – **Data Preparation**
- Outlier Treatment
- Missing Imputation
- Roll Ups and Data Merge

Stage 3 – **Variable Creation**
- Dummy Variable Creation
- Binning and Banding
- Transformations
- Interactions and Groupings

Stage 4 – **Variable Reduction**
- Variable Clustering
- Inter-Correlation Analysis
- Variance Inflation Factor Test

Stage 5 – **Modeling**
- Modeling Technique Selection
- Model Improvements
- Ensemble

Stage 6 – **Validation and Stabilization**
- In-Sample Validation
- Out-of-Time Validation
- Bootstrapping
- Coefficient Blasting

These stages demand lot of manual effort in analyzing and understanding each and every variable

These stages require business sense and out-of-box thinking for brainstorming on creating hypothesis-based variables and dropping redundant features

These stages require good knowledge of statistical techniques for providing high-end quality solutions

* Extended Data Dictionary

Objectives and Scope

Course Goals

- To provide a structured overview of model validation and stabilization techniques used during application of EXL DA methodology
- To introduce trainees to several model performance and stability measures
- To explain metric calculations through illustrations
- Hands on exercises on real life data to practice calculation of validation metrics during the training course
- To provide helpful “tricks of the trade”

Beyond the Scope of this Training

- Comprehensive coaching on model validation
- Derivation of statistical formulas or terms (unless required as part of methodology explanation)

Self Study Goals

- Model validation practice on hypothetical data
- In-depth research on advanced concepts relating to validation and stabilization
- Discussion on advanced concepts can be taken up offline

Table of Contents

Chapter 1: Basics of Model Validation

1.1 Model Validation

1.1.1. Need for Validation

What is Model Validation?

Model validation is a process of determining the degree to which a statistical software generated model (based on input data) is an accurate representation of the real world

Why is Validation Needed?

- *Generalization*

  To ascertain whether predicted values from the model are likely to accurately predict responses on future subjects or subjects not used to develop the model

- *Stability Check*

  To test how consistently the model is going to perform over time

- *Robustness Check*

  To test whether the model is an appropriate representation of the real world for the stated purpose and whether the model is acceptable for its intended use

A model without sufficient validation is only a hypothesis.

1.1.2. Types of Validation

Type	Description	Technique	Validity Strength
Apparent	Performance on sample used to develop model	Apparent	🟩⬜⬜⬜⬜⬜⬜
Internal (Out-of-Sample)	Performance on population underlying the sample	Split Sample	🟩🟩⬜⬜⬜⬜⬜
		Cross Validation	🟩🟩🟩⬜⬜⬜⬜
		Bootstrapping	🟩🟩🟩🟩⬜⬜⬜
External	Performance on related but slightly different population	Out-of-time (OOT)	🟩🟩🟩🟩🟩⬜⬜
		Spatial Validation	🟩🟩🟩🟩🟩🟩⬜
		Fully External Validation	🟩🟩🟩🟩🟩🟩🟩

Apparent Validation

- Measures model performance on modeling data itself; there is no significant value add
- Provides optimistic estimates of model performance
- Very easy to implement
- Validity strength is very low; Implementation of such model in real world may show disappointing results

Internal Validation

- Data for model development and evaluation are both random samples from the same underlying population
- Provides honest and reasonable estimates of model performance
- Sets an upper limit to what may be expected in external validation
- Slightly difficult to implement; All model variables need to be created in the validation set

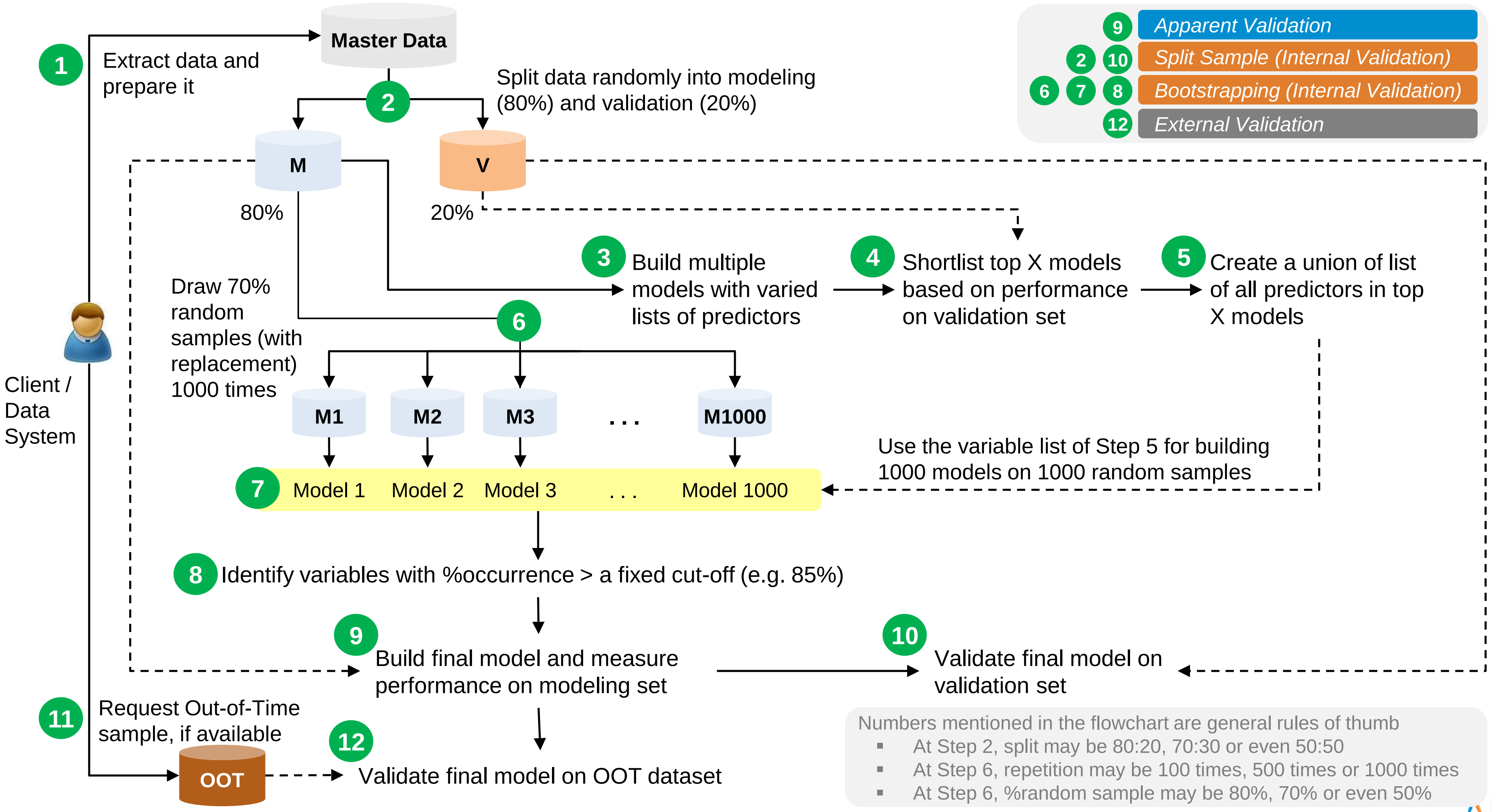
External Validation

- Once the model is developed, it is validated in other settings
- Very strong test of model performance
- Difficult to implement
  - Appropriate population eligibility conditions to be applied for validation population
  - All model variables need to be created in the validation set

Illustration: To predict the probability that a college student pays fees on time

Type	Technique	Modeling Data	Validation Data
Apparent	Apparent	Year 2011 batch students of College XYZ	Same as modeling data
Internal (Out-of-Sample)	Split Sample	X% (e.g. 80%) random sample of year 2011 batch students of College XYZ	Remaining (i.e. 20% of) year 2011 batch students of College XYZ
	Cross Validation (k-fold)	1. Divide data into k equal sized random samples. For example, k = 5 2. Use 4 samples (i.e. 80% data) for modeling and 1 sample (i.e. 20%) for validation 3. Repeat Step 2 five times so that all 5 samples are used for validation once 4. Take average of validation metric across 5 samples	
	Bootstrapping	1. Keep aside a holdout sample for validation 2. Draw 80% random sample (with replacement) for modeling 3. Repeat Step 2 large number of times (m). For example, m = 1000 times 4. Keep those variables in final model whose %occurrence in m models > fixed cut-off (say, 85%)	
External	Out-of-time (OOT)	Year 2011 batch students of College XYZ	**Same population in different time period** **Year 2012 batch** students of College XYZ
	Spatial Validation		**Different population in same time period** Year 2011 batch students of **College ABC**
	Fully External Validation		**Different population in different time period** **Year 2012 batch** students of **College ABC**

1.1.3. EXL's Standard Approach



1 Extract data and prepare it

Master Data

Client / Data System

2 Split data randomly into modeling (80%) and validation (20%)

M 80%

V 20%

3 Build multiple models with varied lists of predictors

4 Shortlist top X models based on performance on validation set

5 Create a union of list of all predictors in top X models

6 Draw 70% random samples (with replacement) 1000 times

M1 M2 M3 . . . M1000

Use the variable list of Step 5 for building 1000 models on 1000 random samples

7 Model 1 Model 2 Model 3 . . . Model 1000

8 Identify variables with %occurrence > a fixed cut-off (e.g. 85%)

9 Build final model and measure performance on modeling set

10 Validate final model on validation set

11 Request Out-of-Time sample, if available

OOT

12 Validate final model on OOT dataset

9	*Apparent Validation*
2 10	*Split Sample (Internal Validation)*
6 7 8	*Bootstrapping (Internal Validation)*
12	*External Validation*

Numbers mentioned in the flowchart are general rules of thumb

- At Step 2, split may be 80:20, 70:30 or even 50:50
- At Step 6, repetition may be 100 times, 500 times or 1000 times
- At Step 6, %random sample may be 80%, 70% or even 50%

1.2 Bias and Variance

1.2.1. Error Decomposition

Consider a model, where error (ε) is normally distributed with zero mean and a constant variance

$$Y = f(X) + \varepsilon \quad \text{such that} \quad E(\varepsilon) = 0 \text{ and } Var(\varepsilon) = \sigma_\varepsilon^2$$

Let $f(X)$ be estimated by model $\hat{f}(X)$

Things to Remember

- Bias is a measure of avg. prediction error across samples
- Variance reflects how much prediction varies from one sample to another

Expected squared prediction error at a point x_0 is given by:

$$\begin{aligned} Err(x_0) &= E[Y - \hat{f}(x_0)]^2 \\ &= \sigma_\varepsilon^2 + [E(\hat{f}(x_0)) - f(x_0)]^2 + E[(\hat{f}(x_0) - E(\hat{f}(x_0))]^2 \\ &= \sigma_\varepsilon^2 + [Bias(\hat{f}(x_0))]^2 + Var(\hat{f}(x_0)) \end{aligned}$$

Noise

Irreducible error on target Y

Bias²

Deviation of the average estimate from the true function's mean

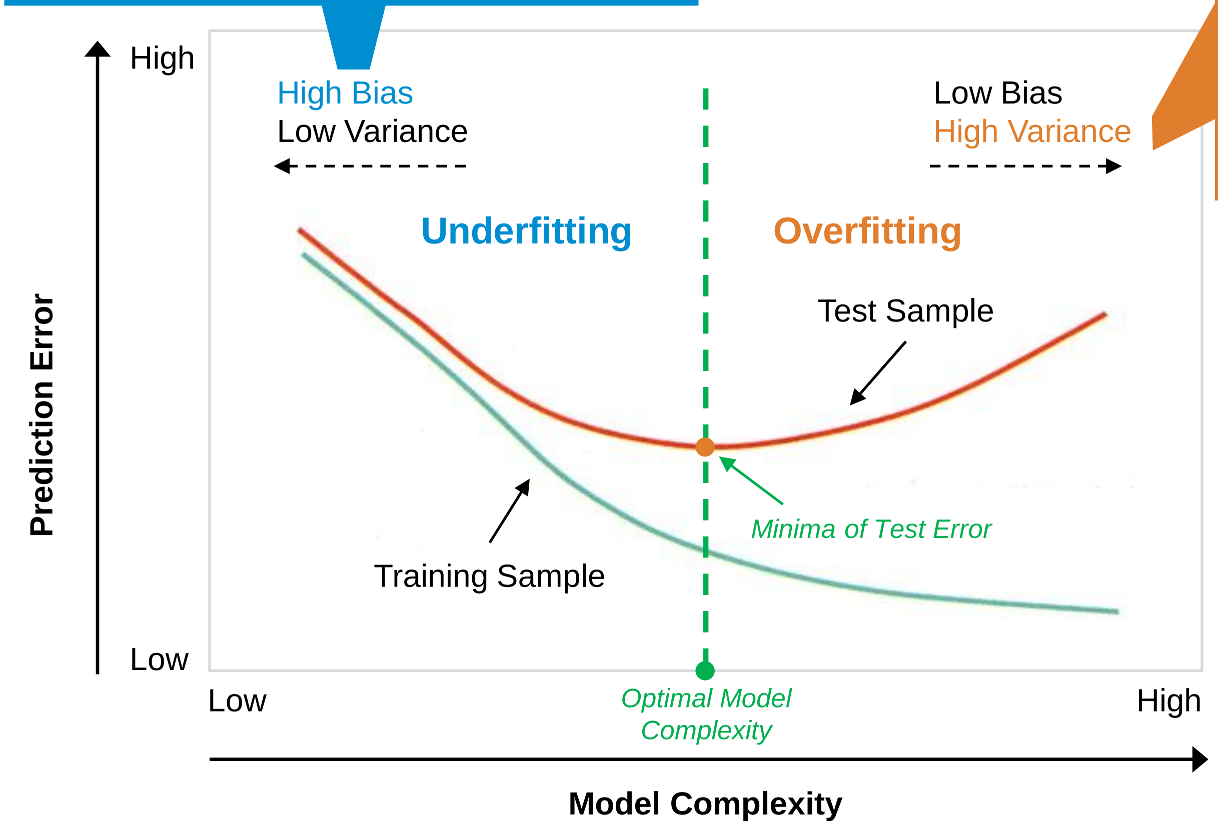
Variance

Expected squared deviation of model's estimate around its mean

1.2.2. Bias and Variance Trade-Off

If a model is too simple, the model would

- Be unable to fit the true structure
- Have a lot of bias (error between the true function and model's approximation)



If a model is too complex, the model would

- Overfit to the noise in training sample
- Become very sensitive to the particular training sample used
- Have a lot of variance across training samples

Things to Remember

- Training error is typically lower than test error
- Training error can be reduced by increasing model complexity, but this risks overfitting
- It is recommended to **minimize the test error to obtain optimal level of model complexity**

1.3 Components of Validation

1.3.1. Sampling Strategies

- Sampling strategies are aimed at addressing the uncertainty that can arise in tests using empirical data
- Examples: Cross Validation, Bootstrapping, Out-of-Sample and Out-of-Time Validation

1.3.2. Power-Testing

- Power-testing techniques are aimed at measuring model's goodness-of-fit
- Examples
  - Classification Table, K-S Statistic, AUC and Concordance for a classification model
  - R^2 for a regression model

1.3.3. Calibration

- Calibration techniques are aimed at assessing how closely the model's predictions match with the actual (i.e. observed) values
- Examples
  - Hosmer-Lemeshow test for a classification model
  - Primary and Secondary Diagonal Metric, ME, MSE, RMSE, MAE, MPE and MAPE for a regression model

While sampling strategies are meant for model stabilization, power testing and calibration measure model performance

Chapter 2: Validation Methods

2.1 Classification Model Performance Measures

2.1.1. Classification Table (Confusion Matrix)

Classification table

- 2x2 matrix of actual and predicted classes
- Also known as Confusion Matrix or Contingency Table
- Greater the sum of primary diagonal (TP + TN), higher the degree of classification accuracy

Positive, because predicted class = 1
True, because prediction is correct

Positive, because predicted class = 1
False, because prediction is wrong

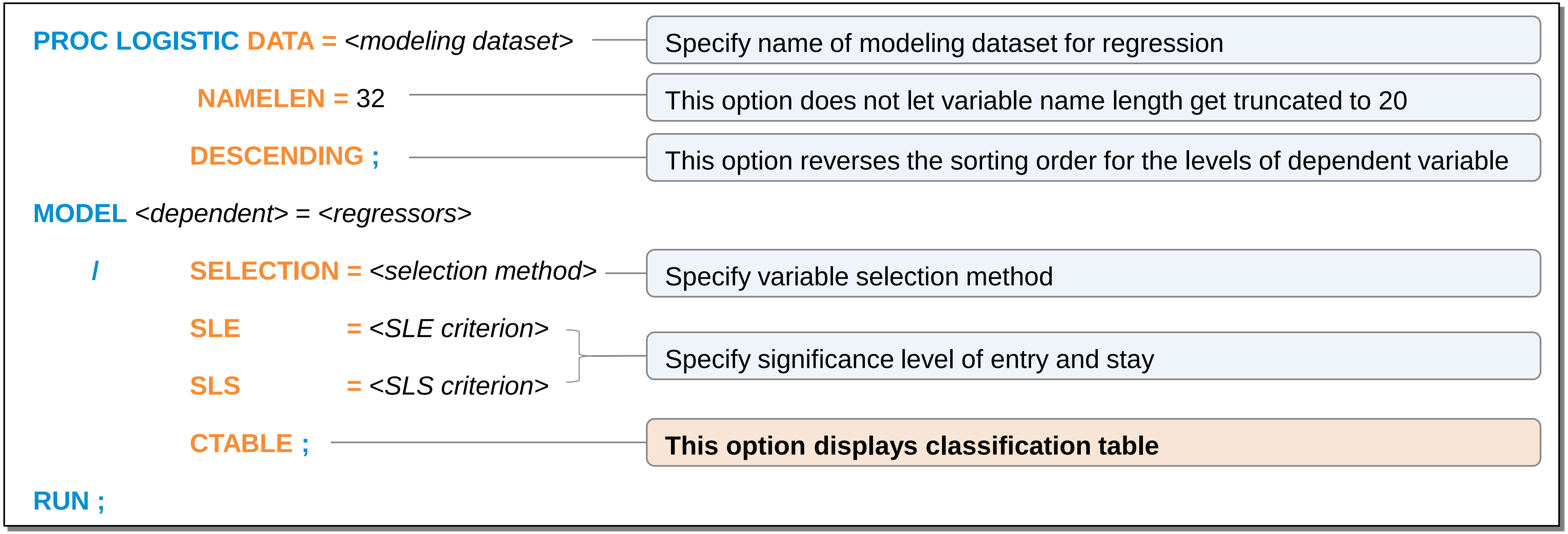
	Target = 1 **(Event)**	**Target = 0** **(Non-Event)**	**Row Total**
Predicted Class = 1	TP True Positive	FP False Positive	**TP + FP** **#Cases predicted as Event**
Predicted Class = 0	FN False Negative	TN True Negative	**FN + TN** **#Cases predicted as Non-Event**
Column Total	**TP + FN = E** **#Events**	**FP + TN = NE** **#Non-Events**	**N = TP + TN + FP + FN** **Total #cases**

Negative, because predicted class = 0
False, because prediction is wrong

Negative, because predicted class = 0
True, because prediction is correct

SAS Implementation

Below is the syntax for generating classification table



```
PROC LOGISTIC DATA = <modeling dataset>
           NAMELEN = 32
          DESCENDING ;
MODEL <dependent> = <regressors>
    /     SELECTION = <selection method>
          SLE           = <SLE criterion>
          SLS           = <SLS criterion>
          CTABLE ;
RUN ;
```

Specify name of modeling dataset for regression

This option does not let variable name length get truncated to 20

This option reverses the sorting order for the levels of dependent variable

Specify variable selection method

Specify significance level of entry and stay

This option displays classification table

- Classification table (generated by `CTABLE` option) provides true positives, false positives, true negatives and false negatives at varied levels of probability z
- An observation is predicted as event if the predicted event probability exceeds z

Classification table generated as a part of SAS output can be used to identify the probability cut-off point for classification decision

Illustrative SAS Output

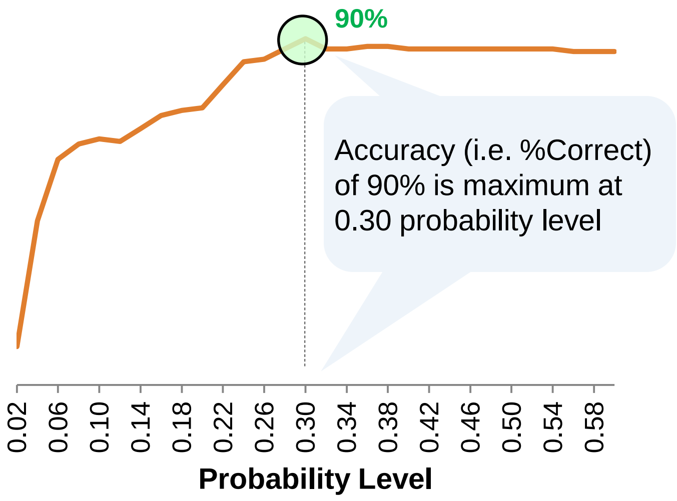
$$\mathit{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

0.30 may be used as the cut-off probability level for assigning classes

- If probability > 0.30, predicted class = 1
- If probability ≤ 0.30, predicted class = 0

Such classification yields 90% accuracy

Accuracy



Accuracy (i.e. %Correct) of 90% is maximum at 0.30 probability level

TP TN FP FN

```
Prob.          Correct           Incorrect        Percentage
Level     Event   Non-Event   Event   Non-Event    Correct
 0.02       15        0        135        0          10.0
 0.04       14       50         85        1          42.7
 0.06       13       75         60        2          58.7
 0.08       12       82         53        3          62.7
 0.10       11       85         50        4          64.0
 0.12       10       85         50        5          63.3
 0.14       10       90         45        5          66.7
 0.16       10       95         40        5          70.0
 0.18        9       98         37        6          71.3
 0.20        8      100         35        7          72.0
 0.22        7      110         25        8          78.0
 0.24        7      119         16        8          84.0
 0.26        7      120         15        8          84.7
 0.28        6      125         10        9          87.3
 0.30        6      129          6        9          90.0
 0.32        2      129          6       13          87.3
 0.34        2      129          6       13          87.3
 0.36        2      130          5       13          88.0
 0.38        2      130          5       13          88.0
 0.40        1      130          5       14          87.3
 0.42        1      130          5       14          87.3
 0.44        1      130          5       14          87.3
 0.46        1      130          5       14          87.3
 0.48        1      130          5       14          87.3
 0.50        1      130          5       14          87.3
 0.52        1      130          5       14          87.3
 0.54        1      130          5       14          87.3
 0.56        0      130          5       15          86.7
 0.58        0      130          5       15          86.7
 0.60        0      130          5       15          86.7
```

2.1.2. Concordance and Discordance

Concordant

- A pair of an event and a non-event is said to be a *concordant pair* if the event observation has higher predicted event probability than the non-event observation

Example:

TARGET	PREDICTION
0	0.90
1	0.95

Discordant

- A pair of an event and a non-event is said to be a *discordant pair* if the event observation has lower predicted event probability than the non-event observation

Example:

TARGET	PREDICTION
0	0.90
1	0.85

Tied

- A pair of an event and a non-event is said to be a *tied pair* if the predicted event probability for both the event and the non-event observations is exactly same

Example:

TARGET	PREDICTION
0	0.90
1	0.90

Illustration

Given Data

ID	TARGET	PREDICTION
1	0	0.36
2	0	0.87
3	0	0.42
4	0	0.13
5	0	0.10
6	1	0.40
7	1	0.87
8	1	0.83

Number of Events : 3

Number of Non-Events : 5

Number of Distinct Pairs of Events and Non-Events

= #Events x #Non-Events

= 3 x 5

= 15

PAIR	ID	TARGET	PREDICTION	RESULT
1	1	0	0.36	Concordant
	6	1	0.40	
2	1	0	0.36	Concordant
	7	1	0.87	
3	1	0	0.36	Concordant
	8	1	0.83	
4	2	0	0.87	Discordant
	6	1	0.40	
5	2	0	0.87	Tied
	7	1	0.87	
6	2	0	0.87	Discordant
	8	1	0.83	
7	3	0	0.42	Discordant
	6	1	0.40	
8	3	0	0.42	Concordant
	7	1	0.87	
9	3	0	0.42	Concordant
	8	1	0.83	
10	4	0	0.13	Concordant
	6	1	0.40	
11	4	0	0.13	Concordant
	7	1	0.87	
12	4	0	0.13	Concordant
	8	1	0.83	
13	5	0	0.10	Concordant
	6	1	0.40	
14	5	0	0.10	Concordant
	7	1	0.87	
15	5	0	0.10	Concordant
	8	1	0.83	

# Pairs = 15

#Concordant Pairs = 11

#Discordant Pairs = 3

#Tied Pairs = 1

Percent Concordance

= 11/15 = **73.3**

Percent Discordance

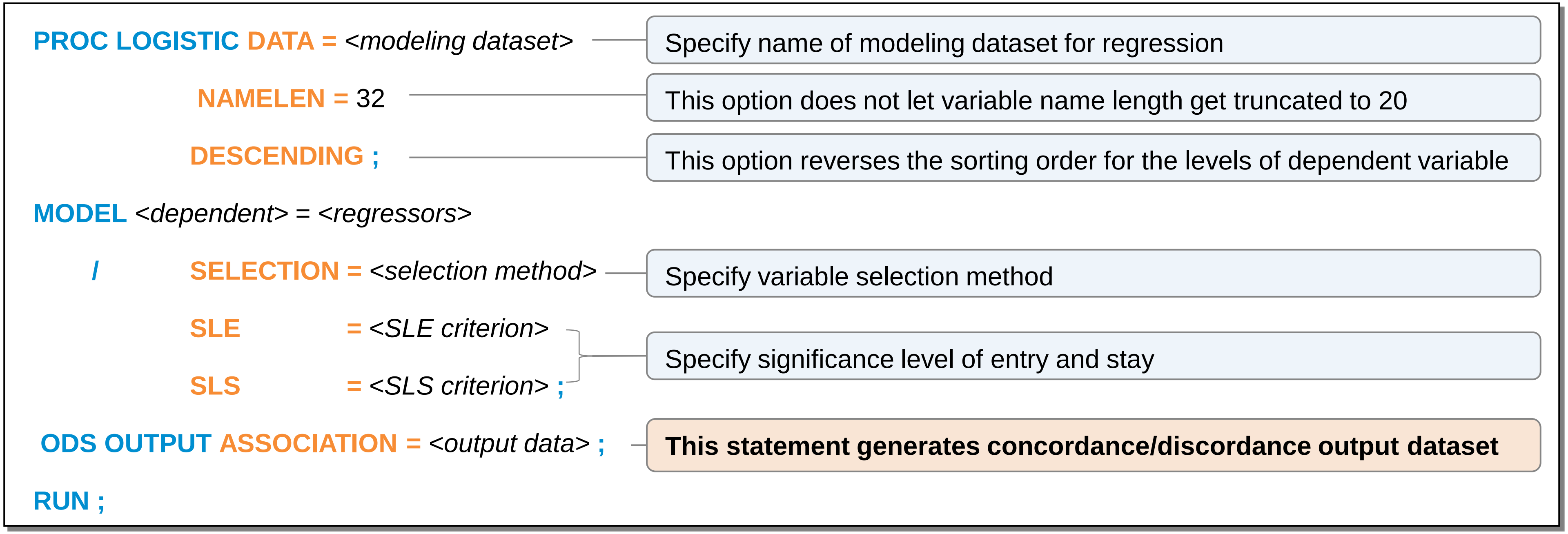
= 3/15 = **20.0**

Percent Tied

= 1/15 = **6.7**

SAS Implementation

Below is the syntax for computing concordance and discordance metrics



```
PROC LOGISTIC DATA = <modeling dataset>
               NAMELEN = 32
              DESCENDING ;
MODEL <dependent> = <regressors>
   /          SELECTION = <selection method>
              SLE          = <SLE criterion>
              SLS          = <SLS criterion> ;
 ODS OUTPUT ASSOCIATION = <output data> ;
RUN ;
```

Specify name of modeling dataset for regression

This option does not let variable name length get truncated to 20

This option reverses the sorting order for the levels of dependent variable

Specify variable selection method

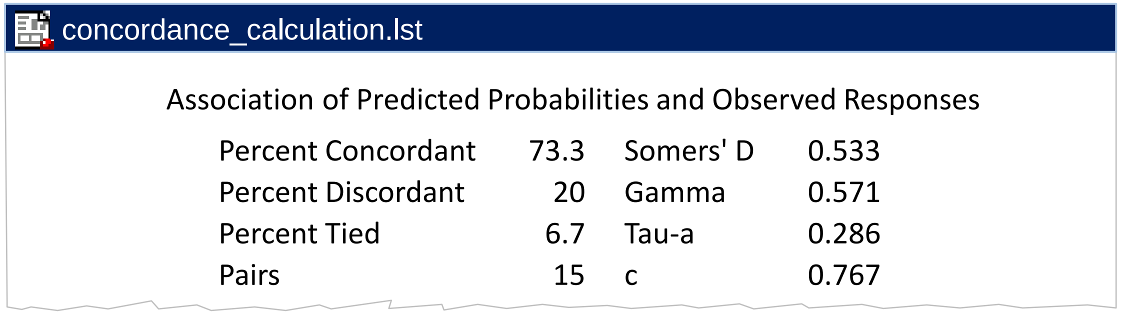
Specify significance level of entry and stay

This statement generates concordance/discordance output dataset

- In addition to percent concordance, percent discordance and percent tied, the `ASSOCIATION` table reports four more metrics:
  - Somer's D
  - Goodman-Kruskal Gamma
  - Kendall's Tau-a
  - c

Illustrative SAS Output

LST File

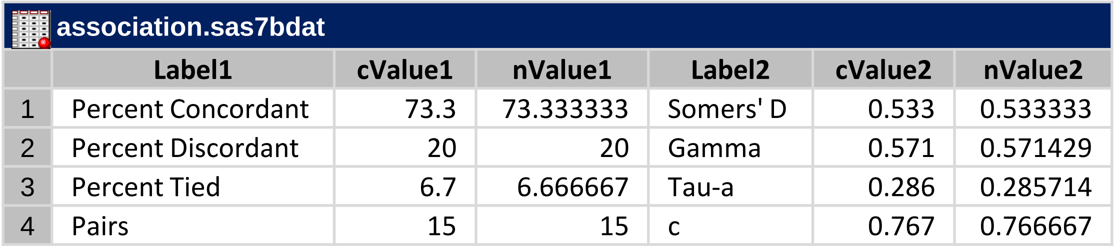


concordance_calculation.lst

```
Association of Predicted Probabilities and Observed Responses

    Percent Concordant     73.3    Somers' D    0.533
    Percent Discordant       20    Gamma        0.571
    Percent Tied            6.7    Tau-a        0.286
    Pairs                    15    c            0.767
```

SAS Dataset



association.sas7bdat

	Label1	cValue1	nValue1	Label2	cValue2	nValue2
1	Percent Concordant	73.3	73.333333	Somers' D	0.533	0.533333
2	Percent Discordant	20	20	Gamma	0.571	0.571429
3	Percent Tied	6.7	6.666667	Tau-a	0.286	0.285714
4	Pairs	15	15	c	0.767	0.766667

$$Somer's\ D = \frac{n_C - n_D}{n_P}$$

Gini Coefficient

$$Gamma = \frac{n_C - n_D}{n_C + n_D}$$

$$Tau - a = \frac{n_C - n_D}{0.5N(N-1)}$$

$$c = \frac{n_C + 0.5n_T}{n_P}$$

Area under the Curve (AUC)

where

$$N = \#observations\ in\ dataset$$

$$n_C = \#\ concordant\ pairs$$

$$n_D = \#\ discordant\ pairs$$

$$n_T = \#\ tied\ pairs$$

$$n_P = total\ \#\ pairs$$

$$i.e.\ n_P = n_C + n_D + n_T$$

Guidelines / Thumb Rules

Percent Concordance	Interpretation
< 70	Poor Discrimination
70-80	Acceptable Discrimination
80-90	Good Discrimination
> 90	Excellent Discrimination

Higher percent concordance indicates better good-bad discrimination power

2.1.3. Receiver Operating Characteristics (ROC)

ROC graph is a 2-dimensional graph in which

- True positive rate is plotted on the Y-axis
- False positive rate is plotted on the X-axis

True Positive Rate (or Sensitivity)

$$\text{True Positive Rate} = \frac{\#Events\ correctly\ classified\ as\ Event}{\#Events} = \frac{TP}{TP + FN}$$

False Positive Rate (or 1 - Specificity)

This image cannot currently be displayed.

ROC Space

Perfect Classification
FP Rate = 0, TP Rate = 1

⇒ FP = FN = 0

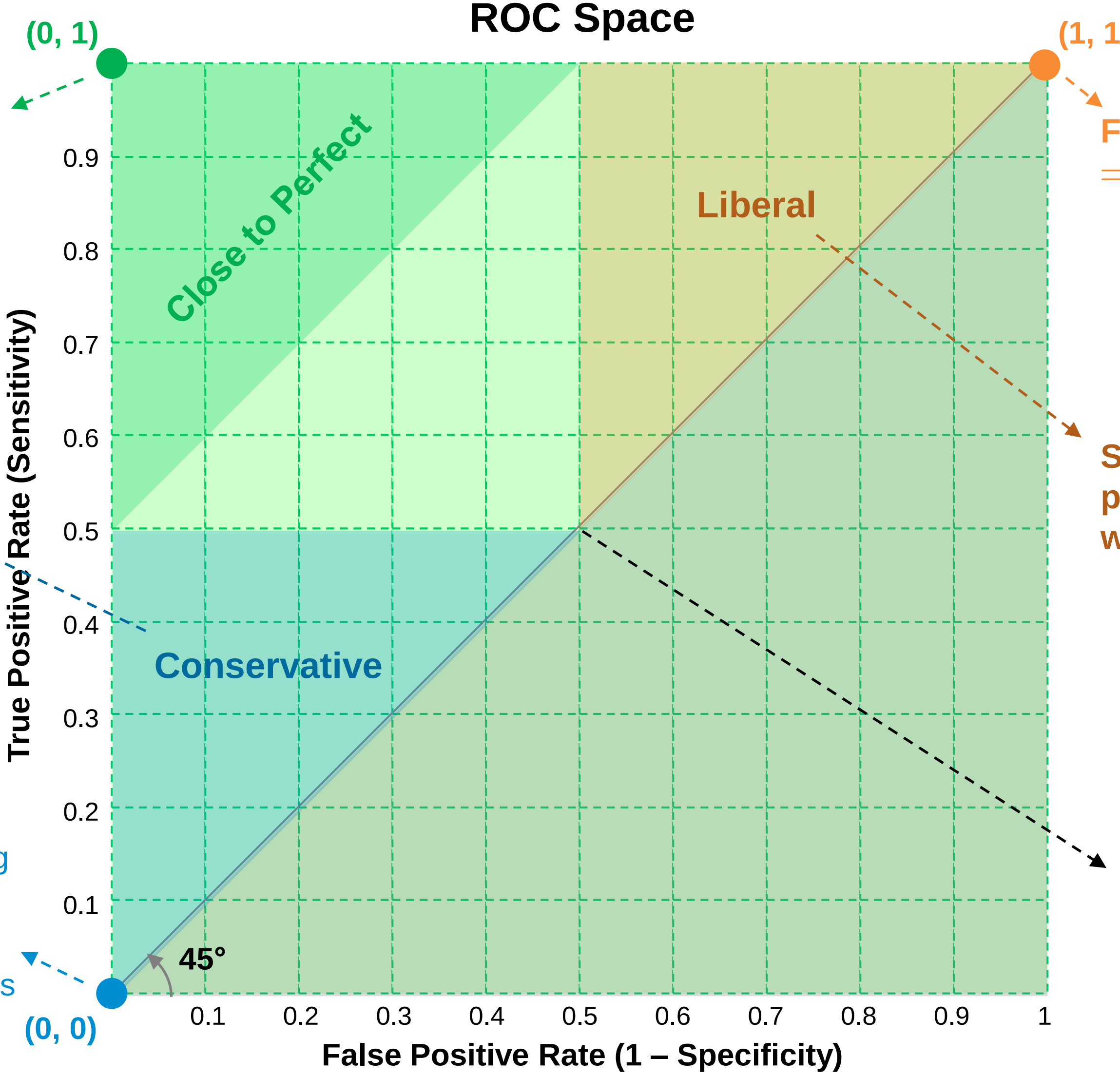
⇒ TN = #Non-Events and
TP = #Events

Such classifiers make positive classifications only with strong evidence

- They make few false positive errors
- But they often have low true positive rates

FP Rate = TP Rate = 0
⇒ FP = TP = 0

- Strategy of never issuing a positive classification
- No false positive errors
- No gains of true positives



FP Rate = TP Rate = 1
⇒ TN = FN = 0

- Strategy of never issuing a negative classification
- No false negative errors
- No gains of true negatives

Such classifiers make positive classifications with weak evidence

- They classify nearly all positives (events) correctly
- But they often have high false positive rates

45° random line

- Random guess
- 50% of total area lies under random line
  i.e. AUC = 0.5

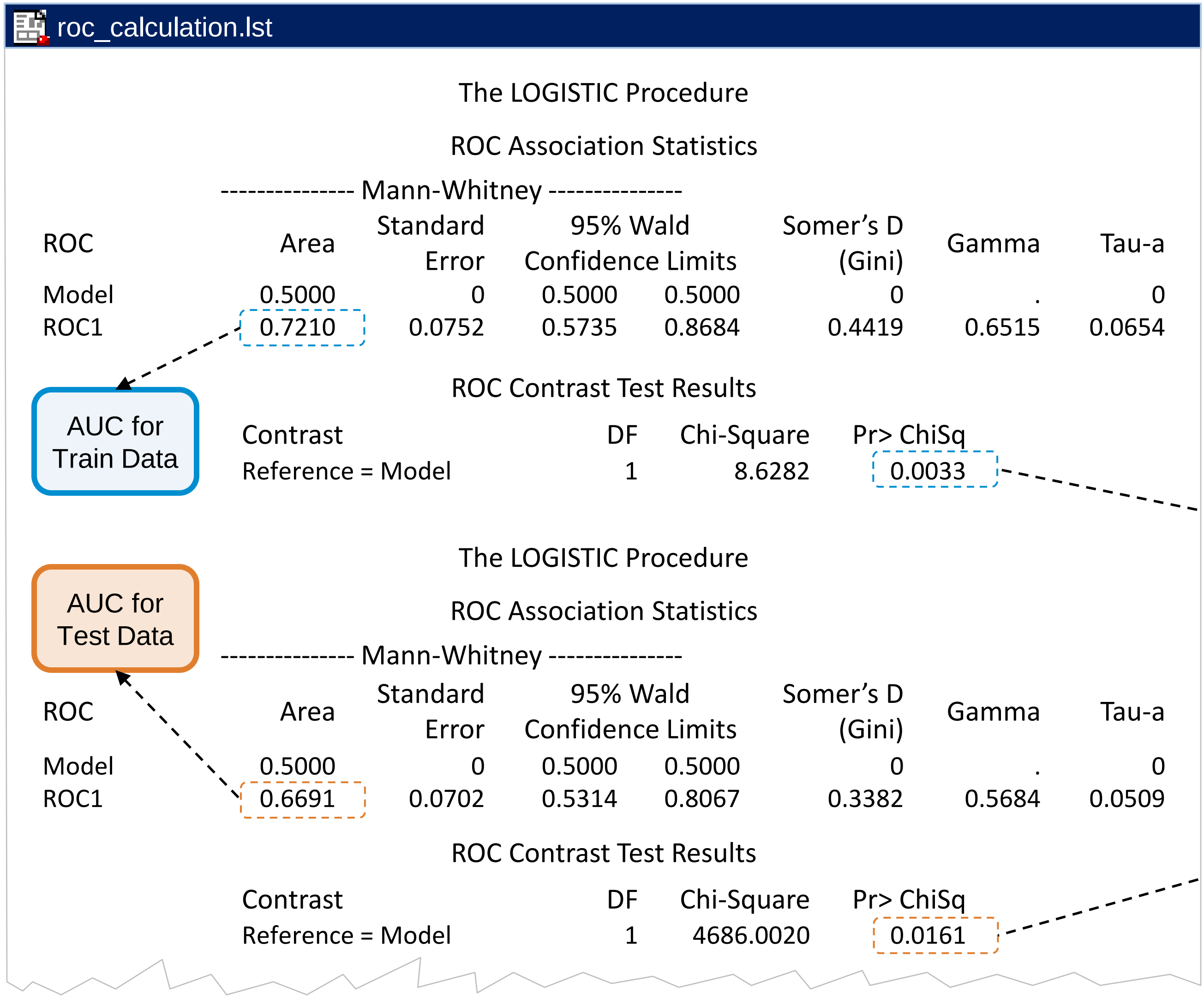
SAS Implementation

PROC LOGISTIC DATA = *<train dataset>*	Specify name of modeling dataset for regression
NAMELEN = 32	This option does not let variable name length get truncated to 20
DESCENDING ;	This option reverses the sorting order for the levels of dependent variable
MODEL *<dependent>* = *<regressors>*	
/ SELECTION = *<selection method>*	Specify variable selection method

SLE = *<SLE criterion>*
SLS = *<SLS criterion>*
	Specify significance level of entry and stay

OUTROC = *<train ROC dataset>* ;	**This option creates ROC output dataset for train data; To be used to plot ROC graph**
OUTPUT OUT = *<train predictions>*	This option generates train scored dataset
P = P_1 ;	This option requests for score variable name. Specify P_1 to denote probability of event
SCORE DATA = *<test dataset>*	This option requests for name of test dataset as input
OUT = <test predictions>	This option generates test scored dataset
OUTROC = *<test ROC dataset>* ;	**This option creates ROC output dataset for test data; To be used to plot ROC graph**
RUN ;	
PROC LOGISTIC DATA = *<train predictions>* DESCENDING ;	
MODEL *<dependent>* = ;	Specify only dependent variable. Do not specify regressors
ROC PRED = P_1 ;	Specify P_1 as score variable name
ROCCONTRAST ;	**This option compares Random AUC (0.5) with train AUC and checks significance**
RUN ;	
PROC LOGISTIC DATA = *<test predictions>* DESCENDING ;	
MODEL *<dependent>* = ;	Specify only dependent variable. Do not specify regressors
ROC PRED = P_1 ;	Specify P_1 as score variable name
ROCCONTRAST ;	**This option compares Random AUC (0.5) with test AUC and checks significance**
RUN ;	

Illustrative SAS Output (Output generated due to `ROC PRED=` and `ROCCONTRAST` options)



roc_calculation.lst

```
                              The LOGISTIC Procedure

                             ROC Association Statistics

            --------------- Mann-Whitney ---------------
                       Standard       95% Wald          Somer's D
ROC           Area        Error   Confidence Limits       (Gini)     Gamma     Tau-a

Model       0.5000            0   0.5000     0.5000            0         .         0
ROC1        0.7210       0.0752   0.5735     0.8684       0.4419    0.6515    0.0654

                             ROC Contrast Test Results

            Contrast                 DF    Chi-Square    Pr> ChiSq
            Reference = Model         1        8.6282       0.0033

                              The LOGISTIC Procedure

                             ROC Association Statistics

            --------------- Mann-Whitney ---------------
                       Standard       95% Wald          Somer's D
ROC           Area        Error   Confidence Limits       (Gini)     Gamma     Tau-a

Model       0.5000            0   0.5000     0.5000            0         .         0
ROC1        0.6691       0.0702   0.5314     0.8067       0.3382    0.5684    0.0509

                             ROC Contrast Test Results

            Contrast                 DF    Chi-Square    Pr> ChiSq
            Reference = Model         1     4686.0020       0.0161
```

AUC for Train Data

AUC for Test Data

Guidelines for Assessment

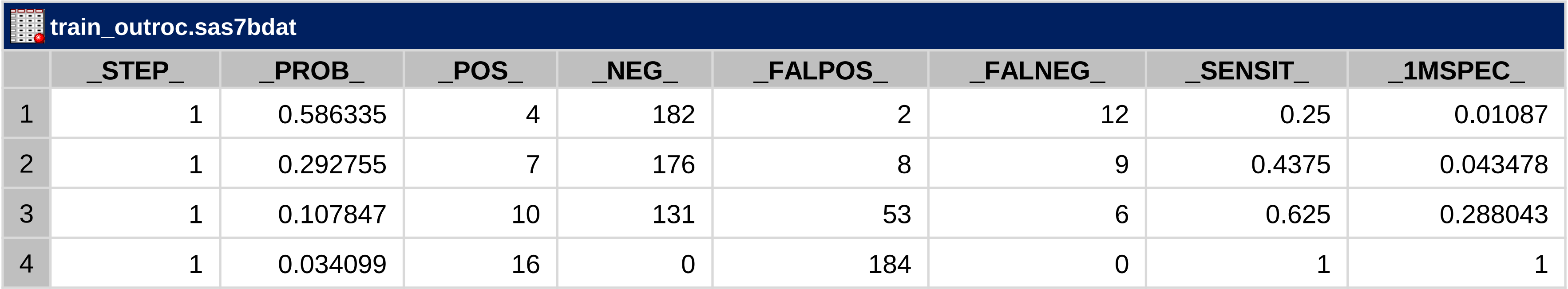
AUC	Classification
0.5	No Discrimination
0.6-0.7	Poor
0.7-0.8	Acceptable
0.8-0.9	Good
> 0.9	Excellent

p-value is quite low (<0.05) and therefore **train data's AUC** is significantly different from 0.5 benchmark (AUC from random guessing)

p-value is quite low (<0.05) and therefore **test data's AUC** is significantly different from 0.5 benchmark (AUC from random guessing)

Illustrative SAS Output (Output generated due to OUTROC=*<train ROC dataset>* option)

Train ROC Dataset

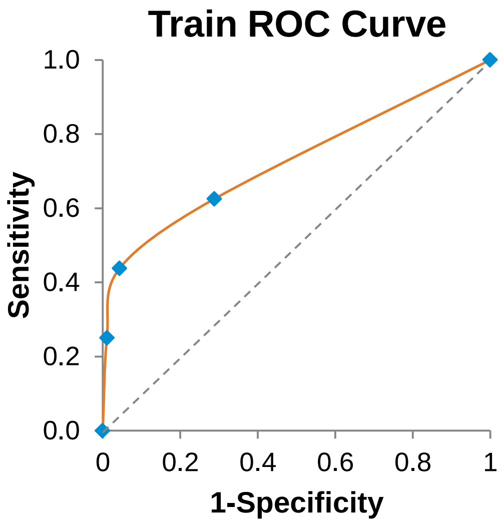


train_outroc.sas7bdat

	STEP	_PROB_	_POS_	_NEG_	_FALPOS_	_FALNEG_	_SENSIT_	_1MSPEC_
1	1	0.586335	4	182	2	12	0.25	0.01087
2	1	0.292755	7	176	8	9	0.4375	0.043478
3	1	0.107847	10	131	53	6	0.625	0.288043
4	1	0.034099	16	0	184	0	1	1

Variable Description

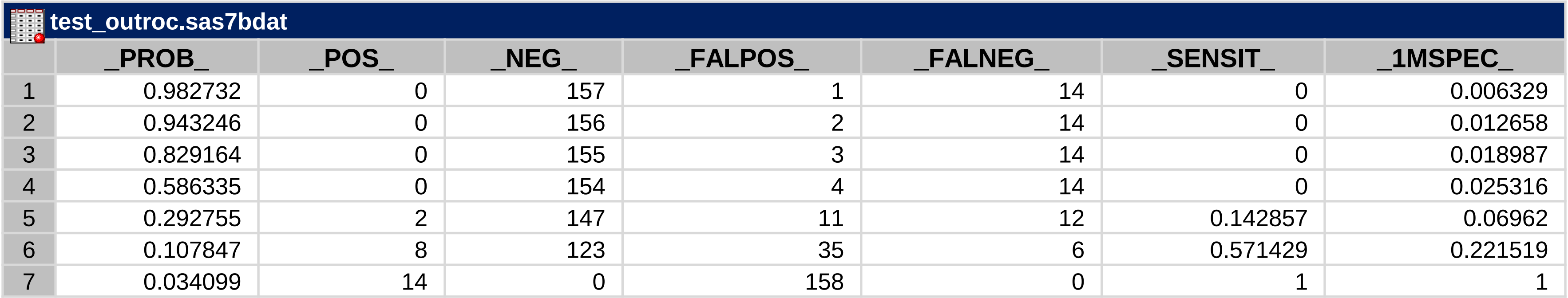
Variable	Meaning
STEP	Model Building Step
PROB	Cut-off Probability Level for Assigning Classes
POS	No. of Correctly Predicted Events
NEG	No. of Correctly Predicted Nonevents
FALPOS	No. of Nonevents Predicted as Events
FALNEG	No. of Events Predicted as Nonevents
SENSIT	Sensitivity
1MSPEC	1 - Specificity



Data used to plot ROC Curve

Illustrative SAS Output (Output generated due to OUTROC=<test ROC dataset> option)

Test ROC Dataset

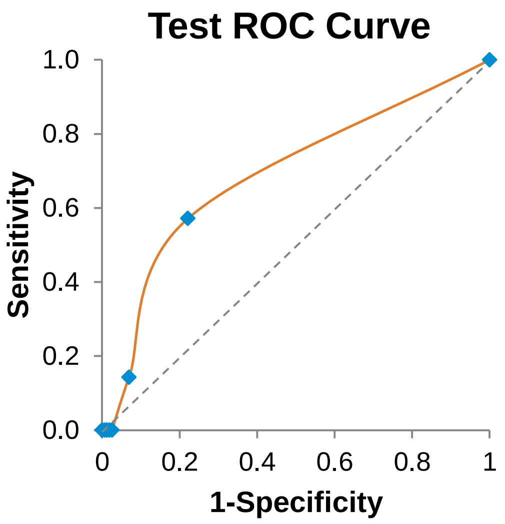


test_outroc.sas7bdat

	PROB	_POS_	_NEG_	_FALPOS_	_FALNEG_	_SENSIT_	_1MSPEC_
1	0.982732	0	157	1	14	0	0.006329
2	0.943246	0	156	2	14	0	0.012658
3	0.829164	0	155	3	14	0	0.018987
4	0.586335	0	154	4	14	0	0.025316
5	0.292755	2	147	11	12	0.142857	0.06962
6	0.107847	8	123	35	6	0.571429	0.221519
7	0.034099	14	0	158	0	1	1

Variable Description

Variable	Meaning
PROB	Cut-off Probability Level for Assigning Classes
POS	No. of Correctly Predicted Events
NEG	No. of Correctly Predicted Nonevents
FALPOS	No. of Nonevents Predicted as Events
FALNEG	No. of Events Predicted as Nonevents
SENSIT	Sensitivity
1MSPEC	1 - Specificity



Data used to plot ROC Curve

Illustration: Manual Computation of Area Under the Curve (AUC) from ROC Data Points

Train AUC Calculation

	A	B	C	D	E	F	G	H
1	**_PROB_**	**_SENSIT_**	**_1MSPEC_**	**LAG_SENSIT_**	**LAG_1MSPEC_**	**(B) + (D)**	**(C) – (E)**	**0.5 x (F) x (G)**
2	0.5863	0.2500	0.0109	0.0000	0.0000	0.2500	0.0109	0.0014
3	0.2928	0.4375	0.0435	0.2500	0.0109	0.6875	0.0326	0.0112
4	0.1078	0.6250	0.2880	0.4375	0.0435	1.0625	0.2446	0.1299
5	0.0341	1.0000	1.0000	0.6250	0.2880	1.6250	0.7120	0.5785
6								**AUC = Σ(H) = 0.7210**

Data from TRAIN_OUTROC dataset

AUC for Train Data

Test AUC Calculation

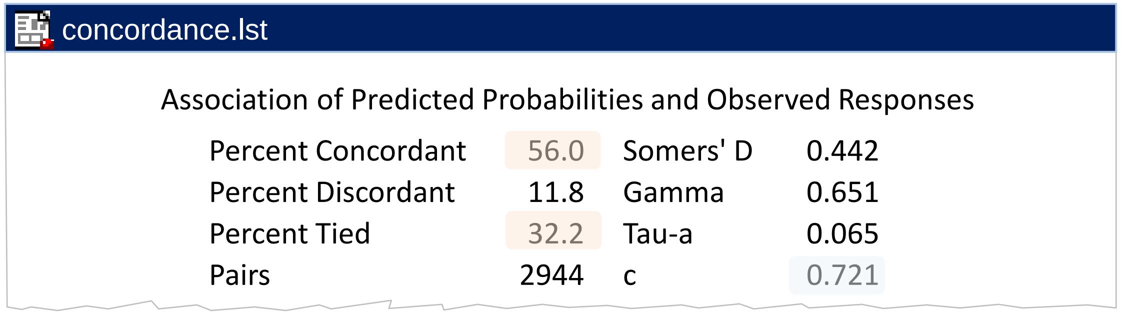
	A	B	C	D	E	F	G	H
1	**_PROB_**	**_SENSIT_**	**_1MSPEC_**	**LAG_SENSIT_**	**LAG_1MSPEC_**	**(B) + (D)**	**(C) – (E)**	**0.5 x (F) x (G)**
2	0.9827	0.0000	0.0063	0.0000	0.0000	0.0000	0.0063	0.0000
3	0.9432	0.0000	0.0127	0.0000	0.0063	0.0000	0.0063	0.0000
4	0.8292	0.0000	0.0190	0.0000	0.0127	0.0000	0.0063	0.0000
5	0.5863	0.0000	0.0253	0.0000	0.0190	0.0000	0.0063	0.0000
6	0.2928	0.1429	0.0696	0.0000	0.0253	0.1429	0.0443	0.0032
7	0.1078	0.5714	0.2215	0.1429	0.0696	0.7143	0.1519	0.0542
8	0.0341	1.0000	1.0000	0.5714	0.2215	1.5714	0.7785	0.6117
9								**AUC = Σ(H) = 0.6691**

Data from TEST_OUTROC dataset

AUC for Test Data

Illustration: Train AUC from SAS Output

LST File



concordance.lst

```
Association of Predicted Probabilities and Observed Responses

Percent Concordant     56.0    Somers' D    0.442
Percent Discordant     11.8    Gamma        0.651
Percent Tied           32.2    Tau-a        0.065
Pairs                  2944    c            0.721
```

Train AUC Calculation

Method 1	AUC = %Concordant + 0.5 (%Tied) = 56.0% + 0.5(32.2%) = 72.1% = 0.721
Method 2	AUC = c = 0.721

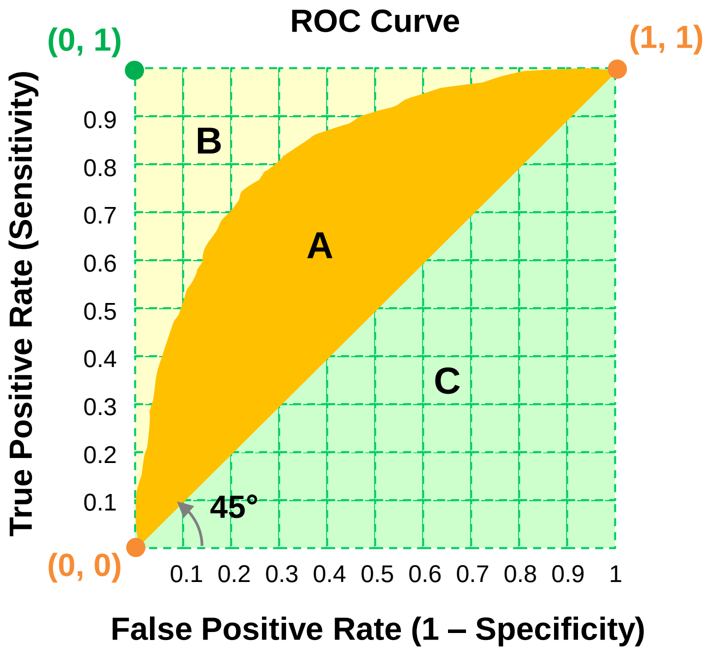
2.1.4. Gini Coefficient

Gini coefficient is a measure of degree of discrimination between goods (non-events) and bads (events)

- Gini coefficient is twice the area between ROC curve and 45° random line of equality
- Gini coefficient varies between 0 and 1
  - Gini = 0 implies no discrimination
  - Gini = 1 implies perfect discrimination

Relation between Gini and AUC

This image cannot currently be displayed.



Relation between Gini and Concordance

Two important points:

- Gini is simply the difference between concordance and discordance
- Gini is equivalent to Somer's D

$$
\begin{aligned}
Gini &= 2AUC - 1 \\
&= 2\left(\frac{n_C + 0.5n_T}{n_P}\right) - 1 \\
&= \frac{2n_C + n_T - n_P}{n_P} \\
&= \frac{2n_C + n_T - (n_C + n_D + n_T)}{n_P} \\
&= \frac{n_C - n_D}{n_P} \\
&= Somer's\ D
\end{aligned}
$$

Recall from Section 2.1.2

$$Somer's\ D = \frac{n_C - n_D}{n_P}$$

$$c\ (\text{i.e. } AUC) = \frac{n_C + 0.5n_T}{n_P}$$

where

$$n_C = \#\ concordant\ pairs$$

$$n_D = \#\ discordant\ pairs$$

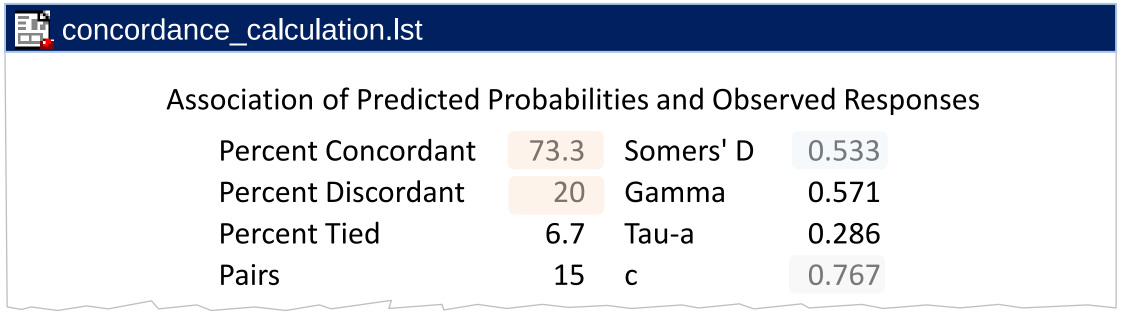
$$n_T = \#\ tied\ pairs$$

$$n_P = total\ \#\ pairs$$

$$i.e.\ n_P = n_C + n_D + n_T$$

Illustration: Gini from SAS Output

LST File (Illustration from Section 2.1.2)



concordance_calculation.lst

```
Association of Predicted Probabilities and Observed Responses

  Percent Concordant     73.3    Somers' D     0.533
  Percent Discordant       20    Gamma         0.571
  Percent Tied            6.7    Tau-a         0.286
  Pairs                    15    c             0.767
```

Gini Calculation

Method 1	Gini = Concordance – Discordance = 73.3% - 20% = 53.3% = 0.533
Method 2	Gini = Somer’s D = 0.533
Method 3	Gini = 2AUC - 1 = 2(0.767) – 1 = 0.533

2.1.5. Cumulative Lift Chart

Cumulative lift chart (also known as cumulative gains chart) is a widely used measure of model's effectiveness in capturing bads (events) by rank-ordering of population based on model's score (predictions)

- Lift is not a single value for overall model. It is calculated at bin level. The bins may be:
  - Deciles (i.e. 10 equal-sized bins); or
  - Demi-Deciles (i.e. 20 equal-sized bins); or
  - Percentiles (i.e. 100 equal-sized bins)
- Lift is computed after rank-ordering of records based on model's score. Scale of score does not matter
- Model performance is generally assessed by examining cumulative lift at top 1, 2 or 3 deciles

Steps for Cumulative Lift Calculation

Step 1 Sort data by predicted value (i.e. model's score) in descending order, given that focus class is TARGET = 1

Step 2 Divide data into 10, 20 or 100 equal sized bins

Step 3 Summarize data at bin level and compute bin population, #events and #non-events for each bin

Step 4 For each bin, calculate bin lift as ratio of #events captured in the bin to total #events in the dataset

Step 5 Calculate cumulative lift as %cumulative events captured at bin level

Step 6 Plot cumulative lift chart with '*%Cumulative Population*' on X-axis and '*%Cumulative Events Captured*' on Y-axis

Illustration: Customer Attrition (Target Variable: IND_ATTR)

Train Dataset

train.sas7bdat

	CUST_ID	IND_ATTR	PRED
1	X00001	0	0.0062
2	X00004	0	0.0084
	<Rows Deleted>		
4000	X08145	0	0.0235
4001	X08147	1	0.0643
	<Rows Deleted>		
19877	X40001	0	0.0463
19878	X40003	0	0.0044
19879	X40004	0	0.0810

Step 1

train_sort.sas7bdat

	CUST_ID	IND_ATTR	PRED
1	X14638	1	0.1663
2	X12184	0	0.1546
	<Rows Deleted>		
4000	X00696	0	0.0266
4001	X01066	1	0.0245
	<Rows Deleted>		
19877	X11431	0	0.0009
19878	X18221	0	0.0005
19879	X00940	0	0.0002

Step 2

train_sort_bin.sas7bdat

	CUST_ID	IND_ATTR	PRED	BIN
1	X14638	1	0.1663	1
2	X12184	0	0.1546	1
	<Rows Deleted>			
4000	X00696	0	0.0266	3
4001	X01066	1	0.0245	3
	<Rows Deleted>			
19877	X11431	0	0.0009	10
19878	X18221	0	0.0005	10
19879	X00940	0	0.0002	10

Step 3

train_bin_summary.sas7bdat

	BIN	OBS	BADS	GOODS
1	1	1987	134	1853
2	2	1988	71	1917
3	3	1988	39	1949
4	4	1988	41	1947
5	5	1988	24	1964
6	6	1988	19	1969
7	7	1988	14	1974
8	8	1988	14	1974
9	9	1988	7	1981
10	10	1988	6	1982

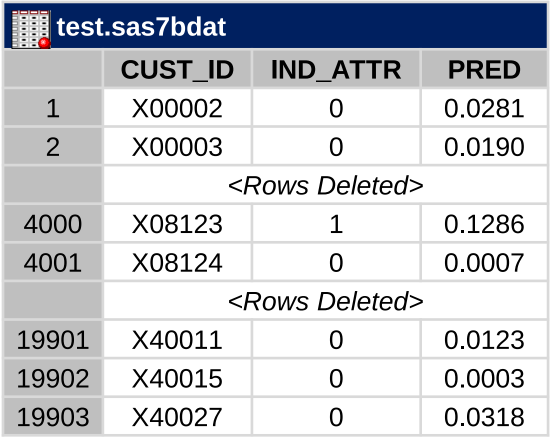
Step 4

Step 5

	A	B	C	D	E	F
1	**BIN**	**OBS**	**BADS**	**GOODS**	**BIN_LIFT = (C) ÷ Σ(C)**	**CUM_LIFT**
2	1	1,987	134	1,853	36.3%	36.3%
3	2	1,988	71	1,917	19.2%	55.6%
4	3	1,988	39	1,949	10.6%	66.1%
5	4	1,988	41	1,947	11.1%	77.2%
6	5	1,988	24	1,964	6.5%	83.7%
7	6	1,988	19	1,969	5.1%	88.9%
8	7	1,988	14	1,974	3.8%	92.7%
9	8	1,988	14	1,974	3.8%	96.5%
10	9	1,988	7	1,981	1.9%	98.4%
11	10	1,988	6	1,982	1.6%	100.0%
12		**Σ(B) = 19,879**	**Σ(C) = 369**	**Σ(D) = 19,510**	**Σ(E) = 100%**	

Illustration: Customer Attrition (Target Variable: IND_ATTR) Continued . . .

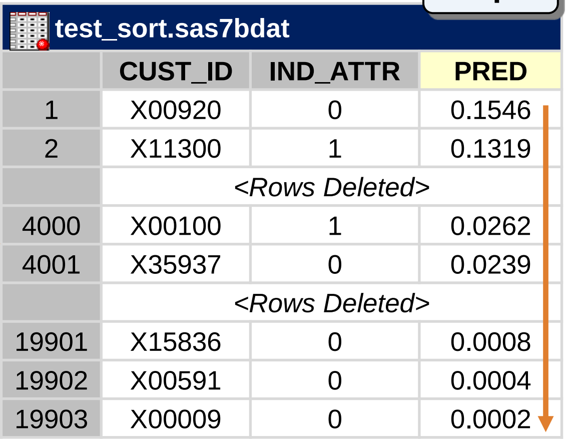
Test Dataset



test.sas7bdat

	CUST_ID	IND_ATTR	PRED
1	X00002	0	0.0281
2	X00003	0	0.0190
	<Rows Deleted>		
4000	X08123	1	0.1286
4001	X08124	0	0.0007
	<Rows Deleted>		
19901	X40011	0	0.0123
19902	X40015	0	0.0003
19903	X40027	0	0.0318

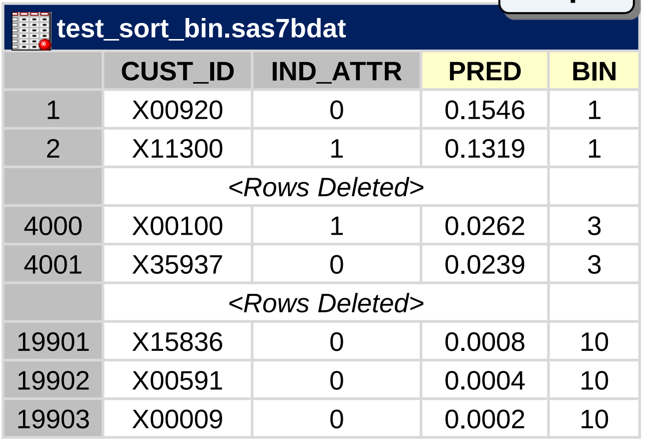
Step 1



test_sort.sas7bdat

	CUST_ID	IND_ATTR	PRED
1	X00920	0	0.1546
2	X11300	1	0.1319
	<Rows Deleted>		
4000	X00100	1	0.0262
4001	X35937	0	0.0239
	<Rows Deleted>		
19901	X15836	0	0.0008
19902	X00591	0	0.0004
19903	X00009	0	0.0002

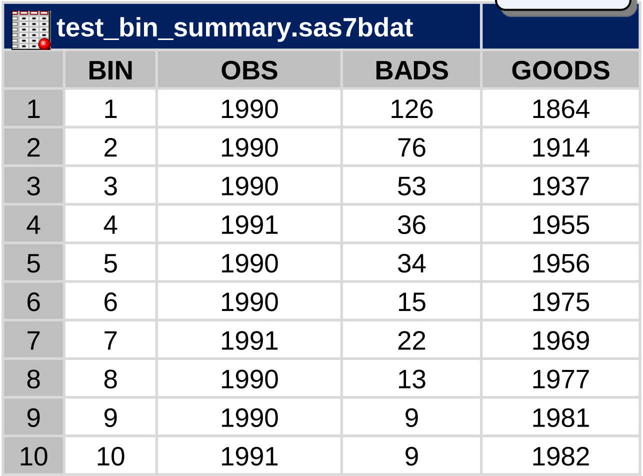
Step 2



test_sort_bin.sas7bdat

	CUST_ID	IND_ATTR	PRED	BIN
1	X00920	0	0.1546	1
2	X11300	1	0.1319	1
	<Rows Deleted>			
4000	X00100	1	0.0262	3
4001	X35937	0	0.0239	3
	<Rows Deleted>			
19901	X15836	0	0.0008	10
19902	X00591	0	0.0004	10
19903	X00009	0	0.0002	10

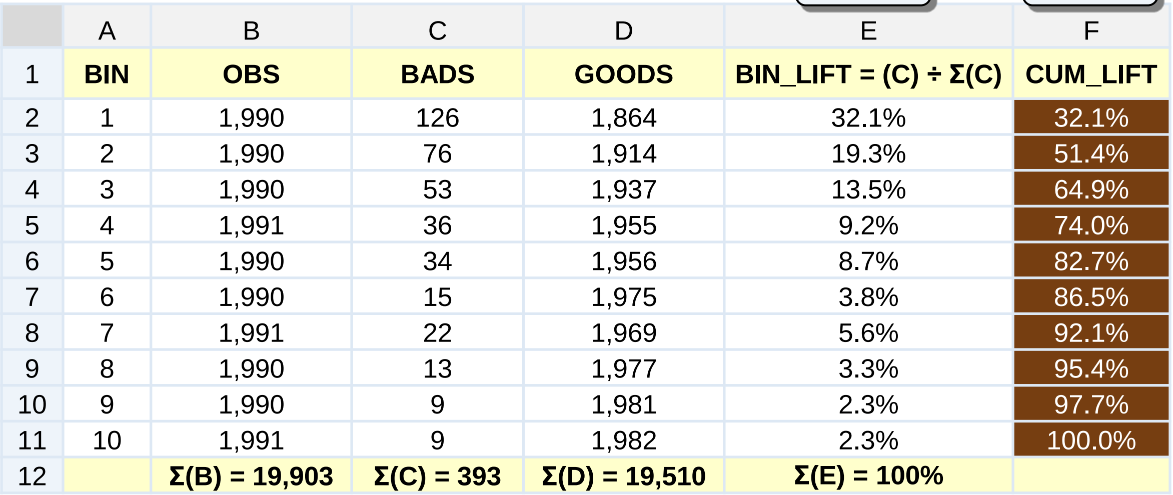
Step 3



test_bin_summary.sas7bdat

	BIN	OBS	BADS	GOODS
1	1	1990	126	1864
2	2	1990	76	1914
3	3	1990	53	1937
4	4	1991	36	1955
5	5	1990	34	1956
6	6	1990	15	1975
7	7	1991	22	1969
8	8	1990	13	1977
9	9	1990	9	1981
10	10	1991	9	1982

Step 4 **Step 5**



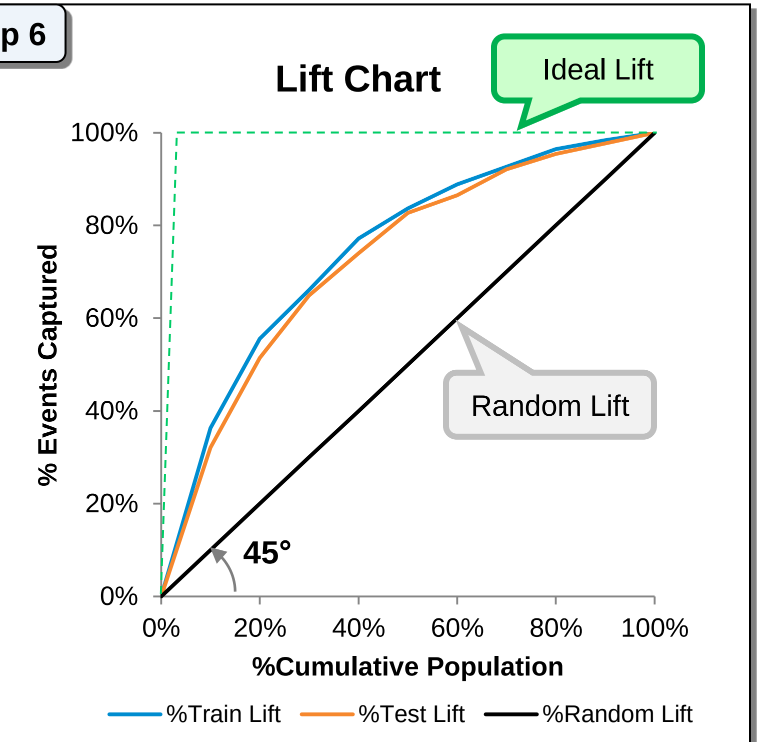
	A	B	C	D	E	F
1	**BIN**	**OBS**	**BADS**	**GOODS**	**BIN_LIFT = (C) ÷ Σ(C)**	**CUM_LIFT**
2	1	1,990	126	1,864	32.1%	32.1%
3	2	1,990	76	1,914	19.3%	51.4%
4	3	1,990	53	1,937	13.5%	64.9%
5	4	1,991	36	1,955	9.2%	74.0%
6	5	1,990	34	1,956	8.7%	82.7%
7	6	1,990	15	1,975	3.8%	86.5%
8	7	1,991	22	1,969	5.6%	92.1%
9	8	1,990	13	1,977	3.3%	95.4%
10	9	1,990	9	1,981	2.3%	97.7%
11	10	1,991	9	1,982	2.3%	100.0%
12		**Σ(B) = 19,903**	**Σ(C) = 393**	**Σ(D) = 19,510**	**Σ(E) = 100%**	

Illustration: Customer Attrition (Target Variable: IND_ATTR) Continued . . .

- **Ideal Lift:** Model is able to rank order all events above non-events. At Event Rate, Ideal Lift = 100%
- **Random Lift:** At X% population, X% events are captured by random guessing. Random Lift Curve is 45° line

	A	B	C	D
1	**Bin**	**%Cumulative Population**	**%Train Cumulative Lift**	**%Test Cumulative Lift**
2	1	10%	36.3%	32.1%
3	2	20%	55.6%	51.4%
4	3	30%	66.1%	64.9%
5	4	40%	77.2%	74.0%
6	5	50%	83.7%	82.7%
7	6	60%	88.9%	86.5%
8	7	70%	92.7%	92.1%
9	8	80%	96.5%	95.4%
10	9	90%	98.4%	97.7%
11	10	100%	100.0%	100.0%

Step 6



Interpretation (based on test dataset results): Any incentive strategy devised for **top 20% customers** (3,980 out of 19,903 customers) is expected to capture **more than 50% attrition cases** (202 out of 393 attrition cases)

2.1.6. Kolmogorov-Smirnov (K-S) Statistic

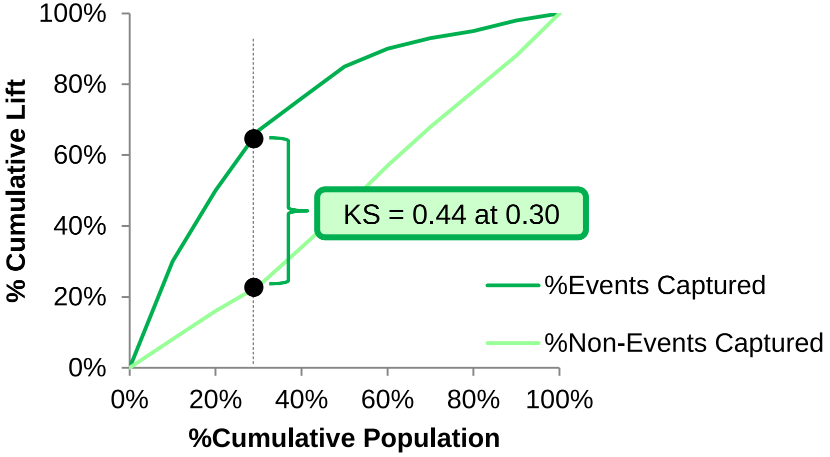
Meaning

- K-S statistic is the maximum vertical difference between the cumulative lift curve for events (goods) and the cumulative lift curve for non-events (bads)

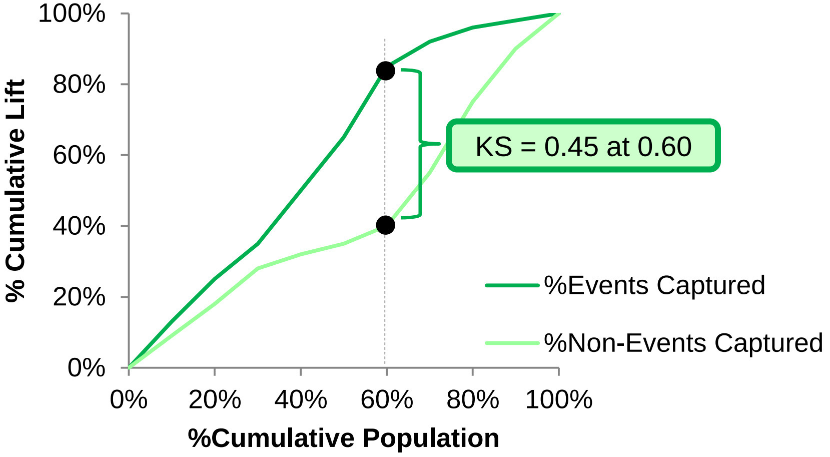
Word of Caution

- K-S is based on a single point on the good and bad distributions – the point where the cumulative distributions are the most different. It shouldn't be relied upon without carefully looking at the distributions

Model 1



Model 2



✓

Acceptable Model

For a reasonable model, KS value (maximum difference) should be attained within top few deciles

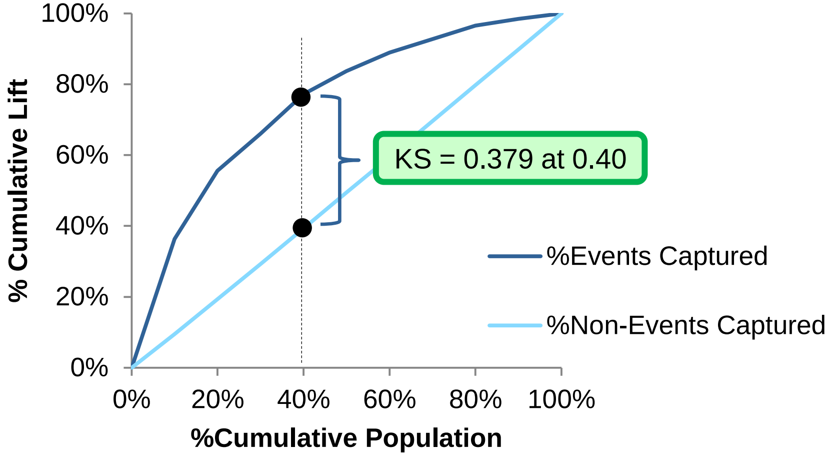
✗

Unacceptable Model

Illustration: Customer Attrition (Target Variable: IND_ATTR)

. . . Continued from Section 2.1.5

Train Dataset K-S Statistic



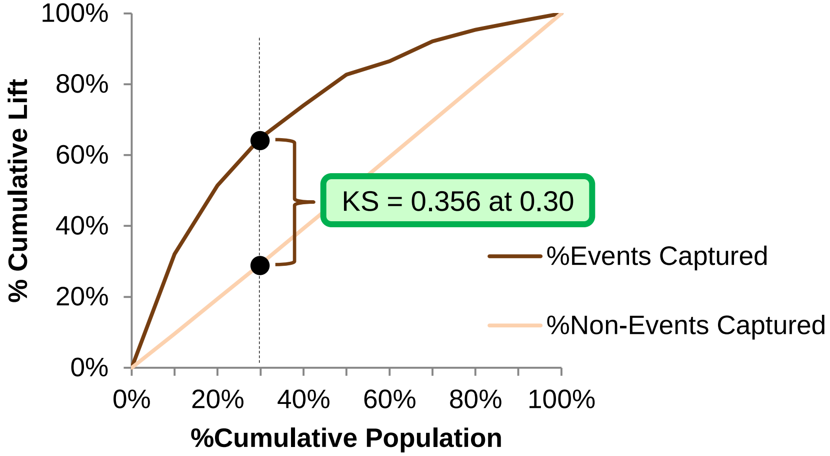
	A	B	C	D	E	F	G	H	I
1	**Bin**	**Cases**	**Bads**	**Goods**	**Bin Lift for Bads (C) ÷ Σ(C)**	**Cumulative Lift for Bads**	**Bin Lift for Goods (D) ÷ Σ(D)**	**Cumulative Lift for Goods**	**(F) – (H)**
2	1	1,987	134	1,853	36.3%	36.3%	9.5%	9.5%	0.268
3	2	1,988	71	1,917	19.2%	55.6%	9.8%	19.3%	0.362
4	3	1,988	39	1,949	10.6%	66.1%	10.0%	29.3%	0.368
5	4	1,988	41	1,947	11.1%	77.2%	10.0%	39.3%	**0.379**
6	5	1,988	24	1,964	6.5%	83.7%	10.1%	49.4%	0.344
7	6	1,988	19	1,969	5.1%	88.9%	10.1%	59.5%	0.294
8	7	1,988	14	1,974	3.8%	92.7%	10.1%	69.6%	0.231
9	8	1,988	14	1,974	3.8%	96.5%	10.1%	79.7%	0.168
10	9	1,988	7	1,981	1.9%	98.4%	10.2%	89.8%	0.085
11	10	1,988	6	1,982	1.6%	100.0%	10.2%	100.0%	0.000
12		**Σ(B) = 19,879**	**Σ(C) = 369**	**Σ(D) = 19,510**	**Σ(E) = 100%**		**Σ(G) = 100%**		

KS

Illustration: Customer Attrition (Target Variable: IND_ATTR)

. . . Continued from Section 2.1.5

Test Dataset K-S Statistic



	A	B	C	D	E	F	G	H	I
1	**Bin**	**Cases**	**Bads**	**Goods**	**Bin Lift for Bads (C) ÷ Σ(C)**	**Cumulative Lift for Bads**	**Bin Lift for Goods (D) ÷ Σ(D)**	**Cumulative Lift for Goods**	**(F) – (H)**
2	1	1,990	126	1,864	32.1%	32.1%	9.6%	9.6%	0.225
3	2	1,990	76	1,914	19.3%	51.4%	9.8%	19.4%	0.320
4	3	1,990	53	1,937	13.5%	64.9%	9.9%	29.3%	**0.356**
5	4	1,991	36	1,955	9.2%	74.0%	10.0%	39.3%	0.347
6	5	1,990	34	1,956	8.7%	82.7%	10.0%	49.3%	0.334
7	6	1,990	15	1,975	3.8%	86.5%	10.1%	59.5%	0.271
8	7	1,991	22	1,969	5.6%	92.1%	10.1%	69.6%	0.226
9	8	1,990	13	1,977	3.3%	95.4%	10.1%	79.7%	0.157
10	9	1,990	9	1,981	2.3%	97.7%	10.2%	89.8%	0.079
11	10	1,991	9	1,982	2.3%	100.0%	10.2%	100.0%	0.000
12		**Σ(B) = 19,903**	**Σ(C) = 393**	**Σ(D) = 19,510**	**Σ(E) = 100%**		**Σ(G) = 100%**		

KS

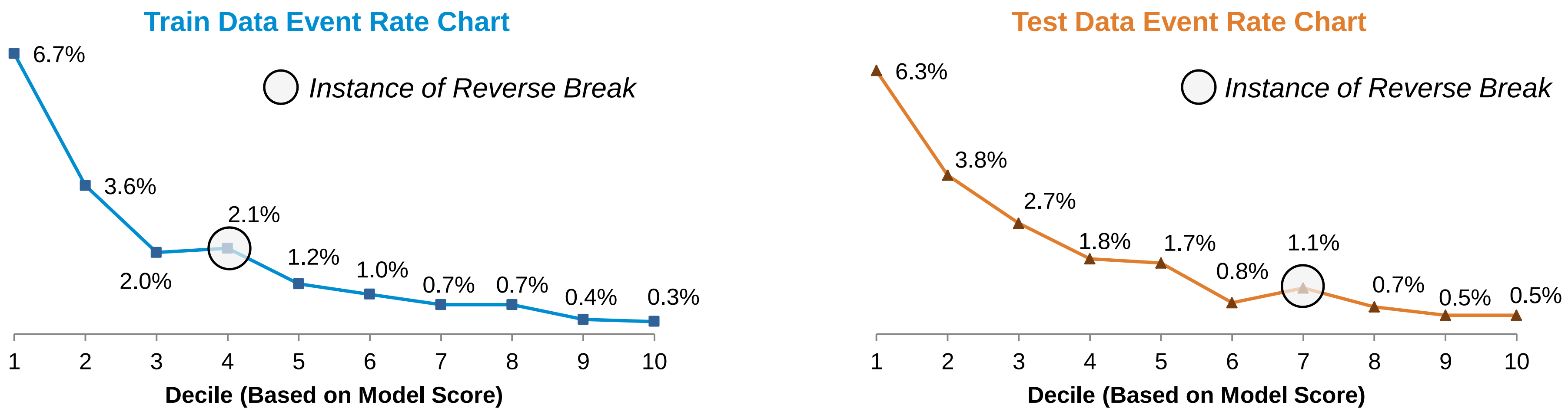
2.1.7. Decile-wise Event Rate Chart

In addition to Lift chart, a decile-wise event rate chart is plotted to gauge if the event rate rank orders well

- Moving down from Decile 1 to Decile 10, average value of target (i.e. event rate) should ideally fall monotonically
- However, in practice, few instances of reverse breaks may be observed. If such breaks exist but if they are neither frequent nor significant, the model may still be accepted

Illustration: Customer Attrition (Target Variable: IND_ATTR)

. . . Continued from Section 2.1.5



In general, there is a declining trend in event rate as we move from Decile 1 to Decile 10

2.1.8. Hosmer-Lemeshow Test

Usage

- Hosmer-Lemeshow test is a goodness-of-fit test for a binary target variable
- Unlike many other goodness-of-fit measures, it does not focus on gauging model's discriminatory power but aims at judging how closely the observed and the predicted values match

Procedure

1. Observations are divided into 10 deciles based on estimated probabilities
2. For each decile, compute
   a. Number of observed events (i.e. number of observations with event flag = 1)
   b. Number of expected events (i.e. total number of observations in decile multiplied by average predicted probability)
3. Discrepancies between observed and expected number of events in the deciles are summarized by the Pearson chi-square statistic, which is compared with a chi-square distribution with DF = 8 (#deciles – 2)
4. A small p-value (<0.05) suggests that the fitted model is not an adequate model

H-L Test Statistic

$$\chi^2_{HL} = \sum_{i=1}^{g} \frac{(O_i - N_i\overline{\pi}_i)^2}{N_i\overline{\pi}_i(1-\overline{\pi}_i)}$$

where

g = Number of groups (g = 10 in case of deciles)

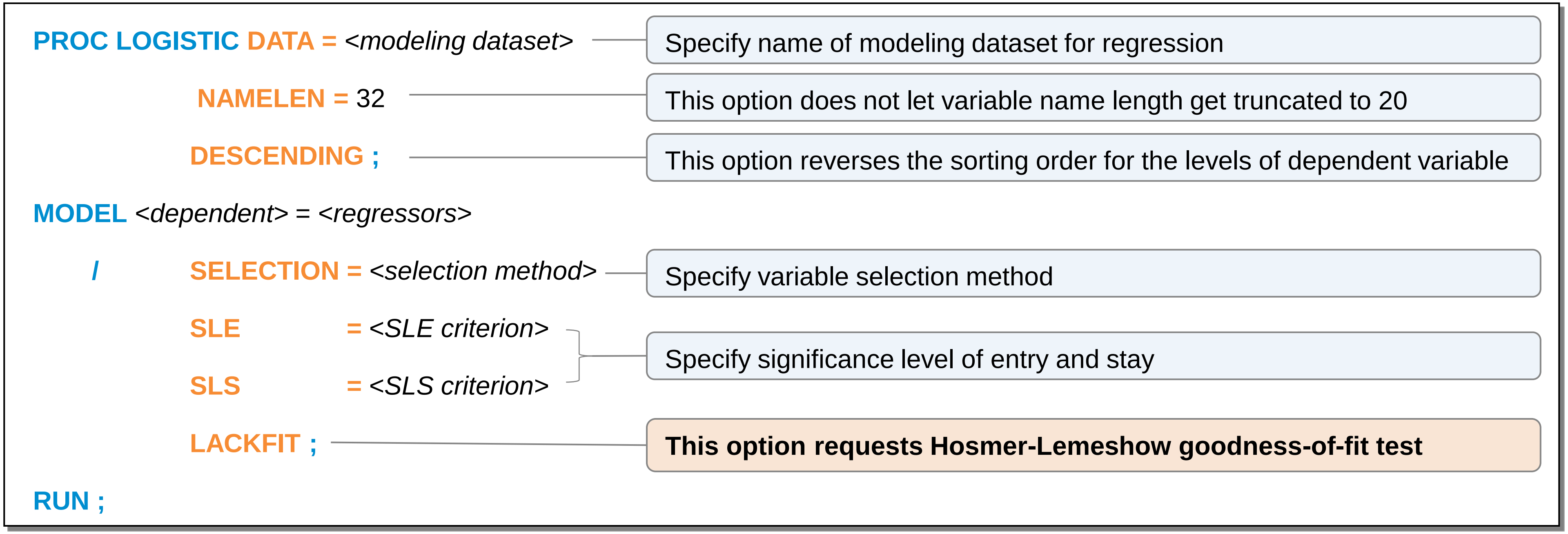
O_i = Observed number of events in group i

N_i = Total number of observations in group i

$\overline{\pi}_i$ = Average predicted probability in group i

SAS Implementation

Below is the syntax for generating Hosmer-Lemeshow test statistic



```
PROC LOGISTIC DATA = <modeling dataset>
              NAMELEN = 32
             DESCENDING ;
MODEL <dependent> = <regressors>
     /       SELECTION = <selection method>
             SLE        = <SLE criterion>
             SLS        = <SLS criterion>
             LACKFIT ;
RUN ;
```

Specify name of modeling dataset for regression

This option does not let variable name length get truncated to 20

This option reverses the sorting order for the levels of dependent variable

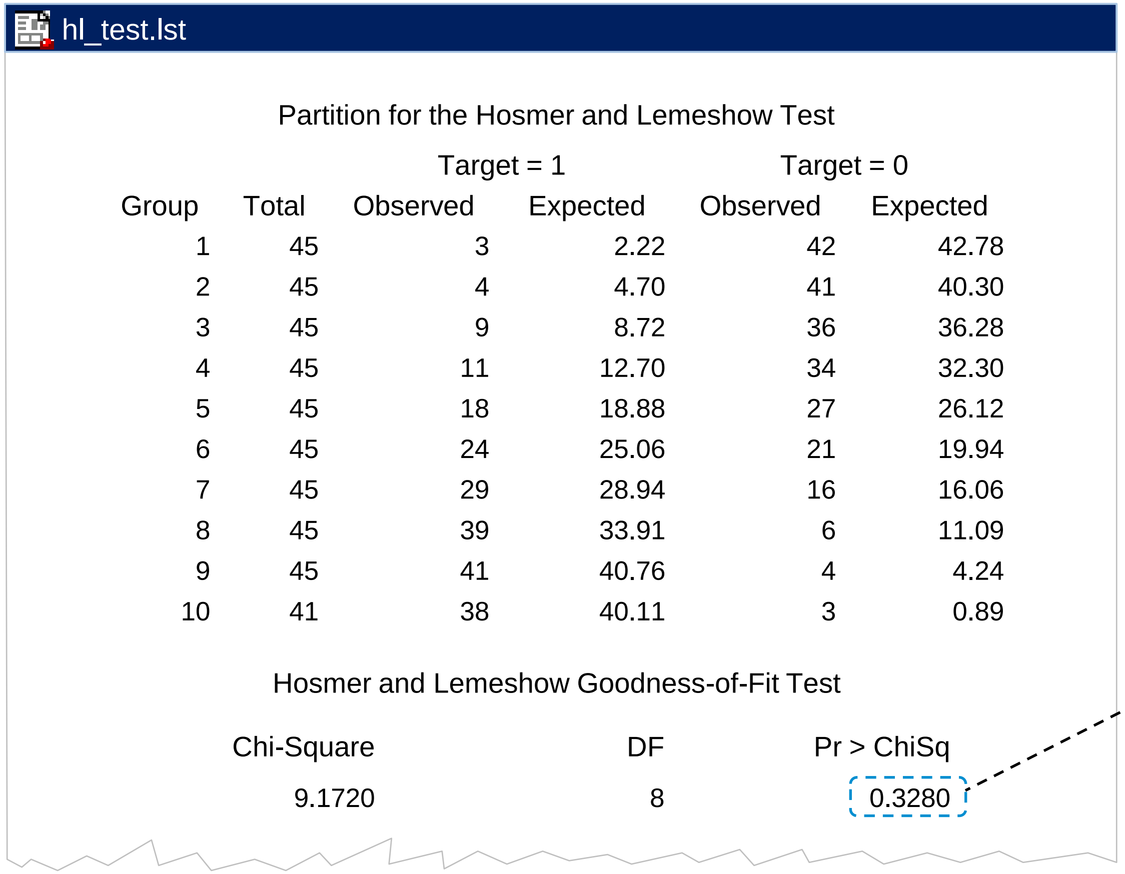
Specify variable selection method

Specify significance level of entry and stay

This option requests Hosmer-Lemeshow goodness-of-fit test

Illustration: Hosmer-Lemeshow Test (SAS Output)

LST File



hl_test.lst

```
                 Partition for the Hosmer and Lemeshow Test

                          Target = 1                 Target = 0
 Group    Total    Observed    Expected    Observed    Expected
     1       45           3        2.22          42       42.78
     2       45           4        4.70          41       40.30
     3       45           9        8.72          36       36.28
     4       45          11       12.70          34       32.30
     5       45          18       18.88          27       26.12
     6       45          24       25.06          21       19.94
     7       45          29       28.94          16       16.06
     8       45          39       33.91           6       11.09
     9       45          41       40.76           4        4.24
    10       41          38       40.11           3        0.89

                 Hosmer and Lemeshow Goodness-of-Fit Test

          Chi-Square              DF          Pr > ChiSq

              9.1720               8              0.3280
```

p-value is quite high (>0.05) and therefore the expected frequencies are not significantly different from the observed frequencies, indicating good model fit

Exercise

Exercise 1. Default Payment Probability Prediction Model

A magazine publication company wants to identify the customers who are likely to default on their subscription payments.

Server : 172.16.70.31
Location : T:\IND004\sas training\methodology\module_5
Train Data : train_sample_1 (Number of Observations: 60,733)
Test Data : test_sample_1 (Number of Observations: 60,188)

	Variable	Type	Label
1	CUST_ID	Num	Customer identification number
2	IND_PAY_DEFAULT	Num	Takes value 1 if customer did not pay dues on time
3	IND_ADDRESS_CHANGED	Num	Takes value 1 if customer changed residential address in past one year
4	IND_CR_STAT_UNPAID_EVER	Num	Takes value 1 if customer credit status has ever been tagged as unpaid
5	ORDER_CNT	Num	Number of orders placed by customer during his tenure
6	MTHS_TO_ORDER_EXPIRATION	Num	Number of months left in expiration of current order
7	PROP_DIRECT_ORDER	Num	Ratio of number of orders placed by customer via direct channel to total number of orders
8	VARIETY_RATIO	Num	Ratio of number of distinct products used by customer to total number of orders
9	IND_SOUTH_REGION	Num	Takes value 1 if customer belongs to south region
10	IND_PROM_MAIL_SENT	Num	Takes value 1 if any promotional mail was sent to the customer in past 1 month
11	CUST_TENURE	Num	Customer tenure in months
12	IND_EAST_REGION	Num	Takes value 1 if customer belongs to east region

Build a logistic regression model

(target variable: IND_PAY_DEFAULT, SLE = SLS = 0.05, selection method: BACKWARD)

Exercise

Exercise 1. Default Payment Probability Prediction Model

. . . Continued

For the developed model,

a. Generate classification table and analyze it to find probability cut-off

b. Report percent concordance and percent discordance for train dataset

c. Calculate AUC and Gini for train and test datasets

d. Calculate Hosmer-Lemeshow statistic for train dataset

e. Plot cumulative lift chart for train and test datasets

f. Compute K-S Statistic for train and test datasets

g. Plot decile-wise default rate for train and test datasets

2.2 Linear Regression Performance Measures

2.2.1. R² (Coefficient of Determination)

k-Variable Linear Regression Equation

Observed : $Y = \beta_0 + \beta_1 X_1 + \dots + \beta_k X_k + \varepsilon$

Model : $\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \dots + \hat{\beta}_k X_k$

R² Interpretation

- Proportion of variation in target variable (Y) explained by the model ($\hat{Y}$)
- R² is a goodness-of-fit measure, which is also known as coefficient of determination

R² Definition 1

$$R^2 = \frac{ESS}{TSS} = 1 - \frac{RSS}{TSS}$$

where

$$ESS = \sum \left(\hat{Y} - \overline{Y}\right)^2 = \text{Explained Sum of Squares}$$ (also known as Regression Sum of Squares)

$$RSS = \sum \left(Y - \hat{Y}\right)^2 = \text{Residual Sum of Squares}$$

$$TSS = \sum \left(Y - \overline{Y}\right)^2 = \text{Total Sum of Squares} = ESS + RSS$$

R² Definition 2

$$R^2 = \left(correlatio\ \ n(Y, \hat{Y})\right)^2$$

Things to Remember

0 ≤ R² ≤ 1

2.2.2. Adjusted R^2

Adjusted R^2 is a modification of R^2 that adjusts for the number of explanatory terms in the model

- Unlike R^2, adjusted R^2 increases only if the new term improves the model more than expected by chance
- Adjusted R^2 can be negative
- Adjusted $R^2 \leq R^2$

$$Adj.R^2 = 1 - \frac{(1 - R^2)(n - m)}{n - (k + m)}$$

where

R^2 = Unadjusted R-Square

n = Number of observations in the sample

k = Number of explanatory variables

$m = 1$ if model has an intercept term; otherwise $m = 0$

Higher R^2 and Adjusted R^2 values indicate better model performance

2.2.3. Root Mean Squared Error (RMSE)

Meaning and Usage

- Estimate of standard deviation of the error term
- Calculated as square root of Mean Squared Error (MSE)
- Scale dependent metric which does not have standalone meaning
- Used for comparison across models for model selection

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

where

Y_i = Observed value

$\hat{Y}_i$ = Predicted value

n = Number of observations

Things to Remember

Similar to RMSE, there are few more metrics that can be used to compare models

1. Mean Error (ME)
2. Mean Squared Error (MSE)
3. Mean Absolute Error (MAE)
4. Mean Percentage Error (MPE)
5. Mean Absolute Percentage Error (MAPE)

Lower RMSE value indicates better model performance

2.2.4. Coefficient of Variation (COV)

Meaning and Usage

- COV is calculated as ratio of RMSE to Dependent Variable Mean, multiplied by 100
- Unlike RMSE, it is a unit-less expression of variation in data

$$COV = \frac{RMSE}{\overline{Y}} \times 100\ \%$$

where

$RMSE$ = Root Mean Squared Error

$\overline{Y}$ = Average Value of Dependent Variable

Lower COV value indicates better model performance

2.2.5. Primary and Secondary Diagonal

Things to Remember

Banding is subjective. Do not manipulate bands for generating over-optimistic results

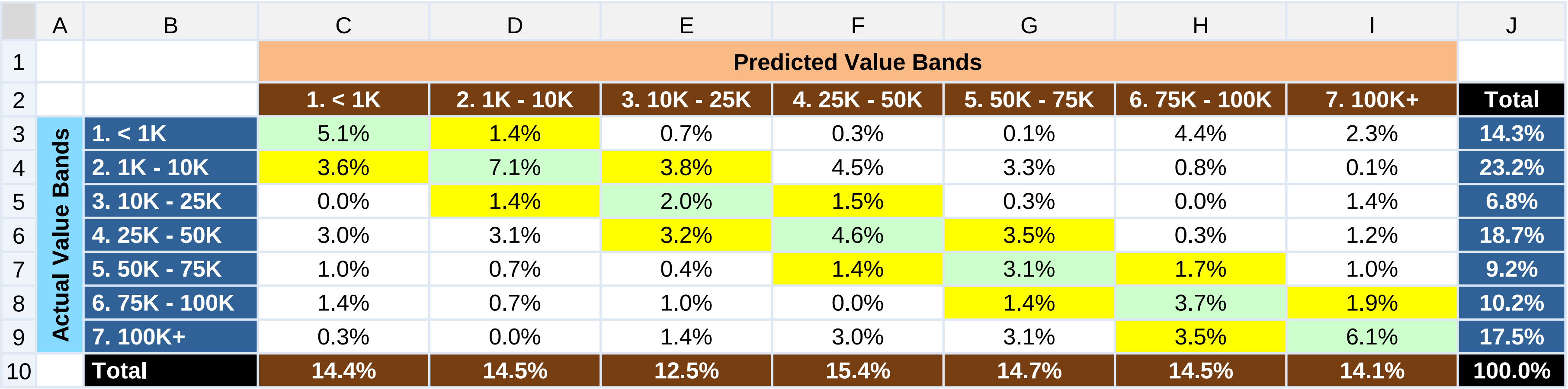
Procedure

- **Step 1** : Create bands based on actual (i.e. observed) and predicted values
- **Step 2** : Cross tabulate actual and predicted value bands and examine frequency distribution
- **Step 3a** : Sum up percentages in primary diagonal cells to report primary diagonal metric
- **Step 3b** : Sum up percentages in secondary diagonal cells to report secondary diagonal metric

Illustration: Credit Card Payment Due Amount (Target Variable: DUE_AMT)

Primary Diagonal Metric : 31.7%
Secondary Diagonal Metric : 28.4%

🟩 *Primary Diagonal*
🟨 *Secondary Diagonal*



	A	B	C	D	E	F	G	H	I	J
1			**Predicted Value Bands**							
2			**1. < 1K**	**2. 1K - 10K**	**3. 10K - 25K**	**4. 25K - 50K**	**5. 50K - 75K**	**6. 75K - 100K**	**7. 100K+**	**Total**
3	**Actual Value Bands**	**1. < 1K**	5.1%	1.4%	0.7%	0.3%	0.1%	4.4%	2.3%	**14.3%**
4		**2. 1K - 10K**	3.6%	7.1%	3.8%	4.5%	3.3%	0.8%	0.1%	**23.2%**
5		**3. 10K - 25K**	0.0%	1.4%	2.0%	1.5%	0.3%	0.0%	1.4%	**6.8%**
6		**4. 25K - 50K**	3.0%	3.1%	3.2%	4.6%	3.5%	0.3%	1.2%	**18.7%**
7		**5. 50K - 75K**	1.0%	0.7%	0.4%	1.4%	3.1%	1.7%	1.0%	**9.2%**
8		**6. 75K - 100K**	1.4%	0.7%	1.0%	0.0%	1.4%	3.7%	1.9%	**10.2%**
9		**7. 100K+**	0.3%	0.0%	1.4%	3.0%	3.1%	3.5%	6.1%	**17.5%**
10		**Total**	**14.4%**	**14.5%**	**12.5%**	**15.4%**	**14.7%**	**14.5%**	**14.1%**	**100.0%**

Higher primary and secondary diagonal values indicate better model performance

2.2.6. SAS Implementation

SAS Syntax

```
PROC REG DATA = <modeling dataset> ;
MODEL <dependent> = <regressors>
      /        SELECTION = <selection method>
               SLE       = <SLE criterion>
               SLS       = <SLS criterion> ;
QUIT;
```

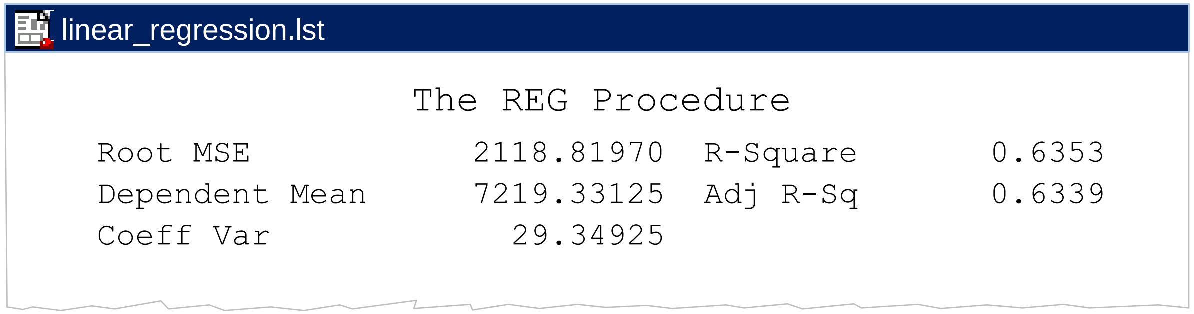
Specify name of modeling dataset for regression

Specify variable selection method

Specify significance level of entry and stay

Illustration

LST File



```
                    The REG Procedure

    Root MSE              2118.81970    R-Square       0.6353
    Dependent Mean        7219.33125    Adj R-Sq       0.6339
    Coeff Var               29.34925
```

2.2.7. Residual Analysis

Need for Residual Analysis

Objective 1: To check whether the residuals are ‘pattern less’ (randomly scattered) centered around zero

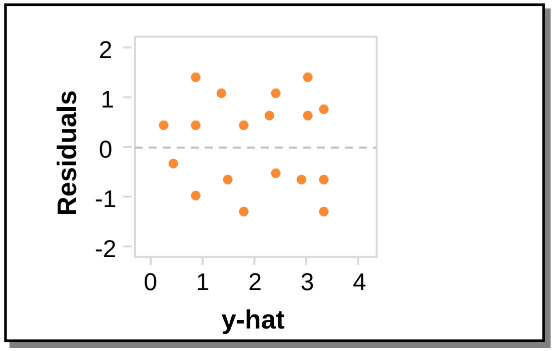
Method of Analysis: Residual Plot

Objective 2: To check whether the residuals follow a normal distribution

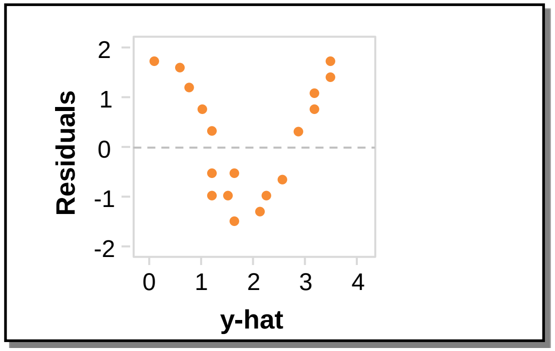
Method of Analysis: Normal Q-Q Plot

Residual Plot

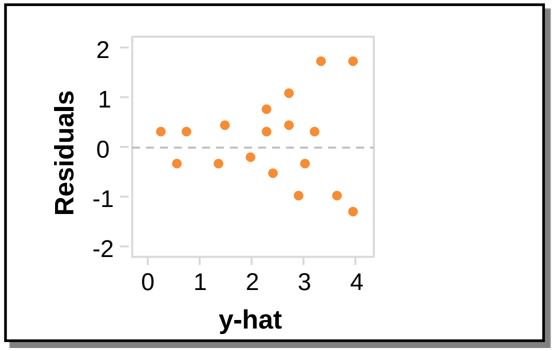
- A graph that shows the residuals on the vertical axis and the fitted values on the horizontal axis
- If the points in a residual plot are randomly dispersed around zero (horizontal axis), a linear regression model is appropriate for the data, otherwise a non-linear model is more appropriate
- Examples:



- Random scatter around zero
- Linear regression Is appropriate



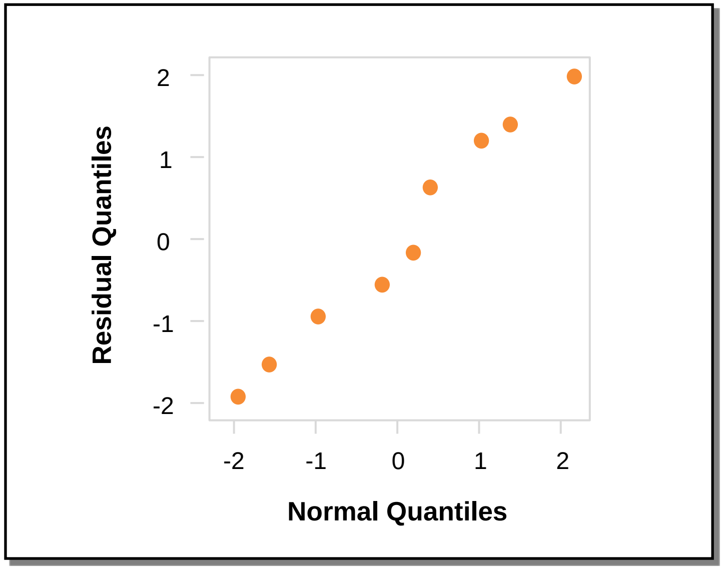
- Distinct curved pattern (U-shaped)
- Linear model is not appropriate (bad fit)
- Non-linear model should be tried out



- Funnel shaped pattern
- More spread for larger fitted values (bad fit)
- Check for Heteroscedasticity

Normal Q-Q Plot

- Quantile-Quantile (Q-Q) plot is a graphical method for comparing two probability distributions by plotting their quantiles against each other
- Normal Q-Q plot shows the observed quantiles of residuals on the vertical axis and the theoretical quantiles of standard normal distribution on the horizontal axis
- If residuals follow normal distribution, the normal Q-Q plot should be a straight line
- Example:

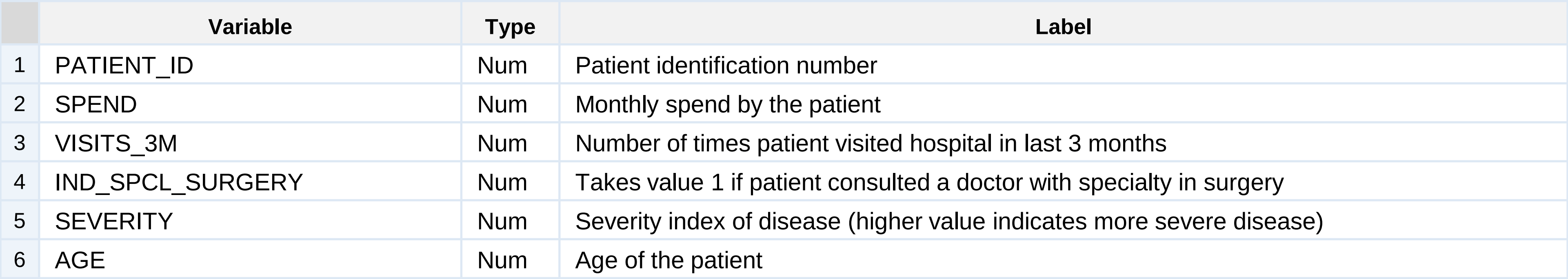


Exercise

Exercise 2. Spend Prediction Model

A hospital management wants to have an estimate of monthly spend (revenue) from each existing patient.

Server	: 172.16.70.31	
Location	: T:\IND004\sas training\methodology\module_5	
Train Data	: train_sample_2	(Number of Observations: 3,500)
Test Data	: test_sample_2	(Number of Observations: 1,500)



	Variable	Type	Label
1	PATIENT_ID	Num	Patient identification number
2	SPEND	Num	Monthly spend by the patient
3	VISITS_3M	Num	Number of times patient visited hospital in last 3 months
4	IND_SPCL_SURGERY	Num	Takes value 1 if patient consulted a doctor with specialty in surgery
5	SEVERITY	Num	Severity index of disease (higher value indicates more severe disease)
6	AGE	Num	Age of the patient

Build a linear regression model

(target variable: SPEND, SLE = SLS = 0.05, selection method: BACKWARD)

Exercise

Exercise 2. Spend Prediction Model

. . . Continued

For the developed model, for train and test datasets compute

a. R^2

b. Adjusted R^2

c. RMSE

d. Coefficient of Variation

e. Primary and Secondary Diagonal Metrics

Chapter 3: Model Stabilization

3.1 Population Stability Analysis

3.1.1. Population Stability Index (PSI)

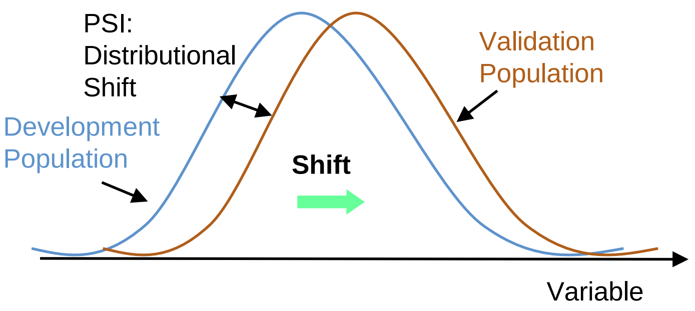
Meaning and Usage

- Widely used stability metric
- Measures the shift in population from development sample to validation sample

Formula

$$PSI = \sum \left[(\% \, Validation - \% \, Developmen \quad t) \times LN \left(\frac{\% \, Validation}{\% \, Developmen \quad t} \right) \right]$$

Frequency Distribution



Guidelines for Assessment

PSI	Interpretation
< 0.10	*Populations are similar*
0.10-0.25	*Some concern over stability*
> 0.25	*Substantial change in populations*

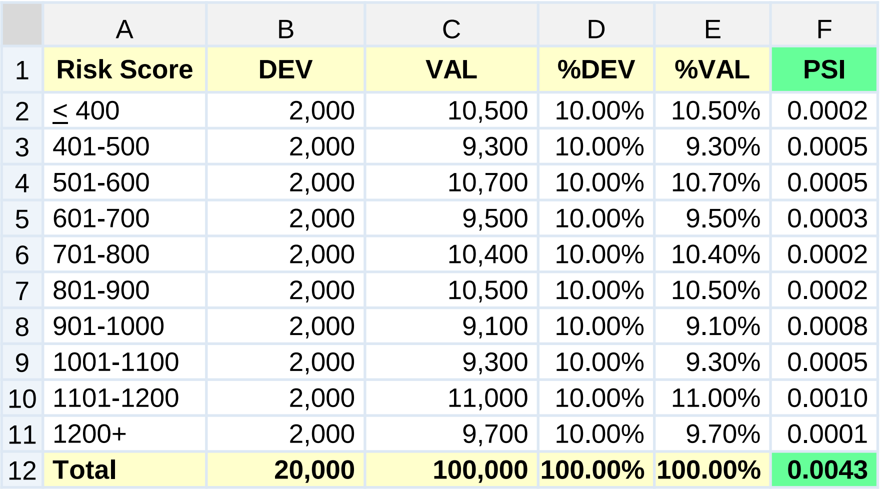
Note: For a continuous variable, the bins are typically created by decile or demi-decile using development sample

3.1.2. PSI Applications

Score Stability Analysis

- PSI metric is calculated based on binning of the model score (predicted outcome)
- Objective is to ascertain if the score distribution shifted and in what direction

Illustration: Credit Risk Score

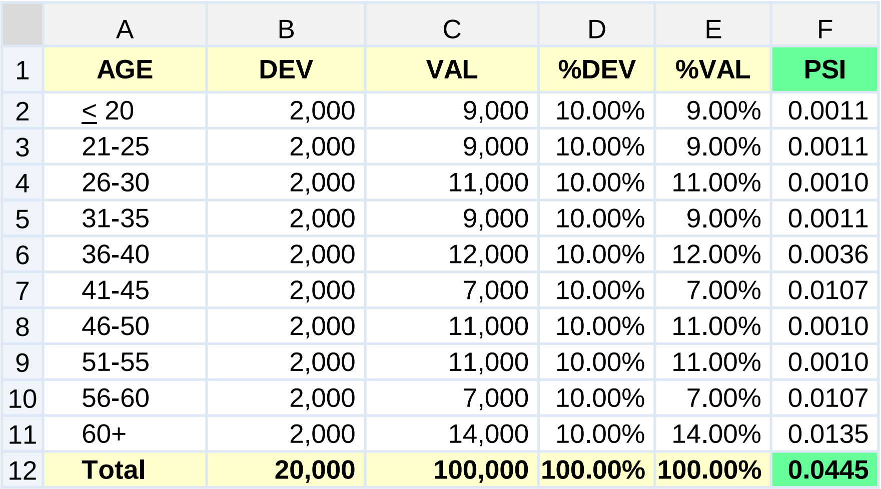


	A	B	C	D	E	F
1	**Risk Score**	**DEV**	**VAL**	**%DEV**	**%VAL**	**PSI**
2	≤ 400	2,000	10,500	10.00%	10.50%	0.0002
3	401-500	2,000	9,300	10.00%	9.30%	0.0005
4	501-600	2,000	10,700	10.00%	10.70%	0.0005
5	601-700	2,000	9,500	10.00%	9.50%	0.0003
6	701-800	2,000	10,400	10.00%	10.40%	0.0002
7	801-900	2,000	10,500	10.00%	10.50%	0.0002
8	901-1000	2,000	9,100	10.00%	9.10%	0.0008
9	1001-1100	2,000	9,300	10.00%	9.30%	0.0005
10	1101-1200	2,000	11,000	10.00%	11.00%	0.0010
11	1200+	2,000	9,700	10.00%	9.70%	0.0001
12	**Total**	**20,000**	**100,000**	**100.00%**	**100.00%**	**0.0043**

Characteristic Stability Analysis

- PSI metric is calculated based on binning of a characteristic (i.e. explanatory variable)
- Objective is to examine shifts in distributions of individual characteristics and to understand if high PSI values of a set of characteristics could explain high PSI value of overall score

Illustration: Demographic Characteristic (AGE)



	A	B	C	D	E	F
1	**AGE**	**DEV**	**VAL**	**%DEV**	**%VAL**	**PSI**
2	≤ 20	2,000	9,000	10.00%	9.00%	0.0011
3	21-25	2,000	9,000	10.00%	9.00%	0.0011
4	26-30	2,000	11,000	10.00%	11.00%	0.0010
5	31-35	2,000	9,000	10.00%	9.00%	0.0011
6	36-40	2,000	12,000	10.00%	12.00%	0.0036
7	41-45	2,000	7,000	10.00%	7.00%	0.0107
8	46-50	2,000	11,000	10.00%	11.00%	0.0010
9	51-55	2,000	11,000	10.00%	11.00%	0.0010
10	56-60	2,000	7,000	10.00%	7.00%	0.0107
11	60+	2,000	14,000	10.00%	14.00%	0.0135
12	**Total**	**20,000**	**100,000**	**100.00%**	**100.00%**	**0.0445**

3.2 Model Stability Boosting Techniques

3.2.1. k-Fold Cross Validation

Purpose

- Cross-validation (CV) is a way to predict the fit of a model to a hypothetical validation set when an explicit validation set is not available
- Cross validation provides a reasonable estimate of model fit. Usage of CV technique at the time of model development provides realistic estimate of benchmark performance and thus infuses stability

Steps

1. Randomly divide data into k folds of equal size
2. Use k-1 folds data for training, and one fold for testing
3. Repeat k times until all folds are used for testing

Things to Remember

Advantage: All observations are used for both training and validation, and each observation is used for validation exactly once

Illustration

In 5-fold cross-validation, the data would be split into five equal sets A, B, C, D and E. Models would be developed on each four-fifths of the data using the remaining one-fifth for testing as follows:

	TRAIN	TEST
1	ABCD	E
2	ABCE	D
3	ACDE	B
4	BCDE	A
5	ABDE	C

→ The results of 5 test datasets A, B, C, D and E are averaged to get the final estimate of model performance

3.2.2. Bootstrapping

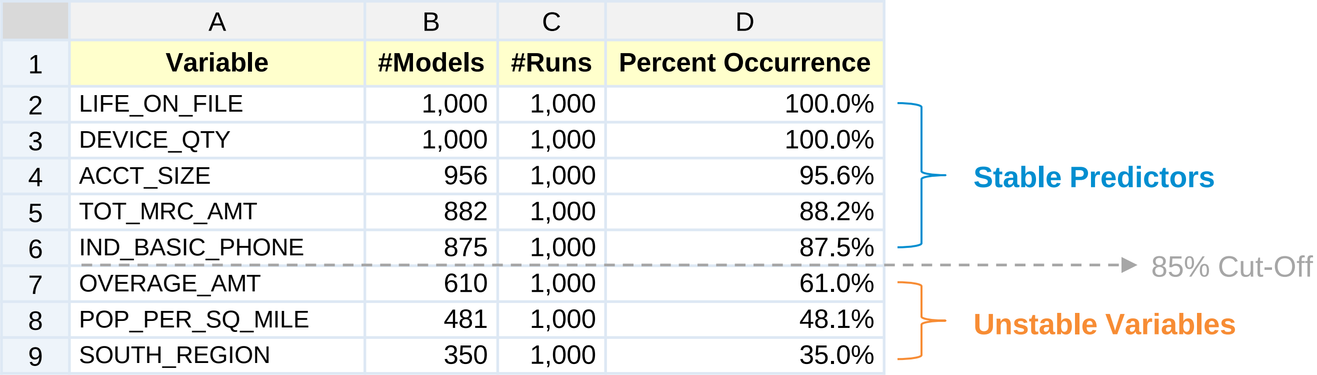
Purpose

- Bootstrapping is a very effective technique to identify stable variables for model development
- It is a time consuming process and hence it is generally applied once a list of potential predictors (not more than 100) has already been identified. The idea is to pick most stable ones out of good performers.

Steps

1. Draw m samples (e.g. m = 1000) with 80% obs. selected randomly (with replacement) from train data
2. Build a model on each sample using a list of predictors and a model selection method (e.g. backward)
3. For each variable, compute ‘percent occurrence’ over all models
4. Apply a cut-off (e.g. 85%) on ‘percent occurrence’ to identify stable variables

Illustration: Telecom Churn (Target Variable: IND_CHURN)



	A	B	C	D
1	**Variable**	**#Models**	**#Runs**	**Percent Occurrence**
2	LIFE_ON_FILE	1,000	1,000	100.0%
3	DEVICE_QTY	1,000	1,000	100.0%
4	ACCT_SIZE	956	1,000	95.6%
5	TOT_MRC_AMT	882	1,000	88.2%
6	IND_BASIC_PHONE	875	1,000	87.5%
7	OVERAGE_AMT	610	1,000	61.0%
8	POP_PER_SQ_MILE	481	1,000	48.1%
9	SOUTH_REGION	350	1,000	35.0%

Stable Predictors
85% Cut-Off
Unstable Variables

Things to Remember

Bootstrapping is also used as a variable reduction technique along with stabilization

3.2.3. Coefficient Blasting

Things to Remember

Coefficient blasting may also be used as an ensemble technique along with stabilization

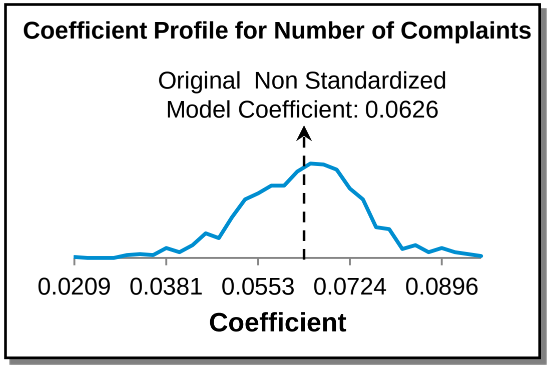
Purpose

- Eliminate variables with inconsistent estimates; or
- Replace beta coefficients of original model with average beta values across samples

Steps

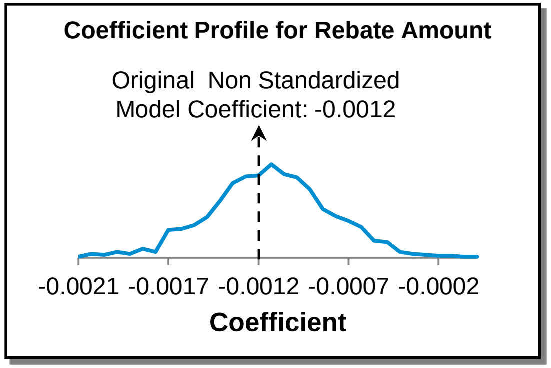
1. Draw m samples (e.g. m = 1000) with 80% obs. selected randomly (with replacement) from train data
2. Build a model on each sample using a 'fixed' list of predictors without any model selection method
3. For each variable, analyze the distribution of coefficients

Illustration: Membership Cancellation (Target Variable: IND_CANCEL)



Sigma	Mean	Median
0.0111	0.0627	0.0630

Estimation of model coefficients over 1000 samples shows that the coefficients of predictors are stable and peak around the value identified in the original model



Sigma	Mean	Median
0.0003	-0.0011	-0.0011

3.2.4. Sensitivity Analysis

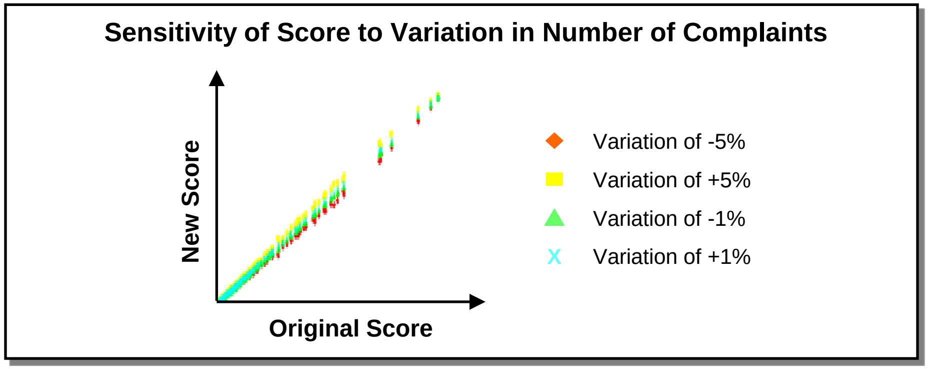
Purpose

- Sensitivity analysis is carried out to gauge sensitivity of the model performance towards variation (+/- 5% and +/- 1%) in a particular variable

Steps

1. Save original model equation and the predicted score
2. Vary a particular predictor by +1% (keeping all other predictors fixed) and regenerate score
3. Repeat step 2 using different percentages (-1%, +5% and -5%)
4. Plot original score against new scores generated by variations in a particular predictor and analyze
5. Repeat steps 2, 3 and 4 for all predictors one by one

Illustration: Membership Cancellation (Target Variable: IND_CANCEL)



The graph shows that the model is not over-sensitive to slight changes in the predictor (*number of complaints*)

References

1. **Chapter 39: The LOGISTIC Procedure** (http://www.math.wpi.edu/saspdf/stat/chap39.pdf)
   SAS OnlineDoc™

2. **Chapter 55: The REG Procedure** (http://www.math.wpi.edu/saspdf/stat/chap55.pdf)
   SAS OnlineDoc™

3. **Logistic Regression Using SAS**, Theory and Application
   by Paul D. Allison

4. **Overfitting and Capacity Control**
   by Sam Roweis

5. **Sensitivity, Specificity, Accuracy, Associated Confidence Interval and ROC Analysis with Practical SAS Implementations**
   by Wen Zhu, Nancy Zeng and Ning Wang

6. **The Applied Use of Population Stability Index (PSI) in SAS Enterprise Miner**
   by Rex Pruitt

7. **Validation of Predictive Regression Models**
   by Ewout W. Steyerberg and Frank E. Harrell

8. **Wikipedia** (http://www.wikipedia.org)

Thanks

For queries, contact Varun Aggarwal at Varun.Aggarwal@exlservice.com