SCORECARD VARIABLE GROUPING AND SELECTION

Dr. Aric LaBarr
Institute for Advanced Analytics

Process Flow

Data Collection
- Variable Selection
- Sample Size
- Sample / Performance Window

Data Cleaning
- Eliminate Duplicates
- Examine / Remove Outliers

Variable Grouping and Selection
- Weights of Evidence (WOE)
- Information Value (IV)
- Gini Criterion

Initial Scorecard Creation
- Logistic Regression
- Accuracy
- Threshold
- Assessment

Reject Inference
- Remove bias resulting from exclusion of rejects

Final Scorecard Creation
- Final Model Assessment

VARIABLE GROUPING

Variable Grouping and Selection

- Scorecards end up with only just groups within a variable.
- Objectives:
  1. Eliminate weak characteristics (variables) or those that do not conform to good business logic.
  2. Group the strongest characteristics' attribute levels in order to produce a model in scorecard format.
- Function/package "scorecard" or "smbinning" in R.
- Package "scorecard" or "OptBinning" in Python.

Variable	Level
MISS	$x < 24$
MISS	$24 \leq x < 36$
MISS	$36 \leq x < 48$
MISS	$x \geq 48$
HOME	OWN
HOME	RENT

Why Grouping (Binning)?

- Goal is to help simplify analysis through grouping:
  - Useful for understanding relationships – no worries about explaining coefficients.
  - Modeling nonlinearities – similar to decision trees.
    (NO MORE LOGISTIC REGRESSION LINEARITY ASSUMPTION!)
  - Dealing with outliers – contained in the smallest / largest group.
  - Missing values typically in own group.

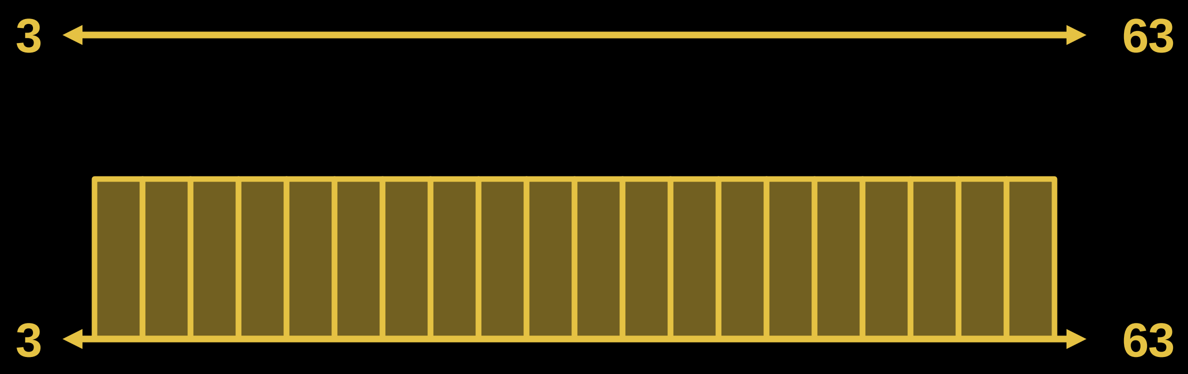
Initial Characteristic Analysis – Pre-binning

- Need a starting point for the grouping / binning.
  - Quantiles are most popular technique.
- Pre-bin the interval variables into a number of user-specified quantiles / buckets for fine detailed groupings.
- Aggregate the fine detailed groupings into a smaller number to produce coarse groupings.
  - Chi-squared tests to combine groups.

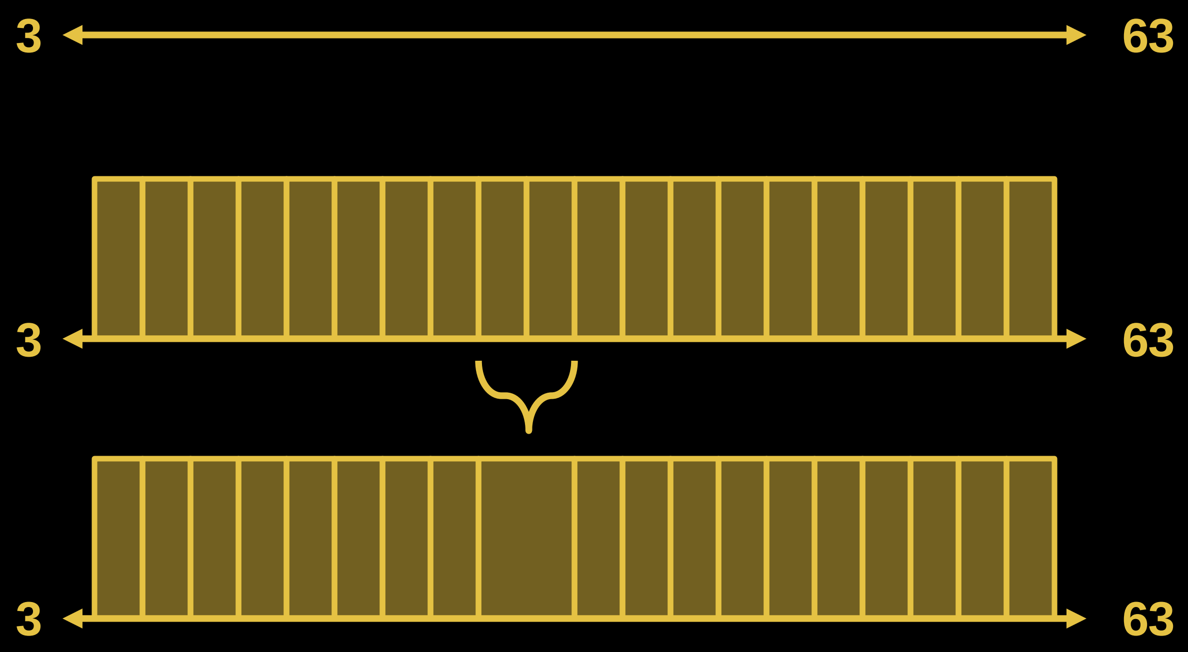
Initial Characteristic Analysis – Pre-binning

3 ⟷ **63**

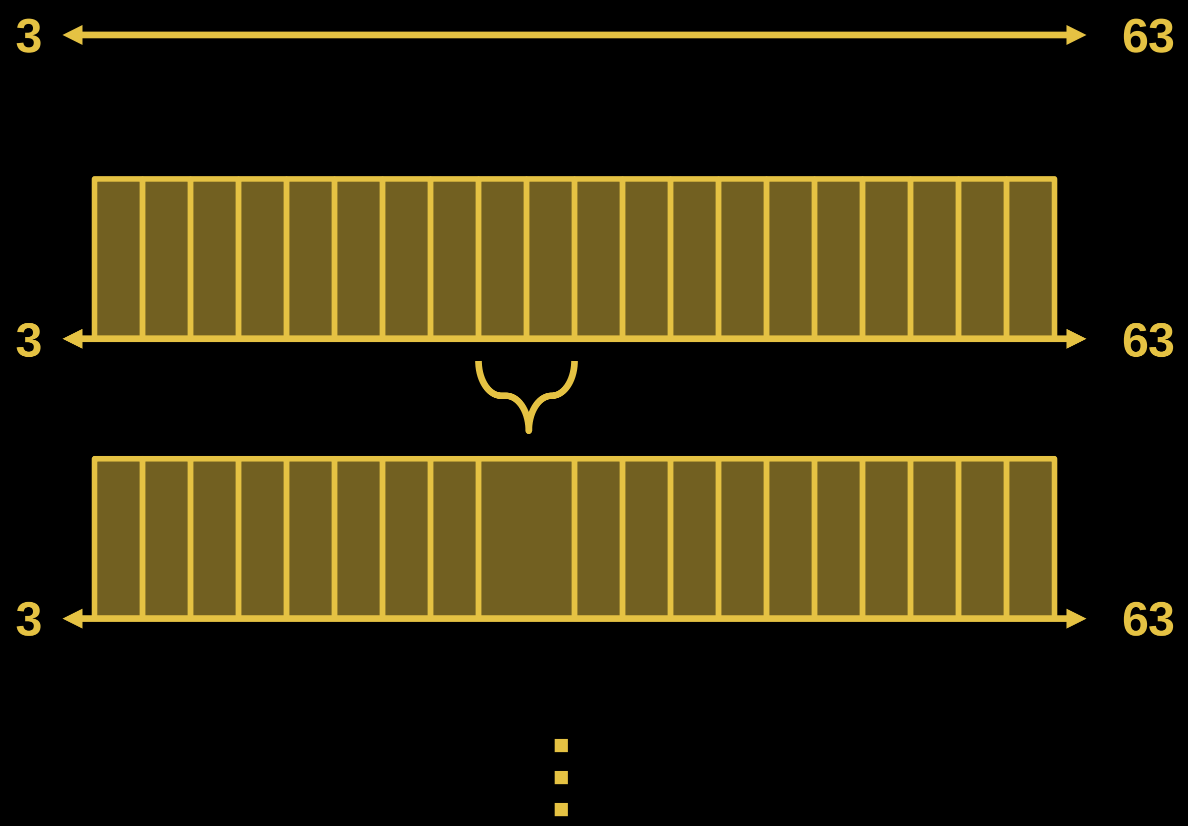
Initial Characteristic Analysis – Pre-binning



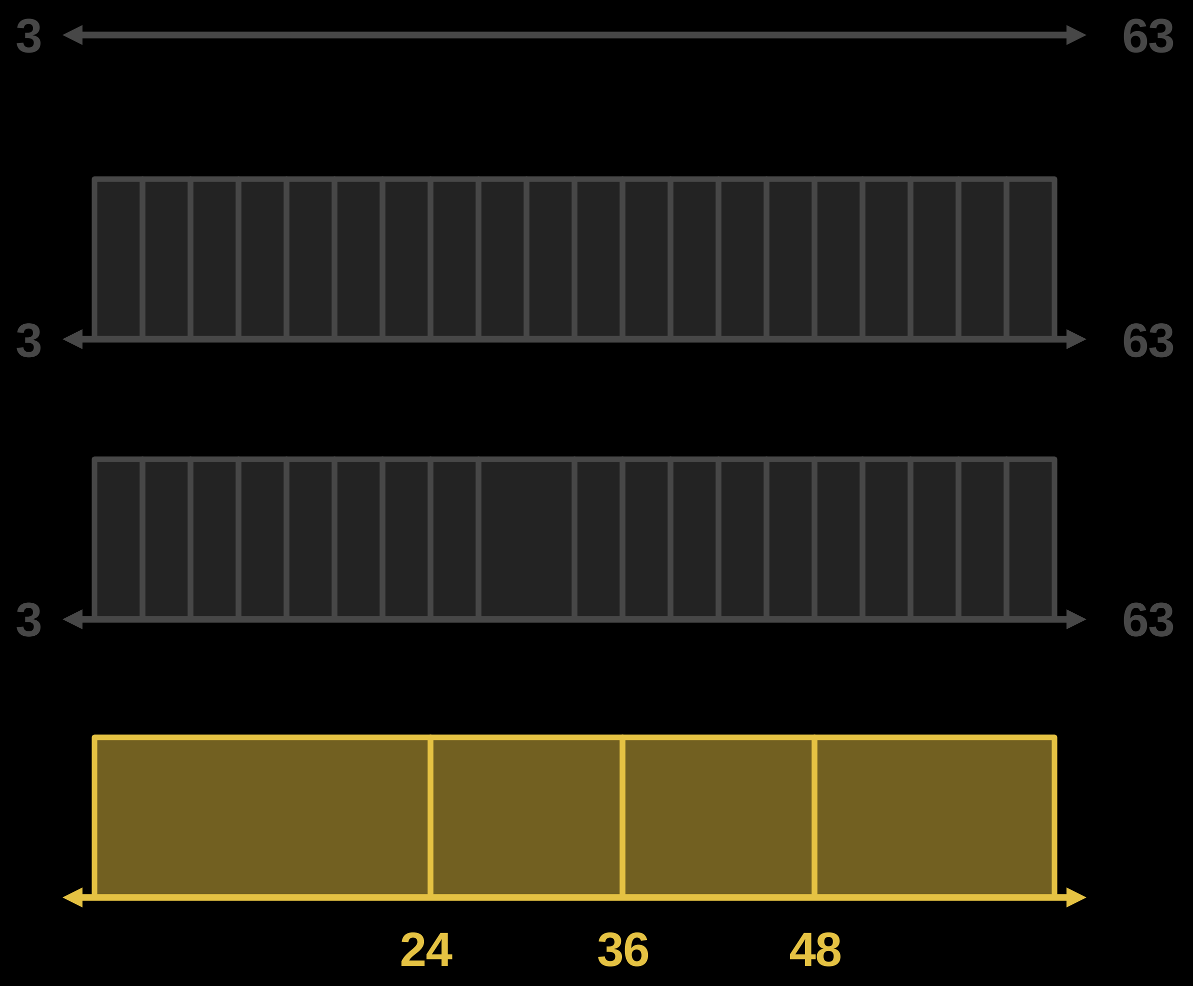
Initial Characteristic Analysis – Pre-binning



Initial Characteristic Analysis – Pre-binning



Initial Characteristic Analysis – Pre-binning



Initial Characteristic Analysis – Tree-based

- Another approach to binning is through tree-based methods like decision trees or conditional inference trees.

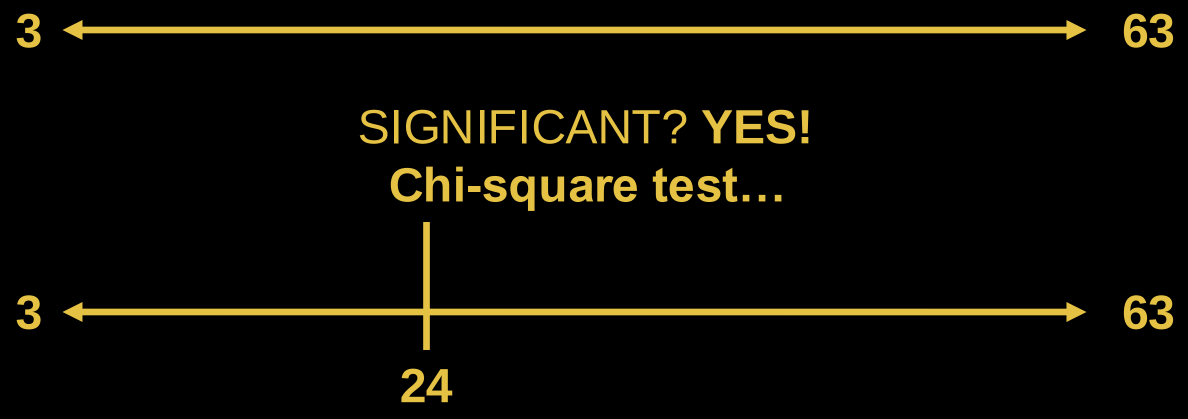
- Conditional Inference Trees (CIT):
  - CART methods potentially have inherent bias – variables with more levels → more likely to be split on if split on Gini and Entropy.
  - CIT method adds extra statistical step before splits occur – statistical tests of significance.
  - What is **MOST** significant variable? → What is the best split (Chi-square) on **THIS** variable? → **REPEAT**.

Initial Characteristic Analysis – CIT

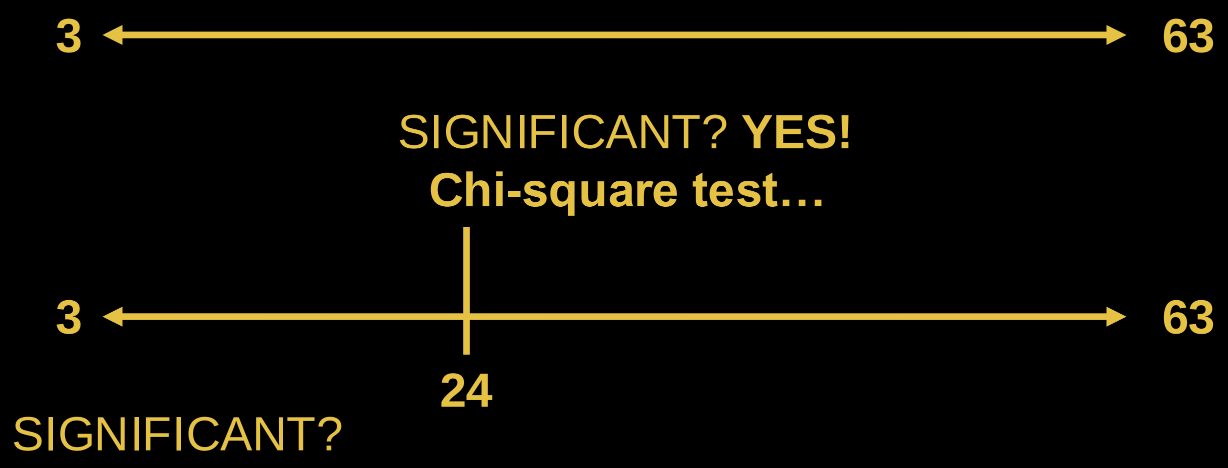
3 ⟷ **63**

SIGNIFICANT?

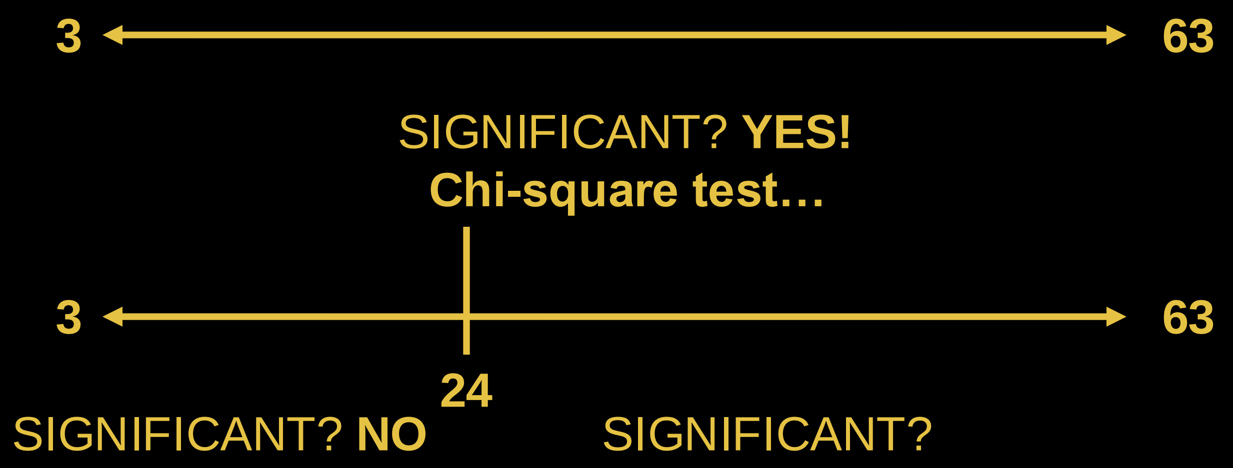
Initial Characteristic Analysis – CIT



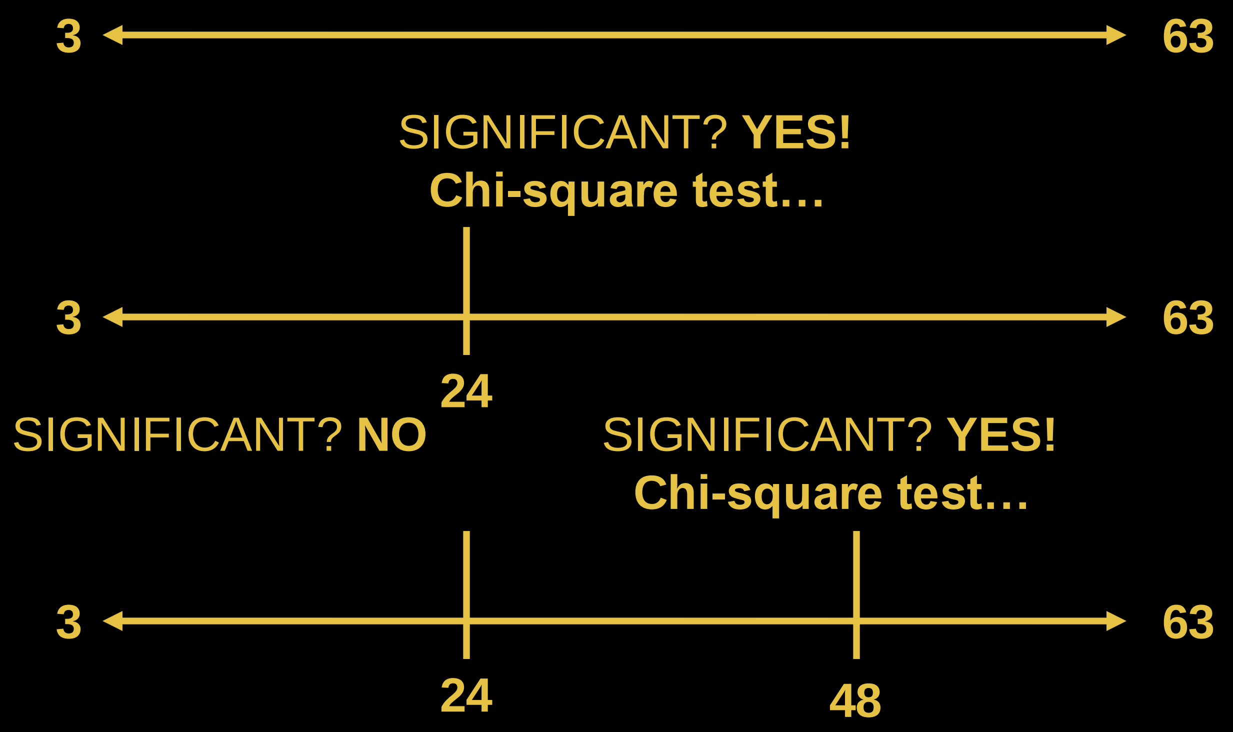
Initial Characteristic Analysis – CIT



Initial Characteristic Analysis – CIT



Initial Characteristic Analysis – CIT



Initial Characteristic Analysis – CIT

Initial Characteristic Analysis

- Cut-offs may be rough from decision tree combining.
- **Optional to override** automatically generated groups to conform to business rules.
- Overrides may make groups suboptimal.

Group Definition
Missing
< $35,200
$35,200 - $60,000
$60,000 - $85,000
$85,000 - $110,000
$110,000 - $142,530
> $142,530

Initial Characteristic Analysis

- Cut-offs may be rough from decision tree combining.
- **Optional to override** automatically generated groups to conform to business rules.
- Overrides may make groups suboptimal.

Group Definition	Override
Missing	Missing
< $35,200	< $35,000
$35,200 - $60,000	$35,000 - $60,000
$60,000 - $85,000	$60,000 - $85,000
$85,000 - $110,000	$85,000 - $110,000
$110,000 - $142,530	$110,000 - $140,000
> $142,530	> $140,000

Initial Characteristic Analysis

- Cut-offs may be rough from decision tree combining.
- **Optional to override** automatically generated groups to conform to business rules.
- Overrides may make groups suboptimal.

Group Definition	Override
Missing	Missing
< $35,200	**< $35,000**
$35,200 - $60,000	**$35,000** - $60,000
$60,000 - $85,000	$60,000 - $85,000
$85,000 - $110,000	$85,000 - $110,000
$110,000 - $142,530	$110,000 - **$140,000**
> $142,530	**> $140,000**

Initial Characteristic Analysis

- Calculate and examine the key assessment metrics:
  - **Weight of Evidence** (WOE) – how well attributes discriminate for each given characteristic
  - **Information Value** (IV) – evaluate a characteristic's overall predictive power
  - **Gini Statistic** – alternate to IV for selecting characteristics for final model.

?

WEIGHT OF EVIDENCE

Weight of Evidence (WOE)

- WOE measures the strength of the attributes of a characteristic in **separating good and bad accounts**.
- WOE is based on comparing the proportion of goods to bads at each attribute level (levels of the predictor variable).

$$WOE_i = \log\left(\frac{Dist.Good_i}{Dist.Bad_i}\right)$$

Weight of Evidence (WOE)

- WOE measures the strength of the attributes of a characteristic in **separating good and bad accounts**.
- WOE is based on comparing the proportion of goods to bads at each attribute level (levels of the predictor variable).

$$WOE_i = \log\left(\frac{Dist.Good_i}{Dist.Bad_i}\right)$$

$$Dist.Good_i = \frac{Number\ Good\ in\ group\ i}{Total\ Number\ Good}$$

Weight of Evidence (WOE)

- WOE measures the strength of the attributes of a characteristic in **separating good and bad accounts**.
- WOE is based on comparing the proportion of goods to bads at each attribute level (levels of the predictor variable).

$$WOE_i = \log\left(\frac{Dist.Good_i}{Dist.Bad_i}\right)$$

$$Dist.Bad_i = \frac{Number\ Bad\ in\ group\ i}{Total\ Number\ Bad}$$

Weight of Evidence (WOE)

- What are we looking for?
  - Looking for “big” differences in WOE between groups.
  - Monotonic changes within an attribute for **interval variables** (not always required).

- Why monotonic increases?
  - Oscillation back and forth of positive to negative values of WOE typically sign of variable that has trouble separating good vs. bad.
  - Not always required **if makes business sense** – credit card utilization for example.

WOE – Example

WOE for Bureau Score				
Group	Values	Event Count	Non-event Count	WOE
1	< 607	129	127	
2	607 – 640	181	285	
3	641 – 653	103	215	
4	654 – 667	86	262	
…	…	…	…	…
12	> 786	5	219	
	MISSING	89	155	
Total		**900**	**3,477**	

WOE – Example

WOE for Bureau Score				
Group	Values	Event Count	Non-event Count	WOE
1	< 607	129	127	
2	607 – 640	181	285	
3	641 – 653	103	215	
4	654 – 667	86	262	
…	…	…	…	…
12	> 786	5	219	
	MISSING	89	155	
Total		**900**	**3,477**	

$$Dist.Good_1 = \frac{127}{3477}$$

$$= 0.0365$$

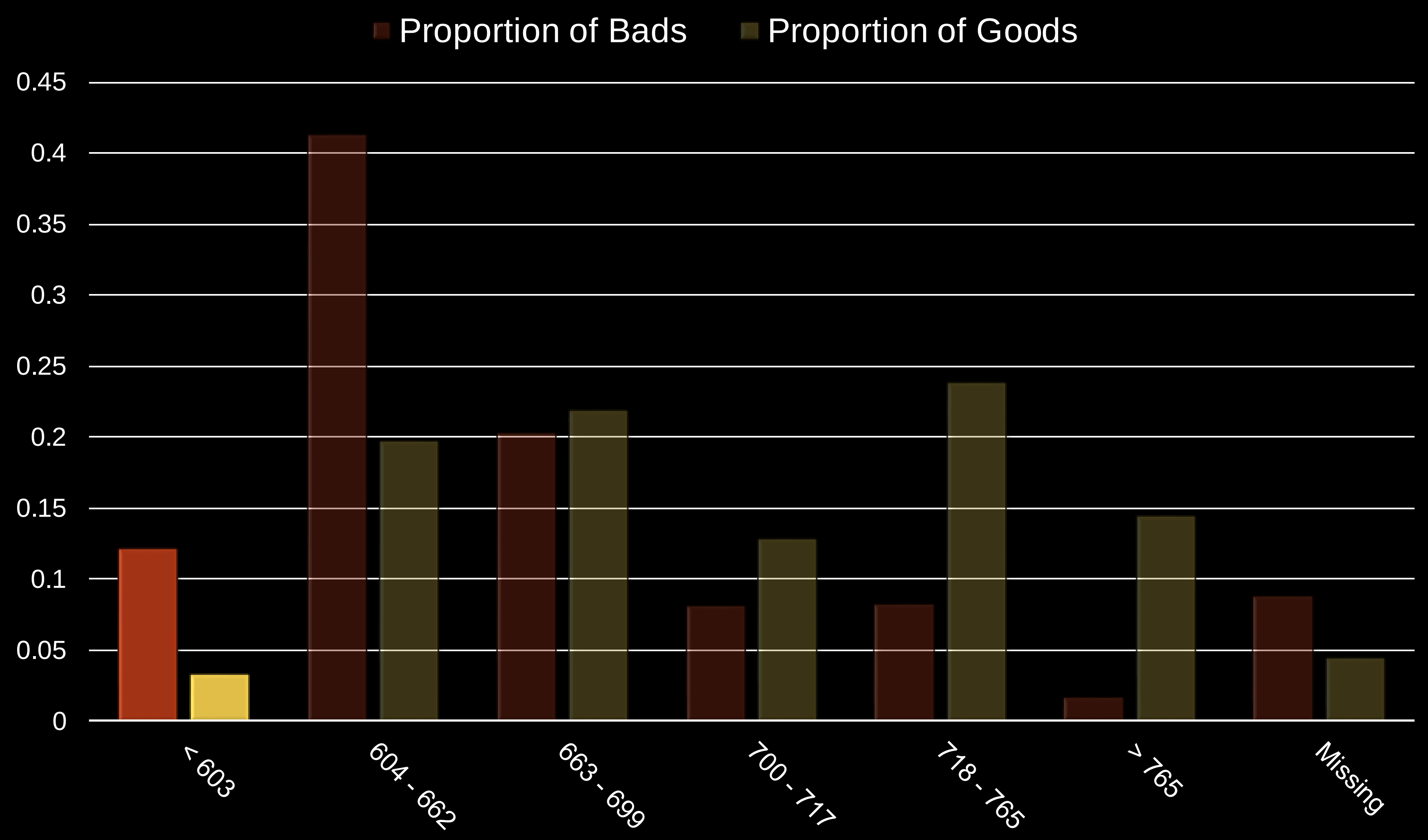
WOE – Example

WOE for Bureau Score				
Group	Values	Event Count	Non-event Count	WOE
1	< 607	129	127	
2	607 – 640	181	285	
3	641 – 653	103	215	
4	654 – 667	86	262	
…	…	…	…	…
12	> 786	5	219	
	MISSING	89	155	
Total		**900**	**3,477**	

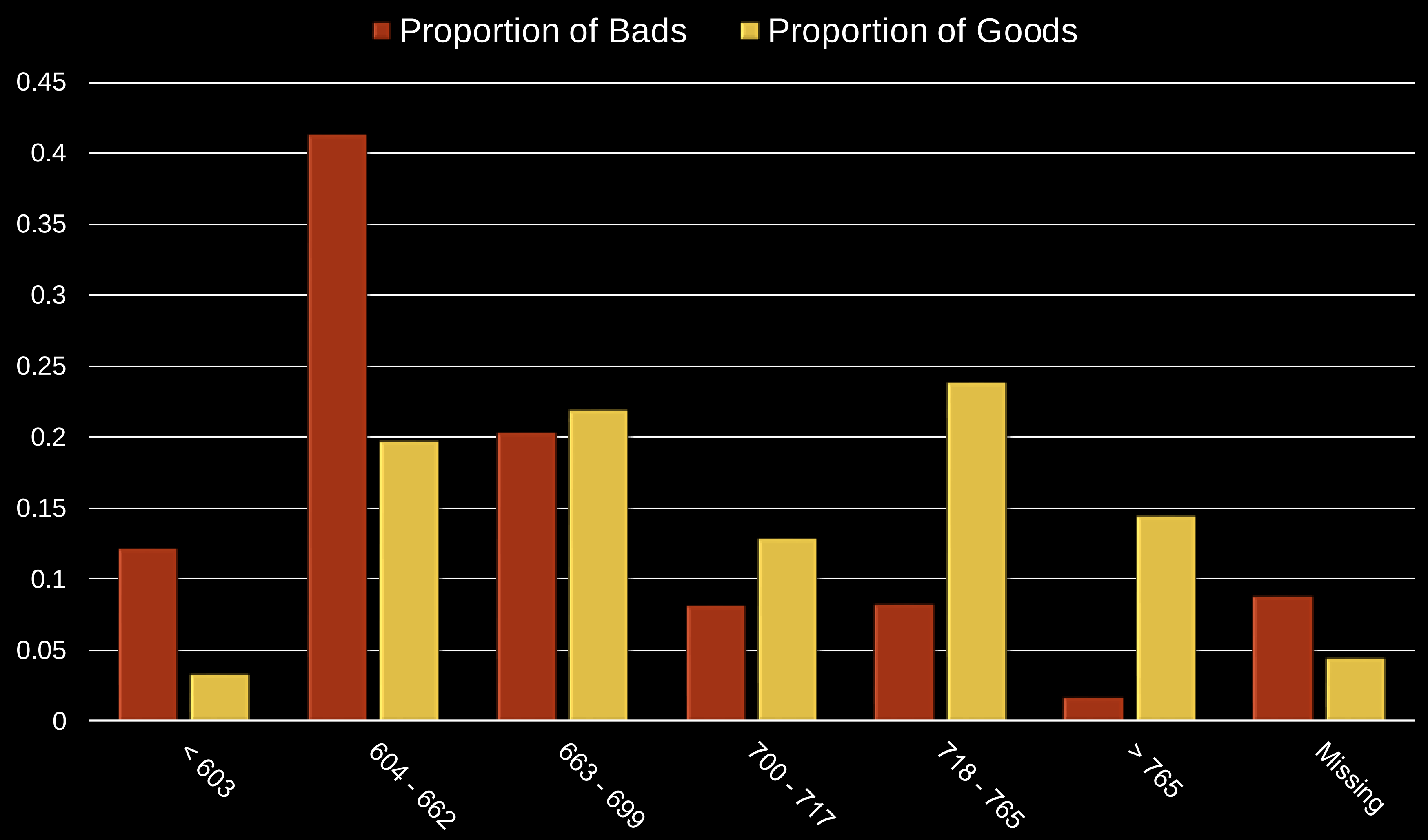
$$Dist.Good_1 = \frac{127}{3477}$$
$$= 0.0365$$

$$Dist.Bad_1 = \frac{129}{900}$$
$$= 0.1433$$

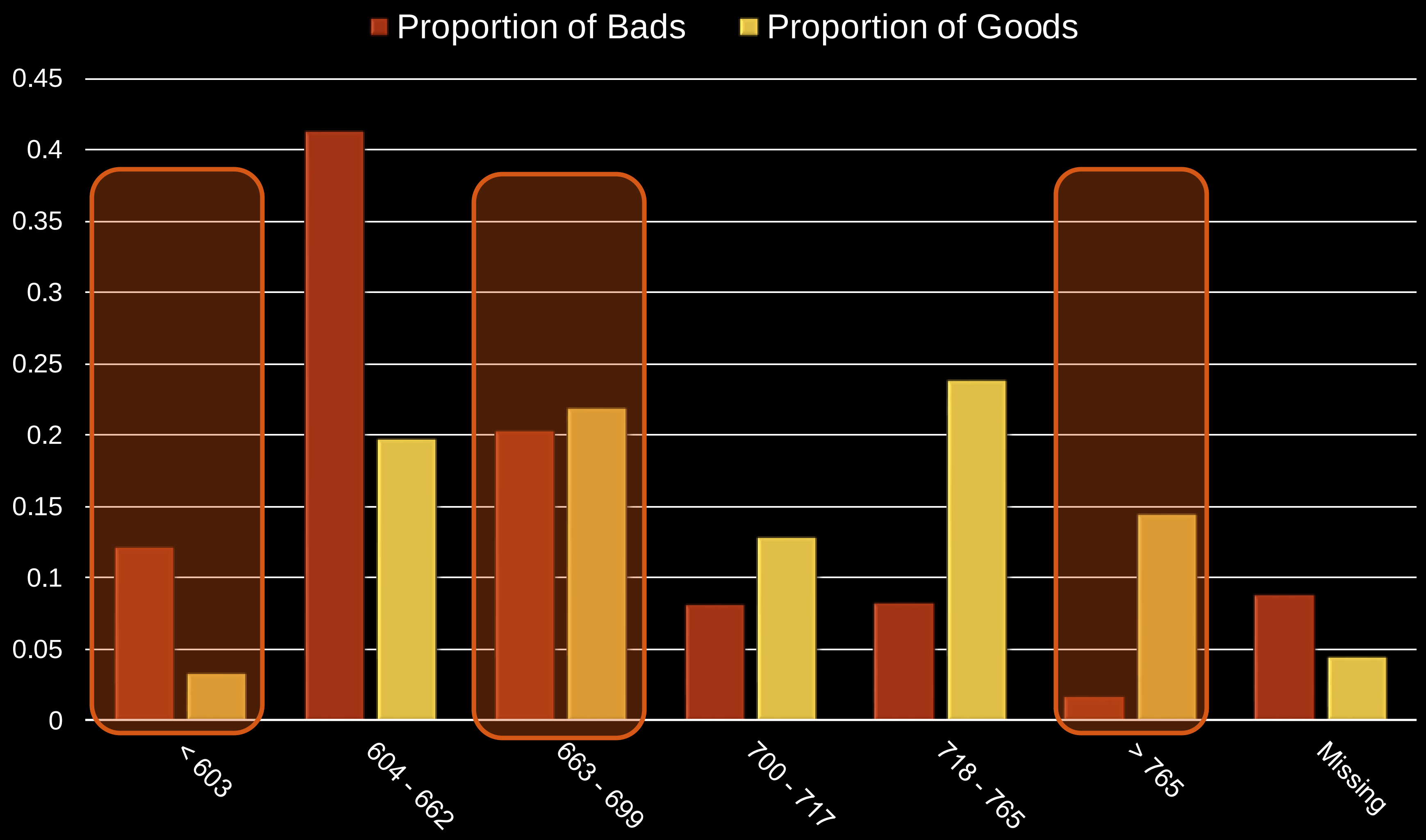
WOE – Example


Proportion of Bads
Proportion of Goods
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
< 603
604 - 662
663 - 699
700 - 717
718 - 765
> 765
Missing

WOE – Example


Proportion of Bads
Proportion of Goods
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
< 603
604 - 662
663 - 699
700 - 717
718 - 765
> 765
Missing

WOE – Example



Proportion of Bads
Proportion of Goods
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
< 603
604 - 662
663 - 699
700 - 717
718 - 765
> 765
Missing

WOE – Example

WOE for Bureau Score				
Group	Values	Event Count	Non-event Count	WOE
1	< 607	129	127	**-1.367**
2	607 – 640	181	285	
3	641 – 653	103	215	
4	654 – 667	86	262	
…	…	…	…	…
12	> 786	5	219	
	MISSING	89	155	
Total		**900**	**3,477**	

$$Dist.Good_1 = \frac{127}{3477}$$

$$= 0.0365$$

$$Dist.Bad_1 = \frac{129}{900}$$

$$= 0.1433$$

$$WOE_1 = \log\left(\frac{0.0365}{0.1433}\right)$$

$$= -1.367$$

WOE – Example

WOE for Bureau Score				
Group	Values	Event Count	Non-event Count	WOE
1	< 607	129	127	**-1.367**
2	607 – 640	181	285	**-0.898**
3	641 – 653	103	215	**-0.616**
4	654 – 667	86	262	**-0.238**
…	…	…	…	**…**
12	> 786	5	219	**2.428**
	MISSING	89	155	**-0.797**
Total		**900**	**3,477**	

Weight of Evidence (WOE)

- WOE measures the strength of the attributes of a characteristic in **separating good and bad accounts**.

$$WOE_i = \log\left(\frac{Dist.Good_i}{Dist.Bad_i}\right)$$

- WOE approximately zero implies what?

Weight of Evidence (WOE)

- WOE measures the strength of the attributes of a characteristic in **separating good and bad accounts**.

$$WOE_i = \log\left(\frac{Dist.Good_i}{Dist.Bad_i}\right)$$

- WOE approximately zero implies % good approximately equal to % bad so group doesn't separate good vs. bad well.

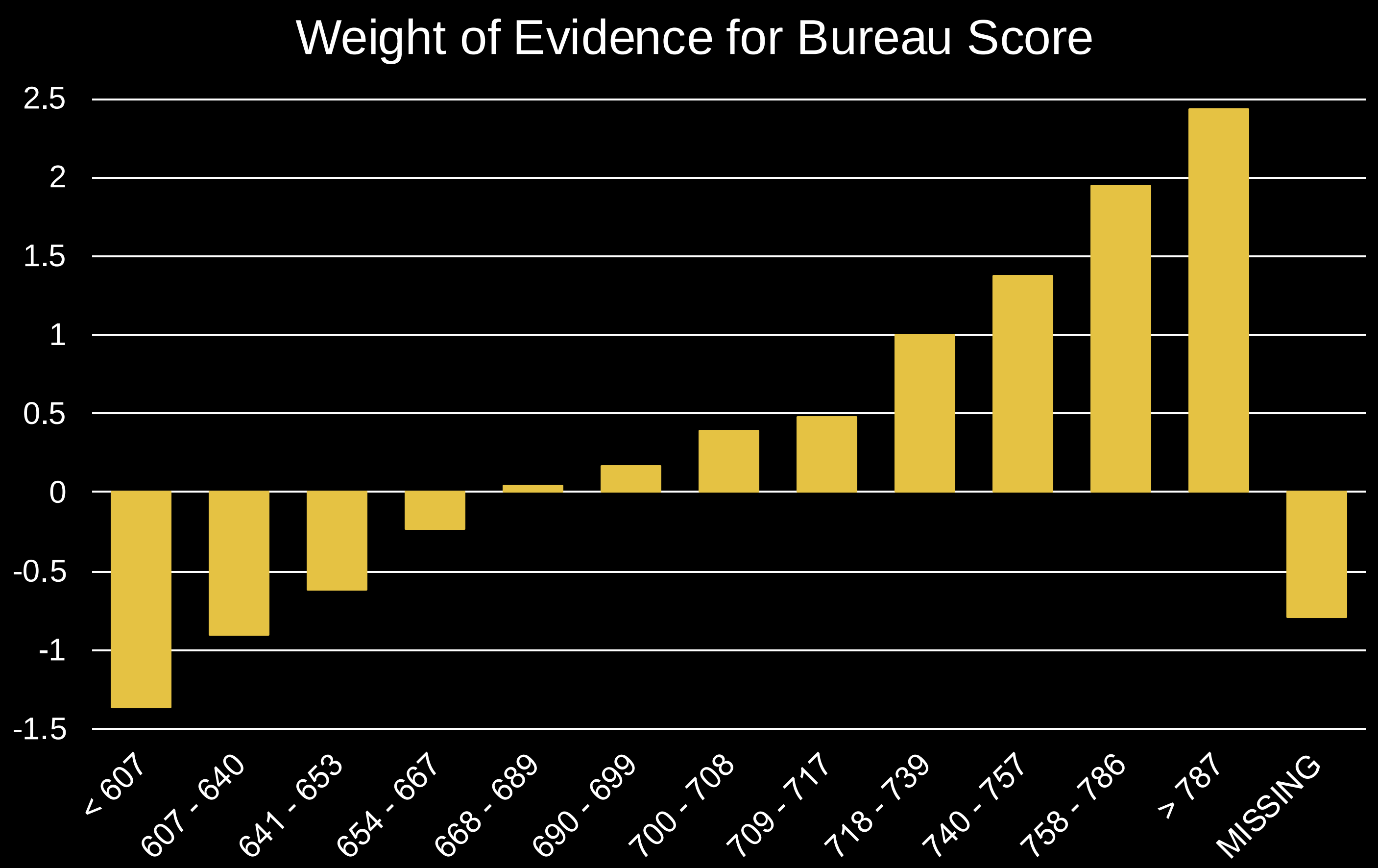
Weight of Evidence (WOE)

- WOE measures the strength of the attributes of a characteristic in **separating good and bad accounts**.

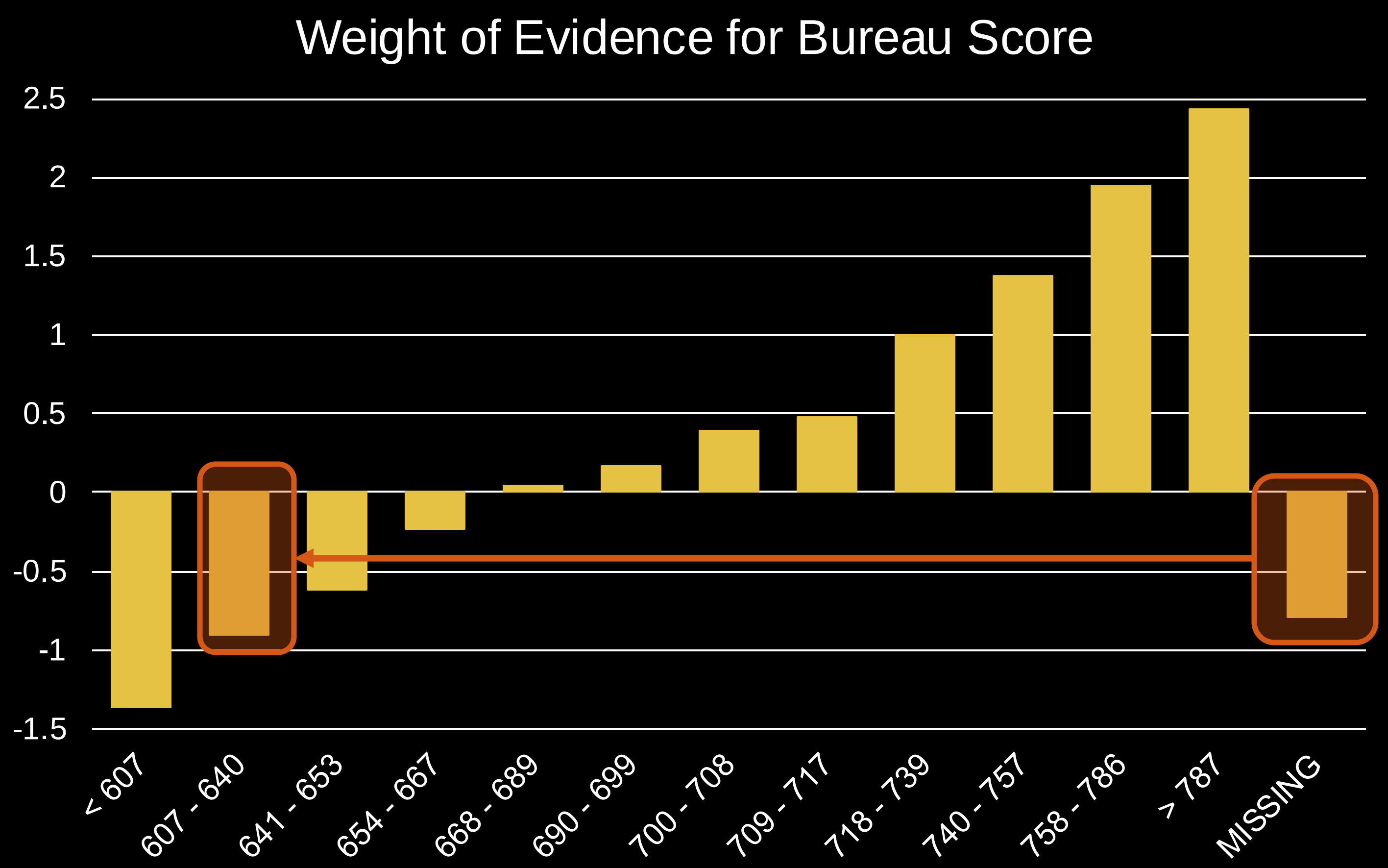
$$WOE_i = \log\left(\frac{Dist.Good_i}{Dist.Bad_i}\right)$$

- WOE approximately zero implies % good approximately equal to % bad so group doesn’t separate good vs. bad well.
- WOE positive implies group identifies people who are good.
- WOE negative implies group identifies people who are bad.

WOE – Example


Weight of Evidence for Bureau Score
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
< 607
607 - 640
641 - 653
654 - 667
668 - 689
690 - 699
700 - 708
709 - 717
718 - 739
740 - 757
758 - 786
> 787
MISSING

WOE – Example



Weight of Evidence for Bureau Score
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
< 607
607 - 640
641 - 653
654 - 667
668 - 689
690 - 699
700 - 708
709 - 717
718 - 739
740 - 757
758 - 786
> 787
MISSING

WOE – Python

```
import numpy as np

from optbinning import OptimalBinning


X = train["bureau_score"]
y = train["bad"]

optbin = OptimalBinning(name = "bureau_score", dtype = "numerical")

optbin.fit(X, y)

optbin.splits
```

```
array([606.5, 640.5, 653.5, 667.5, 689.5, 699.5, 708.5, 717.5, 739.5, 757.5, 786
.5])
```

WOE – Python

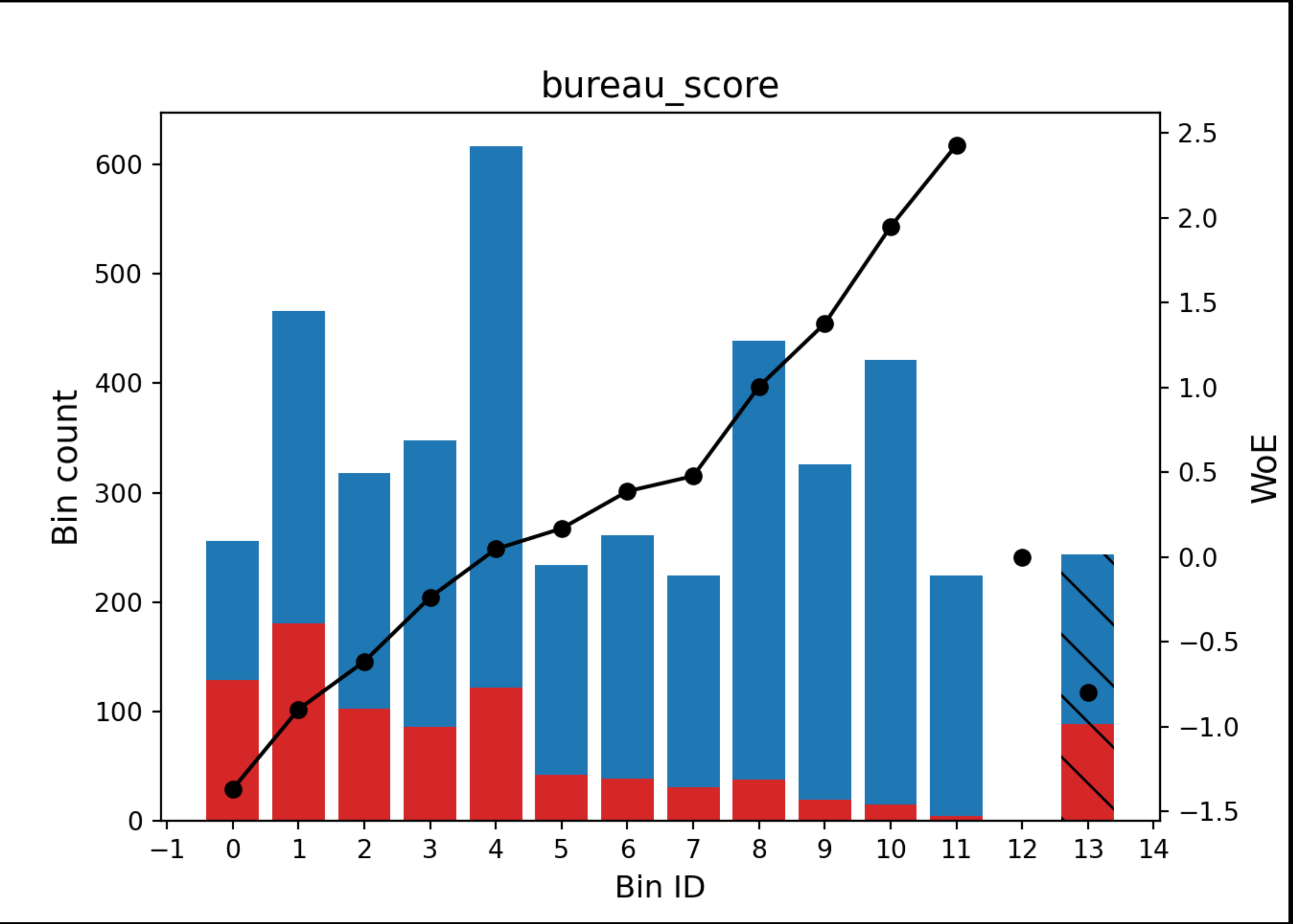
```
iv_table = optbin.binning_table
iv_table.build()
```

WOE – Python

```
                   Bin  Count  Count (%)  ...       WoE        IV        JS
0        (-inf, 606.50)   256   0.058488  ... -1.367156  0.146023  0.016952
1      [606.50, 640.50)   466   0.106466  ... -0.897538  0.106936  0.012936
2      [640.50, 653.50)   318   0.072653  ... -0.615621  0.032388  0.003986
3      [653.50, 667.50)   348   0.079507  ... -0.237533  0.004799  0.000598
4      [667.50, 689.50)   616   0.140736  ...  0.046984  0.000306  0.000038
5      [689.50, 699.50)   234   0.053461  ...  0.168295  0.001439  0.000180
6      [699.50, 708.50)   261   0.059630  ...  0.387585  0.007951  0.000988
7      [708.50, 717.50)   224   0.051177  ...  0.477173  0.010051  0.001245
8      [717.50, 739.50)   439   0.100297  ...  1.004845  0.073461  0.008815
9      [739.50, 757.50)   326   0.074480  ...  1.376322  0.090541  0.010501
10     [757.50, 786.50)   421   0.096185  ...  1.946773  0.194873  0.021120
11        [786.50, inf)   224   0.051177  ...  2.428103  0.139445  0.014113
12              Special     0   0.000000  ...       0.0  0.000000  0.000000
13              Missing   244   0.055746  ... -0.796742  0.043271  0.005270
Totals                   4377   1.000000  ...  0.851485  0.096741
[15 rows x 9 columns]
```

WOE – Python

```
iv_table.plot(metric = “woe”)
```



WOE – Python

```
X = train["purpose"]

y = train["bad"]

optbin = OptimalBinning(name = "purpose", dtype = "categorical")

optbin.fit(X, y)

iv_table = optbin.binning_table

iv_table.build()
```

```
              Bin  Count  Count (%)  ...       WoE        IV        JS
0         [LEASE]   1458   0.333105  ...  0.050268  0.000829  0.000104
1          [LOAN]   2919   0.666895  ... -0.024559  0.000405  0.000051
2         Special      0   0.000000  ...       0.0  0.000000  0.000000
3         Missing      0   0.000000  ...       0.0  0.000000  0.000000
Totals              4377   1.000000  ...  0.001234  0.000154
[5 rows x 9 columns]
```

Separation Issues Remain

- Quasi-complete separation still a problem:

	Non-Event	Event	WOE
A	28	7	**-0.032**
B	16	0	∞
C	94	11	**0.728**
D	23	21	**-1.327**
Total	**161**	**39**	

Adjusted WOE

- Adjust the WOE calculation to account for possible quasi-complete separation:

$$Adjusted\ WOE_i = \log\left(\frac{Dist.Good_i + \eta_1}{Dist.Bad_i + \eta_2}\right)$$

- The η_1 and η_2 parameters are smoothing parameters that correct for potential overfitting and also protect against quasi-complete separation.
- Most software just sets $\eta_1 = \eta_2$ and has one parameter.

Adjusted WOE ($\eta_1 = \eta_2 = 0.005$)

- Quasi-complete separation no longer a problem:

	Non-Event	Event	WOE
A	28	7	**-0.031**
B	16	0	**3.039**
C	94	11	**0.719**
D	23	21	**-1.302**
Total	**161**	**39**	

Smoothed WOE (SWOE)

- SAS has recently proposed a slightly different smoothed version of the WOE calculation to account for possible quasi-complete separation:

$$SWOE_i = \log\left(\frac{\#Bad_i + (Overall\ Prop.Bad) \times c}{\#Good_i + (Overall\ Prop.Good) \times c}\right)$$

- This is just a smoothing parameter put in a slightly different place in the WOE calculation based on more Bayesian inference techniques.
- Haven't seen it really used elsewhere.

INFORMATION VALUE

Information Value (IV)

- How big is a “big” difference when looking across groups for WOE?
- IV measures the ability of the characteristic to separate goods vs. bads.

$$IV = \sum_{i=1}^{L} (Dist.Good_i - Dist.Bad_i) \times \log\left(\frac{Dist.Good_i}{Dist.Bad_i}\right)$$

Information Value (IV)

- How big is a “big” difference when looking across groups for WOE?
- IV measures the ability of the characteristic to separate goods vs. bads.

$$IV = \sum_{i=1}^{L} (Dist.Good_i - Dist.Bad_i) \times \boxed{\log\left(\frac{Dist.Good_i}{Dist.Bad_i}\right)}$$

Weight of Evidence!

Information Value (IV)

- How big is a "big" difference when looking across groups for WOE?
- IV measures the ability of the characteristic to separate goods vs. bads.

$$IV = \sum_{i=1}^{L} (Dist.Good_i - Dist.Bad_i) \times \log\left(\frac{Dist.Good_i}{Dist.Bad_i}\right)$$

- Used to select characteristics with strong predictive value.

Information Value (IV)

- Characteristics of IV:
  - $IV \geq 0$
  - Bigger is Better!

- Rules of Thumb:
  - IV < 0.02 – Not predictive
  - 0.02 < IV < 0.1 – Weak predictor
  - 0.1 < IV < 0.25 – Medium predictor
  - 0.25 < IV – Strong predictor

Information Value (IV) – Python

```
from optbinning import BinningProcess


colnames = list(train.columns[0:20])
X = train[colnames]


selection_criteria = {"iv": {"min": 0.1, "max": 1}}


bin_proc = BinningProcess(colnames,
                          selection_criteria = selection_criteria,
                          categorical_variables = ["bankruptcy",
                                                   "purpose",
                                                   "used_ind"])


iv_all = bin_proc.fit(X, y).summary()
iv_all[iv_all.columns[0:6]].sort_values(by = ["iv"], ascending = False)
```

Information Value (IV) – Python

	name	dtype	status	selected	n_bins	iv
10	bureau_score	numerical	OPTIMAL	True	12	0.851485
8	tot_rev_line	numerical	OPTIMAL	True	10	0.512358
9.	rev_util	numerical	OPTIMAL	True	10	0.363525
4	age_oldest_tr	numerical	OPTIMAL	True	10	0.322788
2	tot_derog	numerical	OPTIMAL	True	6	0.244956
17	ltv	numerical	OPTIMAL	True	7	0.193863
3	tot_tr	numerical	OPTIMAL	True	8	0.187039
7	tot_rev_debt	numerical	OPTIMAL	True	5	0.114815
13	down_pyt	numerical	OPTIMAL	True	5	0.113488
18	tot_income	numerical	OPTIMAL	True	9	0.108861
6	tot_rev_tr	numerical	OPTIMAL	True	6	0.1028
12	msrp	numerical	OPTIMAL	False	8	0.060786
15	loan_term	numerical	OPTIMAL	False	5	0.059987
11	purch_price	numerical	OPTIMAL	False	6	0.046384
5	tot_open_tr	numerical	OPTIMAL	False	5	0.040953
1	app_id	numerical	OPTIMAL	False	6	0.029569
19	used_ind	categorical	OPTIMAL	False	2	0.019464
16	loan_amt	numerical	OPTIMAL	False	4	0.014432
0	bankruptcy	categorical	OPTIMAL	False	2	0.001872
14	purpose	categorical	OPTIMAL	False.	2	0.001234

Information Value (IV) – Python

	name	dtype	status	selected	n_bins	iv
10	bureau_score	numerical	OPTIMAL	True	12	0.851485
8	tot_rev_line	numerical	OPTIMAL	True	10	0.512358
9.	rev_util	numerical	OPTIMAL	True	10	0.363525
4	age_oldest_tr	numerical	OPTIMAL	True	10	0.322788
2	tot_derog	numerical	OPTIMAL	True	6	0.244956
17	ltv	numerical	OPTIMAL	True	7	0.193863
3	tot_tr	numerical	OPTIMAL	True	8	0.187039
7	tot_rev_debt	numerical	OPTIMAL	True	5	0.114815
13	down_pyt	numerical	OPTIMAL	True	5	0.113488
18	tot_income	numerical	OPTIMAL	True	9	0.108861
6	tot_rev_tr	numerical	OPTIMAL	True	6	0.1028
12	msrp	numerical	OPTIMAL	False	8	0.060786
15	loan_term	numerical	OPTIMAL	False	5	0.059987
11	purch_price	numerical	OPTIMAL	False	6	0.046384
5	tot_open_tr	numerical	OPTIMAL	False	5	0.040953
1	app_id	numerical	OPTIMAL	False	6	0.029569
19	used_ind	categorical	OPTIMAL	False	2	0.019464
16	loan_amt	numerical	OPTIMAL	False	4	0.014432
0	bankruptcy	categorical	OPTIMAL	False	2	0.001872
14	purpose	categorical	OPTIMAL	False.	2	0.001234

Information Value (IV) – Python

	name	dtype	status	selected	n_bins	iv
10	bureau_score	numerical	OPTIMAL	True	12	0.851485
8	tot_rev_line	numerical	OPTIMAL	True	10	0.512358
9.	rev_util	numerical	OPTIMAL	True	10	0.363525
4	age_oldest_tr	numerical	OPTIMAL	True	10	0.322788
2	tot_derog	numerical	OPTIMAL	True	6	0.244956
17	ltv	numerical	OPTIMAL	True	7	0.193863
3	tot_tr	numerical	OPTIMAL	True	8	0.187039
7	tot_rev_debt	numerical	OPTIMAL	True	5	0.114815
13	down_pyt	numerical	OPTIMAL	True	5	0.113488
18	tot_income	numerical	OPTIMAL	True	9	0.108861
6	tot_rev_tr	numerical	OPTIMAL	True	6	0.1028
12	msrp	numerical	OPTIMAL	False	8	0.060786
15	loan_term	numerical	OPTIMAL	False	5	0.059987
11	purch_price	numerical	OPTIMAL	False	6	0.046384
5	tot_open_tr	numerical	OPTIMAL	False	5	0.040953
1	app_id	numerical	OPTIMAL	False	6	0.029569
19	used_ind	categorical	OPTIMAL	False	2	0.019464
16	loan_amt	numerical	OPTIMAL	False	4	0.014432
0	bankruptcy	categorical	OPTIMAL	False	2	0.001872
14	purpose	categorical	OPTIMAL	False.	2	0.001234

Information Value (IV)

- Characteristics of IV:
  - $IV \geq 0$
  - Bigger is Better!

- Rules of Thumb:
  - IV < 0.02 – Not predictive
  - 0.02 < IV < 0.1 – Weak predictor
  - 0.1 < IV < 0.25 – Medium predictor
  - 0.25 < IV < 0.5 – Strong predictor
  - IV > 0.5 – Over-predicting?

Information Value (IV)

- Rules of Thumb:
  - IV < 0.02 – Not predictive
  - 0.02 < IV < 0.1 – Weak predictor
  - 0.1 < IV < 0.25 – Medium predictor
  - 0.25 < IV < 0.5 – Strong predictor
  - IV > 0.5 – Over-predicting?
- Over-predicting Example:
  - All previous mortgage decisions have been made only on bureau score so of course bureau score is highly predictive – becomes only significant variable!
  - Create two models – one with bureau score, one without bureau score and **ensemble**.

GINI STATISTIC

Gini Statistic

- **Gini statistic** is optional technique that tries to answer the same question as Information Value – which variables are strong enough to enter the scorecard model?
- IV is more in line with WOE calculation and used more often.
- Characteristics:
  - Range is 0 to 100.
  - Bigger is Better.

Gini Statistic Calculation

- More complicated technique for trying to evaluate how characteristics separate good from bad.
- Majority of the time Gini and IV agree but could be different on the borderline cases.
- Calculation:
  - Sort L groups of variable by descending order of the proportion of all events.

$$Gini = \left(1 - \frac{\left(2\sum_{i=2}^{L}\left(n_{i,E} \times \sum_{i=1}^{i-1} n_{i,NE}\right) + \sum_{i=1}^{L}\left(n_{i,E} \times n_{i,NE}\right)\right)}{N_E \times N_{NE}}\right) \times 100$$

Gini Statistic Calculation

- More complicated technique for trying to evaluate how characteristics separate good from bad.
- Majority of the time Gini and IV agree but could be different on the borderline cases.
- Calculation:
  - Sort *L* groups of variable by descending order of the proportion of all events.

$$Gini = \left(1 - \frac{\left(2\sum_{i=2}^{L}\left(n_{i,E} \times \sum_{i=1}^{i-1} n_{i,NE}\right) + \sum_{i=1}^{L}\left(n_{i,E} \times n_{i,NE}\right)\right)}{N_E \times N_{NE}}\right) \times 100$$

Number of events in group *i*

Gini Statistic Calculation

- More complicated technique for trying to evaluate how characteristics separate good from bad.
- Majority of the time Gini and IV agree but could be different on the borderline cases.
- Calculation:
  - Sort *L* groups of variable by descending order of the proportion of all events.

$$Gini = \left(1 - \frac{\left(2\sum_{i=2}^{L}\left(n_{i,E} \times \sum_{i=1}^{i-1} n_{i,NE}\right) + \sum_{i=1}^{L}\left(n_{i,E} \times n_{i,NE}\right)\right)}{N_E \times N_{NE}}\right) \times 100$$

Number of non-events in group *i*

Gini Statistic Calculation

- More complicated technique for trying to evaluate how characteristics separate good from bad.
- Majority of the time Gini and IV agree but could be different on the borderline cases.
- Calculation:
  - Sort *L* groups of variable by descending order of the proportion of all events.

$$Gini = \left(1 - \frac{\left(2\sum_{i=2}^{L}\left(n_{i,E} \times \sum_{i=1}^{i-1} n_{i,NE}\right) + \sum_{i=1}^{L}\left(n_{i,E} \times n_{i,NE}\right)\right)}{N_E \times N_{NE}}\right) \times 100$$

Total number of events and non-events