

The Essential of Credit Scoring Model Yang Harus dikuasai dalam Pembuatan Model Skoring



Bagus Sartono
2019

Outline

0830-1000	Pengantar mengenai Model Credit Scoring dan menilai kebaikan dari suatu model
1000-1015	Break
1015-1200	Pemodelan prediktif: regresi logistik dan diskretisasi
1200-1300	Lunch Break
1300-1400	Tahapan umum pembuatan model scoring
1400-1500	Penghitungan nilai WoE dan Information Value
1500-1530	Break
1530-1615	Pemodelan awal credit scoring

0830-1000	Validasi model
1000-1015	Break
1015-1200	Penskalaan dan pembuatan scorecard
1200-1300	Lunch Break
1300-1500	Execise
1500-1530	Break
1530-1615	Review dan Diskusi

Pengantar Mengenai Model Credit Scoring dan Menilai Keباikan Dari Suatu Model

bagusco

Apa itu model skoring kredit?

- Model statistika yang berguna dalam menghasilkan skor sebagai bahan untuk mengambil keputusan mengenai kategori resiko kredit (calon) nasabah, baik perorangan maupun perusahaan.
 - Kategori resiko ini hanya merupakan dugaan, sehingga ada kemungkinan bahwa keputusan yang diambil adalah salah.
- bagusco
- Perlu proses yang baik untuk menyusun model skoring agar tingkat kesalahan itu minimum.

Ilustrasi: Scorecard dan Threshold

Attribute	Category	Score
Gender	FEMALE	165
	MALE	107
Age Group	<= 25	108
	25 – 30	123
	31 – 35	121
	36 – 40	133
	41 – 45	128
	> 45	186
Residence Ownership	OTHERS	98
	OWNED	172
	PARENT	115
	RENT	83
Number of Dependants	0	169
	1	140
	2	136
	3	122
	4	89
	> 4	92

Score	Odds (good)
640	200.0
620	100.0
600	50.0
580	25.0
560	12.5
540	6.3
520	3.1
500	1.6
480	0.8
460	0.4
440	0.2

bagusco
“Accepted
(Good) jika
score lebih
dari 540”

Bagaimana menggunakan scorecard?

Attribute	Category	Score
Gender	FEMALE	165
	MALE	107
Age Group	<= 25	108
	25 – 30	123
	31 – 35	121
	36 – 40	133
	41 – 45	128
	> 45	186
Residence Ownership	OTHERS	98
	OWNED	172
	PARENT	115
	RENT	83
Number of Dependants	0	169
	1	140
	2	136
	3	122
	4	89
	> 4	92



Female 165

37 yr old 133

rent house 83

1 dependant 140

Total Score 521

Decision REJECTED

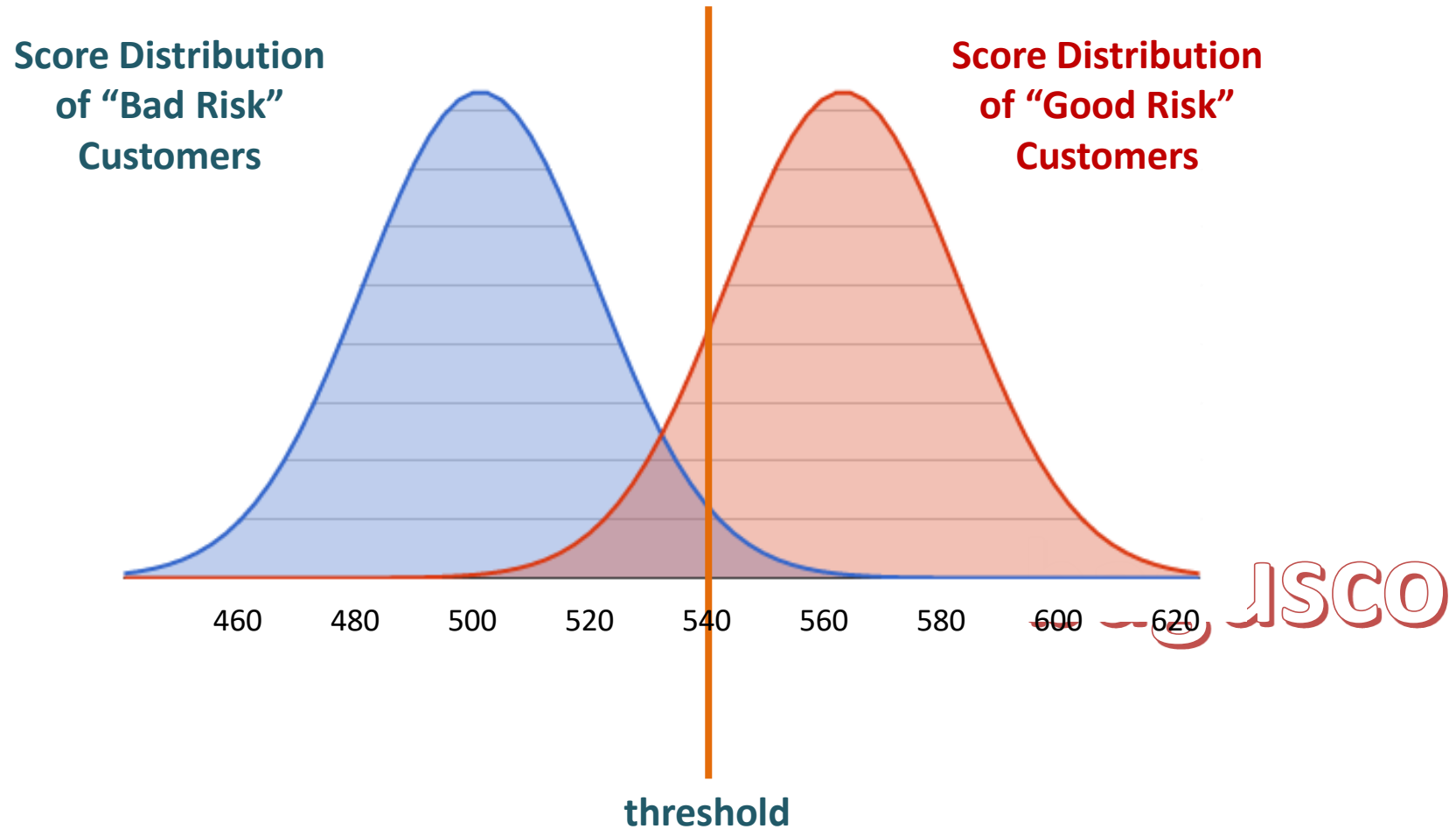
Apakah itu scorecard yang bagus?

Memberikan skor berbeda antara nasabah GOOD dan nasabah BAD

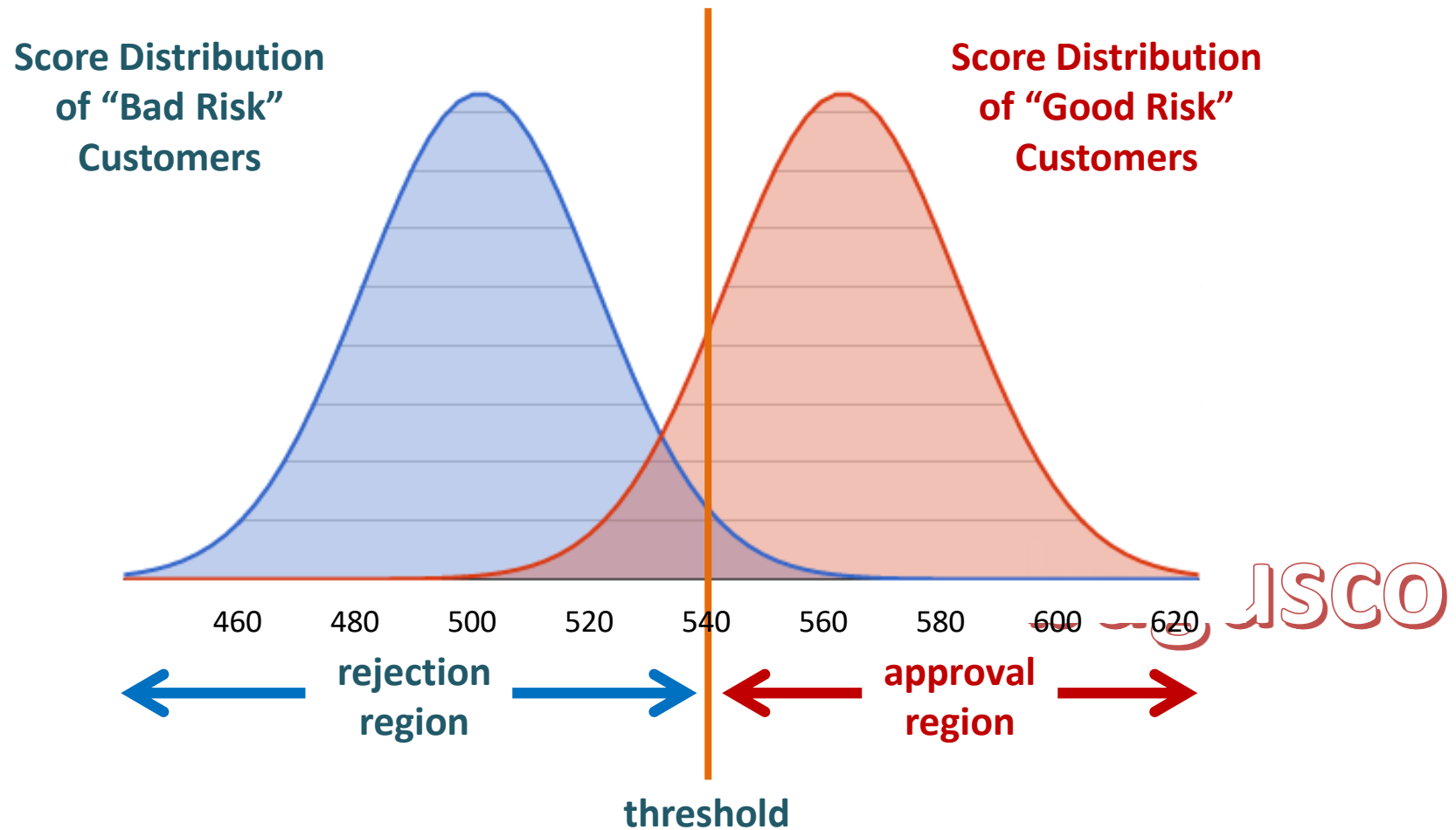
Memberikan padanan skor dan resiko sesuai dengan rancangan pembuatannya

bagusco

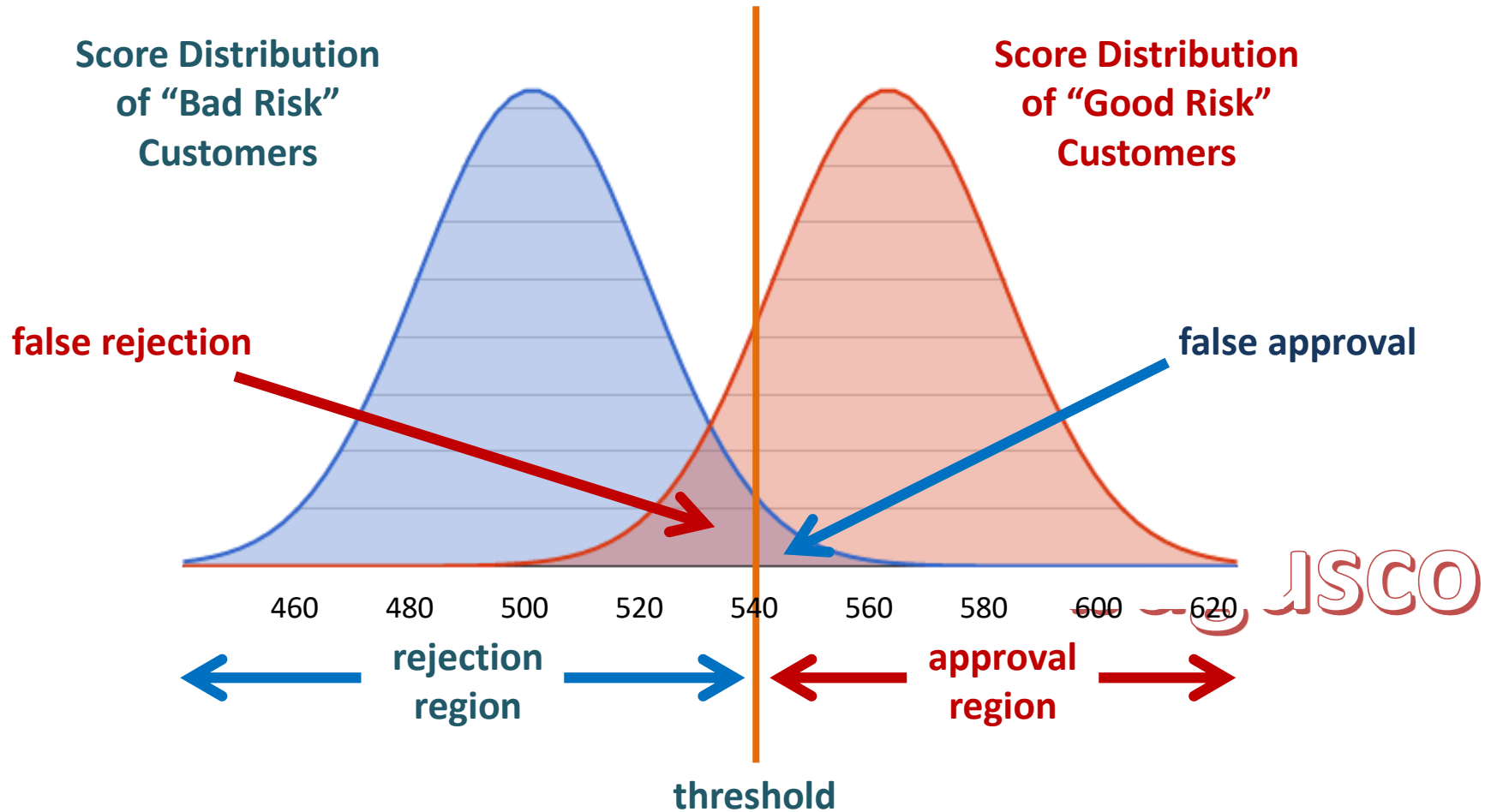
Apakah itu scorecard yang bagus?



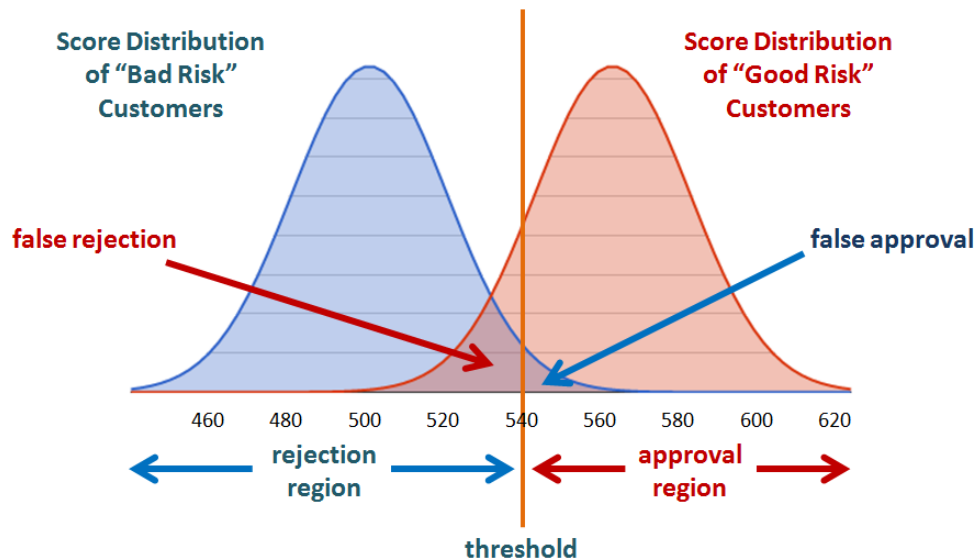
Apakah itu scorecard yang bagus?



Apakah itu scorecard yang bagus?



Apakah itu scorecard yang bagus?



Scorecard yang bagus memiliki false rate yang rendah:

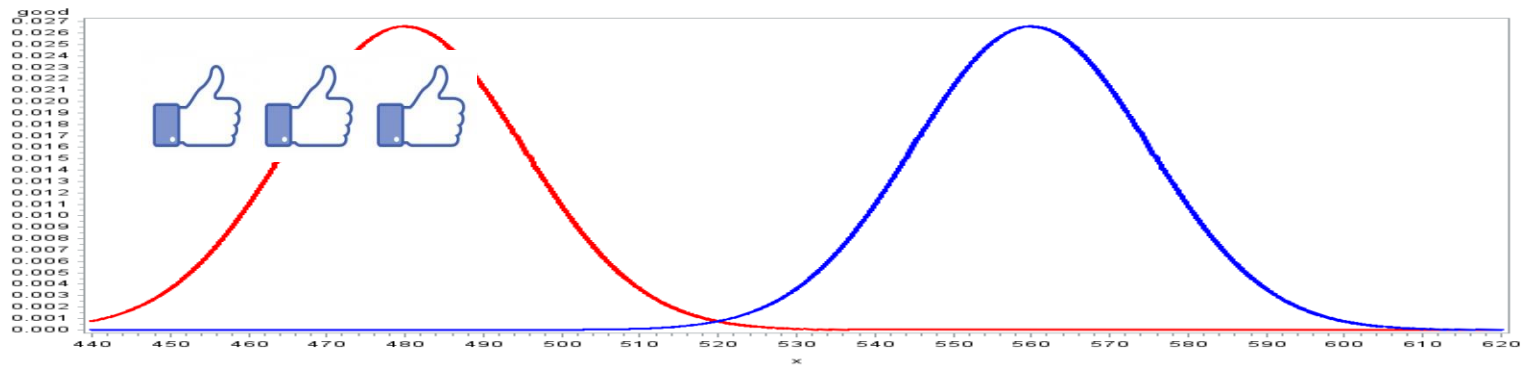
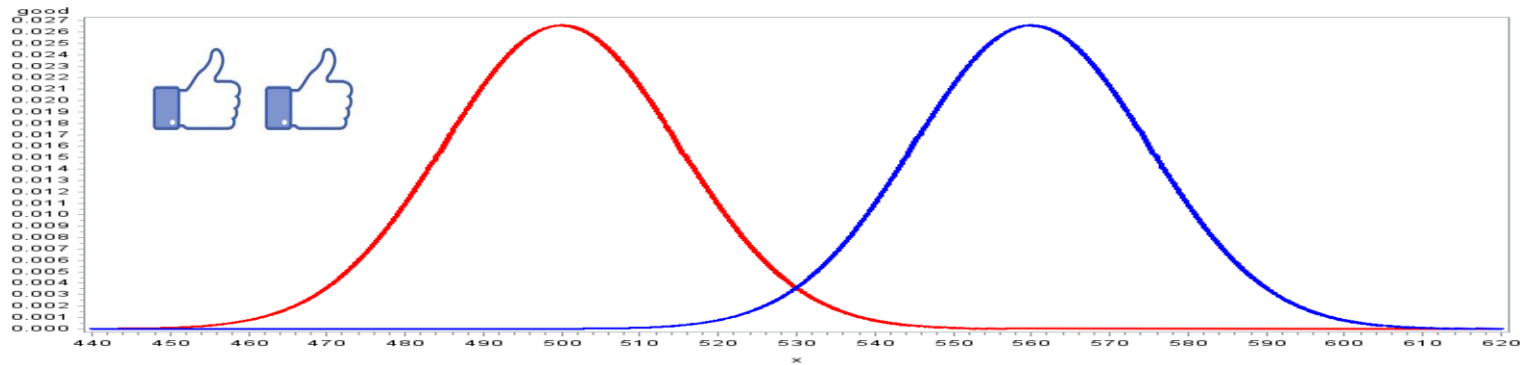
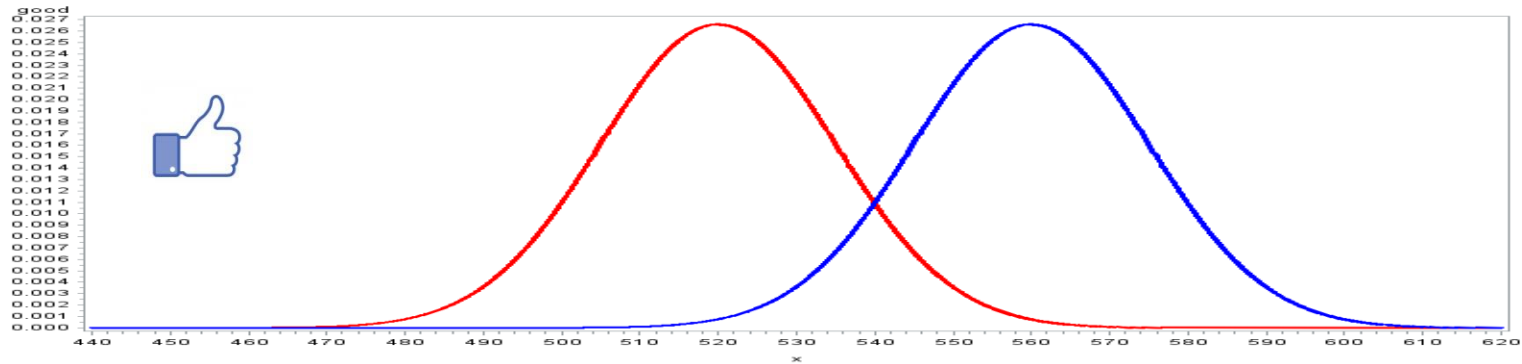
- False approval rate
- False rejection rate
- Total false rate

bagusco

catatan:

- 1 – False approval rate = sensitivity
- 1 – False rejection rate = specificity
- 1 – Total false rate = accuracy

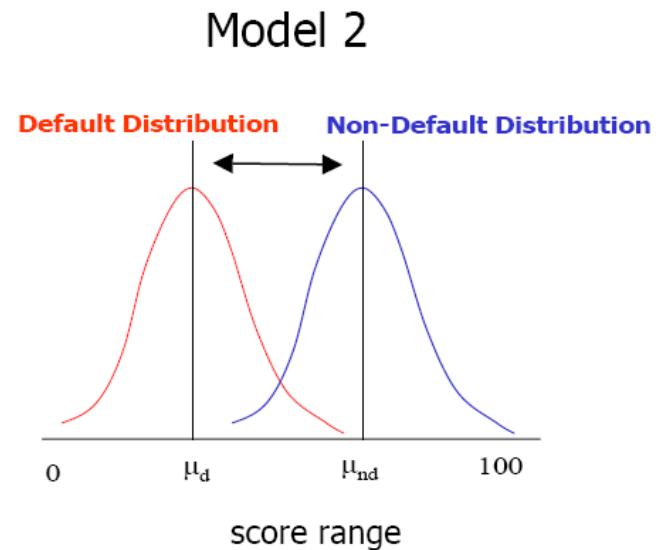
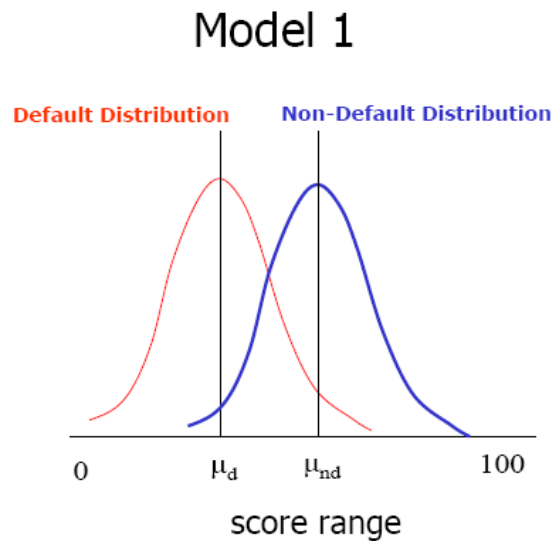
Apakah itu scorecard yang bagus?



CO

Mengevaluasi Model Skoring

- Kolmogorov-Smirnov (KS)
- Melihat apakah model mampu menghasilkan skor yang dapat membedakan Bad-Good



ISCO

- Empirical Distribution Function based test

K-S Statistic

The *empirical distribution function* (EDF) of a sample is defined as the following function:

$$F(x) = \frac{1}{n} (\text{number of } x_j \leq x)$$

If there are two class levels, two-sample Kolmogorov-Smirnov test statistic D as

$$D = \max_j | F_1(x_j) - F_2(x_j) | \quad \text{where } j = 1, 2, \dots, n$$

hagrusco

Model Assessment using K-S Statistic

Band	#Bad	#Good	%Bad	%Good	Cum% Bad	Cum% Good	Diff
1							
2	1		2		3	4	
3							
...							
...							
k							

ISCO

5

KS statistic = maximum diff

Model Assessment using K-S test

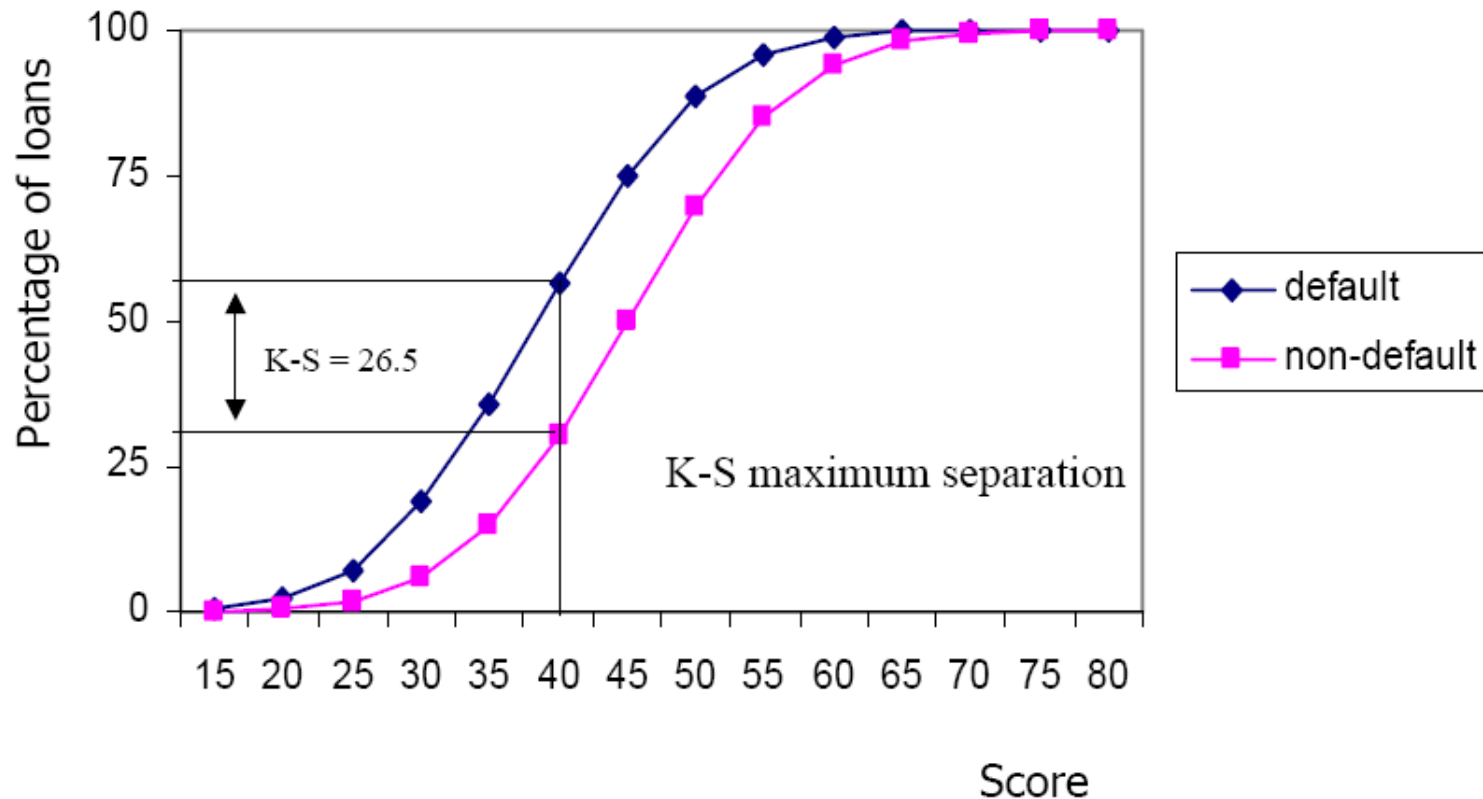
Obs (i)	Score Range		Distributions		Cumulative Distributions		K-S
	lower	upper	Default (#)	Non-Def (#)	Default (%)	Non-Def (%)	
1	0	15	82	458	0.34	0.05	0.29
2	15	20	428	3205	2.12	0.37	1.75
3	20	25	1235	13886	7.24	1.75	5.49
4	25	30	2778	41657	18.77	5.92	12.85
5	30	35	4074	91645	35.69	15.09	20.60
6	35	40	5092	152741	56.82	30.36	26.46
7	40	45	4365	196381	74.94	50.00	24.94
8	45	50	3274	196381	88.53	69.64	18.89
9	50	55	1698	152741	95.58	84.91	10.67
10	55	60	764	91645	98.75	94.08	4.67
11	60	65	232	41657	99.71	98.24	1.47
12	65	70	58	13886	99.95	99.63	0.32
13	70	75	9	3205	99.99	99.95	0.04
14	75	80	1	458	100	100	0.00
15	80	100	1	31	100	100	0

$(82+428)/24092$

Total Bad = 24092



Kolmogorov-Smirnov test visualization



Population Stability Index

- Compares closeness of the predicted defaults to actual defaults by score
 - Exp % corresponds to the predicted default rate in a score band
 - Act % is the actual default rate in a score band
- Stability index =

$$100 * \sum_{i=1}^{NbBands} \left((Exp\%(i) - Act\%(i)) * \ln \left(\frac{Exp\%(i)}{Act\%(i)} \right) \right)$$

- As the index lower, the two population's characteristics become similar

Normal	Caution	Danger
< 10	10 – 25	> 25

Hands-On

- Ada data.... berisi nilai-nilai variabel prediktor dan status good/bad
- Berikan skor sesuai scorecard di atas
- Lihat perbedaan sebarannya antara nasabah good dan nasabah bad... hitung KS-nya
- Lihat peluang setiap band... bandingkan dengan rancangan yang ada pada scorecard

bagusco

PENGENALAN PEMODELAN bagusco

REGRESI LOGISTIK BINER

Pemodelan

- Membangun miniatur dari dunia nyata
 - dinyatakan dalam satu atau beberapa fungsi matematis
- Menyederhanakan fenomenya nyata sehingga mudah memahami pola umum yang ada
 - memberikan penjelasan terhadap perubahan
 - memberikan penjelasan tentang perbedaan yang terjadi
 - menemukan faktor yang menyebabkan perubahan dan perbedaan

bagusco

Komponen Model

$$y = f(x_1, x_2, \dots, x_p) + \varepsilon$$

Y	X
Output Target	Input
Respon	Penjelas (explanatory) Prediktor (predictor) Faktor (factors)
Dependent	Independent

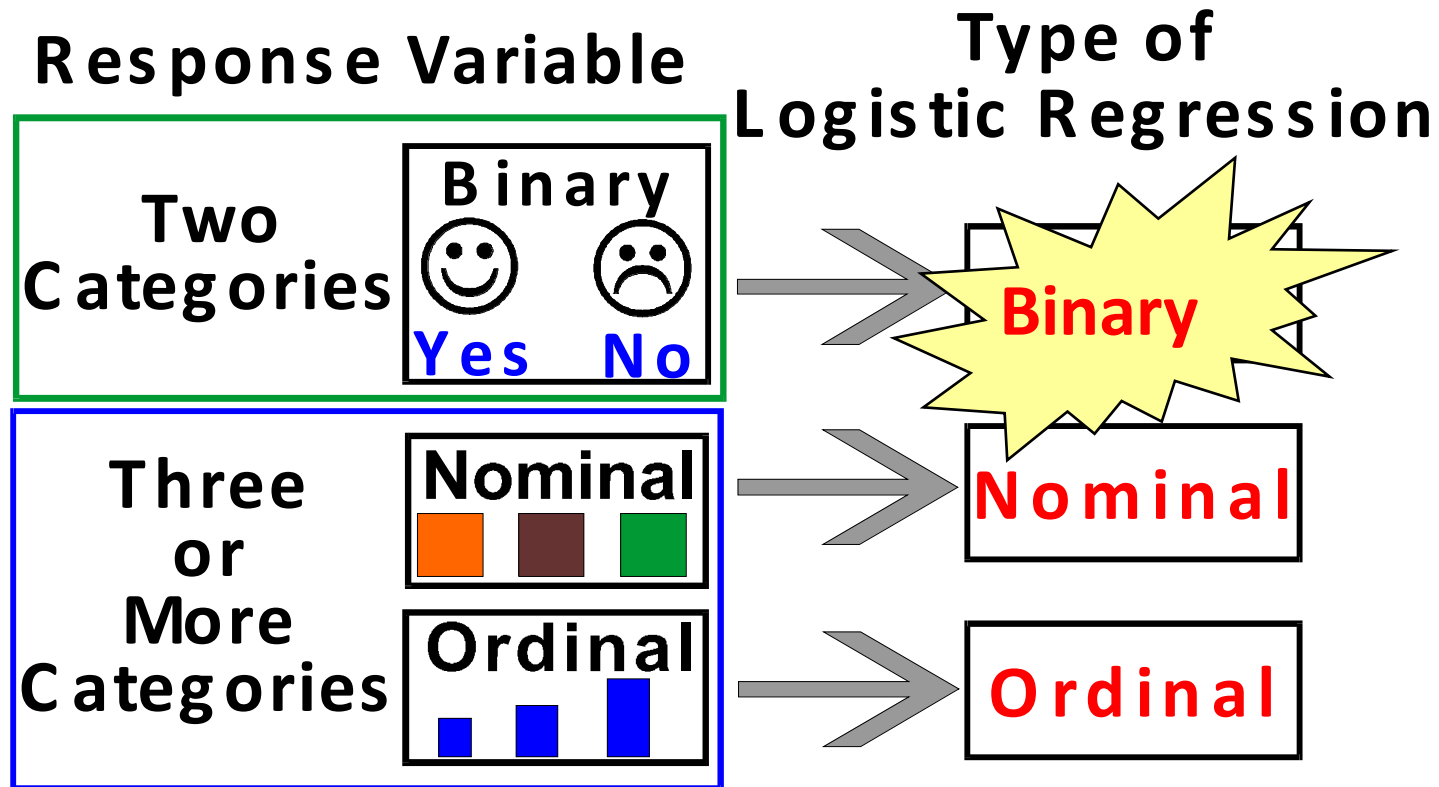
ISCO

Models

- Powerful predictors for optimizing performance
- Powerful summaries for understanding
- Used to explore data set
- Are not perfect
 - “All models are wrong, but some are useful”
 - “Statisticians, like artists, have the bad habit of falling in love with their models”



Types of Logistic Regression



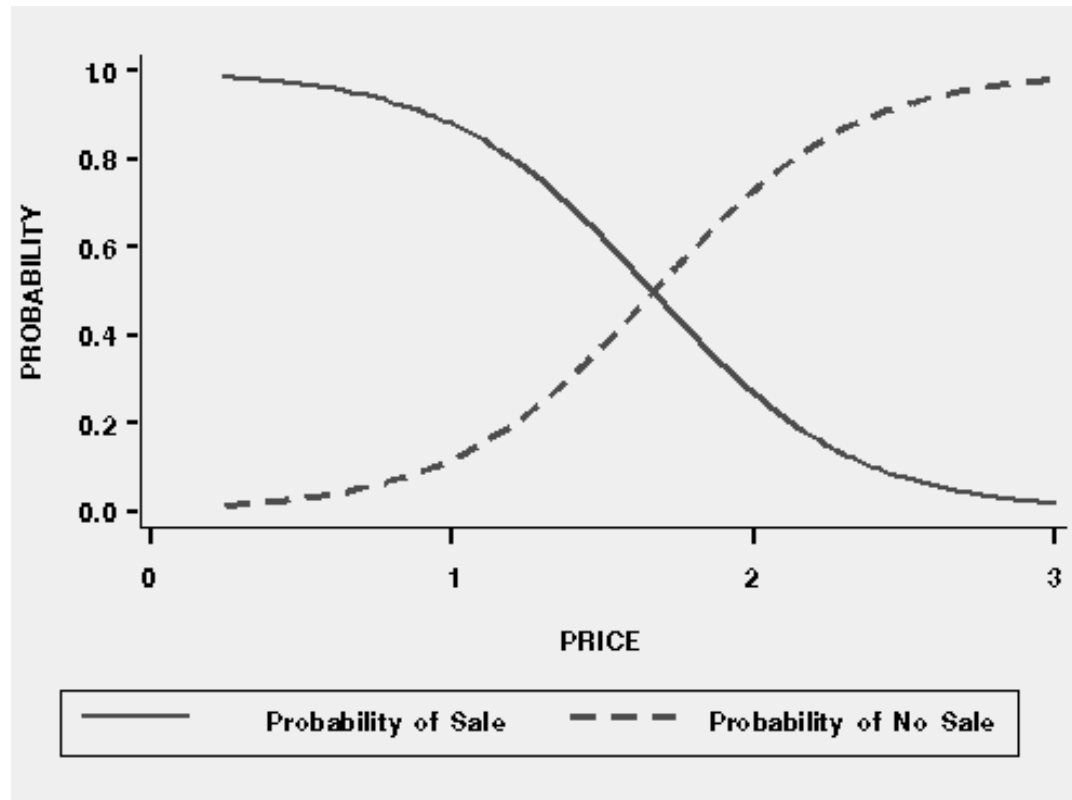
What Does Logistic Regression Do?

The logistic regression model uses the predictor variables, which can be **categorical or continuous**, to predict the probability of specific outcomes.

In other words, logistic regression is designed to describe probabilities associated with the values of the response variable.

bagusco

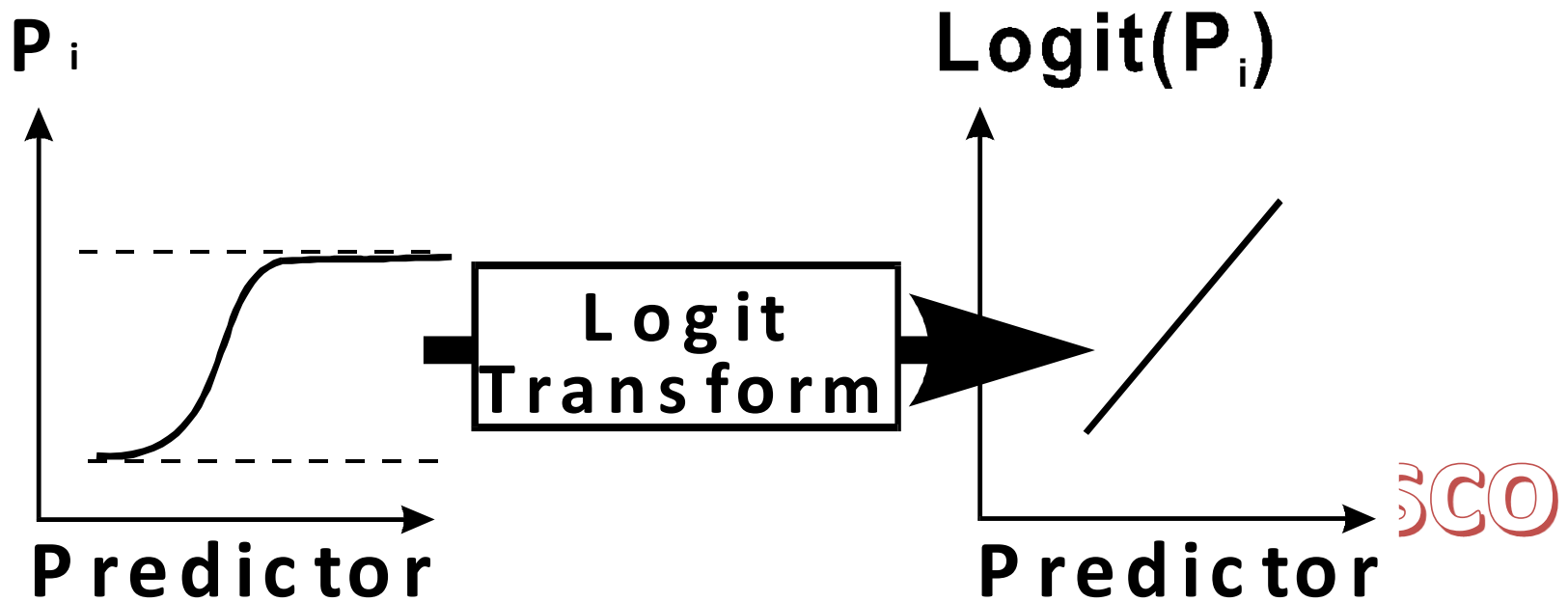
Logistic Regression Curves



gusco

This graph shows the relationship between the probability of SALE to PRICE.

Assumption



Logit Transformation

Logistic regression models transformed probabilities called logits.

where $\text{logit}(p_i) = \log\left(\frac{p_i}{1-p_i}\right)$
 i indexes all cases (observations).

p_i is the probability the event (a sale, for example) occurs in the i^{th} case.

\log is the natural log (to the base e).

bagus.co

Logistic Regression Model

$$\text{logit}(p_i) = \beta_0 + \beta_1 X_1$$

where

$\text{logit}(p_i)$ logit transformation of the probability of the event

β_0 intercept of the regression line

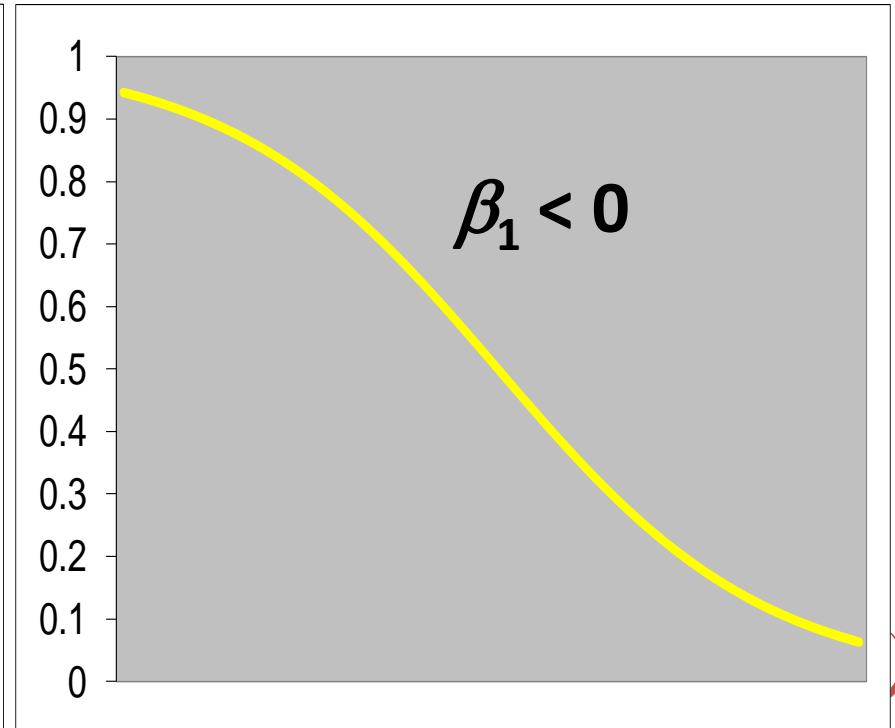
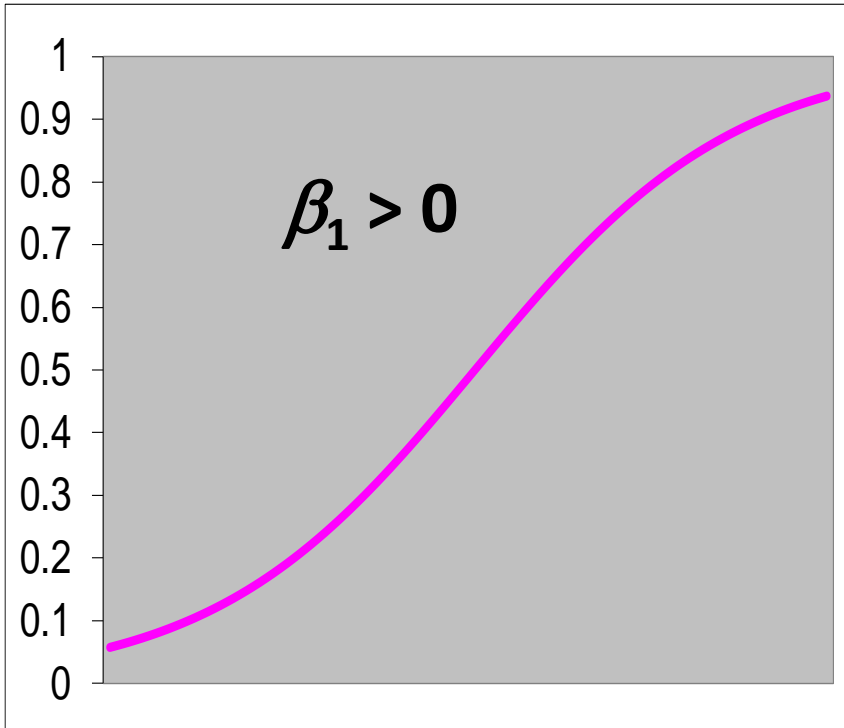
β_1 slope of the regression line. **bagusco**

Logistic Regression Model

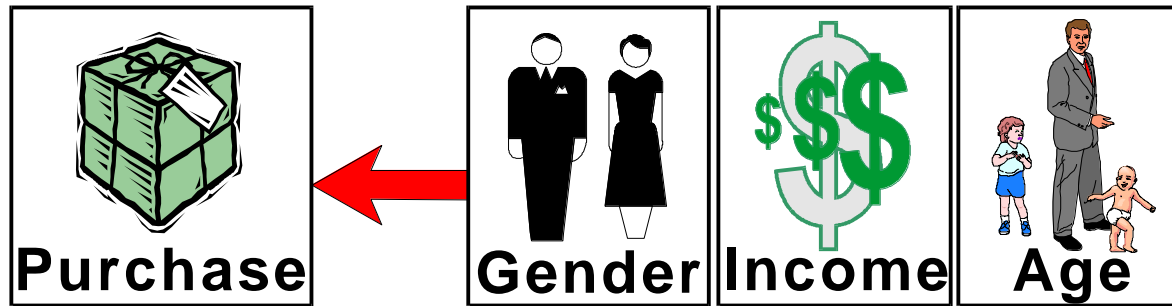
$$P(Y = 1) = \pi = \frac{e^{\beta_0 + \beta_1 x_1}}{1 + e^{\beta_0 + \beta_1 x_1}}$$

bagusco

Logistic Regression Model



Multiple Logistic Regression



$$\text{logit}(p_i) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$$

bagusco

Ilustrasi

- Respon: external rating (OK, NO)
- Prediktor:
 - return on equity
 - return on asset
 - cost to income ratio

bagusco

```
proc logistic data = a.rating outmodel=modelrating;  
model eksternal_rating (event = "OK") =  
    Return_on_equity  
    Return_on_Asset  
    Cost_to_Income_Ratio;  
run;
```

bagusco

Response Profile		
Ordered Value	Eksternal_rating	Total Frequency
1	NO	81
2	OK	29

Probability modeled is Eksternal_rating='OK'.

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-2.9957	1.8408	2.6484	0.1037
Return_on_equity	1	10.4903	4.6121	5.1735	0.0229
Return_on_Asset	1	199.7	71.9641	7.7005	0.0055
Cost_to_Income_Ratio	1	-4.9575	2.9008	2.9208	0.0874

usco

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-2.9957	1.8408	2.6484	0.1037
Return_on_equity	1	10.4903	4.6121	5.1735	0.0229
Return_on_Asset	1	199.7	71.9641	7.7005	0.0055
Cost_to_Income_Ratio	1	-4.9575	2.9008	2.9208	0.0874

$$P(\text{rating} = OK) = \frac{e^{-2.99+10.49ROE+199.7ROA-4.96CIR}}{1 + e^{-2.99+10.49ROE+199.7ROA-4.96CIR}}$$

```
data maudiprediksi;  
input Return_on_equity Return_on_Asset Cost_to_Income_Ratio;  
cards;  
0.1          0.02    0.8  
0.2          0.02    0.6  
;
```

```
proc logistic inmodel=modelrating;  
score data=maudiprediksi out=prediksi;  
run;
```

bagusco

Obs	Return on equity	Return on Asset	Cost to Income Ratio	Eksternal rating	prediksi
1	0.2493	0.0301	0.48030	OK	0.96265
2	0.1764	0.0162	0.61955	OK	0.27260
3	0.1094	0.0055	0.65080	NO	0.01841
4	0.0626	0.0073	0.60580	NO	0.02015
5	0.1800	0.0094	0.56000	NO	0.11853
6	0.0982	0.0084	0.58180	NO	0.04022
7	0.2427	0.0094	0.52160	NO	0.23897
8	0.1493	0.0048	0.78570	NO	0.01254
9	0.1701	0.0056	0.57700	NO	0.04957
10	0.0947	0.0094	0.48440	NO	0.07402
11	0.1748	0.0116	0.52420	OK	0.19090
12	0.2429	0.0063	0.73110	NO	0.05658

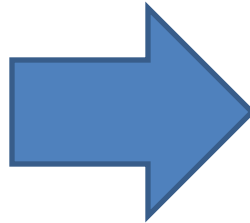
Diskretisasi

Discretization/Binning/Bucketing

Andaikan dataset berisi N observasi, proses diskretisasi terhadap variabel numerik A adalah mengubah nilai variabel tersebut menjadi m interval $D = \{[d_0, d_1], (d_1, d_2], \dots, (d_{m-1}, d_m]\}$, dengan d_0 adalah nilai terkecil, d_m adalah nilai terbesar, dan $d_i < d_{i+1}$, untuk $i = 0, 1, \dots, m-1$.

Diskretisasi

6.58
15.35
14.24
6.22
1.82
2.11
13.77
5.65
15.58
12.46
13.05
11.64
10.91
14.31
7.42



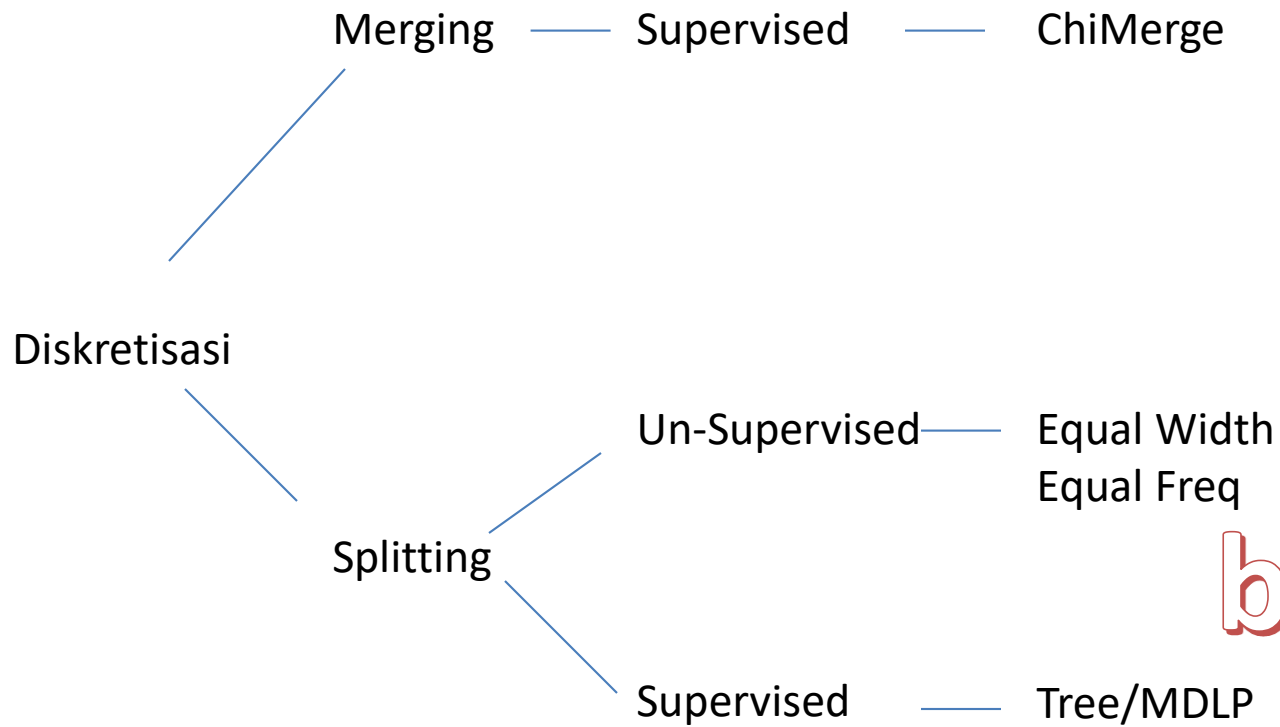
$$X \leq 5$$

$$5 < X \leq 10$$

$$10 < X \leq 15$$

$$X > 15$$

bagusco

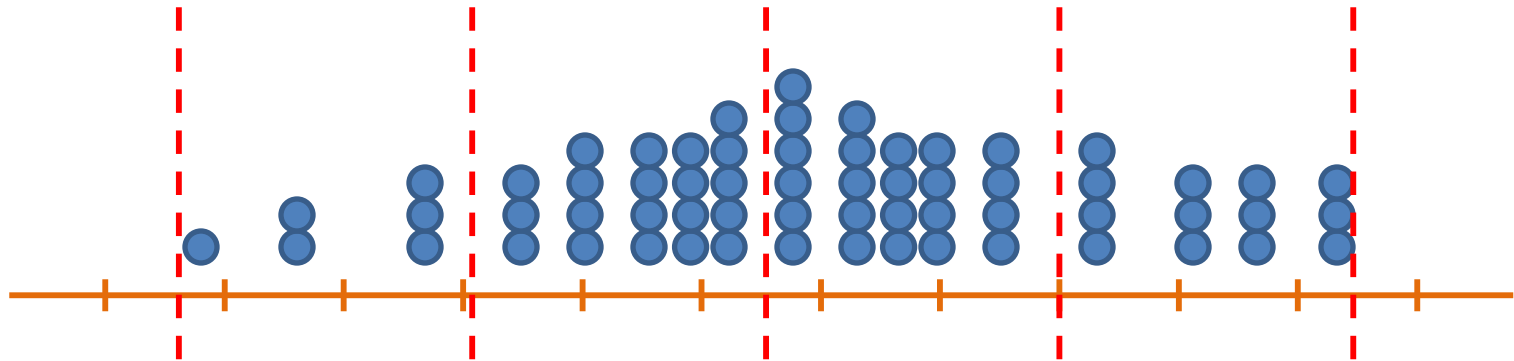


bagusco

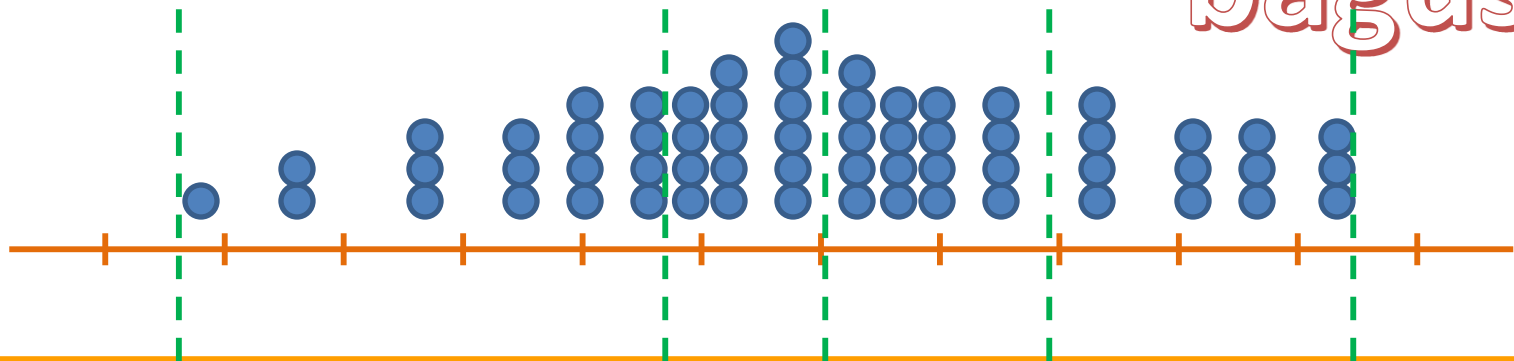
Equal Width dan Equal Frequency

- In equal width, the continuous range of a feature is divided into intervals that have an equal width and each interval represents a bin. The arity can be calculated by the relationship between the chosen width for each interval and the total length of the attribute range.
- In equal frequency, an equal number of continuous values are placed in each bin. Thus, the width of each interval is computed by dividing the length of the attribute range by the desired arity.

Unsupervised Discretization: Equal Width Discretization



Unsupervised Discretization: Equal Freq Discretization



bagusco

Hands-On

- Diskretisasi
- Pake hasil diskretisasi untuk regresi logistik

bagusco

TAHAPAN PEMBUATAN MODEL SKORING

Tahapan

- Eksplorasi data dan Data cleansing
- Pemilihan variabel prediktor
 - WoE (weight of evidence)
 - IV (information value)
- Penyusunan model skoring awal
- Reject Inference (khusus untuk approval scoring)
- Penyusunan model skoring dan scorecard
- Validasi model skoring

Eksplorasi Data

- Mengetahui gambaran umum mengenai karakteristik data yang akan digunakan
 - Jenis variabel: kategorik vs numerik
 - Distribusi nilai variabel
 - Kode dan kesalahan pengkodean
 - Outliers
 - Perlunya menyusun variabel baru (derivative variabel)
 - misal, variabel usia dari variabel tanggal lahir

bagusco

Jenis Variabel

- Numerik
 - Misal: income, age, number of dependants
 - Terhadapnya dapat dilakukan operasi-operasi aritmatika
- Kategorik
 - Misal: gender, occupation, residential ownership
 - Ada yang bersifat ordinal, ada yang bersifat nominal

Jenis Variabel

- Variabel numerik bisa dijadikan variabel kategorik melalui proses discretization/binning/bucketing
- Misal, nilai income dikelompok-kelompokkan menjadi
 - Kurang dari 5 juta per bulan
 - 5 – 10 juta per bulan
 - 10 – 20 juta per bulan
 - Lebih dari 20 juta per bulan

bagusco

Pemilihan Variabel

- Tidak semua variabel yang ada pada data layak untuk jadi prediktor dalam model skoring
- Hanya variabel yang memiliki pengaruh terhadap status good/bad saja yang pantas untuk dijadikan prediktor
- Variabel prediktor memiliki pengaruh jika untuk nilai variabel yang berbeda maka proporsi good/bad-nya berbeda
- Perbedaan tersebut dapat dilihat menggunakan nilai WoE (weight of evidence)

bagusco

Weight of Evidence

$$WoE(X = k) = \log \left(\frac{P(X = k | Good)}{P(X = k | Bad)} \right)$$

<i>Age</i>	<i>Count</i>	<i>P(X=k)</i>	<i>Good P(X=k Good)</i>	<i>Bad P(X=k Bad)</i>	<i>Bad Rate</i>	<i>WoE</i>		
Missing	1,000	2.50%	860	2.38%	140	3.65%	14.00%	-0.428
18–22	4,000	10.00%	3,040	8.41%	960	25.00%	24.00%	-1.089
23–26	6,000	15.00%	4,920	13.61%	1,080	28.13%	18.00%	-0.726
27–29	9,000	22.50%	8,100	22.40%	900	23.44%	10.00%	-0.045
30–35	10,000	25.00%	9,500	26.27%	500	13.02%	5.00%	0.702
35–44	7,000	17.50%	6,800	18.81%	200	5.21%	2.86%	1.284
44+	3,000	7.50%	2,940	8.13%	60	1.56%	2.00%	1.651
Total	40,000	100%	36,160	100%	3,840	100%	9.60%	

Cara membuat kelompok

- Discretization/Binning/Bucketing
- Binning variabel numerik
 - Awali dengan banyak bin/bucket, kemudian gabung bin dengan bad-rate atau WoE yang sama sehingga jumlah bin/bucket menjadi lebih sedikit
 - “Missing” is grouped separately
 - Rule of thumb: “minimum 5% in each bucket”
 - There is no bucket with 0 counts for good or bad.
 - The bad rate and WOE are sufficiently different from one bucket to the next
 - The WOE for nonmissing values also follows a logical distribution, for example: going from negative to positive without any reversals

Information Value

$$IV = \sum_{k=1}^b (P(X = k | Good) - P(X = k | Bad)) * WoE$$

- Mengukur kekuatan pengaruh suatu prediktor terhadap status good/bad
 - Less than 0.02: unresponsive
 - 0.02 to 0.1: weak
 - 0.1 to 0.3: medium
 - 0.3 +: strong
- Digunakan sebagai salah satu ukuran dalam pemilihan variabel untuk model scoring

bagusco

Information Value

<i>Age</i>	<i>Count</i>	<i>P(X=k)</i>	<i>Good P(X=k/Good)</i>	<i>Bad P(X=k/Bad)</i>	<i>Bad Rate</i>	<i>WoE</i>		
Missing	1,000	2.50%	860	2.38%	140	3.65%	14.00%	-0.428
18–22	4,000	10.00%	3,040	8.41%	960	25.00%	24.00%	-1.089
23–26	6,000	15.00%	4,920	13.61%	1,080	28.13%	18.00%	-0.726
27–29	9,000	22.50%	8,100	22.40%	900	23.44%	10.00%	-0.045
30–35	10,000	25.00%	9,500	26.27%	500	13.02%	5.00%	0.702
35–44	7,000	17.50%	6,800	18.81%	200	5.21%	2.86%	1.284
44+	3,000	7.50%	2,940	8.13%	60	1.56%	2.00%	1.651
Total	40,000	100%	36,160	100%	3,840	100%	9.60%	

bagusco

$$IV = (2.38\% - 3.655\%)(-0.428) + (8.41\% - 25.00\%)(-1.089) + \dots$$

$$= 0.668$$

Penyusunan Model Skoring

- Gunakan nilai WoE sebagai pengganti nilai asli dari setiap variabel sesuai dengan bucket-nya masing-masing
- Susun model regresi logistik dengan variabel baru berisi WoE menjadi predictor variable dan status good/bad sebagai response variable
- Jika diperlukan, gunakan teknik-teknik penyeleksian variabel (forward, backward, stepwise) untuk memastikan kelayakan variabel yang disertakan dalam model.

Hands-On

bagusco

Data

#	Variable	Type	Label
1	ID	Num	ID
2	Age	Num	Age
3	Gender	Char	Gender
4	Residence_Ownership	Char	Residence Ownership
5	number_of_dependants	Num	number of dependants
6	status	Char	status



Peran dalam Pemodelan

Age → Input / Independent

Gender → Input / Independent

Residence_Ownership → Input / Independent

number_of_dependants → Input / Independent

status → Target / Dependent

bagusco

Jenis/Tipe Peubah Input

Age → Numerik → perlu binning

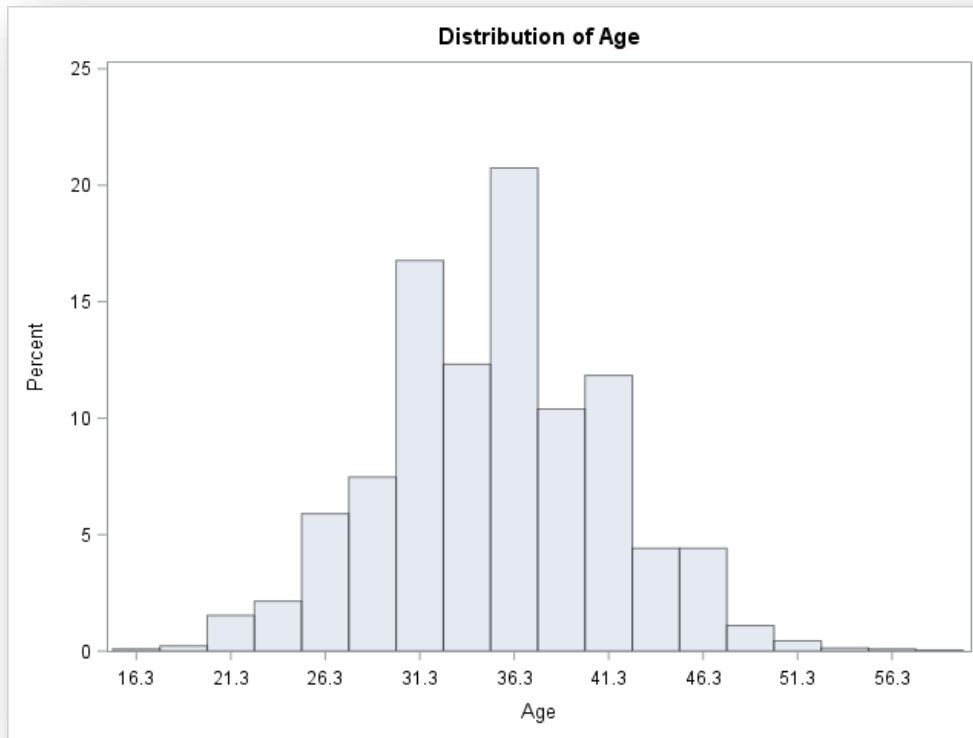
Gender → Kategorik

Residence_Ownership → Kategorik

number_of_dependants → Numerik → perlu binning

bagusco

Melihat Sebaran Nilai Variabel AGE



Quantiles (Definition 5)	
Level	Quantile
100% Max	59
99%	49
95%	45
90%	43
75% Q3	39
50% Median	35
25% Q1	31
10%	28
5%	25
1%	21
0% Min	16

```
proc univariate data=data.datascoring;  
var age;  
histogram age;  
run;
```

Melakukan Binning Variabel AGE

```
data data.datascoreing;  
set data.datascoreing;  
if age <= 25 then agegroup = 1;  
else if age <= 30 then agegroup = 2;  
else if age <= 35 then agegroup = 3;  
else if age <= 40 then agegroup = 4;  
else if age <= 45 then agegroup = 5;  
else agegroup = 6;  
run;  
  
proc tabulate data=data.datascoreing;  
class agegroup;  
table agegroup all, n colpctn;  
run;
```

	N	ColPctN
agegroup		
1	134	5.85
2	394	17.21
3	680	29.69
4	671	29.30
5	311	13.58
6	100	4.37
All	2290	100.00

Melihat Sebaran Nilai number_of_dependants

```
proc tabulate data=data.datascore;  
class number_of_dependants;  
table number_of_dependants all, n colpctn;  
run;
```

	N	ColPctN
number of dependants		
0	533	23.28
1	491	21.44
2	305	13.32
3	331	14.45
4	326	14.24
5	304	13.28
All	2290	100.00

Karena setiap nilai sudah cukup banyak frekuensinya → tidak diperlukan binning/diskretisasi

Setiap nilai menjadi bin

bagusco

Menghitung WOE Gender

**** menghitung WOE dari variabel GENDER ***;

* tahapan: 1. Menghitung $P(\text{Gender} \mid \text{Good})$ dan $P(\text{Gender} \mid \text{Bad})$;

```
proc tabulate data=data.datascoring out=WOEgender;  
class gender status; tables gender, status*colpctn; run;
```

```
proc transpose data=woegender out=woegender;  
var pctn_01; by gender; id status; run;
```

* tahapan: 2. hitung WoE dengan formula $\text{WoE} = \log(P(- \mid \text{GOOD})/P(- \mid \text{BAD}))$;

```
data WOEgender;  
set WOEgender; WOEgender = log(GOOD / BAD); run;
```

* tahapan: 3. Berikan nilai WoE Gender pada data lengkap (datascoring);

```
data woegender (keep = gender woegender);  
set woegender; run;  
proc sort data=data.datascoring;  
by gender; run;
```

```
data data.datascoring;  
merge data.datascoring woegender; by gender; run;
```

Obs	Gender	WOEgender
1	FEMALE	0.88171
2	MALE	-0.46165

Menghitung WOE Residence

**** menghitung WOE dari variabel RESIDENCE ****;

```
proc tabulate data=data.datascoring out=WOEresidence;
```

```
class residence_ownership status; tables residence_ownership, status*colpctn; run;
```

```
proc transpose data=woeresidence out=woeresidence;
```

```
var pctn_01; by residence_ownership; id status; run;
```

```
data WOEresidence;
```

```
set WOEresidence; WOEresidence = log(GOOD / BAD); run;
```

```
data woeresidence keep = residence_ownership woeresidence);
```

```
set woeresidence;
```

```
run;
```

```
proc sort data=data.datascoring;
```

```
by residence_ownership ; run;
```

```
data data.datascoring;
```

```
merge data.datascoring woeresidence;
```

```
by residence_ownership ;
```

```
run;
```

Obs	Residence_Ownership	WOEresidence
1	OTHERS	-0.77202
2	OWNED	1.21297
3	PARENTS	-0.32148
4	RENT	-1.18076



Menghitung WOE Age Group

```
**** menghitung WOE dari variabel agegroup ***;  
proc tabulate data=data.datascoring out=WOEagegroup;  
class agegroup status; tables agegroup, status*colpctn; run;
```

```
proc transpose data=woeagegroup out=woeagegroup;  
var pctn_01; by agegroup; id status; run;
```

```
data WOEagegroup;  
set WOEagegroup; WOEagegroup = log(GOOD / BAD); run;
```

```
data woeagegroup (keep = agegroup woeagegroup);  
set woeagegroup; run;
```

```
proc sort data=data.datascoring;  
by agegroup; run;
```

```
data data.datascoring;  
merge data.datascoring woeagegroup; by agegroup; run;
```

Obs	agegroup	WOEagegroup
1	1	-0.41277
2	2	-0.08443
3	3	-0.13305
4	4	0.12881
5	5	0.01846
6	6	1.27891

Menghitung WOE Number of Dependants

* Menghitung WoE untuk variabel NUMBER OF DEPENDANTS;

```
proc tabulate data=data.datascoring;
```

```
class number_of_dependants; tables number_of_dependants, n colpctn; run;
```

```
proc tabulate data=data.datascoring out=WOEdependants;
```

```
class number_of_dependants status; tables number_of_dependants, status*colpctn; run;
```

```
proc transpose data=woependants out=woependants;
```

```
var pctn_01; by number_of_dependants; id status; run;
```

```
data WOEdependants; set WOEdependants;
```

```
WOEdependants = log(GOOD / BAD); run;
```

```
data woependants
```

```
(keep = number_of_dependants woependants);
```

```
set woependants; run;
```

```
proc sort data=data.datascoring; by number_of_dependants; run;
```

```
data data.datascoring;
```

```
merge data.datascoring woependants; by number_of_dependants; run;
```

Obs	number_of_dependants	WOEdependants
1	0	0.90721
2	1	0.27666
3	2	0.18904
4	3	-0.10972
5	4	-0.82406
6	5	-0.75300

Menghitung Information Value dari Gender

```
**** menghitung INFORMATION VALUE dari GENDER;  
proc tabulate data=data.datascoring out=WOEGender;  
class gender status;  
tables gender, status*colpctn;  
run;  
proc transpose data=woegender out=woegender;  
var pctn_01;  
by gender;  
id status;  
run;  
data WOEGender;  
set WOEGender;  
WOEGender = log(GOOD / BAD);  
IVgender = (GOOD - BAD) * WOEGender /100;  
run;  
proc tabulate data=WOEGender;  
var IVgender;  
tables sum, IVgender;  
run;
```

	IVgender
Sum	0.39

Menghitung Information Value dari Age

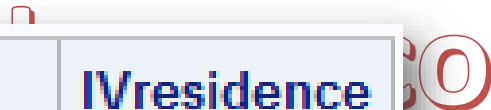
```
proc tabulate data=data.datascoring out=WOEagegroup;  
class agegroup status;  
tables agegroup, status*colpctn;  
run;  
proc transpose data=woeagegroup out=woeagegroup;  
var pctn_01;  
by agegroup;  
id status;  
run;  
data WOEagegroup;  
set WOEagegroup;  
WOEagegroup = log(GOOD / BAD);  
IVagegroup = (GOOD - BAD) * WOEagegroup / 100;  
run;  
proc tabulate data=WOEagegroup;  
var IVagegroup;  
tables sum, IVagegroup;  
run;
```



	IVagegroup
Sum	0.07

Menghitung Information Value dari Residence

```
proc tabulate data=data.datascore out=WOEresidence;  
class residence_ownership status;  
tables residence_ownership, status*colpctn;  
run;  
proc transpose data=woeresidence out=woeresidence;  
var pctn_01;  
by residence_ownership;  
id status;  
run;  
data WOEresidence;  
set WOEresidence;  
WOEresidence = log(GOOD / BAD);  
IVresidence = (GOOD - BAD) * WOEresidence / 100;  
run;  
proc tabulate data=WOEresidence;  
var IVresidence;  
tables sum, IVresidence;  
run;
```



	IVresidence
Sum	1.12

Menghitung Information Value dari Number of Dependants

```
proc tabulate data=data.datascoring out=WOEdependants;  
class number_of_dependants status;  
tables number_of_dependants, status*colpctn;  
run;  
proc transpose data=woe dependants out=woe dependants;  
var pctn_01;  
by number_of_dependants;  
id status;  
run;  
data WOEdependants;  
set WOEdependants;  
WOEdependants = log(GOOD / BAD);  
IVdependants = (GOOD - BAD) * WOEdependants / 100;  
run;  
proc tabulate data=WOEdependants;  
var IVdependants;  
tables sum, IVdependants;  
run;
```

bagusco

	IVdependants
Sum	0.37

Menentukan Bobot Setiap Variabel

***** menentukan bobot masing-masing variabel;

proc logistic data=data.datascoreing outest=bobot;

model status (event = 'GOOD') = WOEgender WOEagegroup WOEresidence
WOEdependants;

run;

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	0.7177	0.0601	142.4263	<.0001
WOEgender	1	1.4986	0.1017	217.0604	<.0001
WOEagegroup	1	1.6055	0.2668	36.2019	<.0001
WOEresidence	1	1.2996	0.0611	452.1798	<.0001
WOEdependants	1	1.6036	0.1067	226.0576	<.0001

gusco

```
data WOEgender (keep = category WOE input);  
set WOEgender; length input $ 20;  
input = 'WOEgender'; category = gender; WOE = WOEgender; run;
```

```
data WOEagegroup (keep = category WOE input);  
set WOEagegroup; length input $ 20;  
input = 'WOEagegroup'; category = compress(agegroup); WOE = WOEagegroup; run;
```

```
data WOEResidence (keep = category WOE input);  
set WOEResidence; length input $ 20;  
input = 'WOEResidence'; category = residence_ownership; WOE = WOEResidence; run;
```

```
data WOEdependants (keep = category WOE input);  
set WOEdependants; length input $ 20;  
input = 'WOEdependants'; category = compress(number_of_dependants); WOE = WOEdependants;  
run;
```

```
data WOEall;  
set WOEgender WOEagegroup WOEResidence WOEdependants;  
run;
```

bagusco

Obs	input	category	WOE
1	WOEGender	FEMALE	0.88171
2	WOEGender	MALE	-0.46165
3	WOEAgegroup	1	-0.41277
4	WOEAgegroup	2	-0.08443
5	WOEAgegroup	3	-0.13305
6	WOEAgegroup	4	0.12881
7	WOEAgegroup	5	0.01846
8	WOEAgegroup	6	1.27891
9	WOEResidence	OTHERS	-0.77202
10	WOEResidence	OWNED	1.21297
11	WOEResidence	PARENT	-0.32148
12	WOEResidence	RENT	-1.18076
13	WOEDependants	0	0.90721
14	WOEDependants	1	0.27666
15	WOEDependants	2	0.18904
16	WOEDependants	3	-0.10972
17	WOEDependants	4	-0.82406
18	WOEDependants	5	-0.75300

Parameter	DF	Estimate
Intercept	1	0.7177
WOEGender	1	1.4986
WOEAgegroup	1	1.6055
WOEResidence	1	1.2996
WOEDependants	1	1.6036

bagusco

The Essential of Credit Scoring Model Yang Harus dikuasai dalam Pembuatan Model Skoring



Hari ke-2

Pembuatan Scorecard

- Model regresi logistik yang diperoleh sebenarnya sudah dapat dipergunakan untuk menghasilkan skor
- Skor yang dihasilkan berupa nilai peluang seorang customer untuk menjadi 'bad'-customer (atau sebaliknya menjadi good-customer). Nilainya antara 0 dan 1.
- Skor tersebut diperoleh dengan memasukkan nilai-nilai variabel prediktor ke dalam model regresi logistik.

bagusco

Penskalaan

- Proses scaling (penskalaan) seringkali diperlukan terhadap hasil regresi logistik
- Penskalaan tidak mempengaruhi power dari model skoring
- Alasan penggunaan penskalaan antara lain:
 - Implementability of the scorecard into application processing software.
 - Ease of understanding by staff (e.g., discrete numbers are easier to work with).
 - Continuity with existing scorecards or other scorecards in the company. This avoids retraining on scorecard usage and interpretation of scores.

bagusco

Penskalaan

$$\text{Score} = \text{Offset} + \text{Factor} \ln(\text{odds})$$

- nilai OFFSET dan FACTOR dapat diperoleh jika telah didefinisikan
 - nilai skor yang diinginkan untuk odds tertentu
 - nilai pdo (points to double the odds), yaitu besarnya kenaikan skor yang menyebabkan odds-nya menjadi dua kali lipat

bagusco

Penskalaan

$$\text{Score} = \text{Offset} + \text{Factor} \ln(\text{odds})$$

$$\text{Score} + \text{pdo} = \text{Offset} + \text{Factor} \ln(2 * \text{odds})$$

- Misal, scorecard yang diinginkan memiliki odds of 50:1 pada nilai 600 dan odds-nya akan dua kali lipat kalau skornya bertambah 20 points ($\text{pdo} = 20$)
- Maka akan diperoleh
$$\text{Factor} = 20 / \ln(2) = 28.8539$$
$$\text{Offset} = 600 - \{28.8539 \ln(50)\} = 487.123$$
- Sehingga
$$\text{Score} = 487.123 + 28.8539 \ln(\text{odds})$$

bagusco

Penskalaan

- Ingat bahwa, dalam model regresi logistik

$$\ln(\text{odds}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p$$

atau

$$\ln(\text{odds}) = \beta_0 + \beta_1 \text{WoE}_1 + \beta_2 \text{WoE}_2 + \dots + \beta_p \text{WoE}_p$$

$$\ln(\text{odds}) = \beta_0 + \sum \beta_i \text{WoE}_i$$

Sehingga

$$\text{Score} = \text{Offset} + \text{Factor } \ln(\text{odds})$$

$$\text{Score} = \text{Offset} + \text{Factor} * (\beta_0 + \sum \beta_i \text{WoE}_i)$$

$$\text{Score} = \sum ((\beta_i \text{WoE}_i + \beta_0/p) * \text{factor} + \text{Offset} / p)$$

bagusco

Scorecard

- Dengan demikian, skor dari suatu individu merupakan penjumlahan dari skor untuk setiap variabel prediktor
- Skor dari setiap variabel prediktor diperoleh menggunakan formula

$$(\beta_i \text{WoE}_i + \beta_0/p) * \text{factor} + \text{Offset}/p$$

bagusco


```
data _null_;  
set bobot;  
if _n_=1 then call symput("b0", intercept);  
if _n_=1 then call symput("bgender", WOEgender);  
if _n_=1 then call symput("bagegroup", WOEagegroup);  
if _n_=1 then call symput("bresidence", WOEResidence);  
if _n_=1 then call symput("bdependants", WOEdependants);  
run;
```

```
data WOEall (drop = factor offset);  
set WOEall;  
Factor = 20 / log (2);  
Offset = 600 - factor * log (50);  
if input = 'WOEgender' then score = (&bgender * WOE + &b0 / 4) * factor + offset / 4;  
if input = 'WOEagegroup' then score = (&bagegroup * WOE + &b0 / 4) * factor + offset / 4;  
if input = 'WOEResidence' then score = (&bresidence * WOE + &b0 / 4) * factor + offset / 4;  
if input = 'WOEdependants' then score = (&bdependants * WOE + &b0 / 4) * factor + offset / 4;  
score = round(score);  
run;
```

```
proc print data=WOEall;  
run;
```

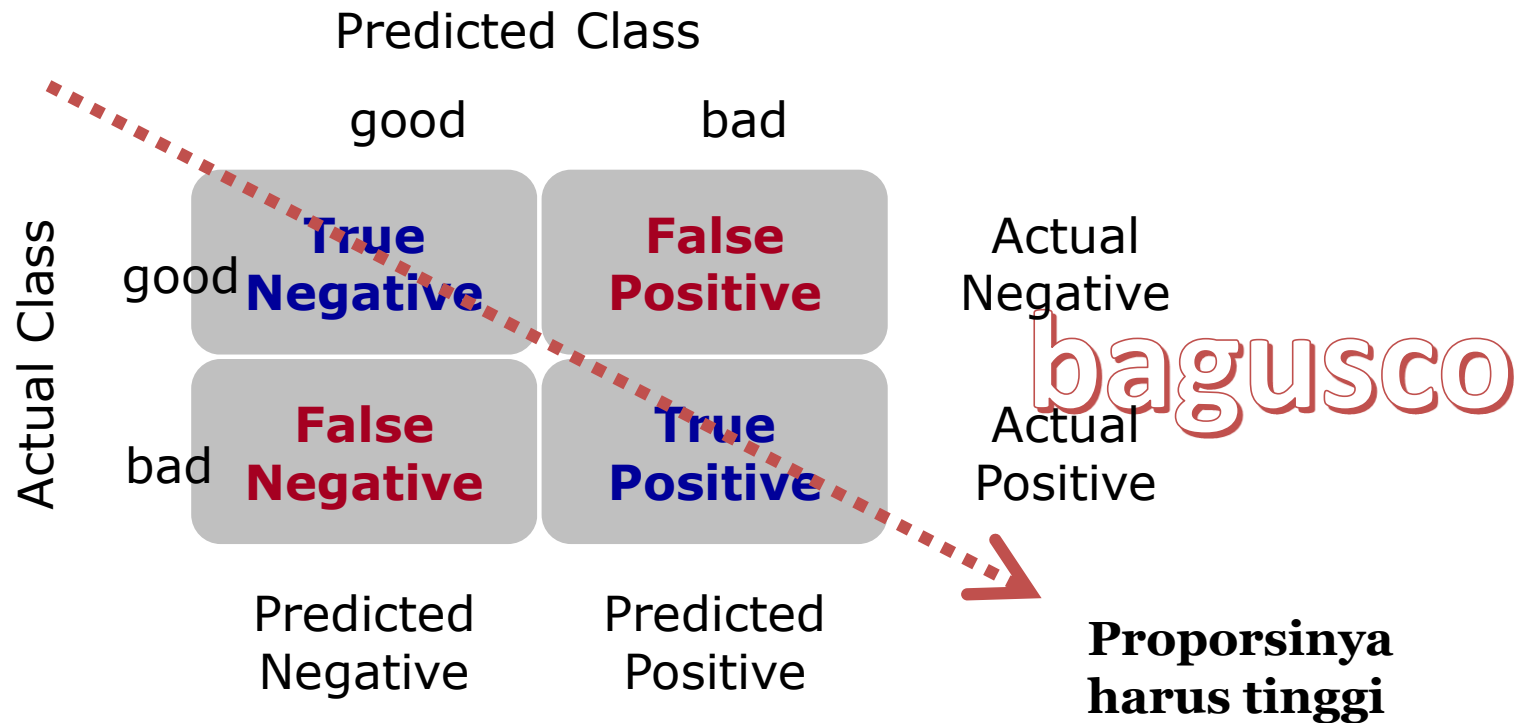
Scorecard yang Dihasilkan

Obs	input	category	WOE	score
1	WOEgender	FEMALE	0.88171	165
2	WOEgender	MALE	-0.46165	107
3	WOEagegroup	1	-0.41277	108
4	WOEagegroup	2	-0.08443	123
5	WOEagegroup	3	-0.13305	121
6	WOEagegroup	4	0.12881	133
7	WOEagegroup	5	0.01846	128
8	WOEagegroup	6	1.27891	186
9	WOEresidence	OTHERS	-0.77202	98
10	WOEresidence	OWNED	1.21297	172
11	WOEresidence	PARENT	-0.32148	115
12	WOEresidence	RENT	-1.18076	83
13	WOEdependant	0	0.90721	169
14	WOEdependant	1	0.27666	140
15	WOEdependant	2	0.18904	136
16	WOEdependant	3	-0.10972	122
17	WOEdependant	4	-0.82406	89
18	WOEdependant	5	-0.75300	92

bagusco

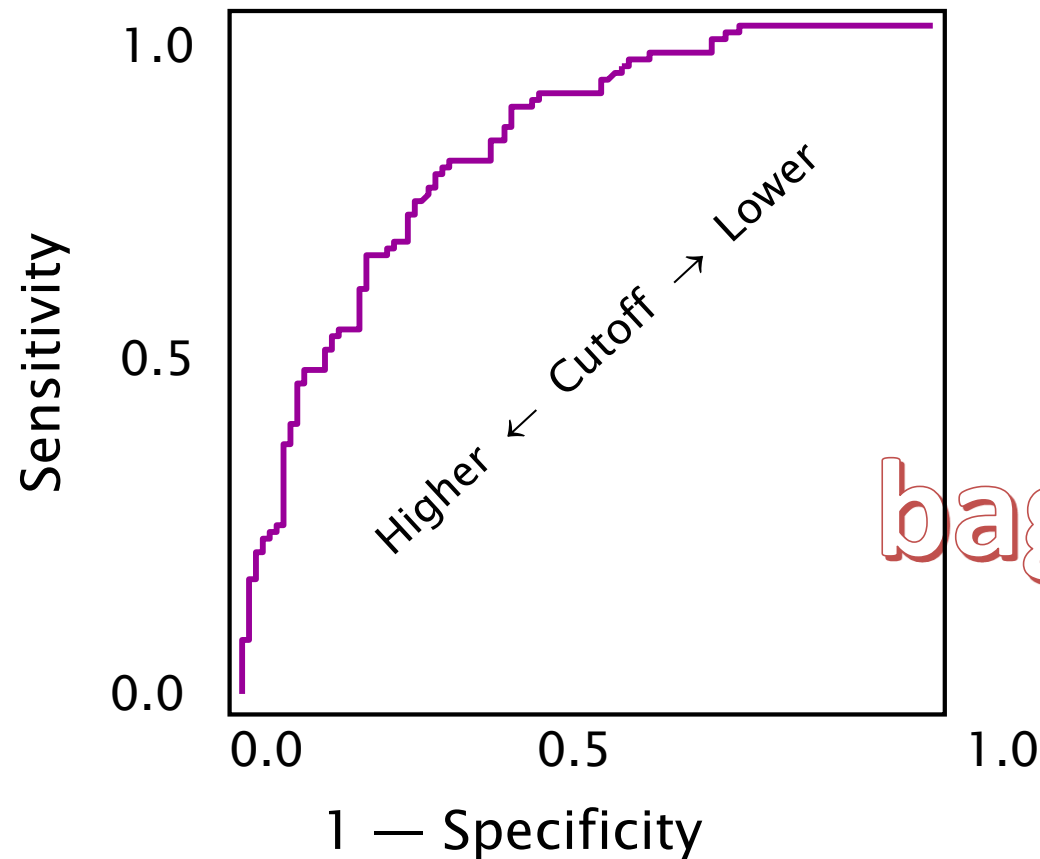
Mengevaluasi Model Skoring

- Misclassification
 - Classification table / Confusion table



Mengevaluasi Model Skoring

- ROC-curve



bagusco

Validating the Model

- Model yang dihasilkan harus dapat memberikan performan yang baik tidak hanya pada data yang digunakan dalam menyusun model (learning set, in-sample), tetapi juga mempunyai kemampuan prediktif yang baik pada gugus data lain (testing set, out-sample)
 - Scoring an alternate data set
 - Population Stability Index

bagusco

Scoring an Alternate Data Set

- Models often deteriorate when scored on a data set that was not used in model development
- A similar campaign from a different time period or geographic area is good for validation
- Methods for scoring
 - Within the model development process
 - modeling new data

bagusco

Menilai seberapa bagus model yang diperoleh

Tahapan:

- Menghitung score dari setiap individu
- Menentukan prediksi kelas status
- Membandingkan prediksi dengan status yang sebenarnya

```
data data.datascoring;
```

```
set data.datascoring;
```

```
Factor = 20 / log (2);
```

```
Offset = 600 - factor * log (50);
```

```
SCOREgender = round((&bgender * WOEgender + &b0 / 4) * factor + offset / 4);
```

```
SCOREagegroup = round((&bagegroup * WOEagegroup + &b0 / 4) * factor + offset / 4);
```

```
SCOREresidence = round((&bresidence * WOEsidence + &b0 / 4) * factor + offset / 4);
```

```
SCOREdependants = round((&bdependants * WOEdependants + &b0 / 4) * factor + offset / 4);
```

```
SCOREtotal = sum(SCOREgender, SCOREagegroup, SCOREresidence, SCOREdependants);
```

```
run;
```

Menilai seberapa bagus model yang diperoleh

```
data data.datascoreing;  
set data.datascoreing;  
if SCOREtotal > 500 then predict = "GOOD";  
else predict = "BAD ";  
run;
```

```
proc tabulate data=data.datascoreing;  
class status predict;  
table status, predict*(n pctl rowpctl);  
run;
```

	predict					
	BAD			GOOD		
	N	Pctl	rowpctl	N	Pctl	rowpctl
status						
BAD	570	24.89	75.70	183	7.99	24.30
GOOD	282	12.31	18.35	1255	54.80	81.65

bagusco

Accuracy = ?

Sensitivity = ?

Specificity = ?

Menilai seberapa bagus model yang diperoleh

```
proc sort data=data.datascore;
```

```
by status;
```

```
proc kde data=data.datascore;
```

```
univar SCOREtotal / out=density bwm=3;
```

```
by status;
```

```
run;
```

```
symbol1 i=join w=2;
```

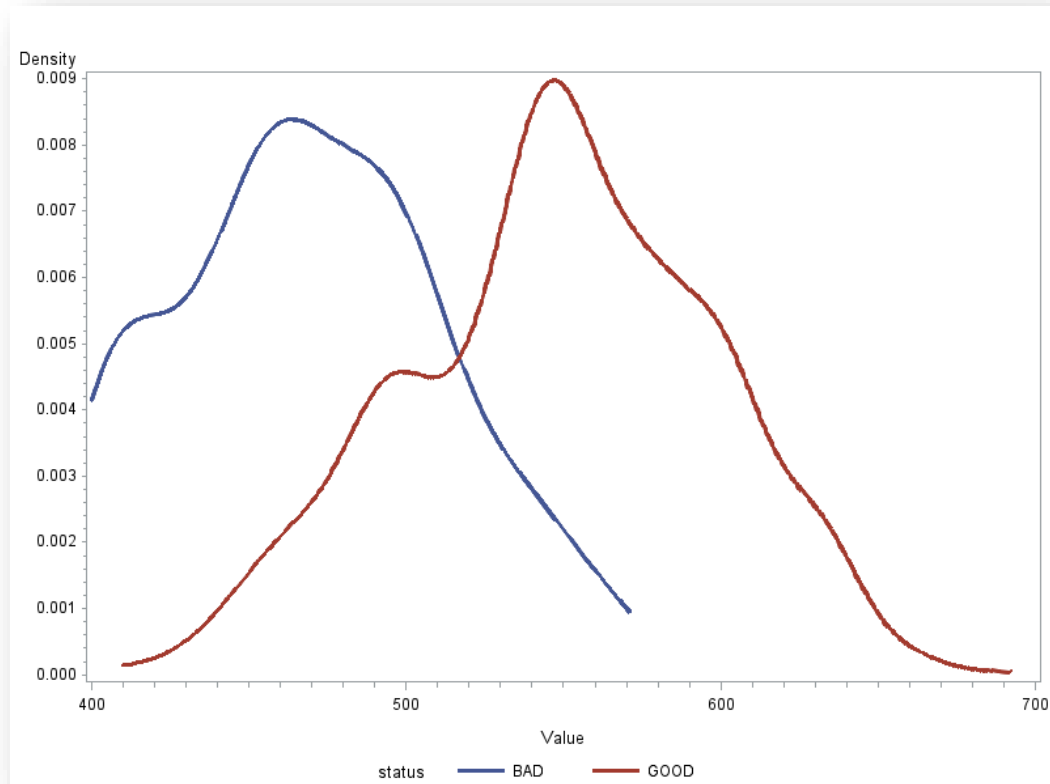
```
symbol2 i=join w=2;
```

```
proc gplot data=density;
```

```
plot density*value=status;
```

```
run;
```

```
quit;
```



CO

Reject Inference

- Pada kasus pembuatan approval scoring, penggunaan data customer penerima kredit dapat menyebabkan bias.
- Hal ini dikarenakan data yang digunakan hanya melibatkan individu yang terpilih (tidak secara acak) oleh proses seleksi approval sebelumnya.
- Dengan demikian, data yang digunakan memiliki sifat keterwakilan (representativeness) yang rendah.

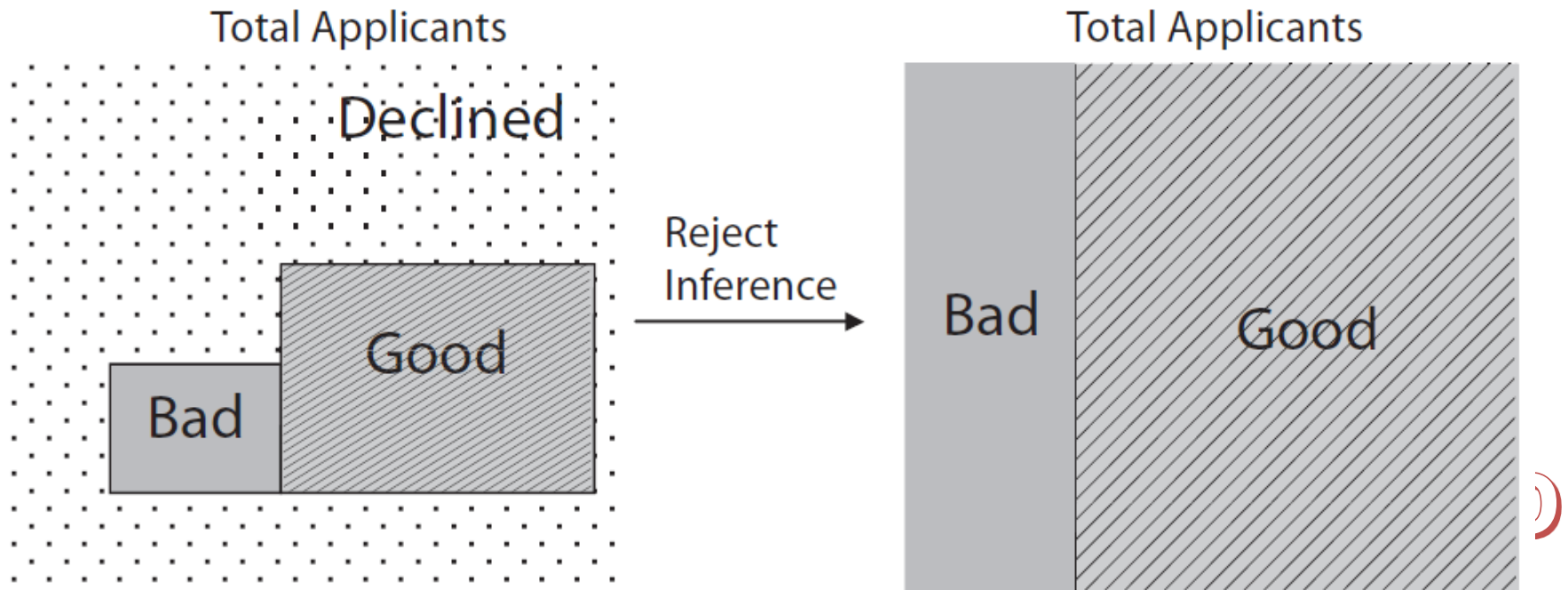
bagusco

Reject Inference

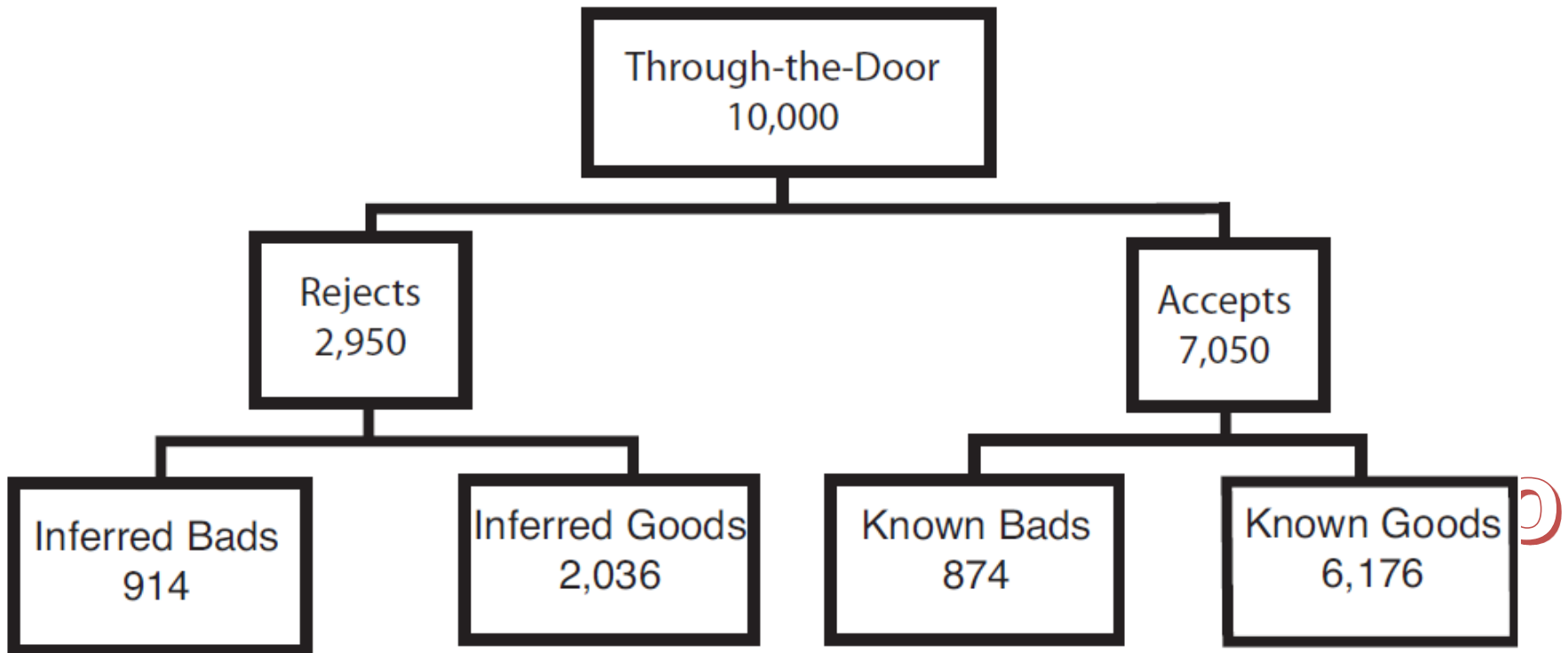
- Karena itu, ada baiknya melibatkan juga data-data individu yang ditolak pada seleksi proses approval sebelumnya.
- Yang menjadi persoalan, individu yang ditolak tersebut tidak diketahui status good/bad-nya karena memang tidak menjadi customer dari produk kredit.
- Perlu upaya untuk memberikan status good/bad pada data individu yang ditolak agar bisa digunakan.

bagusco

Reject Inference



Reject Inference



Teknik melakukan Reject Inference

- **Simple Augmentation (hard cutoff)**
 - Step 1 Build a model using known goods and bads
 - Step 2 Score rejects using this model and establish their expected bad rates, or $p(bad)$.
 - Step 3 Set an expected bad rate level above which an account is deemed “bad”; all applicants below this level are conversely classified as “good.”
 - Step 4 Add the inferred goods and bads to the known goods/bads and remodel.

bagusco

Teknik melakukan Reject Inference

- **Nearest Neighbor (Clustering)**

- Step 1 Create two sets of clusters—one each for known goods and bads.
- Step 2 Run rejects through both clusters.
- Step 3 Compare Euclidean distances to assign most likely performance (i.e., if a reject is closer to a “good” cluster than a “bad” one, then it is likely a good).
- Step 4 Combine accepts and rejects to create inferred dataset, and remodel.

bagusco