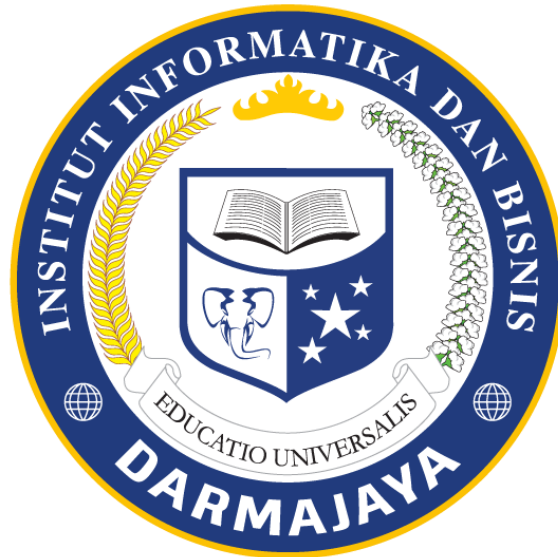


UTS

Case Study for Credit Scorecard Analysis (Step 1 – 6)



Disusun oleh:

Abcdefg..... (2321210002)

**MAGISTER TEKNIK INFORMATIKA
INSTITUT INFORMATIKA DAN BISNIS DARMAJAYA
BANDAR LAMPUNG
2025**

Studi Kasus untuk Analisis Credit Scorecard

Studi kasus Credit Scorecard Analysis dirancang untuk mengajarkan metode mengembangkan model skoring kredit yang kuat dengan analisis data historis peminjam. Tujuannya adalah memprediksi risiko gagal bayar melalui proses transformasi variabel dan pelatihan model yang melibatkan regresi logistik serta kalibrasi skor. Teknik ini mendukung perusahaan dalam membuat keputusan kredit yang lebih cerdas dan mengurangi potensi risiko, dengan memastikan sistem skoring yang akurat dan mudah divalidasi. Berikut ini adalah langkah-langkahnya:

1. Create a creditscorecard object

Tahap pertama dalam membuat model skoring kredit adalah membuat objek `creditscorecard` menggunakan data dari `CreditCardData.mat`, yang merupakan dataset dari Refaat 2011. Objek ini membantu menyusun dan mengelola prediktor utama dalam analisis. Data kredit pelanggan, seperti usia, pendapatan, dan status residensi, disiapkan dalam format tabel. Setelahnya, prediktor yang kurang relevan dapat disaring menggunakan fungsi `screenpredictors` dan `threshold predictors` untuk memastikan fokus pada variabel yang paling signifikan bagi skoring kredit.

Untuk melihat datanya di matlab perlu memasukkan kode berikut:

```
load CreditCardData
head(data)
```

Gambar 1. Kode untuk menampilkan beberapa data awal (head)

Dan akan muncul outputnya sebagai berikut:

CustID	CustAge	TmAtAddress	ResStatus	EmpStatus	CustIncome	TmWBank	OtherCC	AMBalance	UtilRate	status
1	53	62	Tenant	Unknown	50000	55	Yes	1055.9	0.22	0
2	61	22	Home Owner	Employed	52000	25	Yes	1161.6	0.24	0
3	47	30	Tenant	Employed	37000	61	No	877.23	0.29	0
4	50	75	Home Owner	Employed	53000	20	Yes	157.37	0.08	0
5	68	56	Home Owner	Employed	53000	14	Yes	561.84	0.11	0
6	65	13	Home Owner	Employed	48000	59	Yes	968.18	0.15	0
7	34	32	Home Owner	Unknown	32000	26	Yes	717.82	0.02	1
8	50	57	Other	Employed	51000	33	No	3041.2	0.13	0

Gambar 2. Kode untuk menampilkan beberapa data awal (head)

Gambar 2 diatas menampilkan ringkasan data dari dataset `CreditCardData` dengan beberapa kolom berikut:

- **CustID:** ID unik untuk setiap pelanggan.

- **CustAge**: Umur pelanggan.
- **TmAtAddress**: Lama tinggal di alamat saat ini (dalam tahun).
- **ResStatus**: Status tempat tinggal (misalnya, Penyewa, Pemilik Rumah).
- **EmpStatus**: Status pekerjaan (misalnya, Bekerja, Tidak Diketahui).
- **CustIncome**: Pendapatan pelanggan.
- **TmWBank**: Lama berhubungan dengan bank (dalam tahun).
- **OtherCC**: Apakah pelanggan memiliki kartu kredit lain.
- **AMBalance**: Rata-rata saldo bulanan.
- **UtilRate**: Tingkat pemakaian kredit.
- **status**: Status kredit (0 atau 1, kemungkinan menandakan default atau non-default).

Variabel `sc` dalam gambar 3 dibawah ini merujuk pada objek yang dihasilkan dari pemanggilan fungsi `creditscorecard`. Objek ini adalah model *credit scorecard* yang dibuat dengan menggunakan dataset *CreditCardData*. Fungsi `creditscorecard` digunakan untuk membangun model prediksi yang bertujuan untuk menilai kelayakan kredit seseorang berdasarkan atribut-atribut tertentu dari data pelanggan.

```
sc = creditscorecard(data, 'IDVar', 'CustID')
```

Gambar 3. Variabel `sc` untuk menyimpan objek dari fungsi `creditscorecard`

Kemudian hasil outputnya akan tampil seperti gambar 4 dibawah ini:

```
sc =
creditscorecard with properties:

    GoodLabel: 0
    ResponseVar: 'status'
    WeightsVar: ''
    VarNames: {'CustID' 'CustAge' 'TmAtAddress' 'ResStatus' 'EmpStatus' 'CustIncome' 'TmWBank' 'OtherCC' 'AMBalance' 'UtilRate' 'status'}
    NumericPredictors: {'CustAge' 'TmAtAddress' 'CustIncome' 'TmWBank' 'AMBalance' 'UtilRate'}
    CategoricalPredictors: {'ResStatus' 'EmpStatus' 'OtherCC'}
    BinMissingData: 0
    IDVar: 'CustID'
    PredictorVars: {'CustAge' 'TmAtAddress' 'ResStatus' 'EmpStatus' 'CustIncome' 'TmWBank' 'OtherCC' 'AMBalance' 'UtilRate'}
    Data: [1200x11 table]
```

Gambar 4. Output dari variabel `sc`

Objek `creditscorecard` ini adalah model penilaian kredit yang menggunakan berbagai informasi pelanggan untuk memprediksi kelayakan kredit. Berikut penjelasan singkat setiap properti:

- **GoodLabel**: Nilai yang menunjukkan kategori kredit "baik" (di sini, 0).
- **ResponseVar**: Variabel target yang diprediksi, yaitu `status`.
- **WeightsVar**: Tidak terpakai (kosong), biasa digunakan untuk bobot.
- **VarNames**: Daftar semua variabel dalam dataset (misalnya `CustID`, `CustAge`, `CustIncome`).
- **NumericPredictors**: Prediktor numerik seperti `CustAge` dan `CustIncome`.

- **CategoricalPredictors:** Prediktor kategorikal seperti ResStatus dan EmpStatus.
- **BinMissingData:** Mengatur pengelompokan data yang hilang (di sini, 0 berarti tidak ada).
- **IDVar:** Variabel ID unik untuk setiap pelanggan (CustID).
- **PredictorVars:** Semua variabel yang digunakan sebagai input prediktor.
- **Data:** Dataset yang digunakan dengan ukuran 1200x11 (1200 baris, 11 kolom).

Selanjutnya dari variabel sc tadi, kita dapat melakukan analisis binning atau pengelompokan pada variabel kategorikal *ResStatus* dalam dataset. Berikut gambar perintah kodenya:

```
bininfo(sc, 'ResStatus')
```

Gambar 5. Fungsi untuk analisis bin

Hasilnya akan tampil seperti gambar 6 dibawah ini:

ans=4x6 table					
Bin	Good	Bad	Odds	WOE	InfoValue
{'Home Owner' }	365	177	2.0621	0.019329	0.0001682
{'Tenant' }	307	167	1.8383	-0.095564	0.0036638
{'Other' }	131	53	2.4717	0.20049	0.0059418
{'Totals' }	803	397	2.0227	NaN	0.0097738

Gambar 6. Tabel hasil analisis bin

Hasil analisis binning pada variabel *ResStatus* yang terdiri dari tiga kategori: "Home Owner" (Pemilik Rumah), "Tenant" (Penyewa), dan "Other" (Lainnya). Berikut adalah penjelasan kolom-kolom dalam tabel:

1. **Bin:** Menampilkan kategori dalam variabel *ResStatus*.
 - "Home Owner" mengacu pada orang yang memiliki rumah.
 - "Tenant" mengacu pada orang yang menyewa tempat tinggal.
 - "Other" mungkin mengacu pada kategori lain selain pemilik dan penyewa.
 - "Totals" adalah total dari semua kategori.
2. **Good:** Jumlah observasi yang dikategorikan sebagai "Good" (misalnya, pelanggan baik atau memiliki skor risiko rendah) dalam setiap kategori.
 - Contoh: Kategori "Home Owner" memiliki 365 data yang termasuk dalam kategori "Good".

3. **Bad:** Jumlah observasi yang dikategorikan sebagai "Bad" (misalnya, pelanggan buruk atau memiliki skor risiko tinggi) dalam setiap kategori.
 - Contoh: Kategori "Tenant" memiliki 167 data yang termasuk dalam kategori "Bad".
4. **Odds:** Rasio antara jumlah *Good* dan *Bad* untuk setiap kategori. Ini dihitung sebagai $Good / Bad$.
 - Contoh: Untuk kategori "Home Owner", $Odds = 365 / 177 \approx 2.0621$. Ini menunjukkan bahwa ada lebih banyak data "Good" dibandingkan "Bad" dalam kategori ini.
5. **WOE (Weight of Evidence):** Nilai yang mengukur seberapa kuat pengaruh masing-masing kategori dalam memisahkan kelompok *Good* dan *Bad*. Nilai WOE dapat positif atau negatif:
 - Nilai positif menunjukkan bahwa kategori tersebut lebih terkait dengan *Good* dibandingkan *Bad*.
 - Nilai negatif menunjukkan bahwa kategori tersebut lebih terkait dengan *Bad*.
 - Contoh: "Tenant" memiliki WOE -0.095564, yang menunjukkan bahwa kategori ini sedikit lebih terkait dengan *Bad*.
6. **InfoValue (Information Value):** Mengukur kekuatan prediktif variabel *ResStatus*. Semakin besar nilai InfoValue, semakin baik variabel tersebut dalam memprediksi target.
 - Umumnya, nilai InfoValue < 0.02 menunjukkan variabel dengan kekuatan prediksi lemah, $0.02-0.1$ menunjukkan kekuatan prediksi sedang, dan > 0.1 menunjukkan kekuatan prediksi kuat.
 - Dalam tabel ini, InfoValue untuk setiap kategori cukup kecil, dan totalnya 0.0097738. Ini menunjukkan bahwa *ResStatus* mungkin tidak memiliki kekuatan prediksi yang signifikan.

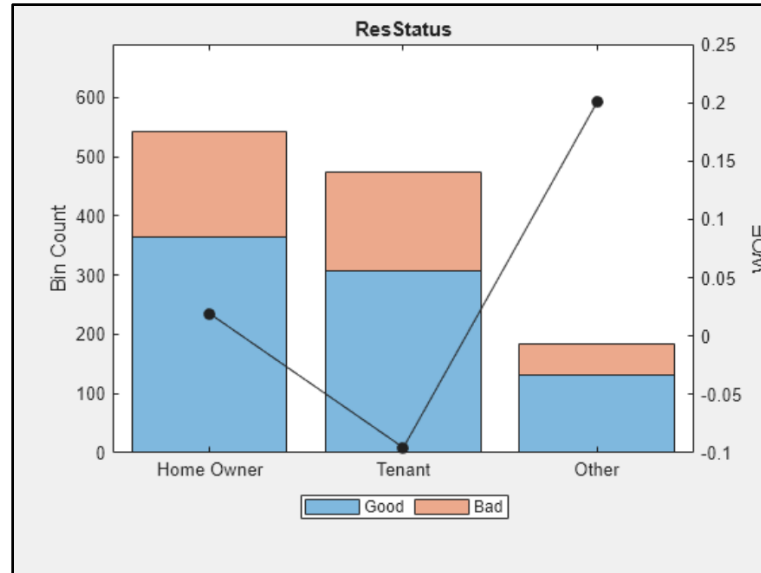
Berdasarkan nilai-nilai ini, kategori "Home Owner" dan "Other" lebih berhubungan dengan "Good", sementara "Tenant" lebih berhubungan dengan "Bad", namun secara keseluruhan, variabel *ResStatus* tidak memiliki kekuatan prediktif yang besar (karena InfoValue rendah).

Kemudian kita lakukan plot untuk melihat grafiknya menggunakan perintah kode di bawah ini:

```
plotbins(sc, 'ResStatus')
```

Gambar 7. Kode plot bin (ResStatus)

Hasil grafiknya akan muncul sebagai berikut:



Gambar 8. grafik plot bin (ResStatus)

Grafik pada gambar 8 diatas menggambarkan distribusi kategori ResStatus (status tempat tinggal) dalam tiga kelompok: Home Owner (Pemilik Rumah), Tenant (Penyewa), dan Other (Lainnya). Setiap kategori menunjukkan jumlah observasi yang termasuk dalam kelompok Good (ditandai dengan warna biru) dan Bad (ditandai dengan warna merah muda).

Pada kategori **Home Owner**, jumlah Good lebih tinggi dibandingkan Bad, dengan nilai WOE (Weight of Evidence) yang sedikit positif, menandakan bahwa kategori ini lebih terkait dengan hasil Good daripada Bad, meskipun pengaruhnya kecil.

Kategori **Tenant** memiliki jumlah Good dan Bad yang hampir seimbang. Nilai WOE-nya sedikit negatif, menunjukkan bahwa kategori Tenant cenderung sedikit lebih terkait dengan hasil Bad dibandingkan dengan Good.

Kategori **Other** memiliki jumlah observasi yang paling sedikit untuk Good dan Bad, tetapi nilai WOE-nya paling tinggi dan positif, menunjukkan bahwa kategori ini lebih terkait dengan hasil Good dibandingkan dua kategori lainnya.

Secara keseluruhan, grafik ini menunjukkan variasi jumlah Good dan Bad di setiap kategori serta bagaimana kategori-kategori tersebut berkaitan dengan hasil Good atau Bad berdasarkan nilai WOE.

Untuk data numerik kita perlu melakukan "pengklasifikasian halus", yaitu mengelompokkan data ke dalam beberapa bin menggunakan grid teratur. Sebagai contoh, gunakan prediktor 'CustIncome'. Dan berikut ini perintah kodenya:

```
cp = 20000:5000:60000;

sc = modifybins(sc, 'CustIncome', 'CutPoints', cp);

bininfo(sc, 'CustIncome')
```

Gambar 9. Kode bin (CustIncome)

Pada gambar 9 diatas, kode tersebut membuat variabel cp yang berisi daftar titik potong dari 20000 hingga 60000 dengan interval 5000, untuk membagi data *CustIncome* menjadi beberapa bin. Perintah *modifybins* kemudian menerapkan titik potong ini pada variabel *CustIncome* dalam dataset *sc*, sehingga data dikelompokkan sesuai rentang yang ditentukan. Terakhir, *bininfo* menampilkan statistik setiap bin, seperti jumlah *Good* dan *Bad*, rasio *Odds*, *WOE*, dan *InfoValue*. Berikut ini hasil tabelnya:

ans=11x6 table					
Bin	Good	Bad	Odds	WOE	InfoValue
{'[-Inf,20000)'} }	3	5	0.6	-1.2152	0.010765
{'[20000,25000)'} }	23	16	1.4375	-0.34151	0.0039819
{'[25000,30000)'} }	38	47	0.80851	-0.91698	0.065166
{'[30000,35000)'} }	131	75	1.7467	-0.14671	0.003782
{'[35000,40000)'} }	193	98	1.9694	-0.026696	0.00017359
{'[40000,45000)'} }	173	76	2.2763	0.11814	0.0028361
{'[45000,50000)'} }	131	47	2.7872	0.32063	0.014348
{'[50000,55000)'} }	82	24	3.4167	0.52425	0.021842
{'[55000,60000)'} }	21	8	2.625	0.26066	0.0015642
{'[60000,Inf]'} }	8	1	8	1.375	0.010235
{'Totals' }	803	397	2.0227	NaN	0.13469

Gambar 10. tabel bin (CustIncome)

Tabel pada gambar 10 diatas memberikan ringkasan statistik untuk setiap kelompok penghasilan. Rentang penghasilan yang lebih tinggi (contohnya 60000 ke atas) cenderung memiliki nilai Odds dan WOE yang lebih tinggi, menunjukkan hubungan antara penghasilan yang lebih tinggi dengan kemungkinan lebih besar dalam kategori Good. Nilai InfoValue

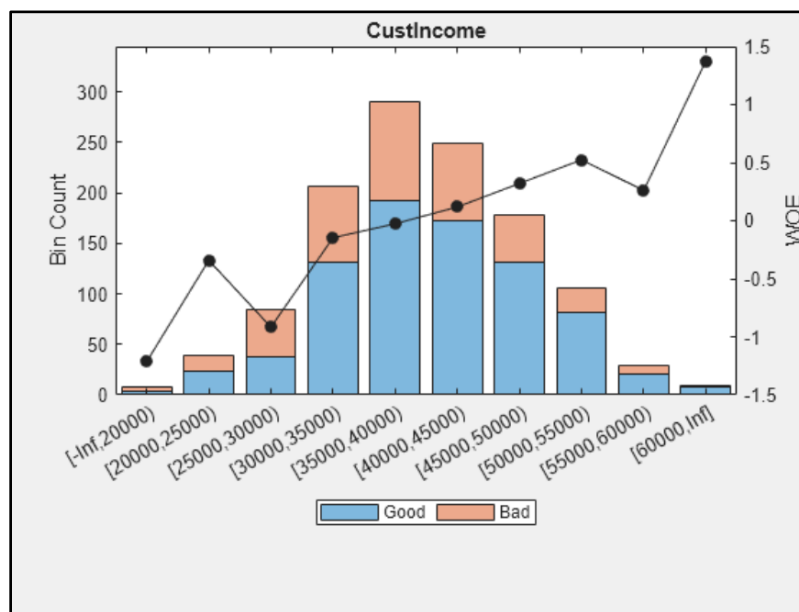
keseluruhan (0.13469) menunjukkan kekuatan prediktif variabel *CustIncome* untuk memisahkan kategori Good dan Bad.

Selanjutnya kita akan plot menggunakan perintah kode sebagai berikut:

```
plotbins(sc, 'CustIncome')
```

Gambar 11. Kode plot bin (CustIncome)

Berikut hasil grafiknya:



Gambar 12. Grafik plot bin (CustIncome)

Secara keseluruhan, grafik diatas menunjukkan bahwa pelanggan dengan pendapatan yang lebih tinggi cenderung memiliki histori pembayaran yang lebih baik (lebih "Good"), sementara pelanggan dengan pendapatan yang lebih rendah atau menengah memiliki distribusi yang lebih merata antara "Good" dan "Bad".

2. Step 2a. Automatically bin the data

Pada langkah 2a dalam analisis credit scorecard, dilakukan pemrosesan otomatis untuk mengelompokkan data setiap variabel prediktor menggunakan fungsi autobinining. Fungsi ini memakai algoritma default "Monotone", yang menjaga agar tren Weight of Evidence (WOE) bersifat monotonik dan idealnya linier, sehingga menghasilkan nilai poin yang linier untuk model. Berikut adalah perintah kodenya:

```
sc = autobinning(sc);
```

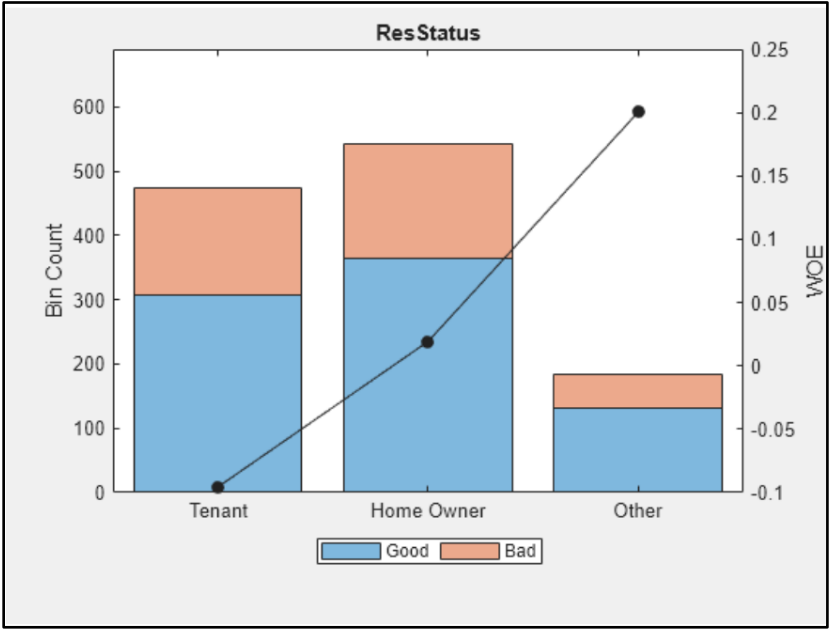
Gambar 13. Kode autobinning

Kemudian kita perlu kode untuk memvisualisasikan tren Weight of Evidence (WOE) dari variabel prediktor bernama ResStatus setelah proses binning otomatis. Dengan fungsi plotbins menampilkan grafik yang menunjukkan distribusi WOE di berbagai kelompok (bin) dari variabel ResStatus, sehingga pengguna dapat melihat apakah tren WOE konsisten dan monotonik. Tren WOE yang monotonik dan linier penting karena membantu meningkatkan interpretasi dan keakuratan model skor kredit dalam membedakan risiko kredit. Berikut kodenya:

```
Predictor = ResStatus ;  
plotbins(sc,Predictor)
```

Gambar 14. Kode plot bin (ResStatus)

Berikut ini adalah hasil grafik plot nya:



Gambar 15. grafik plot bin (ResStatus)

Grafik pada gambar diatas menunjukkan bahwa status tempat tinggal berpengaruh pada risiko kredit. Kategori "Other" memiliki nilai WOE tertinggi, menunjukkan risiko kredit yang lebih rendah dibandingkan "Tenant" dan "Home Owner." Dengan demikian, model kredit

dapat menganggap pelanggan dalam kategori "Other" sebagai kelompok yang lebih aman, sementara "Tenant" mungkin lebih berisiko.

Step 2b. Fine-tune the bins using manual binning

Pada langkah ini, untuk memodifikasi bin secara manual, pertama, gunakan fungsi `bininfo` dengan dua argumen keluaran untuk memperoleh aturan binning awal pada argumen kedua. Setelah itu, lakukan penyesuaian pada aturan binning ini secara manual sesuai kebutuhan. Selanjutnya, terapkan aturan binning yang telah diperbarui menggunakan fungsi `modifybins`, dan tinjau hasilnya dengan menggunakan `plotbins` atau `bininfo` untuk memastikan bahwa perubahan tersebut memberikan distribusi bin yang lebih sesuai dan informatif dalam analisis data.

```
Predictor = CustAge ;  
[bi,cp] = bininfo(sc,Predictor)
```

Gambar 16. Kode bin (CustAge)

Fungsi pada gambar diatas digunakan untuk melakukan binning atau pengelompokan data berdasarkan variabel prediktor (dalam hal ini, 'CustAge' atau usia pelanggan). Proses binning ini membagi variabel kontinu menjadi beberapa interval (bin) dan menghitung beberapa metrik statistik terkait untuk masing-masing bin. Kode tersebut mendapat output sebagai berikut:

```
bi=8x6 table
```

Bin	Good	Bad	Odds	WOE	InfoValue
{'[-Inf,33]'	70	53	1.3208	-0.42622	0.019746
{'[33,37]'	64	47	1.3617	-0.39568	0.015308
{'[37,40]'	73	47	1.5532	-0.26411	0.0072573
{'[40,46]'	174	94	1.8511	-0.088658	0.001781
{'[46,48]'	61	25	2.44	0.18758	0.0024372
{'[48,58]'	263	105	2.5048	0.21378	0.013476
{'[58,Inf]'	98	26	3.7692	0.62245	0.0352
{'Totals'	803	397	2.0227	NaN	0.095205

```
cp = 6x1
```

33
37
40
46
48
58

Gambar 17. Tabel bi dan cp (CustAge)

Output diatas menunjukkan hasil binning pada variabel 'CustAge' (usia pelanggan):

- **Bin**: Rentang usia yang dibagi menjadi kelompok.
- **Good**: Jumlah outcome positif (misal pembelian).
- **Bad**: Jumlah outcome negatif.
- **Odds**: Rasio antara Good dan Bad.
- **WOE** (Weight of Evidence): Ukuran prediktivitas bin berdasarkan odds.
- **InfoValue** Kekuatan informasi dari bin untuk model prediktif.

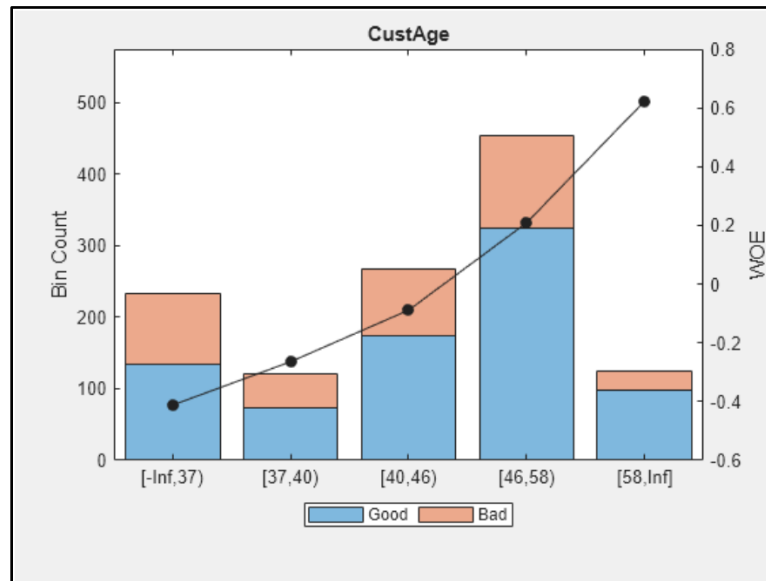
Vektor **cp** berisi titik potong (cut points) yang membagi usia, yaitu 33, 37, 40, 46, 48, 58.

Selanjutnya buat kode untuk merge bin dan modifybins, lalu plot bin nya. Kodenya sebagai berikut:

```
cp([1 5]) = []; % To merge bins 1 and 2, and bins 5 and 6
sc = modifybins(sc, 'CustAge', 'CutPoints', cp);
plotbins(sc, 'CustAge')
```

Gambar 18. Kode merge, modify, dan plot bin (CustAge)

Berikut hasil grafik plot nya:



Gambar 19. Grafik plot bin (CustAge)

Grafik diatas mengindikasikan bahwa usia pelanggan memiliki hubungan yang kuat dengan hasil "Good" atau "Bad". Usia yang lebih tua cenderung menghasilkan lebih banyak outcome

positif ("Good"), sedangkan usia yang lebih muda cenderung menghasilkan lebih banyak outcome negatif ("Bad").

Berdasarkan plot di atas untuk 'CustIncome', sebaiknya menggabungkan bin 3, 4, dan 5 karena nilai WOE-nya hampir sama. Untuk penggabungan bin-binned ini:

```
Predictor = CustIncome ;
[bi,cp] = bininfo(sc,Predictor)
```

Gambar 20. Kode bin (CustIncome)

Kode diatas digunakan untuk melakukan binning atau pengelompokan data pada variabel prediktor 'CustIncome' (pendapatan pelanggan) menggunakan fungsi bininfo. Dan pada fungsi bininfo digunakan untuk melakukan binning pada data sc berdasarkan variabel prediktor 'CustIncome'. Fungsi ini menghasilkan dua output yaitu bi dan cp berdasarkan variabel prediktor 'CustIncome'.

```
bi=8x6 table
```

Bin	Good	Bad	Odds	WOE	InfoValue
{'[-Inf,29000]'	53	58	0.91379	-0.79457	0.06364
{'[29000,33000]'	74	49	1.5102	-0.29217	0.0091366
{'[33000,35000]'	68	36	1.8889	-0.06843	0.00041042
{'[35000,40000]'	193	98	1.9694	-0.026696	0.00017359
{'[40000,42000]'	68	34	2	-0.011271	1.0819e-05
{'[42000,47000]'	164	66	2.4848	0.20579	0.0078175
{'[47000,Inf]'	183	56	3.2679	0.47972	0.041657
{'Totals'	803	397	2.0227	NaN	0.12285

```
cp = 6x1
```

29000
33000
35000
40000
42000
47000

Gambar 21. tabel bi dan cp (CustIncome)

Gambar diatas menunjukkan hasil binning pada variabel 'CustIncome' (pendapatan pelanggan), di mana data dibagi menjadi beberapa bin berdasarkan rentang pendapatan, dengan perhitungan beberapa metrik statistik untuk setiap bin. Tabel bi berisi informasi tentang setiap bin, seperti jumlah "Good" dan "Bad" outcomes, rasio odds antara "Good" dan "Bad", WOE (Weight of Evidence) yang mengukur kekuatan prediktivitas masing-masing bin, dan InfoValue yang menunjukkan seberapa banyak setiap bin berkontribusi terhadap model prediktif. Misalnya, bin dengan rentang [-Inf,29000) memiliki odds 0.91379, WOE -0.79457,

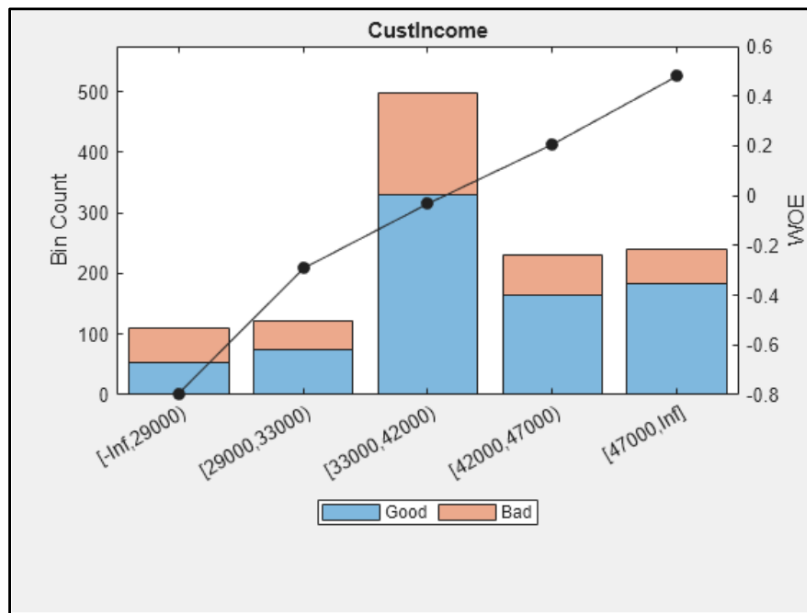
dan InfoValue 0.06364. Vektor cp menunjukkan titik potong yang memisahkan bin-binnya, yaitu 29000, 33000, 35000, 40000, 42000, dan 47000.

Lalu kita buat kode untuk membuat output grafiknya sebagai berikut:

```
cp([3 4]) = []; % To merge bins 3, 4, and 5
sc = modifybins(sc, 'CustIncome', 'CutPoints', cp);
plotbins(sc, 'CustIncome')
```

Gambar 22. Kode merge, modify, dan plot bin (CustIncome)

Berikut hasil grafik plotnya :



Gambar 23. Grafik plot bin (CustIncome)

Grafik pada gambar diatas menunjukkan distribusi jumlah pelanggan (Good dan Bad) berdasarkan kelompok pendapatan (Custincome) dan bobot bukti (Weight of Evidence/WOE). Dalam grafik tersebut, terlihat bahwa kelompok pendapatan di kisaran 33,000-42,000 memiliki jumlah pelanggan tertinggi, baik kategori "Good" maupun "Bad", dan WOE untuk kelompok ini cukup positif. Kelompok pendapatan lainnya memiliki jumlah pelanggan yang lebih rendah, dengan WOE yang bervariasi. Kelompok dengan pendapatan lebih rendah dan lebih tinggi cenderung memiliki WOE yang lebih negatif, menunjukkan hubungan antara pendapatan dan kemungkinan risiko kredit pada kategori pelanggan tertentu.

Untuk 'TmWBank', berdasarkan plot di atas, disarankan untuk menggabungkan bin 2 dan 3 karena keduanya memiliki nilai WOE yang serupa. Untuk menggabungkan bin tersebut menggunakan kode berikut ini:

```
Predictor = TmWBank ;
[bi,cp] = bininfo(sc,Predictor)
```

Gambar 24. Kode bin (TmWBank)

Kode pada gambar diatas menggunakan fungsi bininfo untuk mendapatkan informasi tentang bin dan titik pemotongan (cut points) dari prediktor 'TmWBank'. Outputnya adalah bi, yang berisi detail bin, dan cp, yang berisi titik-titik pemotongan antar bin, sehingga dapat dianalisis lebih lanjut untuk model. Hasil outputnya adalah sebagai berikut:

```
bi=6x6 table
```

Bin	Good	Bad	Odds	WOE	InfoValue
{'[-Inf,12)'} }	141	90	1.5667	-0.25547	0.013057
{'[12,23)'} }	165	93	1.7742	-0.13107	0.0037719
{'[23,45)'} }	224	125	1.792	-0.12109	0.0043479
{'[45,71)'} }	177	67	2.6418	0.26704	0.013795
{'[71,Inf]'} }	96	22	4.3636	0.76889	0.049313
{'Totals' }	803	397	2.0227	NaN	0.084284

```
cp = 4x1
```

```
12
23
45
71
```

Gambar 25. Tabel bi dan cp (TmWBank)

Gambar tabel diatas menunjukkan informasi bin untuk prediktor berdasarkan rentang nilai. Setiap bin memiliki jumlah pelanggan "Good" dan "Bad", rasio "Odds" antara Good dan Bad, serta nilai Weight of Evidence (WOE) yang mencerminkan risiko pada tiap rentang. Bin dengan rentang nilai tertinggi, yaitu [71,Inf], memiliki rasio Odds tertinggi (4.36) dan WOE paling positif (0.77), menandakan risiko yang lebih rendah. Nilai Information Value (IV) untuk tiap bin diakumulasikan menjadi 0.084, menunjukkan kekuatan prediktor ini dalam model. Titik-titik pemotongan (cut points) berada di 12, 23, 45, dan 71.

Selanjutnya kita akan plot dengan menggunakan kode berikut:

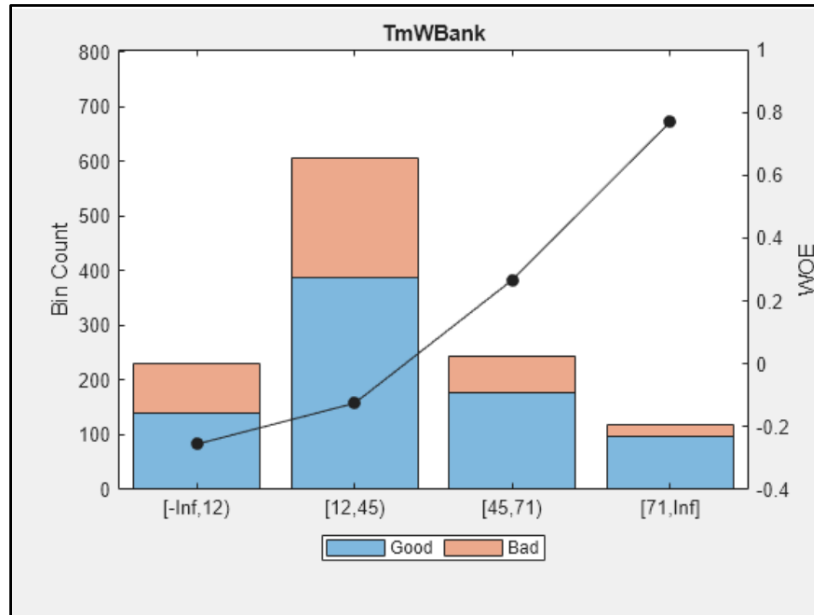
```

cp(2) = []; % To merge bins 2 and 3
sc = modifybins(sc,'TmWBank','CutPoints',cp);
plotbins(sc,'TmWBank')

```

Gambar 26. Kode merge, modify, dan plot bin (TmWBank)

Dan hasil grafiknya dapat dilihat pada gambar di bawah ini:



Gambar 27. Grafik plot bin (TmWBank)

Grafik diatas menunjukkan distribusi jumlah pelanggan (Good dan Bad) berdasarkan rentang nilai prediktor 'TmWBank' dan bobot bukti (Weight of Evidence/WOE). Kelompok dengan rentang nilai [12,45) memiliki jumlah pelanggan tertinggi, sementara rentang tertinggi [71,Inf] memiliki jumlah pelanggan paling sedikit namun nilai WOE paling positif. Nilai WOE meningkat seiring bertambahnya rentang, yang menunjukkan bahwa semakin tinggi nilai 'TmWBank', semakin rendah risiko atau kemungkinan pelanggan termasuk kategori "Bad".

Untuk 'AMBalance', berdasarkan plot di atas, sebaiknya menggabungkan bin 2 dan 3 karena keduanya memiliki nilai WOE yang mirip. Untuk menggabungkan bin tersebut:

```

Predictor = AMBalance ;
[bi,cp] = bininfo(sc,Predictor)

```

Gambar 28. Kode bin (AMBalance)

Kode diatas menggunakan fungsi bininfo untuk mendapatkan informasi bin dan titik pemotongan (cut points) dari prediktor 'AMBalance' dalam dataset sc. Outputnya adalah bi, yang berisi detail setiap bin, dan cp, yang berisi titik pemisah antar bin. Informasi ini membantu memahami distribusi dan pengaruh nilai 'AMBalance' dalam model. Kode tersebut menghasilkan output sebagai berikut ini:

```

bi=5x6 table

```

Bin	Good	Bad	Odds	WOE	InfoValue
{'[-Inf,558.88)'} }	346	134	2.5821	0.24418	0.022795
{'[558.88,1254.28)'} }	309	171	1.807	-0.11274	0.0051774
{'[1254.28,1597.44)'} }	76	44	1.7273	-0.15787	0.0025554
{'[1597.44,Inf]'} }	72	48	1.5	-0.29895	0.0093402
{'Totals' }	803	397	2.0227	NaN	0.039868

```

cp = 3x1
103 x
0.5589
1.2543
1.5974

```

Gambar 29. Tabel bi dan cp (AMBalance)

Gambar diatas menunjukkan distribusi data berdasarkan kategori yang dibagi dalam beberapa interval atau bin. Setiap bin mencakup jumlah "Good" dan "Bad" yang dihitung untuk menentukan Odds (rasio antara jumlah "Good" dan "Bad"). Odds ini, bersama dengan WOE (Weight of Evidence) dan InfoValue, digunakan untuk menganalisis hubungan antara variabel dan hasil yang diinginkan dalam model prediksi. cp (cut points) adalah nilai-nilai pemisah yang membagi data menjadi interval tertentu, yaitu 0.5589, 1.2543, dan 1.5974. Nilai-nilai cp ini menentukan batasan rentang untuk setiap bin, yang kemudian digunakan untuk menghitung Odds, WOE, dan InfoValue di setiap kategori.

Kemudian kita plot dalam bentuk grafik menggunakan kode berikut:

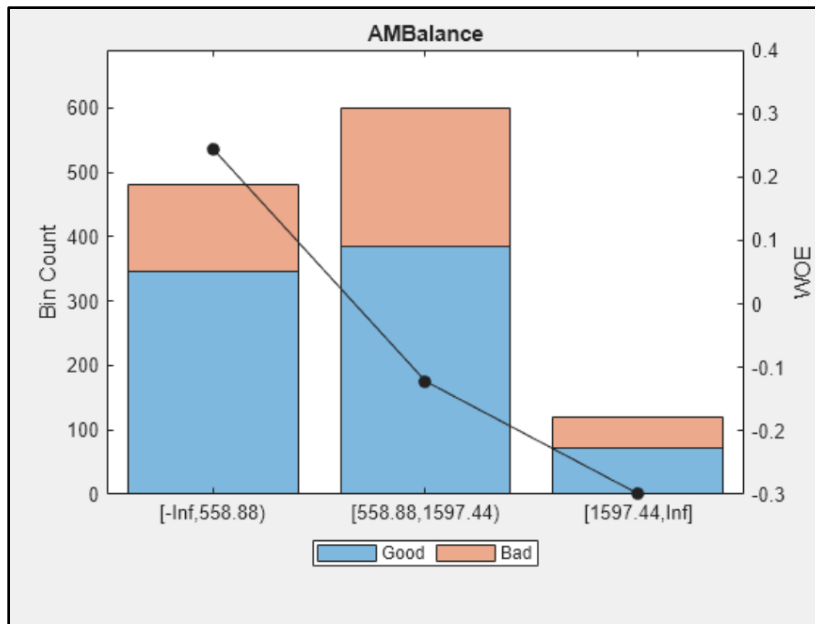
```

cp(2) = []; % To merge bins 2 and 3
sc = modifybins(sc,'AMBalance','CutPoints',cp);
plotbins(sc,'AMBalance')

```

Gambar 30. Kode merge, modify, dan plot bin (AMBalance)

Dan berikut hasil grafiknya:



Gambar 31. Grafik plot bin (AMBalance)

Gambar diatas menunjukkan distribusi binned dari variabel "AMBalance" berdasarkan kelompok "Good" dan "Bad" serta bobot bukti (Weight of Evidence/WOE). Terdapat tiga rentang nilai untuk AMBalance: di bawah 558,88, antara 558,88 dan 1597,44, serta di atas 1597,44. Kelompok "Good" mendominasi rentang nilai rendah dan menengah, sementara jumlah "Bad" lebih tinggi dalam rentang pertama. Nilai WOE turun seiring peningkatan AMBalance, menunjukkan korelasi negatif antara AMBalance dan kemungkinan kategori "Bad". Rentang tertinggi (di atas 1597,44) memiliki nilai WOE negatif terbesar dan jumlah "Bad" yang paling rendah, mengindikasikan risiko rendah pada rentang tersebut.

Sekarang setelah penyesuaian binning selesai, bin untuk semua prediktor menunjukkan tren WOE yang mendekati linear.

3. Fit a logistic regression model

Fungsi fitmodel bertujuan untuk melakukan pemodelan regresi logistik pada data berbasis Weight of Evidence (WOE). Secara internal, fitmodel bekerja dengan membagi data pelatihan ke dalam beberapa bin, mengonversinya menjadi nilai-nilai WOE, serta memetakan variabel respons agar nilai 'Good' dikonversi menjadi 1, sehingga memungkinkan pemodelan regresi logistik linear yang sesuai. Proses pemodelan ini memanfaatkan prosedur stepwise secara default, yang secara bertahap menyeleksi dan menentukan prediktor mana saja yang paling relevan dan sebaiknya disertakan dalam model akhir. Hal ini dilakukan untuk meningkatkan akurasi model dan memastikan bahwa hanya variabel yang signifikan yang dimasukkan dalam analisis.

```
sc = fitmodel(sc);
```

Gambar 32. Fit model sc

Setelah kode diatas di running maka akan menghasilkan output sebagai berikut:

```
1. Adding CustIncome, Deviance = 1490.8954, Chi2Stat = 32.545914, PValue = 1.1640961e-08
2. Adding TmWBank, Deviance = 1467.3249, Chi2Stat = 23.570535, PValue = 1.2041739e-06
3. Adding AMBalance, Deviance = 1455.858, Chi2Stat = 11.466846, PValue = 0.00070848829
4. Adding EmpStatus, Deviance = 1447.6148, Chi2Stat = 8.2432677, PValue = 0.0040903428
5. Adding CustAge, Deviance = 1442.06, Chi2Stat = 5.5547849, PValue = 0.018430237
6. Adding ResStatus, Deviance = 1437.9435, Chi2Stat = 4.1164321, PValue = 0.042468555
7. Adding OtherCC, Deviance = 1433.7372, Chi2Stat = 4.2063597, PValue = 0.040272676

Generalized linear regression model:
  logit(status) ~ 1 + CustAge + ResStatus + EmpStatus + CustIncome + TmWBank + OtherCC + AMBalance
Distribution = Binomial

Estimated Coefficients:

```

	Estimate	SE	tStat	pValue
(Intercept)	0.7024	0.064	10.975	5.0407e-28
CustAge	0.61562	0.24783	2.4841	0.012988
ResStatus	1.3776	0.65266	2.1107	0.034799
EmpStatus	0.88592	0.29296	3.024	0.0024946
CustIncome	0.69836	0.21715	3.216	0.0013001
TmWBank	1.106	0.23266	4.7538	1.9958e-06
OtherCC	1.0933	0.52911	2.0662	0.038806
AMBalance	1.0437	0.32292	3.2322	0.0012285

```
1200 observations, 1192 error degrees of freedom
Dispersion: 1
Chi^2-statistic vs. constant model: 89.7, p-value = 1.42e-16
```

Gambar 33. Output fitmodel sc

Berdasarkan output fitmodel diatas, model regresi logistik dibangun dengan menambahkan variabel prediktor satu per satu, dan setiap kali variabel baru ditambahkan, pengaruhnya terhadap model diukur dengan menggunakan nilai *deviance*, statistik Chi-squared (*Chi2Stat*), dan nilai *p-value*. Berikut adalah detail dari setiap langkah:

- ❖ **Menambahkan *CustIncome***: Penambahan variabel *CustIncome* mengurangi deviance menjadi 1490.8954. Statistik Chi-squared sebesar 32.545914 dan p-value yang sangat kecil (1.1640961e-08) menunjukkan bahwa variabel ini signifikan dan secara substansial meningkatkan model.
- ❖ **Menambahkan *TmWBank***: Setelah menambahkan *TmWBank*, deviance berkurang lebih jauh menjadi 1467.3249, dengan statistik Chi-squared sebesar 23.570535 dan p-value 1.2041739e-06. Ini juga menunjukkan signifikansi tinggi, artinya *TmWBank* menambah nilai prediktif dalam model.
- ❖ **Menambahkan *AMBalance***: Variabel ini menurunkan deviance menjadi 1455.858 dengan statistik Chi-squared sebesar 11.466846 dan p-value 0.00070848829. *AMBalance* juga signifikan dan relevan bagi model.

- ❖ **Menambahkan *EmpStatus***: Deviance lebih lanjut berkurang menjadi 1447.6148. Statistik Chi-squared sebesar 8.2432677 dan p-value 0.0040903428 menunjukkan bahwa *EmpStatus* juga cukup signifikan untuk disertakan.
- ❖ **Menambahkan *CustAge***: Dengan menambahkan *CustAge*, deviance menurun menjadi 1442.06. Statistik Chi-squared sebesar 5.5547849 dan p-value 0.018430237 menunjukkan bahwa *CustAge* memiliki kontribusi yang signifikan.
- ❖ **Menambahkan *ResStatus***: Penambahan variabel ini menurunkan deviance menjadi 1437.9435, dengan statistik Chi-squared sebesar 4.1164321 dan p-value 0.042468555. *ResStatus* juga signifikan, meski kontribusinya sedikit lebih kecil dibanding variabel sebelumnya.
- ❖ **Menambahkan *OtherCC***: Terakhir, menambahkan *OtherCC* menghasilkan deviance sebesar 1433.7372, dengan statistik Chi-squared 4.2063597 dan p-value 0.040272676. Variabel ini juga signifikan dan memberikan kontribusi tambahan pada model.

Secara keseluruhan, semua variabel yang ditambahkan menunjukkan penurunan deviance yang signifikan, sehingga dianggap relevan dan berkontribusi positif dalam memprediksi hasil model.

4. Review and format scorecard points

Setelah model logistik disesuaikan, poin-poin dalam model secara default tidak berskala dan dihasilkan langsung dari kombinasi nilai Weight of Evidence (WOE) dengan koefisien model logistik. Artinya, poin yang diberikan untuk setiap prediktor atau variabel dalam kartu skor berasal dari hasil perkalian nilai WOE dengan koefisien variabel terkait dalam model. Fungsi untuk menampilkan titik atau skor berfungsi untuk merangkum poin-poin yang terdapat pada kartu skor ini, sehingga memudahkan untuk melihat kontribusi masing-masing variabel dalam skala keseluruhan. Dengan demikian, fungsi ini memberikan gambaran menyeluruh tentang bagaimana setiap variabel mempengaruhi skor akhir dalam model logistik.

```
p1 = displaypoints(sc);
disp(p1)
```

Gambar 34. Kode display point

Kode diatas menggunakan fungsi `displaypoints` untuk menampilkan poin kartu skor dari model yang disimpan dalam variabel `sc`, kemudian hasilnya disimpan dalam variabel `p1`. Perintah `disp(p1)` digunakan untuk menampilkan isi dari `p1` pada layar atau output. Fungsi `displaypoints` ini meringkas poin yang telah dihitung untuk setiap prediktor dalam model, berdasarkan kombinasi nilai *Weight of Evidence* (WOE) dan koefisien yang telah dihasilkan dari regresi logistik. Ini membantu dalam memahami bagaimana setiap variabel atau kategori tertentu menyumbang poin ke skor keseluruhan, yang biasanya diimplementasikan dalam

kartu skor kredit atau sistem penilaian risiko. Untuk output nya dapat dilihat pada gambar di bawah ini:

Predictors	Bin	Points
{'CustAge' }	{'[-Inf,37)' }	-0.15314
{'CustAge' }	{'[37,40)' }	-0.062247
{'CustAge' }	{'[40,46)' }	0.045763
{'CustAge' }	{'[46,58)' }	0.22888
{'CustAge' }	{'[58,Inf]' }	0.48354
{'CustAge' }	{'<missing>' }	NaN
{'ResStatus' }	{'Tenant' }	-0.031302
{'ResStatus' }	{'Home Owner' }	0.12697
{'ResStatus' }	{'Other' }	0.37652
{'ResStatus' }	{'<missing>' }	NaN
{'EmpStatus' }	{'Unknown' }	-0.076369
{'EmpStatus' }	{'Employed' }	0.31456
{'EmpStatus' }	{'<missing>' }	NaN
{'CustIncome' }	{'[-Inf,29000)' }	-0.45455
{'CustIncome' }	{'[29000,33000)' }	-0.1037
{'CustIncome' }	{'[33000,42000)' }	0.077768
{'CustIncome' }	{'[42000,47000)' }	0.24406
{'CustIncome' }	{'[47000,Inf]' }	0.43536
{'CustIncome' }	{'<missing>' }	NaN
{'TmWBank' }	{'[-Inf,12)' }	-0.18221
{'TmWBank' }	{'[12,45)' }	-0.038279
{'TmWBank' }	{'[45,71)' }	0.39569
{'TmWBank' }	{'[71,Inf]' }	0.95074
{'TmWBank' }	{'<missing>' }	NaN
{'OtherCC' }	{'No' }	-0.193
{'OtherCC' }	{'Yes' }	0.15868
{'OtherCC' }	{'<missing>' }	NaN
{'AMBalance' }	{'[-Inf,558.88)' }	0.3552
{'AMBalance' }	{'[558.88,1597.44)' }	-0.026797
{'AMBalance' }	{'[1597.44,Inf]' }	-0.21168
{'AMBalance' }	{'<missing>' }	NaN

Gambar 35. Tabel display point (p1)

Output diatas menunjukkan poin kartu skor yang dihitung untuk setiap kategori atau rentang nilai (bin) dari prediktor dalam model logistik. Setiap baris berisi informasi tentang nama prediktor, rentang atau kategori dari prediktor tersebut, serta poin yang dikaitkan. Sebagai contoh, untuk variabel CustAge, jika nilai berada dalam rentang "[46,58)", maka poin yang diberikan adalah 0.22888, sementara rentang "[58,Inf]" mendapat poin 0.48354, menunjukkan hubungan positif dengan status yang diprediksi. Sebaliknya, kategori dengan poin negatif, seperti CustIncome pada rentang "[-Inf,29000)" dengan -0.45455, berkontribusi menurunkan skor keseluruhan. Nilai NaN pada kolom poin menunjukkan data yang hilang (missing), yang tidak diberi skor. Informasi ini membantu membangun skor risiko berdasarkan karakteristik individu, di mana poin-poin ini akan dijumlahkan untuk menghasilkan skor akhir.

Ini adalah waktu yang tepat untuk mengubah label bin, jika ingin melakukan penyesuaian untuk alasan merias. Untuk melakukannya, gunakan fungsi *modifybins* untuk mengubah label bin tersebut. Berikut kodenya:

```

sc = modifybins(sc,'CustAge','BinLabels',...
{'Up to 36' '37 to 39' '40 to 45' '46 to 57' '58 and up'});

sc = modifybins(sc,'CustIncome','BinLabels',...
{'Up to 28999' '29000 to 32999' '33000 to 41999' '42000 to 46999' '47000 and up'});

sc = modifybins(sc,'TmWBank','BinLabels',...
{'Up to 11' '12 to 44' '45 to 70' '71 and up'});

sc = modifybins(sc,'AMBalance','BinLabels',...
{'Up to 558.87' '558.88 to 1597.43' '1597.44 and up'});

p1 = displaypoints(sc);
disp(p1)

```

Gambar 36. Kode modifikasi sc bin

Kode di atas menggunakan fungsi modifybins untuk mengubah binning (pengelompokan data) pada beberapa variabel dalam dataset yang disimpan dalam objek sc. Setiap variabel yang disebutkan (CustAge, CustIncome, TmWBank, dan AMBalance) dikelompokkan ke dalam rentang nilai tertentu dengan label bin yang ditentukan. Misalnya, untuk variabel CustAge, rentang usia dibagi menjadi lima kelompok: "Up to 36", "37 to 39", "40 to 45", "46 to 57", dan "58 and up". Demikian juga, variabel lainnya dikelompokkan berdasarkan rentang nilai yang telah ditentukan, sehingga data lebih terstruktur dan mudah dianalisis berdasarkan kategori tertentu. Dan output akan menjadi seperti gambar di bawah ini:

Predictors	Bin	Points
{'CustAge' }	{'Up to 36' }	-0.15314
{'CustAge' }	{'37 to 39' }	-0.062247
{'CustAge' }	{'40 to 45' }	0.045763
{'CustAge' }	{'46 to 57' }	0.22888
{'CustAge' }	{'58 and up' }	0.48354
{'CustAge' }	{'<missing>' }	NaN
{'ResStatus' }	{'Tenant' }	-0.031302
{'ResStatus' }	{'Home Owner' }	0.12697
{'ResStatus' }	{'Other' }	0.37652
{'ResStatus' }	{'<missing>' }	NaN
{'EmpStatus' }	{'Unknown' }	-0.076369
{'EmpStatus' }	{'Employed' }	0.31456
{'EmpStatus' }	{'<missing>' }	NaN
{'CustIncome' }	{'Up to 28999' }	-0.45455
{'CustIncome' }	{'29000 to 32999' }	-0.1037
{'CustIncome' }	{'33000 to 41999' }	0.077768
{'CustIncome' }	{'42000 to 46999' }	0.24406
{'CustIncome' }	{'47000 and up' }	0.43536
{'CustIncome' }	{'<missing>' }	NaN
{'TmWBank' }	{'Up to 11' }	-0.18221
{'TmWBank' }	{'12 to 44' }	-0.038279
{'TmWBank' }	{'45 to 70' }	0.39569
{'TmWBank' }	{'71 and up' }	0.95074
{'TmWBank' }	{'<missing>' }	NaN
{'OtherCC' }	{'No' }	-0.193
{'OtherCC' }	{'Yes' }	0.15868
{'OtherCC' }	{'<missing>' }	NaN
{'AMBalance' }	{'Up to 558.87' }	0.3552
{'AMBalance' }	{'558.88 to 1597.43' }	-0.026797
{'AMBalance' }	{'1597.44 and up' }	-0.21168
{'AMBalance' }	{'<missing>' }	NaN

Gambar 37. Tabel modifikasi sc bin

Tabel di atas menunjukkan pengaruh berbagai variabel terhadap skor prediktif berdasarkan binning atau pengelompokan data, dengan setiap kombinasi variabel dan bin memiliki nilai poin tertentu. Misalnya, untuk variabel `CustAge`, kelompok usia yang lebih tua (misalnya "58 and up") memiliki nilai poin positif yang lebih tinggi (0.48354), menunjukkan dampak yang lebih besar terhadap model, sementara kelompok usia lebih muda (misalnya "Up to 36") memiliki nilai poin negatif (-0.15314), yang menunjukkan dampak yang lebih rendah. Demikian pula, untuk variabel lain seperti `ResStatus` (status tempat tinggal), `EmpStatus` (status pekerjaan), `CustIncome` (pendapatan pelanggan), `TmWBank` (waktu dengan bank), dan `AMBalance` (saldo akun utama), nilai poin bervariasi sesuai dengan bin yang ditetapkan, dengan beberapa bin memiliki nilai poin positif dan yang lainnya negatif, sementara entri dengan nilai yang hilang (`<missing>`) diberikan nilai `NaN`, yang berarti tidak ada kontribusi terhadap model untuk data yang hilang. Poin-poin ini menunjukkan pentingnya masing-masing bin dalam model prediktif yang lebih besar.

Poin biasanya memiliki skala tertentu dan sering dibulatkan. Untuk mengatur ini, Kita bisa menggunakan fungsi `formatpoints`, yang memungkinkan penyesuaian sesuai dengan tingkat peluang target dan menentukan nilai poin per penggandaan peluang (PDO). Kodenya sebagai berikut:

```
TargetPoints = 500;  
TargetOdds = 2;  
PDO = 50; % Points to double the odds  
  
sc = formatpoints(sc, 'PointsOddsAndPDO', [TargetPoints TargetOdds PDO]);  
p2 = displaypoints(sc);  
disp(p2)
```

Gambar 38. Kode `formatpoints`

Kode di atas mengatur format poin untuk objek `sc` dengan menetapkan target poin, odds, dan nilai PDO. `TargetPoints` diatur ke 500, yang berarti skor target yang diinginkan, `TargetOdds` diatur ke 2, yang menunjukkan odds target, dan `PDO` diatur ke 50, yang berarti setiap peningkatan 50 poin akan menggandakan odds. Fungsi `formatpoints` diterapkan untuk mengonfigurasi pengaturan ini, kemudian `displaypoints` digunakan untuk menampilkan hasil konfigurasi poin tersebut. Hasilnya dapat dilihat pada gambar di bawah ini:

Predictors	Bin	Points
{'CustAge' }	{'Up to 36' }	53.239
{'CustAge' }	{'37 to 39' }	59.796
{'CustAge' }	{'40 to 45' }	67.587
{'CustAge' }	{'46 to 57' }	80.796
{'CustAge' }	{'58 and up' }	99.166
{'CustAge' }	{'<missing>' }	NaN
{'ResStatus' }	{'Tenant' }	62.028
{'ResStatus' }	{'Home Owner' }	73.445
{'ResStatus' }	{'Other' }	91.446
{'ResStatus' }	{'<missing>' }	NaN
{'EmpStatus' }	{'Unknown' }	58.777
{'EmpStatus' }	{'Employed' }	86.976
{'EmpStatus' }	{'<missing>' }	NaN
{'CustIncome' }	{'Up to 28999' }	31.497
{'CustIncome' }	{'29000 to 32999' }	56.805
{'CustIncome' }	{'33000 to 41999' }	69.896
{'CustIncome' }	{'42000 to 46999' }	81.891
{'CustIncome' }	{'47000 and up' }	95.69
{'CustIncome' }	{'<missing>' }	NaN
{'TmWBank' }	{'Up to 11' }	51.142
{'TmWBank' }	{'12 to 44' }	61.524
{'TmWBank' }	{'45 to 70' }	92.829
{'TmWBank' }	{'71 and up' }	132.87
{'TmWBank' }	{'<missing>' }	NaN
{'OtherCC' }	{'No' }	50.364
{'OtherCC' }	{'Yes' }	75.732
{'OtherCC' }	{'<missing>' }	NaN
{'AMBalance' }	{'Up to 558.87' }	89.908
{'AMBalance' }	{'558.88 to 1597.43' }	62.353
{'AMBalance' }	{'1597.44 and up' }	49.016
{'AMBalance' }	{'<missing>' }	NaN

Gambar 39. Tabel setelah di formatpoints

Gambar di atas menunjukkan nilai poin yang diterapkan pada berbagai bin atau kategori untuk setiap variabel prediktor dalam dataset. Setiap kombinasi variabel dan bin memiliki nilai poin yang ditetapkan, yang mencerminkan kontribusi atau pengaruh kategori tersebut terhadap model. Misalnya, pada variabel `CustAge`, kelompok usia yang lebih tua seperti "58 and up" memiliki poin tertinggi (99.166), sedangkan kelompok yang lebih muda seperti "Up to 36" memiliki poin lebih rendah (53.239). Demikian juga untuk variabel lain seperti `ResStatus`, `EmpStatus`, `CustIncome`, `TmWBank`, `OtherCC`, dan `AMBalance`, di mana nilai poin bervariasi tergantung pada kategori yang terpilih. Nilai `NaN` muncul untuk kategori dengan data yang hilang, yang berarti tidak ada kontribusi untuk model dari kategori tersebut.

5. Score the data

Fungsi skor yaitu menghitung skor untuk data pelatihan. Sebagai tambahan, Kita dapat memasukkan data opsional, seperti data validasi, untuk dihitung juga. Selain itu, fungsi ini dapat menghasilkan output opsional yang memberikan jumlah poin yang diperoleh setiap prediktor untuk setiap pelanggan, yang memberikan wawasan lebih lanjut tentang kontribusi masing-masing faktor dalam perhitungan skor.

```
[Scores,Points] = score(sc);  
disp(Scores(1:10))
```

Gambar 40. Kode score

Perintah di atas menjalankan fungsi `score(sc)` untuk menghitung skor berdasarkan model yang telah diterapkan pada objek `sc`. Fungsi ini menghasilkan dua output: `Scores`, yang berisi nilai skor untuk setiap entitas dalam data (misalnya, pelanggan), dan `Points`, yang menunjukkan jumlah poin yang dihitung untuk setiap prediktor. Kemudian, perintah `disp(Scores(1:10))` digunakan untuk menampilkan 10 skor pertama dari output `Scores`, output tersebut dapat dilihat pada gambar dibawah ini.

```
528.2044  
554.8861  
505.2406  
564.0717  
554.8861  
586.1904  
441.8755  
515.8125  
524.4553  
508.3169
```

Gambar 41. Output Scores

Dan untuk output `Points`nya dapat dilihat pada gambar dibawah ini:

```
disp(Points(1:10,:))
```

CustAge	ResStatus	EmpStatus	CustIncome	TmWBank	OtherCC	AMBalance
80.796	62.028	58.777	95.69	92.829	75.732	62.353
99.166	73.445	86.976	95.69	61.524	75.732	62.353
80.796	62.028	86.976	69.896	92.829	50.364	62.353
80.796	73.445	86.976	95.69	61.524	75.732	89.908
99.166	73.445	86.976	95.69	61.524	75.732	62.353
99.166	73.445	86.976	95.69	92.829	75.732	62.353
53.239	73.445	58.777	56.805	61.524	75.732	62.353
80.796	91.446	86.976	95.69	61.524	50.364	49.016
80.796	62.028	58.777	95.69	61.524	75.732	89.908
80.796	73.445	58.777	95.69	61.524	75.732	62.353

Gambar 42. Output Points

Output tabel diatas menunjukkan nilai poin untuk beberapa prediktor pada setiap entitas (misalnya, pelanggan) berdasarkan kategori mereka. Setiap kolom mewakili variabel (seperti CustAge, ResStatus, EmpStatus, dll.) dan setiap baris menunjukkan poin yang dihitung untuk prediktor tersebut pada entitas tertentu. Misalnya, pada baris pertama, pelanggan dengan CustAge "46 to 57" mendapatkan 80.796 poin, ResStatus "Tenant" mendapatkan 62.028 poin, dan seterusnya. Nilai-nilai ini mencerminkan kontribusi setiap variabel terhadap skor keseluruhan pelanggan berdasarkan model yang diterapkan.

6. Calculate the probability of default

Untuk menghitung probabilitas gagal bayar (default), kita menggunakan fungsi `probdefault(sc)` yang menghasilkan nilai probabilitas gagal bayar (`pd`) untuk setiap entitas dalam dataset.

```
pd = probdefault(sc);
```

Gambar 43. Kode `probdefault`

Selanjutnya, probabilitas untuk menjadi "Good" (tidak gagal bayar) didefinisikan sebagai $\text{ProbGood} = 1 - \text{pd}$. Setelah itu, kita memplotkan odds prediksi terhadap skor yang telah diformat, untuk memvisualisasikan hubungan antara keduanya. Melalui plot ini, kita dapat menganalisis secara visual apakah target points dan target odds sesuai dengan yang diharapkan, serta memastikan bahwa hubungan antara poin dan penggandaan odds (Points-to-Double-the-Odds atau PDO) terpenuhi, yaitu perubahan pada skor sebesar 50 poin menggandakan odds.

```

ProbGood = 1-pd;
PredictedOdds = ProbGood./pd;

figure
scatter(Scores,PredictedOdds)
title('Predicted Odds vs. Score')
xlabel('Score')
ylabel('Predicted Odds')

hold on

xLimits = xlim;
yLimits = ylim;

% Target points and odds
plot([TargetPoints TargetPoints],[yLimits(1) TargetOdds],'k:')
plot([xLimits(1) TargetPoints],[TargetOdds TargetOdds],'k:')

% Target points plus PDO
plot([TargetPoints+PDO TargetPoints+PDO],[yLimits(1) 2*TargetOdds],'k:')
plot([xLimits(1) TargetPoints+PDO],[2*TargetOdds 2*TargetOdds],'k:')

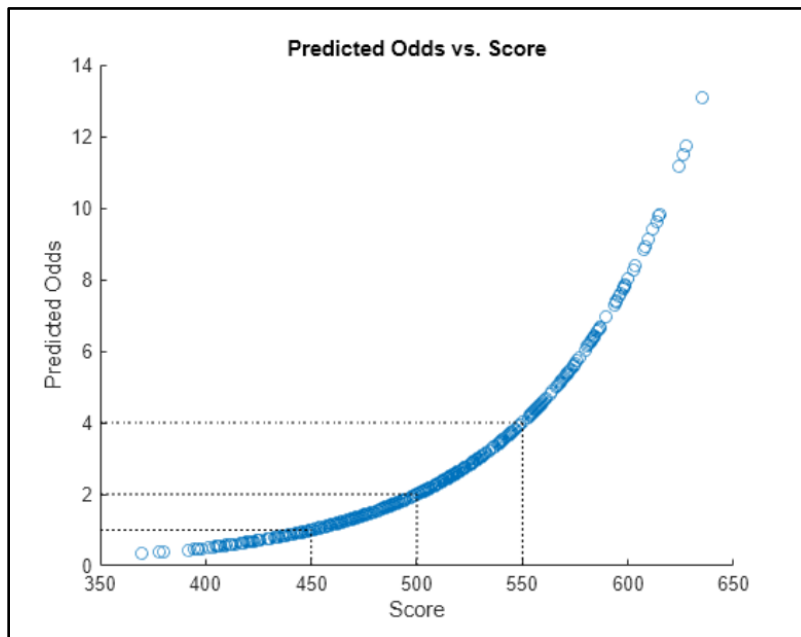
% Target points minus PDO
plot([TargetPoints-PDO TargetPoints-PDO],[yLimits(1) TargetOdds/2],'k:')
plot([xLimits(1) TargetPoints-PDO],[TargetOdds/2 TargetOdds/2],'k:')

hold off

```

Gambar 44. Kode untuk Visualisasi Prediksi Odds dan Skor dengan Target Points dan PDO

Hasil grafiknya dapat dilihat pada gambar dibawah ini:



Gambar 45. Grafik Prediksi Odds dan Skor dengan Target Points dan PDO

Grafik diatas menunjukkan hubungan antara skor (sumbu x) dan odds prediktif (sumbu y). Hubungan tersebut tampak sebagai kurva eksponensial, di mana peningkatan skor menyebabkan peningkatan odds yang cukup signifikan. Pada skor rendah (di bawah 500), perubahan skor hanya meningkatkan odds secara lambat, tetapi setelah melewati nilai sekitar 500, odds meningkat tajam seiring kenaikan skor. Kesimpulannya, semakin tinggi skor, semakin besar odds, dan hubungan ini tidak linear, tetapi cenderung eksponensial pada skor tinggi.