

Case Study for Credit Scorecard Analysis

UTS



Disusun oleh :

Nama:

NPM :

Dosen :

JOKO TRILOKA, S.Kom., M.T., Ph.D

**PROGRAM STUDI MAGISTER TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
INSTITUT INFORMATIKA DAN BISNIS DARMAJAYA
2025**

Studi Kasus Analisis Menilai Resiko Kredit

Studi Kasus untuk Analisis Kartu Skor Kredit menunjukkan sebuah penelitian atau contoh penerapan analisis risiko kredit menggunakan metode scorecard. Kartu skor kredit adalah alat yang digunakan oleh lembaga keuangan untuk menilai kelayakan kredit calon nasabah berdasarkan data historis dan statistik. Dalam studi kasus ini, analisis dilakukan untuk memahami bagaimana metode scorecard dapat digunakan untuk mengukur risiko gagal bayar kredit. Proses ini melibatkan pengumpulan data nasabah, pengelompokan data menjadi kategori tertentu (binning), hingga pembuatan model prediksi menggunakan teknik statistik seperti regresi logistik. Hasil dari analisis ini membantu dalam menentukan skor kredit yang akurat, memprediksi probabilitas gagal bayar, serta memvalidasi model yang digunakan dengan beberapa metrik. Tujuan utamanya adalah untuk menyediakan sistem penilaian yang efektif dan efisien dalam menilai risiko nasabah dan membantu pengambilan keputusan kredit yang lebih baik di dunia perbankan.

1. **Buat objek kartu skor kredit.**

Langkah pertama dalam analisis kartu skor kredit ini adalah membuat objek kartu skor kredit berdasarkan data yang telah disiapkan. Dataset yang digunakan berasal dari kumpulan data kredit kartu, yaitu CreditCardData.mat yang diambil dari referensi Refaat (2011), yang berisi berbagai informasi terkait nasabah dan status pembayaran kredit mereka. Langkah awal dalam proses eksplorasi data (Data Exploration) dataset CreditCardData diawali dengan code berikut:

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

Gambar 1. Kode Head CreditCardData

Ini adalah perintah yang digunakan untuk melihat beberapa baris pertama dari dataset CreditCardData. Tujuannya adalah untuk memberikan cuplikan data sehingga pengguna bisa memahami struktur dataset, melihat jenis variabel yang ada, dan memeriksa apakah data sudah dimuat dengan benar.

Berikut ini adalah hasil dari code tersebut (data CreditCardData.mat):

CustID	CustAge	TmAtAddress	ResStatus	EmpStatus	CustIncome	TmBank	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. Data Pengajuan Kartu Kredit

Setiap baris pada tabel ini mewakili data seorang nasabah, dan setiap kolom berisi informasi atau karakteristik yang terkait dengan nasabah tersebut.

Berikut penjelasan detail dari setiap kolom dalam tabel:

1) **CustID:**

- Nomor identifikasi unik untuk nasabah. Ini adalah ID yang digunakan untuk mengidentifikasi setiap nasabah secara individual.

2) **CustAge:**

- Usia nasabah dalam tahun. Informasi ini dapat digunakan untuk melihat apakah usia memengaruhi kemungkinan gagal bayar kredit.

3) **TmAtAddress:**

- Lama tinggal di alamat saat ini (dalam tahun). Variabel ini menunjukkan stabilitas tempat tinggal nasabah.

4) **ResStatus:**

- Status tempat tinggal nasabah, misalnya:
 - **Tenant:** Nasabah menyewa tempat tinggal.
 - **Home Owner:** Nasabah adalah pemilik rumah.
 - **Other:** Status lainnya yang tidak termasuk dalam kategori utama.
- Status tempat tinggal dapat mempengaruhi tingkat risiko nasabah.

5) **EmpStatus:**

- Status pekerjaan nasabah, seperti:

- **Employed:** Nasabah sedang bekerja.
- **Unknown:** Status pekerjaan tidak diketahui.
- **Other:** Status pekerjaan lainnya.
- Status pekerjaan menunjukkan stabilitas pendapatan, yang penting untuk menilai kemampuan membayar kredit.

6) **CustIncome:**

- Pendapatan tahunan nasabah (dalam satuan mata uang). Pendapatan yang lebih tinggi umumnya berarti risiko gagal bayar lebih rendah.

7) **TmWBank:**

- Lama hubungan nasabah dengan bank (dalam tahun). Hubungan yang lebih lama sering kali menunjukkan loyalitas nasabah.

8) **OtherCC:**

- Apakah nasabah memiliki kartu kredit lain (Yes/No). Memiliki kartu kredit lain bisa menambah risiko, terutama jika nasabah sudah memiliki beban utang yang tinggi.

9) **AMBalance:**

- Saldo rata-rata kartu kredit nasabah. Ini menunjukkan jumlah rata-rata utang atau saldo yang dimiliki nasabah di kartu kredit.

10) **UtilRate:**

- Tingkat pemakaian kredit (credit utilization rate), yaitu rasio penggunaan kredit yang tersedia. Tingkat pemakaian yang tinggi bisa mengindikasikan bahwa nasabah berada pada risiko lebih tinggi.

11) **status:**

- Variabel respons yang menunjukkan status pembayaran kredit:
 - **0:** Nasabah membayar dengan baik (tidak ada masalah pembayaran).
 - **1:** Nasabah gagal membayar atau memiliki masalah pembayaran.

Jadi bisa disimpulkan bahwa dataset ini berisi informasi yang komprehensif tentang nasabah, mencakup karakteristik pribadi, status tempat tinggal, status pekerjaan, serta perilaku penggunaan kartu kredit. Data ini akan digunakan untuk membangun model prediksi dalam analisis kartu skor kredit. Model tersebut akan memanfaatkan variabel-variabel ini untuk menilai risiko gagal bayar dan membantu bank dalam pengambilan keputusan terkait persetujuan kredit.

Ketika perintah codenya diubah tanpa head seperti gambar di bawah ini:

```
load CreditCardData
data
```

Gambar 3. Kode CreditCardData Tanpa Head

Maka akan menampilkan keseluruhan data sampai 1200 data, berikut ini adalah cuplikan tampilan datanya:

	CustID	CustAge	TmAtAddress	ResStatus	EmpStatus	CustIncome	TmWba...	OtherCC
1	1	53	62	Tenant	Unknown	50000	55	Yes
2	2	61	22	Home Owner	Employed	52000	25	Yes
3	3	47	30	Tenant	Employed	37000	61	No
4	4	50	75	Home Owner	Employed	53000	20	Yes
5	5	68	56	Home Owner	Employed	53000	14	Yes
6	6	65	13	Home Owner	Employed	48000	59	Yes
7	7	34	32	Home Owner	Unknown	32000	26	Yes
8	8	50	57	Other	Employed	51000	33	No
9	9	50	10	Tenant	Unknown	52000	25	Yes

	CustID	CustAge	TmAtAddress	ResStatus	EmpStatus	CustIncome	TmWba...	OtherCC
1192	1192	54	36	Home Owner	Unknown	35000	44	Yes
1193	1193	56	21	Home Owner	Employed	44000	53	No
1194	1194	34	54	Home Owner	Unknown	35000	25	Yes
1195	1195	64	9	Home Owner	Employed	49000	18	No
1196	1196	54	92	Home Owner	Unknown	41000	51	Yes
1197	1197	36	19	Tenant	Unknown	45000	4	Yes
1198	1198	29	35	Home Owner	Employed	27000	47	Yes
1199	1199	56	66	Tenant	Employed	38000	48	Yes
1200	1200	44	24	Tenant	Unknown	41000	16	Yes

Gambar 4. Data Pengajuan Kartu Kredit (data 1200)

Selanjutnya adalah pembuatan objek **Credit Scorecard** dalam analisis kartu skor kredit menggunakan fungsi `creditscorecard`, berikut adalah kode nya:

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

Gambar 5. Kode Credit Score

- `sc` Ini adalah variabel yang akan menyimpan objek kartu skor kredit. Objek ini akan digunakan untuk menyimpan informasi tentang model scorecard dan memudahkan proses analisis serta prediksi risiko kredit.

- `creditscorecard` Fungsi `creditscorecard` digunakan untuk membuat objek `scorecard` yang diperlukan dalam analisis kredit. Fungsi ini mengambil dataset dan beberapa argumen opsional untuk mengidentifikasi variabel penting yang digunakan dalam model.
- `data` Ini adalah dataset yang digunakan sebagai input. Pada contoh ini, dataset yang digunakan adalah `data`, yang berisi informasi nasabah (seperti usia, pendapatan, lama tinggal, dan lainnya).
- `IDVar` Argumen `IDVar` digunakan untuk menunjukkan kolom yang berisi variabel identifikasi (ID) unik bagi setiap nasabah. Ini penting untuk melacak setiap observasi secara unik dalam dataset.
- `CustID` `'CustID'` adalah nama kolom yang digunakan sebagai variabel identifikasi dalam dataset. Kolom ini berisi nomor identifikasi unik bagi setiap nasabah, sehingga setiap baris data dapat diidentifikasi dengan benar.

Kode ini membuat objek kartu skor kredit (`sc`) berdasarkan dataset yang telah diberikan, dengan menggunakan kolom `'CustID'` sebagai variabel identifikasi unik. Objek ini kemudian bisa digunakan untuk melakukan analisis lebih lanjut, seperti melakukan binning data, membuat model prediktif, menghitung skor kredit, dan memvalidasi model.

Hasil dari kode tersebut adalah sebagai berikut:

```

sc =
creditscorecard with properties:
    GoodLabel: 0
    ResponseVar: 'status'
    WeightsVar: ''
    VarNames: {'CustID' 'CustAge' 'TmAtAddress' 'ResStatus' 'EmpStatus' 'CustIncome' 'TmWBank' 'OtherCC' 'AMBalance' 'UtilRate'}
    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 6. Properties dari data `creditscorecard`

Detail Properti dalam Objek Credit Scorecard: Setelah objek `creditscorecard` dibuat, properti `sc` berisi informasi berikut:

- **GoodLabel:** 0 berarti label 0 dianggap sebagai kategori baik dalam model ini, mungkin mewakili status pelanggan yang tidak melakukan *default*.
- **ResponseVar:** Variabel target atau respons adalah `'status'`, yang menunjukkan apakah pelanggan telah melakukan *default* atau tidak.
- **WeightsVar:** Tidak ada variabel bobot (`''`), artinya model tidak menggunakan pembobotan pada data.
- **VarNames:** Daftar semua nama variabel dalam data:

- 'CustID', 'CustAge', 'TmAtAddress', 'ResStatus', 'EmpStatus', 'CustIncome', 'TmWBank', 'OtherCC', 'AMBalance', 'UtilRate', 'status'
- **NumericPredictors:** Variabel prediktor numerik yang digunakan untuk membangun model:
 - 'CustAge', 'TmAtAddress', 'CustIncome', 'TmWBank', 'AMBalance', 'UtilRate'
- **CategoricalPredictors:** Variabel prediktor kategorikal:
 - 'ResStatus', 'EmpStatus', 'OtherCC'
- **BinMissingData:** Nilai 0 menunjukkan bahwa data yang hilang tidak dibagi dalam bentuk bin.
- **IDVar:** Variabel ID adalah 'CustID', yang berfungsi untuk mengidentifikasi setiap pelanggan secara unik.
- **PredictorVars:** Daftar variabel prediktor lengkap yang digunakan untuk memprediksi variabel respons:
 - 'CustAge', 'TmAtAddress', 'ResStatus', 'EmpStatus', 'CustIncome', 'TmWBank', 'OtherCC', 'AMBalance', 'UtilRate'
- **Data:** Tabel data berukuran [1200x11], menunjukkan bahwa dataset berisi 1200 baris (pelanggan) dan 11 kolom (variabel).

Langkah selanjutnya adalah melakukan eksplorasi awal pada data, khususnya untuk menganalisis statistik prediktor pada variabel kategorikal seperti 'ResStatus'. Selanjutnya, data akan divisualisasikan dalam bentuk bin untuk variabel 'ResStatus'. Tahapan ini diawali dengan pengumpulan informasi menggunakan kode berikut:

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

Kode tersebut menggunakan fungsi bininfo pada MATLAB untuk menampilkan informasi binning pada variabel ResStatus dalam objek creditcorecard (sc). Berikut penjelasan secara rinci:

- **Fungsi bininfo:**
 - Dalam konteks pemodelan kredit atau analisis data dengan MATLAB, bininfo digunakan untuk mendapatkan rincian binning dari suatu variabel dalam objek creditcorecard.
 - Binning adalah proses mengelompokkan atau mengkategorikan nilai-nilai dari variabel menjadi kelompok-kelompok yang lebih kecil (bin). Ini sering dilakukan untuk mengubah data numerik menjadi kategori atau untuk mengelompokkan nilai-nilai yang berbeda menjadi kelompok tertentu untuk keperluan analisis.

- **Parameter sc:**
 - sc adalah objek *creditscorecard* yang berisi data dan model *scorecard* kredit. Objek ini berisi berbagai variabel prediktor, termasuk *ResStatus*.
- **Parameter 'ResStatus':**
 - 'ResStatus' adalah nama variabel dalam objek *sc* yang ingin ditampilkan informasi binning-nya. *ResStatus* biasanya adalah variabel kategorikal yang mungkin menunjukkan status tempat tinggal (seperti "menyewa", "memiliki rumah", dll).
- **Hasil Eksekusi:**
 - Dengan menjalankan perintah `bininfo(sc, 'ResStatus')`, MATLAB akan menampilkan informasi binning dari variabel *ResStatus*. Informasi ini biasanya mencakup rincian kategori atau interval dalam bin, nilai batas untuk bin numerik (jika ada), jumlah data dalam setiap bin, serta distribusi data untuk tiap bin.
 - Informasi binning ini penting untuk memahami bagaimana variabel *ResStatus* telah dikategorikan dalam model dan untuk analisis lebih lanjut dalam pengelompokan risiko kredit.

Contoh Keluaran: Misalnya, hasilnya mungkin menunjukkan bahwa *ResStatus* dikelompokkan menjadi beberapa kategori seperti:

- Bin 1: "Menyewa"
- Bin 2: "Memiliki Rumah"
- Bin 3: "Lainnya"

Dengan setiap bin menunjukkan jumlah atau persentase data yang jatuh ke dalam kategori tersebut.

Hasil dari kode tersebut adalah sebagai berikut:

ans = 4x6 table

	Bin	Good	Bad	Odds	WOE	InfoValue
1	'Home Own...	365	177	2.0621	0.0193	1.6820e-04
2	'Tenant'	307	167	1.8383	-0.0956	0.0037
3	'Other'	131	53	2.4717	0.2005	0.0059
4	'Totals'	803	397	2.0227	NaN	0.0098

Gambar 7. Properties dari 'ResStatus'

Gambar 7 menunjukkan hasil dari perintah `bininfo(sc, 'ResStatus')` dalam bentuk tabel berukuran 4x6. Tabel ini menampilkan informasi mengenai binning dari variabel *ResStatus* (status tempat tinggal) pada dataset yang digunakan dalam model *creditscorecard*. Berikut adalah penjelasan setiap kolom dalam tabel:

1. **Bin:**

- Kolom ini berisi kategori dari variabel ResStatus, yaitu:
 - 'Home Owner': Pelanggan yang memiliki rumah.
 - 'Tenant': Pelanggan yang menyewa tempat tinggal.
 - 'Other': Kategori lainnya (misalnya, status tempat tinggal yang tidak termasuk kepemilikan atau penyewaan).
 - 'Totals': Menunjukkan total jumlah di seluruh kategori.

2. **Good:**

- Kolom ini menunjukkan jumlah pelanggan dalam setiap kategori Bin yang tergolong dalam kategori "Good" (tidak mengalami *default*).
- Sebagai contoh:
 - 365 pelanggan dalam kategori 'Home Owner' termasuk dalam status "Good".
 - 307 pelanggan dalam kategori 'Tenant' termasuk dalam status "Good".
 - 131 pelanggan dalam kategori 'Other' termasuk dalam status "Good".
- Total pelanggan "Good" adalah 803.

3. **Bad:**

- Kolom ini menunjukkan jumlah pelanggan dalam setiap kategori Bin yang tergolong dalam kategori "Bad" (mengalami *default*).
- Sebagai contoh:
 - 177 pelanggan dalam kategori 'Home Owner' termasuk dalam status "Bad".
 - 167 pelanggan dalam kategori 'Tenant' termasuk dalam status "Bad".
 - 53 pelanggan dalam kategori 'Other' termasuk dalam status "Bad".
- Total pelanggan "Bad" adalah 397.

4. **Odds:**

- Kolom ini menunjukkan rasio atau *odds* dari jumlah "Good" terhadap "Bad" dalam setiap bin.
- Perhitungan Odds untuk setiap bin adalah jumlah "Good" dibagi jumlah "Bad".

- Contoh:
 - Pada kategori 'Home Owner', *odds* adalah 2.0621, yang berarti terdapat sekitar 2,06 pelanggan "Good" untuk setiap 1 pelanggan "Bad".
 - Pada kategori 'Other', *odds* lebih tinggi, yaitu 2.4717, yang menunjukkan rasio "Good" yang lebih tinggi dibandingkan kategori lainnya.

5. WOE (Weight of Evidence):

- Kolom ini menunjukkan *Weight of Evidence* (WOE) untuk setiap bin. WOE adalah metrik yang sering digunakan dalam analisis kredit untuk mengukur kekuatan pemisahan antara kelompok "Good" dan "Bad" dalam setiap kategori.
- Nilai WOE dihitung berdasarkan proporsi pelanggan "Good" dan "Bad" dalam setiap bin. Nilai WOE dapat positif atau negatif:
 - Nilai WOE positif menunjukkan bahwa proporsi "Good" lebih besar daripada "Bad" di bin tersebut.
 - Nilai WOE negatif menunjukkan bahwa proporsi "Bad" lebih besar daripada "Good" di bin tersebut.
- Contoh:
 - Pada kategori 'Home Owner', WOE adalah 0.0193, menunjukkan sedikit lebih banyak pelanggan "Good" daripada "Bad".
 - Pada kategori 'Tenant', WOE adalah -0.0956, menunjukkan sedikit lebih banyak pelanggan "Bad" daripada "Good".

6. InfoValue (Information Value):

- Kolom ini menunjukkan *Information Value* (IV) dari setiap bin. IV adalah metrik yang mengukur kekuatan variabel prediktor dalam memisahkan kelas "Good" dan "Bad". Semakin tinggi nilai IV, semakin kuat variabel tersebut dalam memprediksi risiko kredit.
- Contoh:
 - Kategori 'Home Owner' memiliki IV sebesar 0.0001682.
 - Kategori 'Tenant' memiliki IV sebesar 0.0037.
 - Total IV untuk seluruh kategori ResStatus adalah 0.0098.

Informasi bin dari 'ResStatus', seperti yang ditunjukkan pada gambar di atas, akan dilanjutkan dengan tahap plotting. Sebelum itu, Anda dapat mencoba mengganti variabel pada fungsi `bininfo()` dari 'ResStatus' menjadi `CustIncom` karena keduanya merupakan prediktor. Kode untuk melakukan perubahan ini ditampilkan sebagai berikut:

```
bininfo(sc, 'CustIncome')
```

ans = 46x6 table

	Bin	Good	Bad	Odds	WOE	InfoValue
1	'18000'	2	3	0.6667	-1.1099	0.0056
2	'19000'	1	2	0.5000	-1.3976	0.0053
3	'20000'	4	2	2	-0.0113	6.3641e-07
4	'21000'	6	3	2	-0.0113	9.5462e-07
5	'22000'	4	2	2	-0.0113	6.3641e-07
6	'23000'	4	4	1	-0.7044	0.0036
7	'24000'	5	5	1	-0.7044	0.0045
8	'25000'	4	9	0.4444	-1.5153	0.0268
9	'26000'	4	11	0.3636	-1.7160	0.0390

Gambar 8. Properties dari 'CustIncome'

Gambar 8 menunjukkan hasil binning dari variabel CustIncome (pendapatan pelanggan) dalam model *creditscorecard* menggunakan fungsi `bininfo` di MATLAB. Tabel ini berukuran 46x6, yang berarti ada 46 *bin* atau kategori yang dibuat untuk variabel CustIncome. Setiap baris dalam tabel memberikan informasi statistik untuk pendapatan dalam rentang tertentu.

Selanjutnya adalah memplotting data dari variable ResStatus yang telah didapatkan informasinya untuk menampilkan visualisasi atau plot dari binning variabel ResStatus dalam objek *creditscorecard* (sc). Secara keseluruhan, kode ini memvisualisasikan distribusi dan pengaruh variabel ResStatus dalam model, memberikan wawasan tambahan bagi analis dalam mengevaluasi bagaimana status tempat tinggal pelanggan berkorelasi dengan risiko kredit.

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

Fungsi plotbins:

Fungsi `plotbins` digunakan untuk membuat grafik yang memvisualisasikan distribusi bin dari suatu variabel dalam model *creditscorecard*. Visualisasi ini membantu memahami pembagian data di setiap bin dan memberikan wawasan tentang bagaimana variabel tersebut memengaruhi model.

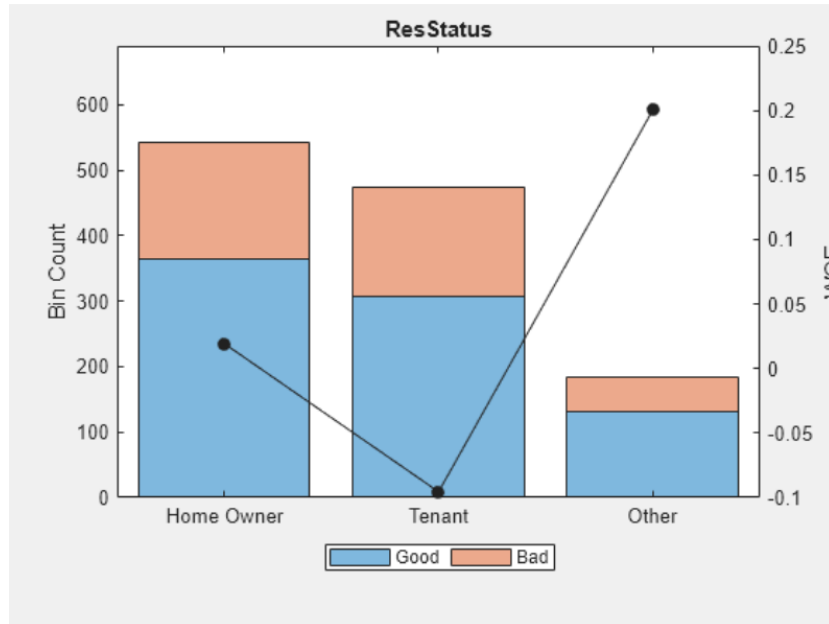
Parameter sc:

sc adalah objek *creditscorecard* yang telah dibuat sebelumnya, yang menyimpan data dan struktur model *scorecard* termasuk binning untuk setiap variabel.

Parameter 'ResStatus':

'ResStatus' adalah variabel yang akan divisualisasikan binning-nya. ResStatus biasanya merupakan variabel kategorikal yang menunjukkan status tempat tinggal pelanggan (misalnya, "Home Owner", "Tenant", atau "Other").

Hasil plotnya adalah sebagai berikut:



Gambar 9. Hubungan antara jumlah Bin 'ResStatus' dengan kurva WOE

Diagram ini adalah visualisasi dari distribusi variabel ResStatus (status tempat tinggal) pada model *creditscorecard*. Diagram ini menunjukkan jumlah pelanggan yang tergolong "Good" (tidak *default*) dan "Bad" (melakukan *default*) untuk masing-masing kategori ResStatus, serta nilai *Weight of Evidence* (WOE) untuk setiap kategori. Berikut adalah penjelasan rinci dari setiap elemen dalam diagram ini:

Komponen Diagram

1. Sumbu Vertikal Kiri (Bin Count):

- Sumbu vertikal kiri menunjukkan jumlah atau frekuensi pelanggan dalam setiap kategori ResStatus.
- Setiap kategori (Home Owner, Tenant, Other) memiliki batang yang dibagi menjadi bagian biru (jumlah pelanggan "Good") dan bagian merah muda (jumlah pelanggan "Bad").

2. Sumbu Vertikal Kanan (WOE):

- Sumbu vertikal kanan menunjukkan nilai *Weight of Evidence* (WOE) untuk setiap kategori ResStatus.
- WOE direpresentasikan dengan garis hitam yang menghubungkan titik-titik pada setiap kategori (Home Owner, Tenant, Other).
- Nilai WOE memberikan indikasi kekuatan pemisahan antara "Good" dan "Bad" di setiap kategori. WOE positif menunjukkan proporsi pelanggan "Good" lebih tinggi, sedangkan WOE negatif menunjukkan proporsi "Bad" lebih tinggi.

3. Batang Warna Biru dan Merah Muda:

- Setiap batang terdiri dari dua bagian:
 - Bagian **biru** menunjukkan jumlah pelanggan "Good" di kategori tersebut.
 - Bagian **merah muda** menunjukkan jumlah pelanggan "Bad" di kategori tersebut.
- Dengan melihat proporsi biru dan merah muda dalam setiap batang, kita bisa memahami distribusi risiko di masing-masing kategori ResStatus.

4. Kategori ResStatus:

- Terdapat tiga kategori:
 - **Home Owner**: Kategori untuk pelanggan yang memiliki rumah.
 - **Tenant**: Kategori untuk pelanggan yang menyewa tempat tinggal.
 - **Other**: Kategori untuk pelanggan dengan status tempat tinggal lain atau yang tidak didefinisikan sebagai pemilik atau penyewa.

Interpretasi Diagram

1. Home Owner:

- Terdapat lebih banyak pelanggan "Good" daripada "Bad" dalam kategori ini, yang dapat dilihat dari ukuran bagian biru yang lebih besar daripada bagian merah muda.
- Nilai WOE untuk kategori ini sedikit positif, menunjukkan bahwa proporsi "Good" sedikit lebih besar daripada "Bad" bagi pemilik rumah.

2. Tenant:

- Jumlah pelanggan "Good" dan "Bad" relatif seimbang di kategori ini, tetapi dengan sedikit lebih banyak pelanggan "Bad" dibandingkan "Good".
- Nilai WOE untuk kategori ini negatif, yang mengindikasikan bahwa proporsi "Bad" lebih tinggi daripada "Good" untuk penyewa tempat tinggal. Hal ini menunjukkan bahwa menjadi penyewa mungkin terkait dengan sedikit peningkatan risiko kredit dibandingkan pemilik rumah.

3. Other:

- Jumlah pelanggan di kategori ini lebih sedikit dibandingkan kategori lainnya, baik untuk "Good" maupun "Bad".

- o WOE di kategori ini adalah positif dan cukup tinggi, yang menunjukkan proporsi "Good" yang lebih besar dibandingkan "Bad". Ini mungkin berarti bahwa pelanggan dengan status tempat tinggal "Other" dianggap kurang berisiko secara relatif.

Dapat disimpulkan bahwa pada variabel 'ResStatus', peminjam yang memiliki rumah, ditambah kepemilikan aset lain seperti apartemen, cenderung mendapatkan pertimbangan lebih baik. Sebaliknya, jika peminjam tidak memiliki rumah atau hanya menyewa, hal ini dapat memberikan dampak negatif. Namun, jika nilai dari variabel lain menunjukkan hasil yang baik, hal tersebut dapat mendukung penilaian secara keseluruhan. Kode yang digunakan dalam tahap ini adalah sebagai berikut:

```
cp = 20000:5000:60000;  
sc = modifybins(sc, 'CustIncome', 'CutPoints', cp);  
bininfo(sc, 'CustIncome')
```

Kode ini melakukan modifikasi binning atau pengelompokan nilai dari variabel *CustIncome* (pendapatan pelanggan) dalam objek *creditscorecard* (*sc*). Binning dilakukan dengan menentukan titik potong baru (cut points) untuk variabel tersebut. Berikut adalah penjelasan detail dari setiap baris kode:

cp = 20000:5000:60000;

Baris ini mendefinisikan variabel *cp* sebagai array dengan rentang dari 20,000 hingga 60,000 dengan interval 5,000.

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

Baris ini memodifikasi binning pada variabel *CustIncome* dalam objek *creditscorecard* (*sc*) dengan menggunakan titik potong yang ditentukan di *cp*.

Fungsi *modifybins* digunakan untuk mengubah aturan binning dari suatu variabel dalam *creditscorecard*. Dalam hal ini, binning dari *CustIncome* akan diubah sehingga pendapatan dikelompokkan berdasarkan interval yang didefinisikan dalam *cp*.

'CutPoints' menentukan bahwa nilai-nilai dalam *cp* akan digunakan sebagai batas atau titik potong untuk membagi rentang pendapatan (*CustIncome*) ke dalam beberapa kelompok/bin.

bininfo(sc, 'CustIncome')

Baris ini memanggil fungsi *bininfo* untuk menampilkan informasi binning terbaru dari variabel *CustIncome*.

Setelah modifikasi bin dilakukan, *bininfo* akan menampilkan hasil binning baru yang mengikuti titik potong yang telah ditentukan di *cp*.

Setelah menjalankan kode tersebut, hasil yang diperoleh adalah sebagai berikut:

ans = 11x6 table

	Bin	Good	Bad	Odds	WOE	InfoValue
1	'[-Inf,20000]'	3	5	0.6000	-1.2152	0.0108
2	'[20000,25000]'	23	16	1.4375	-0.3415	0.0040
3	'[25000,30000]'	38	47	0.8085	-0.9170	0.0652
4	'[30000,35000]'	131	75	1.7467	-0.1467	0.0038
5	'[35000,40000]'	193	98	1.9694	-0.0267	1.7359e-04
6	'[40000,45000]'	173	76	2.2763	0.1181	0.0028
7	'[45000,50000]'	131	47	2.7872	0.3206	0.0143
8	'[50000,55000]'	82	24	3.4167	0.5242	0.0218
9	'[55000,60000]'	21	8	2.6250	0.2607	0.0016

Gambar 10. Predictor CustIncome

Gambar 10 adalah hasil binning terbaru dari variabel CustIncome (pendapatan pelanggan) setelah memodifikasi bin dengan titik potong yang didefinisikan dalam kode sebelumnya. Hasil binning ini disajikan dalam tabel berukuran 11x6, yang menampilkan informasi statistik untuk setiap rentang pendapatan (bin). Untuk mempermudah dalam melihat dan memahami data yang telah diklasifikasikan, fungsi `plotbins()` dapat digunakan. Berikut adalah kode untuk memplot data bin yang telah diklasifikasikan pada tahap sebelumnya:

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

Fungsi plotbins:

plotbins adalah fungsi yang digunakan untuk membuat grafik atau plot binning dari suatu variabel dalam model *creditscorecard*. Grafik ini membantu memvisualisasikan distribusi data dalam setiap bin, baik untuk pelanggan "Good" (tidak *default*) maupun "Bad" (melakukan *default*).

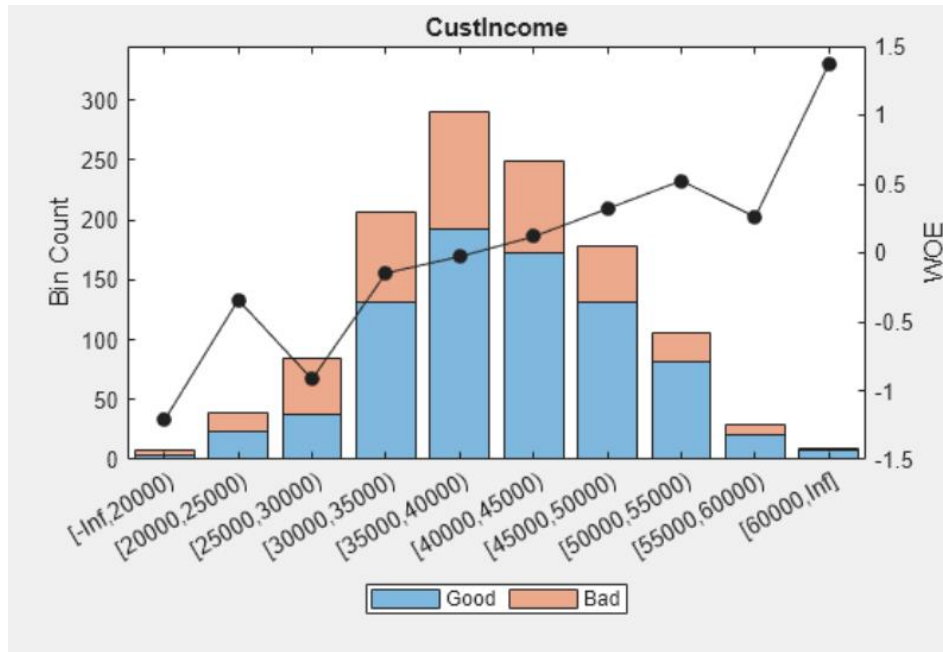
Parameter sc:

sc adalah objek *creditscorecard* yang menyimpan data dan model untuk analisis risiko kredit. Objek ini mencakup struktur binning yang telah ditentukan sebelumnya untuk berbagai variabel prediktor, termasuk CustIncome.

Parameter 'CustIncome':

'CustIncome' adalah nama variabel yang ingin divisualisasikan binning-nya. Variabel ini menunjukkan pendapatan pelanggan, yang telah dikelompokkan dalam beberapa rentang nilai (seperti yang terlihat pada hasil bininfo sebelumnya).

Visualisasi data 'CustIncome' ditampilkan dalam bentuk diagram batang seperti pada gambar berikut:



Gambar 11. Hubungan antara jumlah Bin 'CustIncome' dengan kurva WOE

Diagram ini menggambarkan bagaimana pendapatan pelanggan (CustIncome) berhubungan dengan risiko kredit. Pada pendapatan yang sangat rendah (kurang dari 20,000), terlihat bahwa jumlah pelanggan yang gagal membayar (kategori "Bad") lebih banyak dibandingkan yang tidak gagal (kategori "Good"), dan *Weight of Evidence* (WOE) bernilai negatif. Ini menunjukkan bahwa pendapatan rendah cenderung berisiko lebih tinggi.

Ketika pendapatan meningkat ke rentang menengah (sekitar 30,000 hingga 50,000), jumlah pelanggan "Good" dan "Bad" mulai lebih seimbang. Namun, WOE sedikit positif, menunjukkan bahwa risiko berkurang seiring naiknya pendapatan.

Pada pendapatan yang lebih tinggi (lebih dari 55,000), jumlah pelanggan "Good" jauh lebih besar daripada "Bad", dan WOE bernilai positif tinggi. Ini menunjukkan bahwa pelanggan berpendapatan tinggi memiliki risiko yang jauh lebih rendah untuk gagal membayar.

Secara keseluruhan, diagram ini menunjukkan pola bahwa semakin tinggi pendapatan pelanggan, semakin rendah risiko gagal bayar dalam model kredit.

2. a. Otomatisasi Bin Data

Langkah berikutnya adalah mengklasifikasikan data bin secara otomatis untuk semua variabel prediktor guna menyederhanakan proses analisis. Fungsi yang dapat digunakan untuk melakukan proses ini adalah `autobinning()`. Fungsi ini secara otomatis melakukan pembagian data menjadi beberapa bin berdasarkan setiap variabel prediktor, dengan menggunakan algoritma default 'Monoton', yang memastikan hasil binning memiliki hubungan monoton terhadap target.

Algoritma 'Monoton' sangat berguna untuk menjaga konsistensi pola data dalam hubungannya dengan variabel target, sehingga mempermudah interpretasi hasil analisis. Selain itu, fungsi ini memungkinkan pengguna untuk menyesuaikan opsi algoritma sesuai dengan kebutuhan analisis.

Berikut adalah kode untuk memanggil dan menjalankan fungsi `autobinning`:

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

Perintah `sc = autobinning(sc);` digunakan untuk menjalankan fungsi `autobinning()` pada keseluruhan source data (`sc`) guna melakukan proses binning otomatis. Proses ini membagi variabel prediktor ke dalam bin secara sistematis berdasarkan algoritma default, seperti 'Monoton', untuk memastikan hasil binning sesuai dengan pola data.

Setelah proses binning otomatis selesai, setiap bin dari variabel prediktor perlu ditinjau kembali menggunakan fungsi `bininfo()` untuk memeriksa detail bin dan `plotbins()` untuk memvisualisasikan hasilnya. Peninjauan ini penting untuk memastikan bahwa pola yang dihasilkan menunjukkan tren monoton dan, idealnya, linier dalam Weight of Evidence (WOE). Tren WOE yang monoton dan linier sangat diinginkan dalam pengembangan credit scorecard karena tren ini memungkinkan konversi nilai prediktor menjadi skor yang lebih akurat dan stabil.

Tren WOE dapat divisualisasikan dengan lebih jelas menggunakan fungsi `plotbins()`. Visualisasi ini membantu dalam menilai apakah binning sudah sesuai dengan kebutuhan analisis atau perlu disesuaikan. Berikut adalah kode untuk memvisualisasikan tren WOE menggunakan fungsi `plotbins()`:

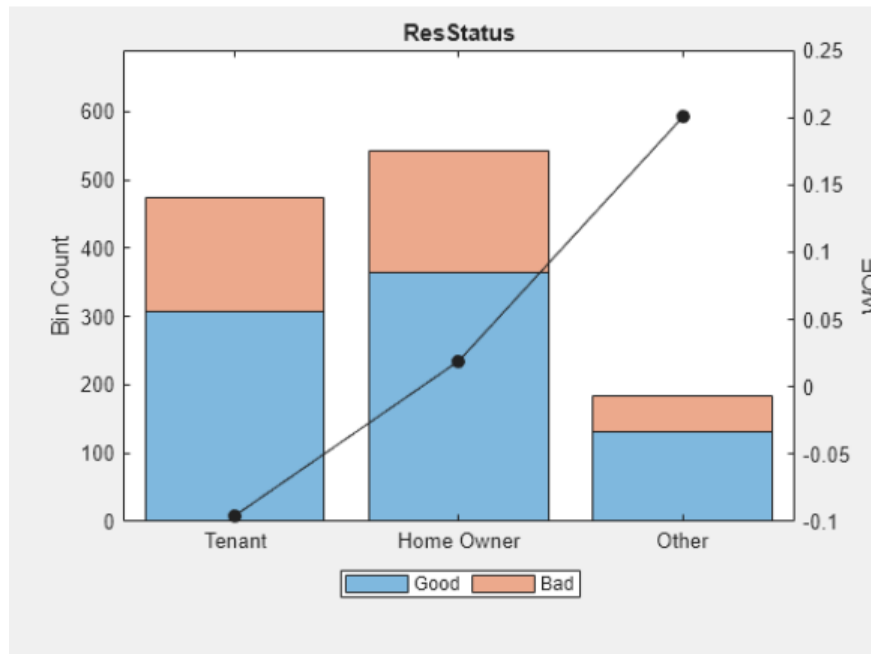
```
Predictor = ResStatus ;  
plotbins(sc, Predictor)
```

Visualisasi data prediktor *ResStatus* ditampilkan dalam bentuk diagram batang, seperti yang terlihat pada gambar di atas. Grafik ini memberikan beberapa informasi penting yang dapat dipahami dari tampilan data tersebut.

Pada prediktor *ResStatus*, terdapat tiga kategori bin, yaitu *Tenant*, *Home Owner*, dan *Other*. Setiap kategori memiliki jumlah *cin count* (frekuensi) yang berbeda, dengan klasifikasi berdasarkan kategori *Good* dan *Bad* untuk setiap bin. Informasi ini menunjukkan distribusi data dalam setiap kategori serta hubungan antara status residensi dan hasil klasifikasi.

Selain itu, tren *Weight of Evidence* (WOE) juga ditampilkan pada grafik ini. WOE memberikan gambaran tentang hubungan monoton antara variabel prediktor dan variabel target, yang sangat penting dalam pengembangan model *credit scoring*. Tren WOE yang stabil atau linier menunjukkan kualitas binning yang baik.

Hasil visualisasi ini dihasilkan melalui fungsi `plotbins()` yang membantu dalam memahami distribusi dan pola data pada setiap kategori bin. Berikut adalah hasil plotting dengan `plotbins()`:



Gambar 12. Tampilan 'ResStatus' melalui proses bin otomatis

Pada awalnya, saat kartu skor dibuat, plot untuk variabel '**ResStatus**' menunjukkan distribusi tren yang cenderung datar atau bahkan fluktuatif. Namun, ketika plot diperbarui menggunakan fungsi **autobinning**, terlihat tren **WOE** (Weight of Evidence) yang meningkat secara signifikan. Perubahan ini disebabkan oleh cara kerja **autobinning** yang secara default mengurutkan kategori berdasarkan peluang kumulatif, yang akhirnya menghasilkan tren yang lebih tajam dan lebih konsisten dalam hal pengaruh setiap kategori terhadap target yang diprediksi.

Autobinning adalah teknik yang mengelompokkan kategori dalam variabel secara otomatis untuk memaksimalkan pemisahan antara kategori-kategori tersebut dengan target yang sedang dianalisis, dalam hal ini menggunakan **WOE** untuk mengukur kekuatan hubungan. Proses ini berfungsi dengan baik dalam memperlihatkan tren yang lebih jelas pada data ini, terutama dalam kasus variabel '**ResStatus**' yang sebelumnya tidak menunjukkan pola yang jelas.

Selanjutnya, penggunaan algoritma **Monotone** dalam proses **binning** menunjukkan hasil yang sangat positif. Algoritma ini secara efektif memastikan bahwa tren **WOE** yang terdeteksi bersifat monoton, artinya semakin besar atau kecilnya kategori akan konsisten dengan perubahan pada target, baik itu peningkatan atau penurunan peluang. Ini adalah aspek penting dalam kartu skor dan model prediktif lainnya, karena tren yang monoton sangat berguna untuk memberikan interpretasi yang lebih jelas tentang pengaruh setiap kategori terhadap hasil prediksi.

Meskipun demikian, meskipun **Monotone** bekerja dengan baik dalam kebanyakan kasus, beberapa prediktor mungkin membutuhkan penyesuaian lebih lanjut. Oleh karena itu, **modifybins** digunakan untuk melakukan penyesuaian manual pada beberapa kategori yang tidak sepenuhnya sesuai dengan pola monoton yang diinginkan. Proses ini memungkinkan model untuk lebih tepat menggambarkan

hubungan antara variabel dan target, sekaligus menjaga interpretabilitas dari kartu skor yang dihasilkan.

Secara keseluruhan, meskipun fungsi **autobinning** dan algoritma **Monotone** sudah berhasil menunjukkan tren yang lebih stabil dan meningkat pada variabel '**ResStatus**', proses **binning** masih memerlukan penyesuaian manual untuk memastikan bahwa model benar-benar mencerminkan realitas data dan menghindari potensi overfitting atau kesalahan dalam pemetaan kategori.

2. b Sempurnakan Binning Menggunakan Pemilahan Manual

Setelah **autobinning** menghasilkan pengelompokan otomatis, langkah berikutnya adalah menyempurnakan hasil pengelompokan ini dengan melakukan **pemilahan manual**. Tujuan dari langkah ini adalah untuk meningkatkan kualitas model dengan memastikan bahwa pengelompokan yang dihasilkan benar-benar mencerminkan hubungan yang konsisten dan bermakna antara variabel dan target, serta meningkatkan interpretabilitas model.

Kode yang digunakan untuk melakukan manual binning adalah sebagai berikut:

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

Kode yang digunakan di atas berguna untuk menampilkan data yang telah dikelompokkan ke dalam format tabel, yang memudahkan kita untuk memvisualisasikan dan menganalisis hasil pengelompokan. Dengan kode ini, kita dapat dengan mudah menyesuaikan variabel prediktor yang ingin dianalisis sesuai dengan kebutuhan model atau analisis kita.

Fungsi **Predictor()** digunakan untuk memilih prediktor yang akan dimuat dalam proses binning. Pada tahap ini, prediktor yang dipilih adalah '**CustAge**' (usia pelanggan). Ini berarti bahwa kita akan melakukan analisis terhadap variabel usia pelanggan untuk melihat bagaimana pengelompokannya dapat membantu memprediksi target.

Selanjutnya, dengan menggunakan perintah **[bi, cp] = bininfo(sc, Predictor)**, kita memanggil fungsi **bininfo()** untuk mengambil informasi mengenai bin yang telah dibuat sebelumnya pada prediktor yang dipilih, yaitu '**CustAge**'. Argumen pertama (**sc**) merujuk pada sumber data yang telah dipilih, sementara argumen kedua (**Predictor**) menentukan prediktor yang ingin kita analisis, dalam hal ini adalah usia pelanggan ('**CustAge**'). Fungsi ini mengembalikan dua nilai:

- **bi**: yang berisi informasi tentang nilai-nilai dalam setiap bin yang telah dibuat.
- **cp**: yang berisi titik potong (cutpoints) yang digunakan untuk mengelompokkan data dalam bin tersebut.

Hasil dari langkah ini adalah kumpulan informasi yang memberikan gambaran yang jelas mengenai cara data usia pelanggan dikelompokkan. Dengan memanfaatkan

informasi ini, kita dapat meninjau pengelompokan dan menentukan apakah ada perubahan atau modifikasi yang perlu dilakukan pada aturan binning untuk meningkatkan performa model.

```

bi = 8x6 table

```

	Bin	Good	Bad	Odds	WOE	InfoValue
1	'[-Inf,33]'	70	53	1.3208	-0.4262	0.0197
2	'[33,37]'	64	47	1.3617	-0.3957	0.0153
3	'[37,40]'	73	47	1.5532	-0.2641	0.0073
4	'[40,46]'	174	94	1.8511	-0.0887	0.0018
5	'[46,48]'	61	25	2.4400	0.1876	0.0024
6	'[48,58]'	263	105	2.5048	0.2138	0.0135
7	'[58,Inf]'	98	26	3.7692	0.6225	0.0352
8	'Totals'	803	397	2.0227	NaN	0.0952

```

cp = 6x1
 33
 37
 40
 46
 48
 58

```

Pada variabel **bi**, ditampilkan informasi terkait bin dari prediktor '**CustAge**', yang mencakup berbagai detail penting seperti rentang nilai pada setiap bin, kategori **good** dan **bad**, serta **WOE** (Weight of Evidence) yang terkait dengan masing-masing bin. Informasi ini memberikan gambaran yang lebih mendalam tentang bagaimana data '**CustAge**' telah dikelompokkan dan bagaimana hubungan antara kelompok usia dengan target prediksi.

Sementara itu, **cp** menampilkan informasi mengenai **cutpoint**, yaitu titik-titik pemotongan yang digunakan untuk membagi rentang data dalam bin yang telah dibuat. Cutpoint ini menunjukkan batasan antara kategori atau kelompok yang berbeda dalam '**CustAge**'. Sebagai contoh, informasi cutpoint ini dapat ditemukan di kolom **bin** dalam tabel **bi**, dengan rentang data tertentu seperti 33, 37, 40, hingga 58, yang menunjukkan bagaimana data usia dibagi menjadi kategori-kategori yang lebih kecil berdasarkan nilai-nilai pemotongan tersebut.

Setelah mendapatkan informasi mengenai bin dan cutpoint, langkah selanjutnya adalah untuk meninjau dan memverifikasi data yang telah diklasifikasikan. Proses ini dilakukan dengan menggunakan fungsi **modifybins()** dan **plotbins()**. Fungsi **modifybins()** memungkinkan kita untuk melakukan perubahan manual pada aturan binning, misalnya dengan menggabungkan atau memodifikasi bin yang ada untuk menghasilkan pengelompokan yang lebih bermakna. Sementara itu, fungsi **plotbins()** digunakan untuk memvisualisasikan distribusi **WOE** dari setiap bin yang ada, sehingga kita dapat dengan mudah melihat apakah pengelompokan yang dilakukan sudah tepat atau perlu ada penyesuaian lebih lanjut.

Kode yang digunakan untuk memanggil kedua fungsi ini akan memperlihatkan hasil klasifikasi data yang lebih jelas dan memungkinkan kita untuk melakukan perubahan jika diperlukan. Dengan demikian, proses ini memastikan bahwa

pengelompokan data lebih optimal dan dapat memberikan hasil yang lebih baik dalam model prediksi.

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

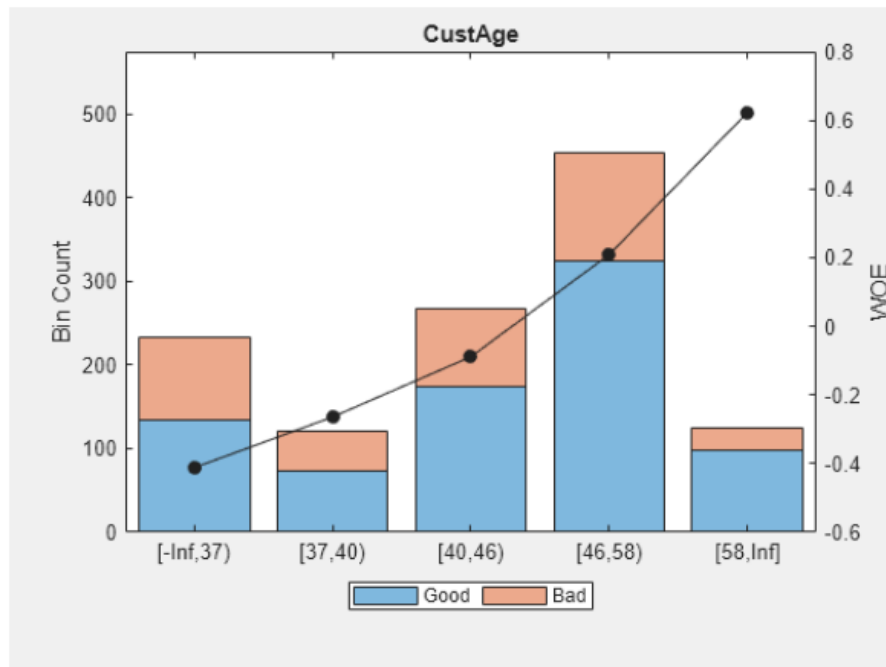
Perintah `cp([1 5]) = []` digunakan untuk menggabungkan (merge) beberapa bin yang ada. Dalam hal ini, bin 1 dan bin 2 digabungkan menjadi satu, serta bin 5 dan bin 6 juga digabungkan. Dengan menghapus elemen pada indeks 1 dan 5 dari daftar cutpoint, kita secara efektif menghilangkan pemisah antara bin-biner tersebut, sehingga nilai-nilai yang berada dalam rentang tersebut akan digabungkan dalam satu bin yang lebih luas.

Setelah melakukan perubahan pada daftar cutpoint, langkah selanjutnya adalah memodifikasi bin berdasarkan aturan yang telah diperbarui menggunakan fungsi `modifybins()`. Perintah `sc = modifybins(sc, 'CustAge', 'CutPoints', cp)` ini diterapkan pada `sc`, yang merupakan sumber data atau struktur dari prediktor `'CustAge'`. Dengan menggunakan daftar cutpoint `cp` yang sudah diperbarui, fungsi ini akan memodifikasi binning untuk prediktor `'CustAge'`, memastikan bahwa pengelompokan yang baru mencerminkan perubahan yang telah kita tentukan.

Terakhir, untuk memverifikasi hasil dari modifikasi bin, kita dapat menggunakan fungsi `plotbins()` dengan perintah `plotbins(sc, 'CustAge')`. Fungsi ini akan memplot distribusi **WOE** dari prediktor `'CustAge'` setelah penggabungan bin dilakukan. Visualisasi ini memungkinkan kita untuk menilai apakah perubahan yang dilakukan pada bin telah menghasilkan pengelompokan yang lebih baik dan lebih konsisten dalam hal hubungan dengan target.

Dengan mengikuti langkah-langkah ini, kita dapat memastikan bahwa pengelompokan data lebih optimal, memberikan gambaran yang lebih jelas tentang bagaimana variabel `'CustAge'` berinteraksi dengan target prediksi, serta meningkatkan performa model secara keseluruhan.

Hasil dari pemrograman plotting yang dilakukan dengan kode di atas menghasilkan visualisasi seperti berikut:



Dari data visual yang ditampilkan dalam bentuk diagram batang (bar chart), kita dapat melihat bahwa grafik ini menggambarkan informasi yang sudah ada sebelumnya dalam tabel pengelompokan bin untuk prediktor ‘**CustAge**’. **Bin count** merujuk pada jumlah bin yang termasuk dalam rentang nilai tertentu, dan masing-masing bin dapat diklasifikasikan dalam dua kategori, yaitu “**Good**” dan “**Bad**”, sesuai dengan rentang yang telah ditentukan, seperti pada rentang antara 37 hingga 58. Plot ini secara jelas menunjukkan bahwa algoritma ‘**Monotone**’ telah bekerja dengan efektif dalam mengidentifikasi dan menunjukkan tren **WOE** yang konsisten dan monoton pada set data ini.

Selanjutnya, untuk prediktor ‘**CustIncome**’, berdasarkan analisis pada plot sebelumnya, disarankan untuk menggabungkan bin 3, 4, dan 5. Hal ini disebabkan oleh fakta bahwa ketiga bin tersebut memiliki nilai **WOE** yang serupa, sehingga penggabungan bin-biner tersebut akan menyederhanakan model tanpa mengorbankan informasi yang relevan. Berikut adalah kode yang digunakan untuk menggabungkan bin-biner tersebut:

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

Fungsi **Predictor()** digunakan untuk memilih prediktor yang akan dimasukkan ke dalam proses analisis. Pada tahap ini, prediktor yang dipilih adalah ‘**CustIncome**’ (pendapatan pelanggan). Dengan memilih ‘**CustIncome**’, kita akan fokus pada variabel pendapatan pelanggan untuk melihat bagaimana pengelompokan dan pemotongan data dapat mempengaruhi model.

Selanjutnya, perintah **[bi, cp] = bininfo(sc, Predictor)** digunakan untuk memanggil fungsi **bininfo()**, yang akan mengekstrak informasi terkait bin yang dibuat untuk prediktor ‘**CustIncome**’ dari sumber data yang dipilih (**sc**). Fungsi ini mengembalikan dua nilai penting:

- **bi**: yang berisi detail informasi tentang setiap bin, seperti rentang nilai dan kategori yang ada di dalamnya.
- **cp**: yang berisi informasi tentang **cutpoint**, yaitu titik pemotongan yang digunakan untuk membagi data ‘**CustIncome**’ menjadi bin-bin yang lebih kecil.

Dengan informasi ini, kita dapat mengatur dan menyesuaikan proses binning untuk prediktor ‘**CustIncome**’, memastikan bahwa pengelompokan data yang dihasilkan sesuai dengan tujuan analisis dan model prediksi yang ingin dibangun.

Hasil dari tahap ini adalah sebagai berikut: informasi mengenai **bin** untuk prediktor ‘**CustIncome**’ yang mencakup rentang nilai, kategori **Good** dan **Bad**, serta **WOE** (Weight of Evidence) yang dihasilkan untuk setiap bin. Selain itu, daftar **cutpoint** juga ditampilkan, menunjukkan pemotongan pada nilai-nilai tertentu yang membagi data menjadi kelompok-kelompok yang lebih kecil. Data ini memberikan dasar untuk memodifikasi atau memverifikasi pengelompokan yang telah dilakukan, dan menjadi langkah awal dalam proses binning yang lebih lanjut untuk meningkatkan akurasi model prediksi.

```
bi = 8x6 table
```

	Bin	Good	Bad	Odds	WOE	InfoValue
1	'[-Inf,29000]'	53	58	0.9138	-0.7946	0.0636
2	'[29000,33000]'	74	49	1.5102	-0.2922	0.0091
3	'[33000,35000]'	68	36	1.8889	-0.0684	4.1042e-04
4	'[35000,40000]'	193	98	1.9694	-0.0267	1.7359e-04
5	'[40000,42000]'	68	34	2	-0.0113	1.0819e-05
6	'[42000,47000]'	164	66	2.4848	0.2058	0.0078
7	'[47000,Inf]'	183	56	3.2679	0.4797	0.0417
8	'Totals'	803	397	2.0227	NaN	0.1228

```
cp = 6x1
29000
33000
35000
40000
42000
47000
```

Pada **bi**, ditampilkan informasi mengenai bin untuk prediktor ‘**CustIncome**’, yang mencakup berbagai detail seperti **rentang nilai bin**, kategori **good** dan **bad**, serta **WOE** (Weight of Evidence) untuk masing-masing bin. **WOE** ini memberikan gambaran tentang hubungan antara nilai-nilai dalam bin dengan target prediksi. Sementara itu, **cp** menampilkan **cutpoint**, yaitu titik pemotongan yang digunakan untuk membagi rentang data menjadi bin-biner yang lebih kecil. Informasi cutpoint ini dapat dilihat juga pada tabel **bi**, tepatnya di kolom **bin**, yang menunjukkan rentang data yang dimulai dari 29.000 hingga 47.000, yang merupakan batasan untuk pemotongan data ‘**CustIncome**’.

Setelah memperoleh informasi ini, langkah selanjutnya adalah untuk mereview dan memverifikasi data yang telah diklasifikasikan. Hal ini dilakukan dengan menggunakan fungsi **modifybins()** dan **plotbins()**. Fungsi **modifybins()** memungkinkan kita untuk melakukan penyesuaian lebih lanjut pada

pengelompokan bin, misalnya dengan menggabungkan bin-biner yang memiliki **WOE** serupa atau mengubah titik potong jika diperlukan. Sementara itu, **plotbins()** digunakan untuk memvisualisasikan hasil pengelompokan, sehingga kita dapat melihat distribusi **WOE** untuk prediktor '**CustIncome**' dan menilai apakah pengelompokan yang ada sudah optimal.

Kode yang digunakan untuk melakukan langkah-langkah ini adalah sebagai berikut:

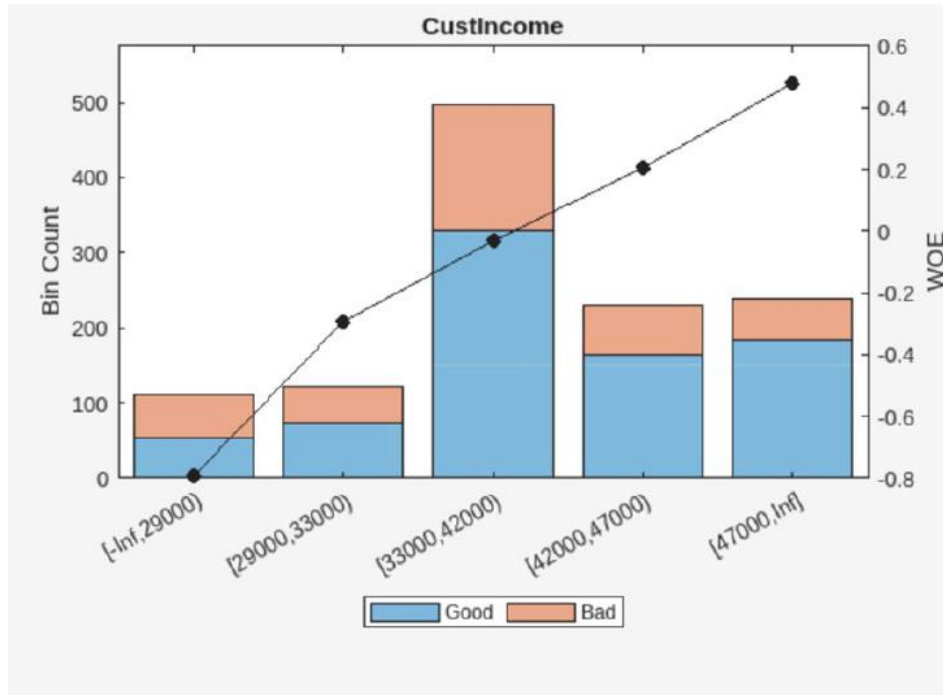
```
cp([3 4]) = []; % To merge bins 3, 4, and 5
sc = modifybins(sc, 'CustIncome', 'CutPoints', cp);
plotbins(sc, 'CustIncome')
```

Perintah **cp([3 4]) = []** digunakan untuk menggabungkan (merge) bin 3, bin 4, dan bin 5. Dengan menghapus elemen pada indeks 3 dan 4 dalam daftar **cutpoint (cp)**, kita menghapus pemisah antara bin-bin tersebut, sehingga nilai-nilai dalam rentang tersebut akan digabungkan menjadi satu bin yang lebih besar.

Selanjutnya, untuk menerapkan perubahan yang telah dilakukan pada cutpoint, kita menggunakan perintah **sc = modifybins(sc, 'CustIncome', 'CutPoints', cp)**. Fungsi **modifybins()** ini memodifikasi pengelompokan bin pada **sc**, yang merupakan sumber data untuk prediktor '**CustIncome**', berdasarkan **cutpoints** yang telah diperbarui. Dengan demikian, pengelompokan data '**CustIncome**' akan disesuaikan sesuai dengan aturan baru yang telah ditetapkan.

Terakhir, untuk memverifikasi hasil dari modifikasi tersebut, kita menggunakan **plotbins(sc, 'CustIncome')**. Fungsi ini memvisualisasikan distribusi **WOE** dari prediktor '**CustIncome**' berdasarkan bin yang telah diperbarui. Dengan melihat plot ini, kita dapat menilai apakah pengelompokan yang baru sudah sesuai dengan harapan dan memberikan wawasan yang lebih jelas tentang hubungan antara pendapatan pelanggan dan target prediksi.

Hasil dari pemrograman plotting yang dilakukan dengan kode di atas menghasilkan visualisasi sebagai berikut:



Plot yang ditampilkan menunjukkan distribusi **WOE** untuk prediktor '**CustIncome**' setelah penggabungan bin dilakukan. Dari visualisasi ini, kita dapat mengamati bagaimana pengelompokan data yang telah dimodifikasi memengaruhi distribusi **WOE** dan apakah perubahan yang dilakukan pada bin menghasilkan tren yang lebih jelas dan konsisten dengan target prediksi. Dengan melihat hasil plotting ini, kita bisa mengevaluasi apakah modifikasi bin telah berhasil meningkatkan pemahaman tentang hubungan antara '**CustIncome**' dan target yang diprediksi.

Dari data visual yang ditampilkan dalam bentuk **bar chart**, kita dapat menyimpulkan bahwa grafik ini mencerminkan informasi yang telah tersedia dalam tabel pengelompokan bin untuk prediktor '**CustIncome**'. **Bin count** menunjukkan jumlah bin yang termasuk dalam rentang nilai tertentu, yang masing-masing dapat dikategorikan sebagai "**Good**" atau "**Bad**", sesuai dengan klasifikasi yang telah ditetapkan pada rentang **CustIncome**, yaitu antara 29.000 hingga 47.000. Visualisasi ini juga mengungkapkan bahwa algoritme '**Monotone**' bekerja dengan efektif dalam menemukan tren **WOE** yang konsisten dan monoton pada set data ini, menunjukkan kemampuan algoritme untuk mengidentifikasi hubungan yang kuat antara **CustIncome** dan target yang diprediksi.

Untuk prediktor '**TmWBank**' (Tenure with Bank), berdasarkan analisis pada plot sebelumnya, disarankan untuk menggabungkan bin 2 dan bin 3. Hal ini disebabkan oleh fakta bahwa kedua bin tersebut memiliki **WOE** yang serupa, yang menunjukkan bahwa penggabungan keduanya akan lebih menyederhanakan model tanpa mengorbankan informasi penting. Dengan menggabungkan bin-bini ini, kita dapat meningkatkan kestabilan model dan mengurangi potensi overfitting yang dapat terjadi akibat pemisahan yang tidak diperlukan.

Kode untuk menggabungkan bin 2 dan 3 adalah sebagai berikut:

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

Fungsi **Predictor()** digunakan untuk memilih prediktor yang akan dimasukkan ke dalam proses analisis. Pada tahap ini, prediktor yang dipilih adalah **'TmWBank'** (Tenure with Bank), yang merujuk pada lamanya waktu seorang pelanggan telah terdaftar di bank. Pemilihan prediktor ini bertujuan untuk memahami bagaimana durasi hubungan pelanggan dengan bank memengaruhi model prediksi.

Selanjutnya, perintah **[bi, cp] = bininfo(sc, Predictor)** digunakan untuk memanggil fungsi **bininfo()**, yang akan mengembalikan informasi terkait pengelompokan bin untuk prediktor **'TmWBank'** dari sumber data **sc** yang telah dipilih. Fungsi ini menghasilkan dua nilai penting:

- **bi**: yang berisi informasi terkait setiap bin, termasuk rentang nilai dan distribusi kategori **Good** dan **Bad** untuk setiap bin.
- **cp**: yang berisi informasi **cutpoint**, yaitu titik pemotongan yang digunakan untuk membagi data **'TmWBank'** ke dalam bin yang lebih kecil.

Hasil dari tahap ini adalah informasi terkait bin dan cutpoint yang digunakan untuk mengelompokkan data **'TmWBank'**, yang akan memberikan dasar bagi penyesuaian dan analisis lebih lanjut terkait hubungan antara **'TmWBank'** dan target prediksi.

```
bi = 6x6 table
```

	Bin	Good	Bad	Odds	WOE	InfoValue
1	'[-Inf,12]'	141	90	1.5667	-0.2555	0.0131
2	'[12,23]'	165	93	1.7742	-0.1311	0.0038
3	'[23,45]'	224	125	1.7920	-0.1211	0.0043
4	'[45,71]'	177	67	2.6418	0.2670	0.0138
5	'[71,Inf]'	96	22	4.3636	0.7689	0.0493
6	'Totals'	803	397	2.0227	NaN	0.0843

```
cp = 4x1
    12
    23
    45
    71
```

Pada **bi**, ditampilkan informasi mengenai bin untuk prediktor **'TmWBank'**, yang mencakup berbagai detail seperti **rentang nilai bin**, kategori **good** dan **bad**, serta **WOE** (Weight of Evidence) untuk masing-masing bin. **WOE** memberikan gambaran tentang pengaruh nilai-nilai dalam bin terhadap prediksi target. Selain itu, **cp** menampilkan **cutpoint**, yaitu titik pemotongan yang digunakan untuk membagi data **'TmWBank'** menjadi beberapa bin berdasarkan rentang nilai tertentu. Informasi cutpoint ini juga dapat dilihat dalam tabel **bi**, khususnya di kolom **bin**, yang menunjukkan rentang data yang dimulai dari **12** hingga **71**.

Setelah memperoleh informasi mengenai bin dan cutpoint, langkah selanjutnya adalah untuk mereview data yang telah diklasifikasikan. Ini dilakukan dengan

menggunakan fungsi **modifybins()** dan **plotbins()**. Fungsi **modifybins()** memungkinkan kita untuk melakukan penyesuaian lebih lanjut terhadap pengelompokan bin yang ada, seperti menggabungkan bin yang memiliki **WOE** yang serupa atau mengubah titik pemotongan jika diperlukan. Kemudian, **plotbins()** digunakan untuk memvisualisasikan hasil pengelompokan, memberikan gambaran yang lebih jelas tentang distribusi **WOE** dari prediktor '**TmWBank**' dan menilai apakah perubahan pada pengelompokan telah memberikan hasil yang lebih sesuai untuk model prediksi.

Kode yang digunakan untuk melakukan langkah-langkah ini adalah sebagai berikut:

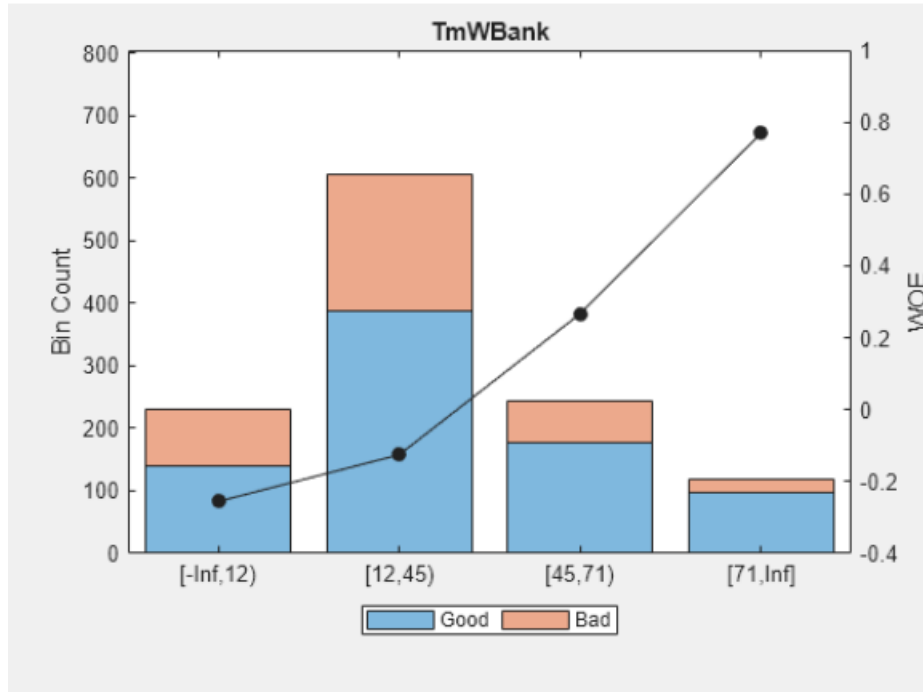
```
cp(2) = []; % To merge bins 2 and 3
sc = modifybins(sc, 'TmWBank', 'CutPoints', cp);
plotbins(sc, 'TmWBank')
```

Perintah **cp(2) = []** digunakan untuk menggabungkan (merge) bin 2 dan bin 3. Dengan menghapus elemen kedua dalam daftar **cutpoint (cp)**, kita menghilangkan pemisah antara kedua bin tersebut, sehingga rentang nilai dari bin 2 dan bin 3 akan digabungkan menjadi satu bin yang lebih besar.

Setelah perubahan cutpoint dilakukan, kita menggunakan **sc = modifybins(sc, 'TmWBank', 'CutPoints', cp)** untuk menerapkan perubahan ini pada **sc**, yang merupakan sumber data untuk prediktor '**TmWBank**'. Fungsi **modifybins()** akan memodifikasi pengelompokan data berdasarkan cutpoint yang baru.

Selanjutnya, untuk memverifikasi hasil perubahan pengelompokan, kita menggunakan **plotbins(sc, 'TmWBank')**. Fungsi ini akan memvisualisasikan data '**TmWBank**' yang telah diperbarui, memungkinkan kita untuk menilai apakah pengelompokan yang telah dimodifikasi memberikan hasil yang diinginkan dalam hal distribusi **WOE** dan keterkaitannya dengan target yang diprediksi.

Hasil dari plotting kode di atas adalah sebagai berikut:



Dari data visual yang ditampilkan dalam bentuk **bar chart**, kita dapat melihat bahwa grafik ini merepresentasikan informasi yang sebelumnya telah ada dalam tabel pengelompokan bin untuk prediktor ‘**TmWBank**’. **Bin count** menunjukkan jumlah bin yang terdaftar dalam setiap rentang tertentu, dan masing-masing bin tersebut dapat diklasifikasikan dalam kategori “**Good**” dan “**Bad**” berdasarkan distribusi yang ada dalam rentang nilai, yang dimulai dari **12** hingga **71**. Visualisasi ini memberikan gambaran yang jelas mengenai distribusi **WOE** dan bagaimana data di dalam setiap bin memberikan kontribusi terhadap model prediksi. Plot ini juga mengindikasikan bahwa algoritme ‘**Monotone**’ berhasil mengidentifikasi tren **WOE** yang konsisten dan monoton dalam set data ini, yang menunjukkan bahwa hubungan antara prediktor ‘**TmWBank**’ dan target prediksi terstruktur dengan baik.

Selanjutnya, untuk prediktor ‘**AMBalance**’ (Account Balance), berdasarkan analisis pada plot sebelumnya, disarankan untuk menggabungkan bin 2 dan bin 3, karena keduanya memiliki nilai **WOE** yang serupa, yang mengindikasikan bahwa penggabungan bin tersebut tidak akan mengurangi informasi yang relevan. Dengan menggabungkan bin-bin ini, kita dapat menyederhanakan model tanpa mengorbankan kualitas prediksi.

Kode untuk menggabungkan bin 2 dan 3 pada prediktor ‘**AMBalance**’ adalah sebagai berikut:

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

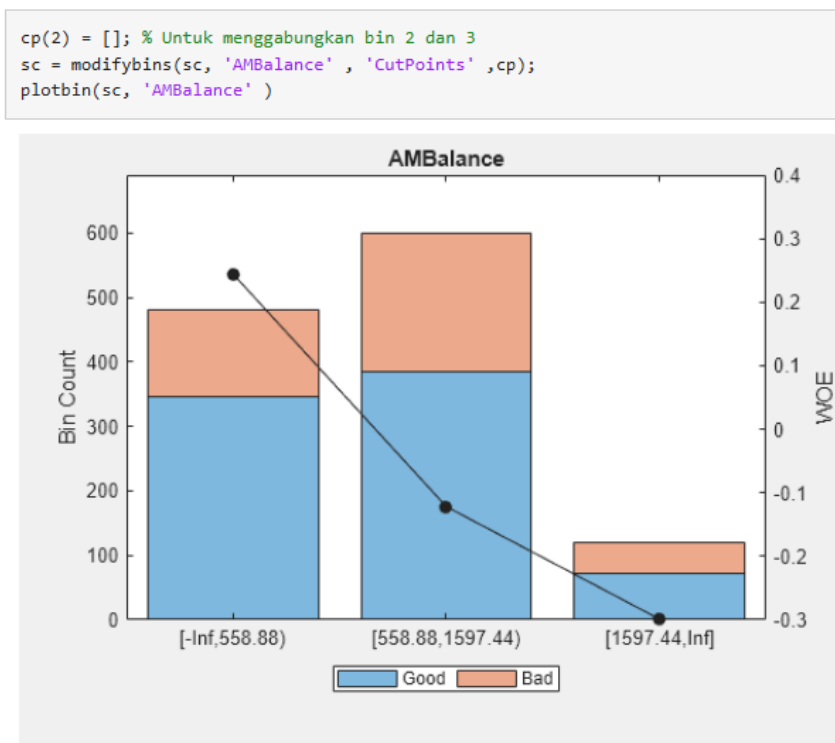
bi = 5x6 table

	Bin	Good	Bad	Odds	WOE	InfoValue
1	'[-Inf,558.88)'	346	134	2.5821	0.2442	0.0228
2	'[558.88,1254.28)'	309	171	1.8070	-0.1127	0.0052
3	'[1254.28,1597.44)'	76	44	1.7273	-0.1579	0.0026
4	'[1597.44,Inf]'	72	48	1.5000	-0.2990	0.0093
5	'Totals'	803	397	2.0227	NaN	0.0399

cp = 3x1
 $10^3 \times$
 0.5589
 1.2543
 1.5974

Pada **bi**, ditampilkan informasi mengenai bin untuk prediktor ‘**AMBalance**’ (Account Balance). Informasi yang disediakan mencakup berbagai aspek penting, seperti **rentang nilai bin**, distribusi kategori **good** dan **bad**, serta nilai **WOE** (Weight of Evidence) yang menggambarkan kekuatan hubungan antara nilai bin dan target prediksi. Rentang nilai ini menunjukkan bagaimana data ‘**AMBalance**’ terbagi dalam beberapa kelompok berdasarkan nilai yang dimiliki oleh masing-masing individu.

Selain itu, **cp** menampilkan informasi terkait **cutpoint**, yaitu titik pemotongan yang digunakan untuk membagi data ‘**AMBalance**’ menjadi beberapa bin. Cutpoint ini mencerminkan rentang nilai pada data yang menjadi batas pemisah antara kelompok-kelompok yang lebih kecil. Informasi cutpoint ini dapat ditemukan pada tabel **bi**, khususnya pada kolom **bin**, yang menunjukkan rentang mulai dari **0.5589** hingga **1.5974**.



3. Penyesuaian model regresi dengan WOE

Setelah nilai **WOE** (Weight of Evidence) dihitung untuk setiap prediktor yang telah dianalisis, langkah selanjutnya adalah melakukan penyesuaian nilai **WOE** terhadap model **regresi**. Nilai **WOE** ini akan digunakan untuk mengubah kategori prediktor menjadi variabel numerik yang dapat dimasukkan ke dalam model regresi, yang memungkinkan model untuk mengukur dampak masing-masing prediktor terhadap probabilitas target.

Dalam model regresi, setiap nilai yang diklasifikasikan sebagai “**Good**” akan dipetakan menjadi angka **1**. Hal ini berarti bahwa data yang termasuk dalam kategori “**Good**” akan diberi bobot numerik yang mewakili hasil yang diinginkan dalam model, yaitu kejadian yang lebih diinginkan atau lebih menguntungkan. Begitu nilai **WOE** untuk setiap bin telah dimasukkan dalam model, sistem akan memperhitungkan pengaruh setiap prediktor terhadap kemungkinan hasil yang diinginkan.

Secara default, model regresi akan menggunakan prosedur **bertahap** (stepwise procedure) untuk memilih prediktor mana yang akan dimasukkan ke dalam model. Prosedur ini bekerja dengan menilai kontribusi masing-masing prediktor terhadap akurasi dan kekuatan model. Proses ini dapat dilakukan dalam dua arah: **forward selection**, yang menambah prediktor satu per satu, dan **backward elimination**, yang menghapus prediktor yang tidak signifikan untuk meningkatkan efisiensi model.

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

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

Model regresi linier umum:

```
logit(status) ~ 1 + CustAge + ResStatus + EmpStatus + CustIncome + TmWBank + OtherCC + ABalance  
Distribusi = Binomial
```

Model yang digunakan dalam analisis ini adalah **model regresi linier** dengan **8 suku** dan **7 prediktor**, serta distribusi yang diterapkan adalah **binomial**. Model ini dirancang untuk memetakan hubungan antara prediktor yang telah dimodifikasi dengan nilai **WOE** dan variabel target dalam konteks klasifikasi dua kelas, yakni **Good** dan **Bad**.

Hasil estimasi koefisien yang diperoleh dari model regresi ini ditampilkan pada **Gambar 20**, yang memberikan gambaran jelas tentang kontribusi masing-masing prediktor terhadap prediksi model. Dari total **1200 data** yang digunakan, terdapat **1192 derajat kebebasan** untuk menghitung kesalahan (degrees of freedom), yang menunjukkan jumlah data yang tersedia untuk menghitung model setelah memperhitungkan variabel prediktor yang dimasukkan.

Untuk mengukur kualitas model, digunakan **dispersion** yang bernilai **1**, yang berarti model tidak memiliki varians yang berlebihan atau kekurangan dalam hal kesalahan prediksi. Selain itu, model ini menghasilkan **Chi-squared statistic vs. model konstan sebesar 89,7**, yang menunjukkan seberapa jauh model regresi ini lebih baik dibandingkan dengan model yang hanya memprediksi rata-rata (model konstan). Nilai **p-value** yang diperoleh adalah **1,42e-16**, yang menunjukkan bahwa model ini sangat signifikan secara statistik, karena p-value yang sangat kecil menunjukkan bahwa model yang dihasilkan lebih baik daripada model konstan dengan tingkat kepercayaan yang tinggi.

```

Koefisien yang Diperkirakan:
      Perkiraan SE tStat pValue
-----
(Intersepsi) 0,7024 0,064 10,975 5,0407e-28
Usia Pelanggan 0,61562 0,24783 2,4841 0,012988
Status Res 1,3776 0,65266 2,1107 0,034799
Status Kosong 0,88592 0,29296 3,024 0,0024946
Pendapatan Cust 0,69836 0,21715 3,216 0,0013001
TmWBank 1.106 0.23266 4.7538 1.9958e-06
CC Lainnya 1,0933 0,52911 2,0662 0,038806
Keseimbangan AM 1,0437 0,32292 3,2322 0,0012285

1200 observasi, 1192 derajat kebebasan kesalahan
Dispersi: 1
Statistik Chi^2 vs. model konstan: 89,7, nilai-p = 1,42e-16

```

Dalam analisis regresi linier, metode ini digunakan sebagai dasar untuk melakukan prediksi terhadap data prediktor yang dimiliki. Tujuan dari analisis ini adalah untuk menentukan sejauh mana seorang pengguna (user) layak untuk diterima atau ditolak pengajuan kartu kreditnya, berdasarkan sejumlah variabel yang relevan.

Dari hasil analisis koefisien estimasi pada gambar **20**, kita dapat melihat hubungan antar variabel prediktor dalam model regresi. **Prediktor ‘ResStatus’** menunjukkan nilai koefisien estimasi yang paling baik, yaitu **1.3776**, yang menandakan bahwa status pekerjaan atau status residen seseorang memiliki pengaruh yang sangat signifikan terhadap keputusan penerimaan atau penolakan pengajuan kartu kredit. Setelah itu, **‘TmWBank’** muncul sebagai prediktor penting dengan koefisien **1.106**, menunjukkan hubungan yang kuat antara durasi hubungan peminjam dengan bank dan kelayakan pengajuan kartu kredit.

Faktor-faktor seperti **status tempat tinggal** dan **lama kebersamaan peminjam di bank** juga berkontribusi secara signifikan dalam menjelaskan keputusan model, menjadikannya prediktor yang paling berhubungan satu sama lain. Hal ini menunjukkan bahwa hubungan jangka panjang antara peminjam dan bank dapat mempengaruhi keputusan persetujuan pengajuan.

Selain itu, untuk nilai **error**, **‘CustIncome’** (0.21715) dan **‘TmWBank’** (0.23266) muncul sebagai prediktor dengan error yang paling rendah, menandakan bahwa kedua prediktor ini memberikan estimasi yang lebih akurat dalam model dibandingkan dengan yang lainnya. Semakin rendah nilai error, semakin baik prediktor tersebut dalam memprediksi hasil yang diinginkan.

Terakhir, untuk nilai **p-value**, '**TmWBank**' dan '**AMBalance**' menunjukkan nilai p-value yang paling baik, mengindikasikan bahwa kedua prediktor ini sangat signifikan secara statistik dalam mempengaruhi keputusan model. Nilai p-value yang kecil berarti ada hubungan yang kuat antara prediktor dan variabel dependen, dan oleh karena itu, prediktor-prediktor ini sangat relevan dalam model untuk memprediksi kelayakan pengajuan kartu kredit.

4. Meriview dan memformat poin scorecard

Setelah melakukan penyesuaian pada model regresi, penting untuk dicatat bahwa secara **default**, poin **scorecard** tidak langsung diskalakan, dan data yang digunakan berasal langsung dari **kombinasi nilai WOE** dan **koefisien model regresi** yang telah dihitung sebelumnya. Dalam hal ini, model regresi memberikan kontribusi untuk menentukan pengaruh setiap prediktor terhadap kelayakan pengajuan kartu kredit, sedangkan nilai **WOE** (Weight of Evidence) mengukur hubungan antara prediktor dan hasil yang diinginkan.

Gambar tersebut memberikan gambaran umum mengenai **ringkasan poin scorecard**, yang menunjukkan bagaimana kombinasi antara WOE dan koefisien regresi tersebut diubah menjadi skor yang lebih mudah dipahami dan diinterpretasikan. Poin-poin pada scorecard ini akan membantu dalam membuat keputusan apakah seorang peminjam layak untuk diterima atau ditolak dalam proses pengajuan kartu kredit berdasarkan nilai yang telah dihitung sebelumnya.

Dalam scorecard tersebut, setiap prediktor akan memiliki nilai poin tertentu yang dihitung dengan mengalikan koefisien model regresi dengan nilai WOE yang sesuai. Poin ini akan memberikan kontribusi terhadap skor total peminjam, yang kemudian akan digunakan sebagai dasar untuk pengambilan keputusan.

```

{'Usia Pelanggan' } {'[-Inf,37)'} -0.15314
{'Umur Pelanggan' } {'[37,40)'} -0,062247
{'Usia Cust' } {'[40,46)'} 0,045763
{'Usia Cust' } {'[46,58)'} 0,22888
{'Usia Pelanggan' } {'[58,Inf]'} 0.48354
{'CustAge' } {'<hilang>'} NaN
{'ResStatus' } {'Penyewa' } -0,031302
{'ResStatus' } {'Pemilik Rumah' } 0.12697
{'ResStatus' } {'Lainnya' } 0,37652
{'ResStatus' } {'<hilang>'} NaN
{'EmpStatus' } {'Tidak Diketahui' } -0,076369
{'EmpStatus' } {'Dipekerjakan' } 0.31456
{'EmpStatus' } {'<hilang>'} NaN
{'CustIncome' } {'[-Inf,29000)'} -0,45455
{'CustIncome' } {'[29000,33000)'} -0,1037
{'CustIncome' } {'[33000,42000)'} 0,077768
{'Pendapatan Pelanggan' } {'[42000,47000)'} 0,24406
{'CustIncome' } {'[47000,Inf]'} 0,43536
{'CustIncome' } {'<hilang>'} NaN
{'TmlwBank' } {'[-Inf,12)'} -0,18221
{'Bank Tmlw' } {'[12,45)'} -0,038279
{'Bank Tmlw' } {'[45,71)'} 0,39569
{'TmlwBank' } {'[71,Inf]'} 0,95074
{'TmlwBank' } {'<hilang>'} NaN
{'OtherCC' } {'Tidak' } -0,193
{'OtherCC' } {'Ya' } 0,15868
{'OtherCC' } {'<hilang>'} NaN
{'Keseimbangan AMB' } {'[-Inf,558.88)'} 0,3552
{'Keseimbangan AMB' } {'[558.88,1597.44)'} -0.026797
{'Keseimbangan AMB' } {'[1597.44,Inf]'} -0.21168
{'AMBalance' } {'<hilang>'} NaN

```

Tampilan Gambar 21 masih belum begitu menarik perlu dilakukan beberapa penyesuaian agar hasilnya menjadi lebih menarik dan lebih enak untuk dilihat agar lebih mudah untuk dipahami.

```

sc = modifybins(sc, 'CustAge' , 'BinLabels' , ...
{ 'Sampai dengan 36' '37 sampai 39' '40 sampai 45' '46 sampai 57' '58 ke atas' });

sc = modifybins(sc, 'CustIncome' , 'BinLabels' , ...
{ 'Sampai dengan 28999' '29000 sampai 32999' '33000 sampai 41999' '42000 sampai 46999' '47000 dan seterusnya' });

sc = modifybins(sc, 'TmlwBank' , 'BinLabels' , ...
{ 'Sampai dengan 11' '12 sampai 44' '45 sampai 70' '71 ke atas' });

sc = modifybins(sc, 'AMBalance' , 'BinLabels' , ...
{ 'Sampai dengan 558,87' '558,88 sampai 1597,43' '1597,44 dan seterusnya' });

p1 = titik_tampilan(sc);
menampilkan(p1)

```

Poin Bin Prediktor

```
{'CustAge' } {'Hingga 36' } -0,15314
{'Usia Pelanggan' } {'37 hingga 39' } -0,062247
{'Usia Pelanggan' } {'40 hingga 45' } 0,045763
{'Usia Pelanggan' } {'46 hingga 57' } 0,22888
{'CustAge' } {'58 ke atas' } 0,48354
{'CustAge' } {'<hilang>' } NaN
{'ResStatus' } {'Penyewa' } -0,031302
{'ResStatus' } {'Pemilik Rumah' } 0,12697
{'ResStatus' } {'Lainnya' } 0,37652
{'ResStatus' } {'<hilang>' } NaN
{'EmpStatus' } {'Tidak Diketahui' } -0,076369
{'EmpStatus' } {'Dipekerjakan' } 0,31456
{'EmpStatus' } {'<hilang>' } NaN
{'CustIncome' } {'Hingga 28999' } -0,45455
{'CustIncome' } {'29000 hingga 32999' } -0,1037
{'CustIncome' } {'33000 hingga 41999' } 0,077768
{'CustIncome' } {'42000 hingga 46999' } 0,24406
{'CustIncome' } {'47000 ke atas' } 0,43536
{'CustIncome' } {'<hilang>' } NaN
{'TmWBank' } {'Hingga 11' } -0,18221
{'TmWBank' } {'12 hingga 44' } -0,038279
{'TmWBank' } {'45 hingga 70' } 0,39569
{'TmWBank' } {'71 ke atas' } 0,95074
{'TmWBank' } {'<hilang>' } NaN
{'OtherCC' } {'Tidak' } -0,193
{'OtherCC' } {'Ya' } 0,15868
{'OtherCC' } {'<hilang>' } NaN
{'AMBBalance' } {'Hingga 558,87' } 0,3552
{'AMBBalance' } {'558,88 hingga 1597,43' } -0,026797
{'AMBBalance' } {'1597,44 dan seterusnya' } -0,21168
{'AMBBalance' } {'<hilang>' } NaN
```

Poin biasanya diskalakan dan juga sering dibulatkan. Untuk melakukannya, gunakan `formatpoints` fungsi ini. Misalnya, Anda dapat menetapkan target level poin yang sesuai dengan target level peluang dan juga menetapkan poin yang diperlukan untuk menggandakan peluang (PDO).

```
TitikTarget = 500;
TargetPeluang = 2;
PDO = 50; % Poin untuk menggandakan peluang

sc = formatpoints(sc, 'PointsOddsDanPDO', [TargetPoints TargetOdds PDO]);
p2 = titik_tampilan(sc);
menampilkan(p2)
```

Poin Bin Prediktor

```
{'CustAge' } {'Hingga 36' } 53.239
{'Usia Pelanggan' } {'37 hingga 39' } 59.796
{'Usia Pelanggan' } {'40 hingga 45' } 67.587
{'Usia Pelanggan' } {'46 hingga 57' } 80.796
{'CustAge' } {'58 ke atas' } 99.166
{'CustAge' } {'<hilang>' } NaN
{'ResStatus' } {'Penyewa' } 62.028
{'ResStatus' } {'Pemilik Rumah' } 73.445
{'ResStatus' } {'Lainnya' } 91.446
{'ResStatus' } {'<hilang>' } NaN
{'EmpStatus' } {'Tidak Diketahui' } 58.777
{'EmpStatus' } {'Dipekerjakan' } 86.976
{'EmpStatus' } {'<hilang>' } NaN
{'CustIncome' } {'Hingga 28999' } 31.497
{'CustIncome' } {'29000 hingga 32999' } 56,805
{'CustIncome' } {'33000 hingga 41999' } 69,896
{'CustIncome' } {'42000 hingga 46999' } 81,891
{'CustIncome' } {'47000 ke atas' } 95,69
{'CustIncome' } {'<hilang>' } NaN
{'TmWBank' } {'Hingga 11' } 51.142
{'TmWBank' } {'12 hingga 44' } 61.524
{'TmWBank' } {'45 hingga 70' } 92.829
{'TmWBank' } {'71 ke atas' } 132,87
{'TmWBank' } {'<hilang>' } NaN
{'OtherCC' } {'Tidak' } 50.364
{'OtherCC' } {'Ya' } 75.732
{'OtherCC' } {'<hilang>' } NaN
{'AMBalance' } {'Hingga 558,87' } 89.908
{'AMBalance' } {'558,88 hingga 1597,43' } 62,353
{'AMBalance' } {'1597.44 dan seterusnya' } 49.016
{'AMBalance' } {'<hilang>' } NaN
```

menjelaskan bahwa scatterplot membentuk pola eksponensial dimana kurva tidak pernah menyinggung sumbu x karena tidak ada nilainya yang 0. Untuk garis putus putus itu menunjukkan TargetPoints yang telah dibagi menjadi 3 kategori. Dari kurva ini dapat ditarik garis besar bahwa semakin besar scorenya maka peluang yang diprediksinya juga semakin besar. Atau keduanya memiliki hubungan yang searah. Tidak terdapat nilai yang outlier pada scatterplot atau nilai ekstrim yang berada jauh diluar kumpulan data. 5. Beri skor pada data.

Fungsi ini scoremenghitung skor untuk data pelatihan. dataInput opsional juga dapat diberikan ke score, misalnya, data validasi. Poin per prediktor untuk setiap pelanggan diberikan sebagai output opsional.

```
[Skor, Poin] = skor(sc);  
menampilkan(Skor(1:10))
```

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

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

```
CustAge ResStatus EmpStatus CustIncome TmWBank LainnyaCC ABalance
```

```
-----  
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
```

6. Hitung kemungkinan gagal bayar.

Untuk menghitung kemungkinan gagal bayar, gunakan `probdefault` fungsi:

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

Tentukan probabilitas menjadi "Baik" dan plot peluang yang diprediksi versus skor yang diformat. Analisis secara visual bahwa poin target dan peluang target cocok dan bahwa hubungan poin-ke-ganda-peluang (PDO) berlaku.

```

ProbBaik = 1-pd;
Peluang yang Diprediksi = ProbBaik./pd;

angka
scatter(Skor,Peluang yang Diprediksi)
judul( 'Prediksi Peluang vs. Skor' )
xlabel( 'Skor' )
ylabel( 'Peluang yang Diprediksi' )

tunggu

Batas x = xlim;
Batas y = ylim;

% Plot titik target dan peluang
([TargetPoints TargetPoints],[yLimits(1) TargetOdds], 'k:' )
plot([xLimits(1) TitikTarget],[PeluangTargetPeluangTarget], 'k:' )

% Titik target ditambah
plot PDO([TargetPoints+PDO TargetPoints+PDO],[yLimits(1) 2*TargetOdds], 'k:' )
plot([xLimits(1) TitikTarget+PDO],[2*PeluangTarget 2*PeluangTarget], 'k:' )

% Titik target dikurangi plot PDO
([TargetPoints-PDO TargetPoints-PDO],[yLimits(1) TargetOdds/2], 'k:' )
plot([xLimits(1) TitikTarget-PDO],[TargetOdds/2 TargetOdds/2], 'k:' )

bertahan

```

Gambar itu menjelaskan bahwa scatterplot membentuk pola eksponensial dimana kurva tidak pernah menyinggung sumbu x karena tidak ada nilainya yang 0. Untuk garis garis putus putus itu menunjukkan TargetPoints yang telah dibagi menjadi 3 kategori. Dari kurva ini dapat ditarik garis besar bahwa semakin besar scorenya maka peluang yang diprediksinya juga semakin besar. Atau keduanya memiliki hubungan yang searah. Tidak terdapat nilai yang outlier pada scatterplot atau nilai ekstrim yang berada jauh diluar kumpulan data.

