Laporan Tugas Besar Pembelajaran Mesin
Task Classification

Dzikri Al-Kautsar Sinatria A – 1301183498
Andika Elang Dirgantara – 1301184153

A. Formulasi Masalah

Diberikan sebuah dataset pembelian kendaraan oleh pelanggan. Permasalahan yang harus diselesaikan adalah bagaimana agar data tersebut dapat digunakan untuk memprediksi apakah pelanggan tertarik untuk membeli kendaraan baru atau tidak berdasarkan data pelanggan di dealer. Namun pada tugas besar tahap kedua kali ini permasalahannya adalah bagaimana dataset tersebut dapat dilakukan klasifikasi secara supervised.

B. Eksperimen Eksplorasi dan Persiapan Data

1. Missing Value

Dalam tahapan persiapan data (*pre-processing data*) yang pertama kali kita lakukan adalah mengisi setiap value yang bernilai kosong (*missing value*) pada *Training dataset* dan *Testing dataset*. Dalam tahap ini, setelah kita mencari di kolom mana saja letak missing value nya, baru kita lakukan imputasi terhadap value yang kosong tersebut berdasarkan dari tipe data kolom tersebut. Apabila tipe data dari kolom tersebut adalah *categorical*, maka kita isi *missing value* tersebut dengan modus dari keseluruhan data yang ada pada kolom tersebut. Dimana dalam kasus ini yang termasuk data *categorical* adalah kolom : ‘Jenis_Kelamin’, ‘SIM’, ‘Kode_Daerah’, ‘Sudah_Asuransi’, ‘Umur_Kendaraan’, ‘Kendaraan_Rusak’, dan ‘Kanal_Penjualan’. Apabila tipe data dari kolom tersebut adalah *numerical*, maka kita isi *missing value* tersebut dengan mean atau median dari keseluruhan data yang ada pada kolom tersebut. Dimana dalam kasus ini yang termasuk data *numerical* adalah kolom : ‘Umur’, ‘Premi, dan ‘Lama_Berlangganan’.

Berikut merupakan *Training* dan *Testing dataset* sebelum dilakukan imputasi *missing value* :

Data Train

id	Jenis_Kelamin	Umur	SIM	Kode_Daerah	Sudah_Asuransi	Umur_Kendaraan	Kendaraan_Rusak	Premi	Kanal_Penjualan	Lama_Berlangganan	Tertarik
1	Wanita	30.0	1.0	33.0	1.0	< 1 Tahun	Tidak	28029.0	152.0	97.0	0
2	Pria	48.0	1.0	39.0	0.0	> 2 Tahun	Pernah	25800.0	29.0	158.0	0
3	NaN	21.0	1.0	46.0	1.0	< 1 Tahun	Tidak	32733.0	160.0	119.0	0
4	Wanita	58.0	1.0	48.0	0.0	1-2 Tahun	Tidak	2630.0	124.0	63.0	0
5	Pria	50.0	1.0	35.0	0.0	> 2 Tahun	NaN	34857.0	88.0	194.0	0
...	...	...	...	...	...	...	...	...	...	...	...
285827	Wanita	23.0	1.0	4.0	1.0	< 1 Tahun	Tidak	25988.0	152.0	217.0	0
285828	Wanita	21.0	1.0	46.0	1.0	< 1 Tahun	Tidak	44686.0	152.0	50.0	0
285829	Wanita	23.0	1.0	50.0	1.0	< 1 Tahun	Tidak	49751.0	152.0	226.0	0
285830	Pria	68.0	1.0	7.0	1.0	1-2 Tahun	Tidak	30503.0	124.0	270.0	0
285831	Pria	45.0	1.0	28.0	0.0	1-2 Tahun	Pernah	36480.0	26.0	44.0	0

Data Test

	Jenis_Kelamin	Umur	SIM	Kode_Daerah	Sudah_Asuransi	Umur_Kendaraan	Kendaraan_Rusak	Premi	Kanal_Penjualan	Lama_Berlangganan	Tertarik
0	Wanita	49	1	8	0	1-2 Tahun	Pernah	46963	26	145	0
1	Pria	22	1	47	1	< 1 Tahun	Tidak	39624	152	241	0
2	Pria	24	1	28	1	< 1 Tahun	Tidak	110479	152	62	0
3	Pria	46	1	8	1	1-2 Tahun	Tidak	36266	124	34	0
4	Pria	35	1	23	0	1-2 Tahun	Pernah	26963	152	229	0
...	...	...	...	...	...	...	...	...	...	...	...
47634	Pria	61	1	46	0	> 2 Tahun	Pernah	31039	124	67	0
47635	Pria	41	1	15	0	1-2 Tahun	Pernah	2630	157	232	0
47636	Pria	24	1	29	1	< 1 Tahun	Tidak	33101	152	211	0
47637	Pria	59	1	30	0	1-2 Tahun	Pernah	37788	26	239	1
47638	Pria	52	1	31	0	1-2 Tahun	Tidak	2630	124	170	0

Berikut merupakan jumlah *missing value* setiap kolom pada *Training dataset* dan *Testing dataset*:

Data Train

```
Premi                14569
Jenis_Kelamin        14440
SIM                  14404
Kode_Daerah          14306
Kanal_Penjualan      14299
Umur_Kendaraan       14275
Sudah_Asuransi       14229
Umur                 14214
Kendaraan_Rusak      14188
Lama_Berlangganan    13992
Tertarik                 0
id                       0
dtype: int64
```

Data Test

```
Tertarik             0
Lama_Berlangganan    0
Kanal_Penjualan      0
Premi                0
Kendaraan_Rusak      0
Umur_Kendaraan       0
Sudah_Asuransi       0
Kode_Daerah          0
SIM                  0
Umur                 0
Jenis_Kelamin        0
dtype: int64
```

Dapat kita lihat bahwa pada *Testing dataset* tidak terdapat missing value, sehingga yang harus dilakukan imputasi *missing valu*e hanya pada *Training dataset* saja. Berikut merupakan *Training dataset* setelah dilakukan imputasi *missing value* :

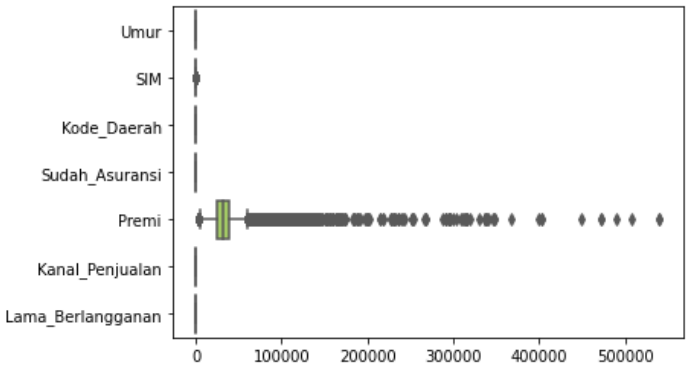
Data Train

	id	Jenis_Kelamin	Umur	SIM	Kode_Daerah	Sudah_Asuransi	Umur_Kendaraan	Kendaraan_Rusak	Premi	Kanal_Penjualan	Lama_Berlangganan	Tertarik
1		Wanita	30.0	1.0	33.0	1.0	< 1 Tahun	Tidak	28029.0	152.0	97.0	0
2		Pria	48.0	1.0	39.0	0.0	> 2 Tahun	Pernah	25800.0	29.0	158.0	0
3		Pria	21.0	1.0	46.0	1.0	< 1 Tahun	Tidak	32733.0	160.0	119.0	0
4		Wanita	58.0	1.0	48.0	0.0	1-2 Tahun	Tidak	2630.0	124.0	63.0	0
5		Pria	50.0	1.0	35.0	0.0	> 2 Tahun	Pernah	34857.0	88.0	194.0	0
...		...	...	...	...	...	...	...	...	...	...	...
285827		Wanita	23.0	1.0	4.0	1.0	< 1 Tahun	Tidak	25988.0	152.0	217.0	0
285828		Wanita	21.0	1.0	46.0	1.0	< 1 Tahun	Tidak	44686.0	152.0	50.0	0
285829		Wanita	23.0	1.0	50.0	1.0	< 1 Tahun	Tidak	49751.0	152.0	226.0	0
285830		Pria	68.0	1.0	7.0	1.0	1-2 Tahun	Tidak	30503.0	124.0	270.0	0
285831		Pria	45.0	1.0	28.0	0.0	1-2 Tahun	Pernah	36480.0	26.0	44.0	0

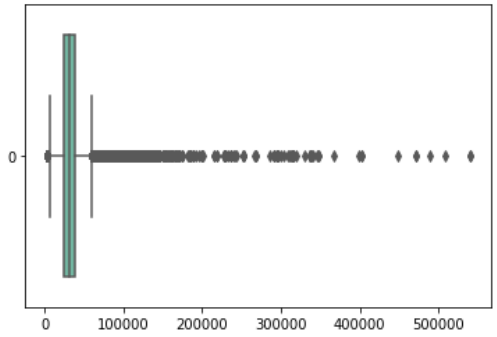
2. Outlier

Setelah kita mengisi semua *missing value*, dalam tahap ini kita mencari kolom mana saja yang memiliki data pencilan (*outlier*) dan kemudian dapat dimodelkan dengan menggunakan *boxplot* agar dapat lebih terlihat dimana saja letak *outlier* tersebut, lalu setelah itu kita lakukan imputasi terhadap kolom yang memiliki *outlier* dengan menggunakan hasil *mean* dari isi data kolom tersebut. Berdasarkan model *boxplot*, dapat dilihat bahwa dalam kasus ini, yang memiliki *outlier* adalah kolom 'Premi'.

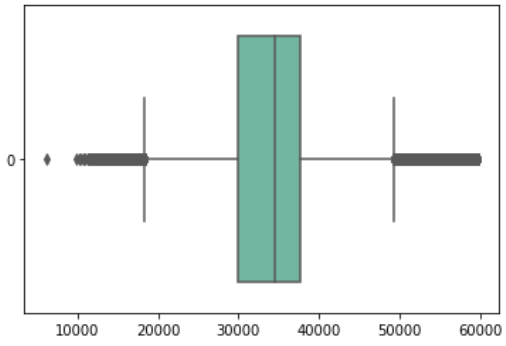
Data Train



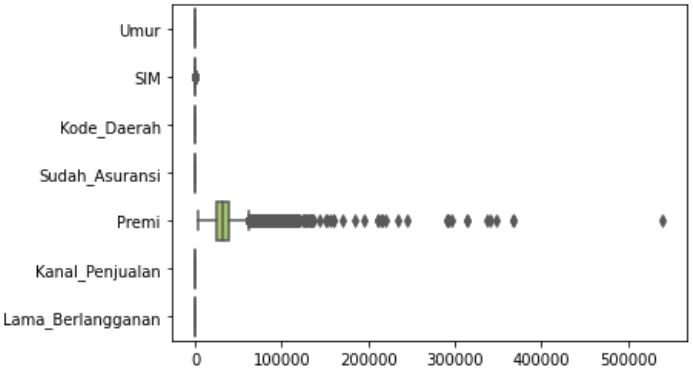
Boxplot sebelum imputasi kolom Premi



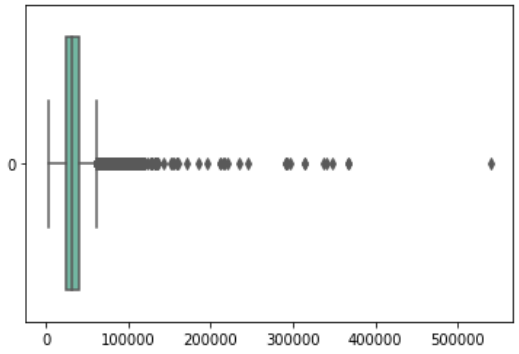
Setelah imputasi outlier pada kolom Premi



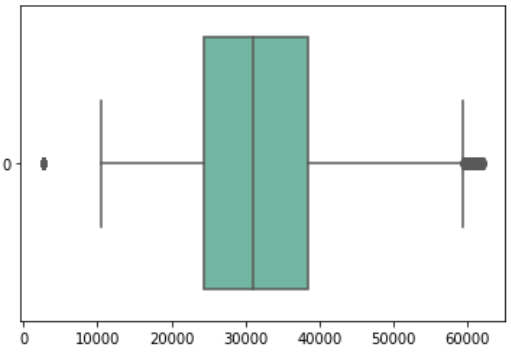
Data Test



Boxplot sebelum imputasi kolom Premi



Setelah imputasi outlier pada kolom Premi



3. Feature Engineering

Setelah kita telah mengatasi *outlier*. Selanjutnya dalam tahap Feature Engineering ini kita melakukan 3 tahap pada data, yaitu Categorical Encoding, Scaling, dan Feature Selection. Tahapan-tahapan tersebut penting untuk dilakukan agar data dapat lebih mudah untuk dimodelkan, dimana nantinya model tersebut dapat digunakan untuk keperluan selanjutnya.

a. Categorical Encoding

Dalam tahap ini kita melakukan konversi untuk setiap data yang berbentuk *categorical* menjadi *numerical*. Dimana dalam kasus ini yang merupakan data *categorical* adalah data pada kolom Jenis_Kelamin, Umur_Kendaraan, dan Kendaraan_Rusak. Hasilnya, setelah melakukan Categorical Encoding ini, kolom Jenis_Kelamin yang awalnya isinya adalah ‘Pria’ dan ‘Wanita’ berubah menjadi data numeric 0 untuk ‘Pria’ dan 1 untuk ‘Wanita’. Lalu untuk kolom Umur_Kendaraan yang awalnya isinya adalah ‘< 1 Tahun’, ‘1-2 Tahun’, dan ‘> 2 Tahun’ berubah menjadi data numeric 0 untuk ‘< 1 Tahun’, 1 untuk ‘1-2 Tahun’, dan 2 untuk ‘> 2 Tahun’. Dan yang terakhir pada kolom Kendaraan_Rusak yang awalnya isinya adalah ‘Pernah’ dan ‘Tidak’ berubah menjadi data numeric 0 untuk ‘Pernah’ dan 1 untuk ‘Tidak’.

Berikut adalah data sebelum Categorical Encoding

Data Train

	id	Jenis_Kelamin	Umur	SIM	Kode_Daerah	Sudah_Asuransi	Umur_Kendaraan	Kendaraan_Rusak	Premi	Kanal_Penjualan	Lama_Berlangganan	Tertarik
0	1	Wanita	30.0	1.0	33.0	1.0	< 1 Tahun	Tidak	28029.000000	152.0	97.0	0
1	2	Pria	48.0	1.0	39.0	0.0	> 2 Tahun	Pernah	25800.000000	29.0	158.0	0
2	3	Pria	21.0	1.0	46.0	1.0	< 1 Tahun	Tidak	32733.000000	160.0	119.0	0
3	4	Wanita	58.0	1.0	48.0	0.0	1-2 Tahun	Tidak	34526.832758	124.0	63.0	0
4	5	Pria	50.0	1.0	35.0	0.0	> 2 Tahun	Pernah	34857.000000	88.0	194.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...
285826	285827	Wanita	23.0	1.0	4.0	1.0	< 1 Tahun	Tidak	25988.000000	152.0	217.0	0
285827	285828	Wanita	21.0	1.0	46.0	1.0	< 1 Tahun	Tidak	44686.000000	152.0	50.0	0
285828	285829	Wanita	23.0	1.0	50.0	1.0	< 1 Tahun	Tidak	49751.000000	152.0	226.0	0
285829	285830	Pria	68.0	1.0	7.0	1.0	1-2 Tahun	Tidak	30503.000000	124.0	270.0	0
285830	285831	Pria	45.0	1.0	28.0	0.0	1-2 Tahun	Pernah	36480.000000	26.0	44.0	0

Data Test

	Jenis_Kelamin	Umur	SIM	Kode_Daerah	Sudah_Asuransi	Umur_Kendaraan	Kendaraan_Rusak	Premi	Kanal_Penjualan	Lama_Berlangganan	Tertarik
0	Wanita	49	1	8	0	1-2 Tahun	Pernah	46963.000000	26	145	0
1	Pria	22	1	47	1	< 1 Tahun	Tidak	39624.000000	152	241	0
2	Pria	24	1	28	1	< 1 Tahun	Tidak	29319.082061	152	62	0
3	Pria	46	1	8	1	1-2 Tahun	Tidak	36266.000000	124	34	0
4	Pria	35	1	23	0	1-2 Tahun	Pernah	26963.000000	152	229	0
...	...	...	...	...	...	...	...	...	...	...	...
47634	Pria	61	1	46	0	> 2 Tahun	Pernah	31039.000000	124	67	0
47635	Pria	41	1	15	0	1-2 Tahun	Pernah	2630.000000	157	232	0
47636	Pria	24	1	29	1	< 1 Tahun	Tidak	33101.000000	152	211	0
47637	Pria	59	1	30	0	1-2 Tahun	Pernah	37788.000000	26	239	1
47638	Pria	52	1	31	0	1-2 Tahun	Tidak	2630.000000	124	170	0

dan berikut merupakan data setelah dilakukan Categorical Encoding:

Data Train

	id	Jenis_Kelamin	Umur	SIM	Kode_Daerah	Sudah_Asuransi	Umur_Kendaraan	Kendaraan_Rusak	Premi	Kanal_Penjualan	Lama_Berlangganan	Tertarik
0	1	1	30.0	1.0	33.0	1.0	1	1	28029.000000	152.0	97.0	0
1	2	0	48.0	1.0	39.0	0.0	2	0	25800.000000	29.0	158.0	0
2	3	0	21.0	1.0	46.0	1.0	1	1	32733.000000	160.0	119.0	0
3	4	1	58.0	1.0	48.0	0.0	0	1	34526.832758	124.0	63.0	0
4	5	0	50.0	1.0	35.0	0.0	2	0	34857.000000	88.0	194.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...
285826	285827	1	23.0	1.0	4.0	1.0	1	1	25988.000000	152.0	217.0	0
285827	285828	1	21.0	1.0	46.0	1.0	1	1	44686.000000	152.0	50.0	0
285828	285829	1	23.0	1.0	50.0	1.0	1	1	49751.000000	152.0	226.0	0
285829	285830	0	68.0	1.0	7.0	1.0	0	1	30503.000000	124.0	270.0	0
285830	285831	0	45.0	1.0	28.0	0.0	0	0	36480.000000	26.0	44.0	0

Data Test

	id	Jenis_Kelamin	Umur	SIM	Kode_Daerah	Sudah_Asuransi	Umur_Kendaraan	Kendaraan_Rusak	Premi	Kanal_Penjualan	Lama_Berlangganan	Tertarik
0	1	1	30.0	1.0	33.0	1.0	1	1	28029.000000	152.0	97.0	0
1	2	0	48.0	1.0	39.0	0.0	2	0	25800.000000	29.0	158.0	0
2	3	0	21.0	1.0	46.0	1.0	1	1	32733.000000	160.0	119.0	0
3	4	1	58.0	1.0	48.0	0.0	0	1	34526.832758	124.0	63.0	0
4	5	0	50.0	1.0	35.0	0.0	2	0	34857.000000	88.0	194.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...
285826	285827	1	23.0	1.0	4.0	1.0	1	1	25988.000000	152.0	217.0	0
285827	285828	1	21.0	1.0	46.0	1.0	1	1	44686.000000	152.0	50.0	0
285828	285829	1	23.0	1.0	50.0	1.0	1	1	49751.000000	152.0	226.0	0
285829	285830	0	68.0	1.0	7.0	1.0	0	1	30503.000000	124.0	270.0	0
285830	285831	0	45.0	1.0	28.0	0.0	0	0	36480.000000	26.0	44.0	0

b. Scaling

Dalam tahap ini kita melakukan normalisasi pada data dengan menggunakan metode MinMax Scaling. Dimana tujuannya adalah untuk menyamakan rentang dari setiap data menjadi terkecil 0 sampai yang terbesar 1. Dapat dilihat setelah melakukan normalisasi MinMax pada kasus ini, maka rentang dari setiap data untuk terkecil (*min*) telah berubah menjadi 0 dan data terbesar (*max*) menjadi 1.

Data Train

	id	Jenis_Kelamin	Umur	SIM	Kode_Daerah	Sudah_Asuransi	Umur_Kendaraan	Kendaraan_Rusak	Premi	Kanal_Penjualan	Lama_Berlangganan	Tertarik
count	285831.000000	285831.000000	285831.000000	285831.000000	285831.000000	285831.000000	285831.000000	285831.000000	285831.000000	285831.000000	285831.000000	285831.000000
mean	0.500000	0.436317	0.289913	0.997957	0.509331	0.435939	0.245271	0.470628	0.530181	0.697664	0.499261	0.122471
std	0.288677	0.495929	0.232794	0.045155	0.248490	0.495880	0.287140	0.499137	0.142626	0.330514	0.282425	0.327830
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.250000	0.000000	0.076923	1.000000	0.288462	0.000000	0.000000	0.000000	0.444760	0.333333	0.259516	0.000000
50%	0.500000	0.000000	0.276923	1.000000	0.538462	0.000000	0.000000	0.000000	0.530181	0.932099	0.499261	0.000000
75%	0.750000	1.000000	0.446154	1.000000	0.673077	1.000000	0.500000	1.000000	0.588650	0.932099	0.737024	0.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

	id	Jenis_Kelamin	Umur	SIM	Kode_Daerah	Sudah_Asuransi	Umur_Kendaraan	Kendaraan_Rusak	Premi	Kanal_Penjualan	Lama_Berlangganan	Tertarik
0	0.000000	1.0	0.153846	1.0	0.634615	1.0	0.5	1.0	0.409000	0.932099	0.301038	0.0
1	0.000003	0.0	0.430769	1.0	0.750000	0.0	1.0	0.0	0.367431	0.172840	0.512111	0.0
2	0.000007	0.0	0.015385	1.0	0.884615	1.0	0.5	1.0	0.496727	0.981481	0.377163	0.0
3	0.000010	1.0	0.584615	1.0	0.923077	0.0	0.0	1.0	0.530181	0.759259	0.183391	0.0
4	0.000014	0.0	0.461538	1.0	0.673077	0.0	1.0	0.0	0.536338	0.537037	0.636678	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...
285826	0.999986	1.0	0.046154	1.0	0.076923	1.0	0.5	1.0	0.370937	0.932099	0.716263	0.0
285827	0.999990	1.0	0.015385	1.0	0.884615	1.0	0.5	1.0	0.719643	0.932099	0.138408	0.0
285828	0.999993	1.0	0.046154	1.0	0.961538	1.0	0.5	1.0	0.814103	0.932099	0.747405	0.0
285829	0.999997	0.0	0.738462	1.0	0.134615	1.0	0.0	1.0	0.455139	0.759259	0.899654	0.0
285830	1.000000	0.0	0.384615	1.0	0.538462	0.0	0.0	0.0	0.566606	0.154321	0.117647	0.0

Data Test

	Jenis_Kelamin	Umur	SIM	Kode_Daerah	Sudah_Asuransi	Umur_Kendaraan	Kendaraan_Rusak	Premi	Kanal_Penjualan	Lama_Berlangganan	Tertarik
count	47639.000000	47639.000000	47639.000000	47639.000000	47639.000000	47639.000000	47639.000000	47639.000000	47639.000000	47639.000000	47639.000000
mean	0.456958	0.289469	0.997922	0.506375	0.457608	0.258759	0.495350	0.448850	0.686300	0.499819	0.123029
std	0.498149	0.239213	0.045540	0.254103	0.498205	0.288473	0.499984	0.245132	0.334338	0.289419	0.328474
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.076923	1.000000	0.288462	0.000000	0.000000	0.000000	0.366047	0.172840	0.249135	0.000000
50%	0.000000	0.246154	1.000000	0.538462	0.000000	0.000000	0.000000	0.478011	0.827160	0.501730	0.000000
75%	1.000000	0.446154	1.000000	0.673077	1.000000	0.500000	1.000000	0.601529	0.932099	0.750865	0.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

	Jenis_Kelamin	Umur	SIM	Kode_Daerah	Sudah_Asuransi	Umur_Kendaraan	Kendaraan_Rusak	Premi	Kanal_Penjualan	Lama_Berlangganan	Tertarik
0	1.0	0.446154	1.0	0.153846	0.0	0.0	0.0	0.745581	0.154321	0.467128	0.0
1	0.0	0.030769	1.0	0.903846	1.0	0.5	1.0	0.622156	0.932099	0.799308	0.0
2	0.0	0.061538	1.0	0.538462	1.0	0.5	1.0	0.448850	0.932099	0.179931	0.0
3	0.0	0.400000	1.0	0.153846	1.0	0.0	1.0	0.565682	0.759259	0.083045	0.0
4	0.0	0.230769	1.0	0.442308	0.0	0.0	0.0	0.409226	0.932099	0.757785	0.0
...	...	...	...	...	...	...	...	...	...	...	...
47634	0.0	0.630769	1.0	0.884615	0.0	1.0	0.0	0.477775	0.759259	0.197232	0.0
47635	0.0	0.323077	1.0	0.288462	0.0	0.0	0.0	0.000000	0.962963	0.768166	0.0
47636	0.0	0.061538	1.0	0.557692	1.0	0.5	1.0	0.512454	0.932099	0.695502	0.0
47637	0.0	0.600000	1.0	0.576923	0.0	0.0	0.0	0.591278	0.154321	0.792388	1.0
47638	0.0	0.492308	1.0	0.596154	0.0	0.0	1.0	0.000000	0.759259	0.553633	0.0

c. Feature Selection

Dalam tahap ini kita hanya melakukan drop pada kolom SIM. Alasannya adalah karena data pada kolom tersebut yang terbilang aneh dan kami anggap tidak berguna untuk kedepannya.

Data Train

	id	Jenis_Kelamin	Umur	Kode_Daerah	Sudah_Asuransi	Umur_Kendaraan	Kendaraan_Rusak	Premi	Kanal_Penjualan	Lama_Berlangganan	Tertarik
0	0.000000	1.0	0.153846	0.634615	1.0	0.5	1.0	0.409000	0.932099	0.301038	0.0
1	0.000003	0.0	0.430769	0.750000	0.0	1.0	0.0	0.367431	0.172840	0.512111	0.0
2	0.000007	0.0	0.015385	0.884615	1.0	0.5	1.0	0.496727	0.981481	0.377163	0.0
3	0.000010	1.0	0.584615	0.923077	0.0	0.0	1.0	0.530181	0.759259	0.183391	0.0
4	0.000014	0.0	0.461538	0.673077	0.0	1.0	0.0	0.536338	0.537037	0.636678	0.0
...	...	...	...	...	...	...	...	...	...	...	...
285826	0.999986	1.0	0.046154	0.076923	1.0	0.5	1.0	0.370937	0.932099	0.716263	0.0
285827	0.999990	1.0	0.015385	0.884615	1.0	0.5	1.0	0.719643	0.932099	0.138408	0.0
285828	0.999993	1.0	0.046154	0.961538	1.0	0.5	1.0	0.814103	0.932099	0.747405	0.0
285829	0.999997	0.0	0.738462	0.134615	1.0	0.0	1.0	0.455139	0.759259	0.899654	0.0
285830	1.000000	0.0	0.384615	0.538462	0.0	0.0	0.0	0.566606	0.154321	0.117647	0.0

Data Test

	Jenis_Kelamin	Umur	Kode_Daerah	Sudah_Asuransi	Umur_Kendaraan	Kendaraan_Rusak	Premi	Kanal_Penjualan	Lama_Berlangganan	Tertarik
0	1.0	0.446154	0.153846	0.0	0.0	0.0	0.745581	0.154321	0.467128	0.0
1	0.0	0.030769	0.903846	1.0	0.5	1.0	0.622156	0.932099	0.799308	0.0
2	0.0	0.061538	0.538462	1.0	0.5	1.0	0.448850	0.932099	0.179931	0.0
3	0.0	0.400000	0.153846	1.0	0.0	1.0	0.565682	0.759259	0.083045	0.0
4	0.0	0.230769	0.442308	0.0	0.0	0.0	0.409226	0.932099	0.757785	0.0
...	...	...	...	...	...	...	...	...	...	...
47634	0.0	0.630769	0.884615	0.0	1.0	0.0	0.477775	0.759259	0.197232	0.0
47635	0.0	0.323077	0.288462	0.0	0.0	0.0	0.000000	0.962963	0.768166	0.0
47636	0.0	0.061538	0.557692	1.0	0.5	1.0	0.512454	0.932099	0.695502	0.0
47637	0.0	0.600000	0.576923	0.0	0.0	0.0	0.591278	0.154321	0.792388	1.0
47638	0.0	0.492308	0.596154	0.0	0.0	1.0	0.000000	0.759259	0.553633	0.0

C. Pemodelan

Pada tahap ini kita menggunakan model *Decision Tree Classifier* karena menurut kami termasuk salah satu algoritma klasifikasi yang paling mudah dan populer untuk dipahami dan diinterpretasikan. *Decision Tree Classifier* itu sendiri adalah teknik supervised learning yang membangun representasi aturan klasifikasi berstruktur sekuensial hirarki dengan cara mempartisi himpunan data latih secara rekursif (Suyanto, 2018).

D. Eksperimen dan Evaluasi

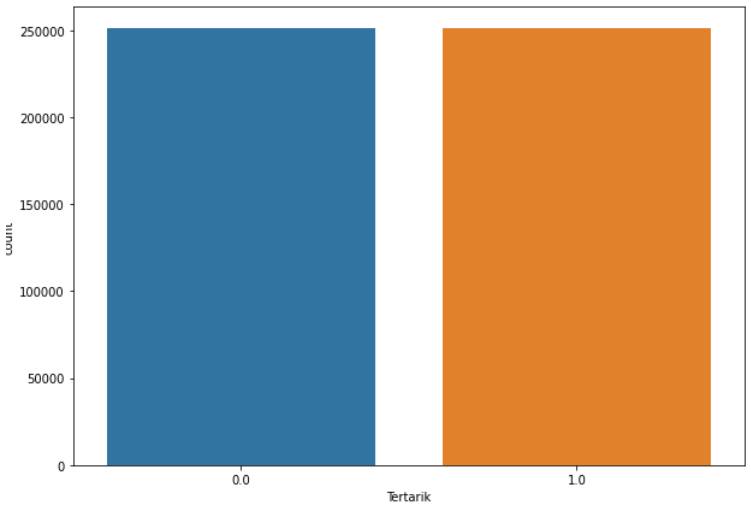
Dalam proses ini, yang dilakukan adalah klasifikasi terhadap model yang telah dibentuk sebelumnya. Kita menggunakan *Confusion Matrix* untuk mengukur kinerja dari model yang digunakan. Pada eksperimen yang kami lakukan, kami akan mencari nilai akurasi terbaik dari dataset.

Pertama kali kami akan cek terlebih dahulu nilai akurasi dari *Training dataset* yang ada. Setelah kami cek, ternyata nilai akurasi yang diperoleh sebesar 0.8214863600185424. Sebenarnya nilai tersebut sudah dapat dikatakan lumayan baik, tetapi kami masih bisa mendapatkan nilai akurasi yang lebih baik lagi dari nilai tersebut karena dapat dilihat pada distribusi jumlah label yang tidak rata sehingga berpengaruh pada akurasi. Cara yang kami

lakukan untuk mengatasi hal tersebut dan untuk mendapatkan akurasi yang lebih baik lagi adalah dengan menggunakan teknik SMOT(Synthetic Minority Oversampling Technique).

Dengan teknik SMOT(Synthetic Minority Oversampling Technique) ini kita akan menyamaratakan distribusi label yang tidak seimbang. Pada tahap ini kita melakukan duplikat pada setiap row data yang memiliki label *minority* pada Training dataset, sehingga jumlah dari masing-masing label akan sama. Berikut merupakan Training dataset setelah dilakukan teknik SMOT, dan dapat kita lihat bahwa sekarang jumlah data menjadi bertambah dan distribusi jumlah label sudah sama rata.

	id	Jenis_Kelamin	Umur	Kode_Daerah	Sudah_Asuransi	Umur_Kendaraan	Kendaraan_Rusak	Premi	Kanal_Penjualan	Lama_Berlangganan	Tertarik
0	0.000000	1.0	0.153846	0.634615	1.0	0.5	1.0	0.409000	0.932099	0.301038	0.0
1	0.000003	0.0	0.430769	0.750000	0.0	1.0	0.0	0.367431	0.172840	0.512111	0.0
2	0.000007	0.0	0.015385	0.884615	1.0	0.5	1.0	0.496727	0.981481	0.377163	0.0
3	0.000010	1.0	0.584615	0.923077	0.0	0.0	1.0	0.530181	0.759259	0.183391	0.0
4	0.000014	0.0	0.461538	0.673077	0.0	1.0	0.0	0.536338	0.537037	0.636678	0.0
...	...	...	...	...	...	...	...	...	...	...	...
501645	0.805564	1.0	0.079679	0.043464	0.0	0.5	0.0	0.459725	0.944944	0.569975	1.0
501646	0.506833	0.0	0.469187	0.538462	0.0	0.0	0.0	0.558378	0.749983	0.255117	1.0
501647	0.088553	1.0	0.048139	0.255893	0.0	0.5	0.0	0.562782	0.959976	0.599007	1.0
501648	0.570744	0.0	0.305128	0.538462	0.0	1.0	0.0	0.492918	0.154321	0.237291	1.0
501649	0.293577	1.0	0.154747	0.882739	0.0	0.0	0.0	0.286633	0.932340	0.437134	1.0



Selain itu juga, setelah dilakukan teknik SMOT tersebut maka pada *Training dataset* diperoleh akurasi sebesar 0.8873567228147115 dan pada *Testing dataset* diperoleh akurasi sebesar 0.8851988151871466. Dari sini, maka dapat dibuktikan bahwa teknik SMOT dapat dilakukan untuk meningkatkan nilai akurasi pada *Training dataset* dari yang awalnya 0.8214863600185424 menjadi 0.8873567228147115.

E. Kesimpulan

Supervised Learning adalah salah satu metode klasifikasi. Terdapat banyak metode yang ada dalam klasifikasi Supervised Learning diantaranya adalah diantaranya Regresi Logistik, K-nearest Neighbor, Super Vector Machine, Naive Bayes, Decision Tree dan Random Forest.

Dimulai dari tahap pre-processing data yang dilakukan pada Training dataset dan Testing dataset dengan mengatasi missing value, mengatasi outlier, dan melakukan feature engineering seperti categorical encoding, scaling, dan seleksi fitur. Kemudian, setelah selesai melakukan pre-processing data, maka baru data tersebut dapat dimodelkan dan dapat digunakan untuk keperluan klasifikasi.

Dalam proses klasifikasi yang kami lakukan, kami menggunakan model Decision Tree Classifier karena kelebihannya pada implementasi yang paling populer dan lebih mudah dibandingkan metode yang lain. Kemudian, setelah itu kami melakukan eksperimen agar mendapatkan nilai akurasi tertinggi dari dataset yang ada. Pertama kali, kami mengecek akurasi dari *Training dataset* dan diperoleh nilai akurasi sebesar 0.8214863600185424. Nilai tersebut sebenarnya sudah dapat dikatakan cukup baik, namun dengan teknik SMOT(Synthetic Minority Oversampling Technique) yang kami lakukan, maka kami dapat memperoleh nilai akurasi yang lebih baik karena dataset yang ada memiliki distribusi label yang tidak sama dan jauh jaraknya. Sehingga oleh karena itu teknik SMOT terbukti dapat dilakukan untuk meningkatkan nilai akurasi pada *Training dataset* dari yang awalnya 0.8214863600185424 menjadi 0.8873567228147115.

Setelah dilakukan semua proses tersebut, kesimpulannya kami memperoleh nilai akurasi yang cukup tinggi pada kedua dataset. Pada *Training dataset* diperoleh akurasi sebesar 0.8873567228147115 dan pada *Testing dataset* diperoleh akurasi sebesar 0.8851988151871466. Sehingga dari hal tersebut dapat dikatakan bahwa proses yang telah kami lakukan telah menghasilkan kualitas data yang cukup baik.