JOINTS UGM DATA SCIENCE COMPETITION

PREDIKSI KELANJUTAN ASURANSI PELANGGAN J INSURANCE

ElsiX

Atmavidya Virananda

Johannes Joseph Billie Christian

M Sammy Ivan Kurniawan

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2021

DAFTAR ISI

BAB I
PENDAHULUAN

1.1 Latar Belakang

Kehidupan yang sekarang manusia jalani sudah jelas adalah kondisi yang memiliki banyak sekali ketidakpastian. Lingkungan yang beragam, latar belakang yang berbeda-beda, dan orientasi kehidupan yang sangat bervariasi membuat kalanya seorang manusia menerima musibah tidak dapat diprediksi lagi. Oleh karena itu, munculah suatu hal yang sekarang kita sebut dengan asuransi. Asuransi pada dasarnya adalah sebuah pertanggungan yang mana dengan sejumlah premi tertentu, suatu aspek dari kehidupan kita akan dijamin olehnya, seperti jiwa, aset, rumah, mobil, dan lain-lain. Apalagi, di era pandemi ini orang-orang mulai mengkhawatirkan kesehatan mereka, dan perusahaan asuransi akan berperan sangat besar di kondisi seperti ini, baik untuk tujuan yang mulai ataupun untuk mengejar potensi keuntungan yang ada.

Perusahaan "J Insurance" merupakan sebuah perusahaan asuransi yang sudah berdiri sejak lama. Perusahaan ini memberikan layanan berupa asuransi jiwa, asuransi kesehatan, dan asuransi kendaraan. Reputasinya sudah tidak diragukan lagi di negaranya sehingga para *top management* menginginkan untuk mempertahankan prestasi tersebut. Terdapat banyak cara untuk melakukan hal ini, seperti *marketing* yang lebih gencar, menetapkan strategi organisasi yang lebih sesuai, dan salah satu cara yang paling *tangible* dan dapat dikuantifikasi adalah dengan mempertahankan pelanggan lama.

Dengan perkembangan teknologi yang pesat, terdapat berbagai macam cara untuk melakukan prediksi terhadap kecenderungan perilaku seorang pelanggan, termasuk perilakunya untuk memperpanjang / mendaftar kembali asuransi. Untuk mendukung upaya "J insurance" dalam mempertahankan prestasi tersebut, maka dapat dilakukan *targeted marketing* yang berarti mengeluarkan usaha yang lebih untuk menggiring pelanggan yang cenderung ingin melanjutkan asuransi agar benar-benar melanjutkan asuransinya. Oleh karena itu, pemanfaatan ranah *data science* dapat digunakan untuk membangun sebuah model berbasis *machine learning* yang dapat melakukan prediksi kecenderungan seseorang akan melanjutkan asuransi atau tidak, berdasarkan beberapa karakteristik inti. Tentunya, penelaahan karakteristik yang dimiliki oleh "J Insurance" terhadap pelanggannya perlu ditinjau lebih lanjut kepentingan dan efeknya terhadap keberhasilan prediksi.

1.2 Tujuan & Manfaat

Berdasarkan latar belakang yang telah dipaparkan, terdapat beberapa tujuan beserta manfaatnya yang dapat dideskripsikan pada analisis dan prediksi *dataset* asuransi ini, yakni:

1. Menentukan cara pengolahan data terbaik sebagai *input* untuk model machine learning
2. Menentukan model terbaik untuk melakukan prediksi terhadap kemauan seseorang untuk melanjutkan asuransinya
3. Menentukan hasil nilai validasi yang didapatkan dari model terbaik

BAB II
EXPLORATORY DATA ANALYSIS

Exploratory data analysis (EDA) dapat diartikan sebagai proses dimana dilakukan langkah-langkah kritis untuk menginvestigasi data untuk menemukan pola-pola tidak wajar, abnormalitas, dan informasi menarik. Dalam keberjalanannya, EDA juga mengutilisasikan ilmu statistik khususnya dalam *hypothesis testing* untuk memeriksa asumsi, serta visualisasi yang beragam untuk mengidentifikasi hal-hal yang tidak mudah terlihat hanya dengan mengamati tabel data yang ada. Pertama-tama, dilakukan peninjauan terhadap deskripsi dari kedua dataset. Dua tabel pada Gambar 2 dan Gambar 3 dihasilkan dari *describe() method* yang mendeskripsikan karakteristik singkat dari dataset.

```
Index(['id', 'Gender', 'Umur', 'Izin_Mengemudi', 'Kode_Wilayah',
       'Tanggal_Asuransi', 'Tahun_Kendaraan', 'Biaya', 'Sourcing_Channel',
       'Hari_Diasuransikan', 'Target'],
      dtype='object')
```

Gambar 1 Daftar kolom pada dataset

	id	Umur	Izin_Mengemudi	Kode_Wilayah	Biaya	Sourcing_Channel	Hari_Diasuransikan	Target
count	382154.000000	285896.000000	305507.000000	298080.000000	255617.000000	298509.000000	306488.000000	382154.000000
mean	234392.953477	38.916592	0.998815	26.406032	31183.756781	110.872007	154.168995	0.163811
std	139527.487326	16.706800	0.034402	13.163179	18392.305587	57.862621	83.720850	0.370104
min	1.000000	20.000000	0.000000	0.000000	2630.000000	1.000000	10.000000	0.000000
25%	115006.250000	24.000000	1.000000	15.000000	24426.000000	26.000000	81.000000	0.000000
50%	230461.500000	33.000000	1.000000	28.000000	31887.000000	152.000000	154.000000	0.000000
75%	345434.750000	52.000000	1.000000	35.000000	40007.000000	152.000000	227.000000	0.000000
max	508145.000000	85.000000	1.000000	52.000000	540165.000000	163.000000	299.000000	1.000000

Gambar 2 *Train dataset description*

	id	Umur	Izin_Mengemudi	Kode_Wilayah	Biaya	Sourcing_Channel	Hari_Diasuransikan
count	78273.000000	62692.000000	62560.000000	58720.000000	57426.000000	65034.000000	61053.000000
mean	233232.431656	41.207092	0.998449	26.415497	31888.958329	108.169696	154.825103
std	138985.576956	16.854298	0.039346	13.099933	18185.146144	57.092660	83.467490
min	5.000000	20.000000	0.000000	0.000000	2630.000000	1.000000	10.000000
25%	115187.000000	25.000000	1.000000	15.000000	24948.000000	26.000000	83.000000
50%	228150.000000	40.000000	1.000000	28.000000	32412.500000	151.000000	155.000000
75%	343485.000000	54.000000	1.000000	35.000000	40913.750000	152.000000	227.000000
max	508136.000000	85.000000	1.000000	52.000000	489663.000000	163.000000	299.000000

Gambar 3 *Test dataset description*

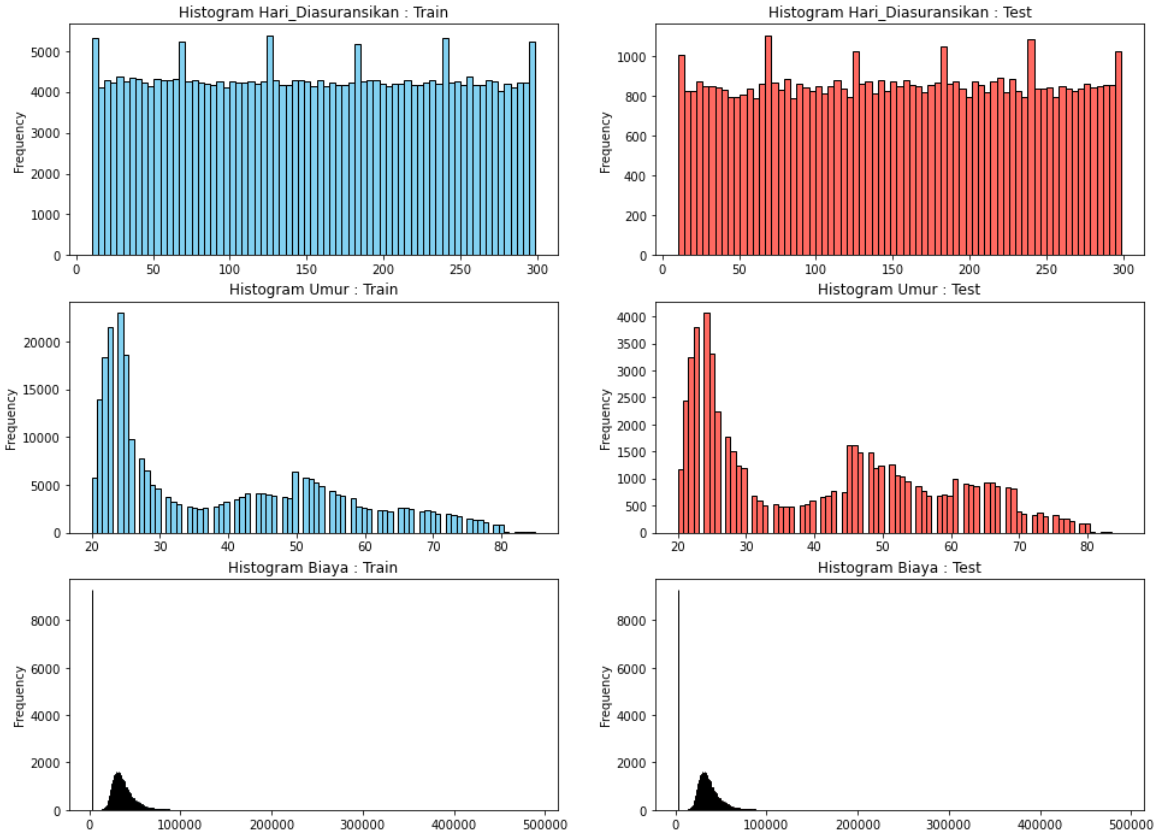
Dari kedua tabel diatas, diperoleh beberapa poin berikut:

- Pada dataset *train*, terdapat 11 kolom dengan 3 kolom (Hari_Diasuransikan, Umur, Biaya) bersifat numerik dan 5 kolom (Gender, Izin_Mengemudi, Tahun_Kendaraan, Kode_Wilayah, Sourcing_Channel) bersifat kategorikal. Kolom Id, Target, dan Tanggal_Diasuransikan merupakan 3 kolom lainnya
- Orang termuda dan tertua yang saat ini terdaftar dalam asuransi ini adalah 20 dan 85 tahun
- Nilai maksimum dari biaya sangat jauh dari rata-ratanya, mengindikasikan adanya *outlier*

Setelah diperoleh beberapa informasi berikut, selanjutnya adalah untuk melakukan proses EDA spesifik, yang dipecah berdasarkan poin-poin pembahasan.

2.1 Distribusi Data Numerik

Distribusi data numerik merupakan salah satu cara terbaik untuk mengetahui karakteristik dari data numerik. Karakteristik yang dimaksud adalah pola distribusi data, *uniformity* dari data, *outlier postulation*, dan tentunya, dapat digunakan untuk membandingkan proporsi data antara dataset *train* dan *test*. Maka dari itu, dilakukan *histogram plotting* untuk kedua dataset di bawah ini, khususnya untuk data yang bersifat numerik.



Gambar 4 Histogram berbagai data numerik

Dengan pengamatan visual murni terhadap distribusi nilai-nilai fitur numerik yang telah dipetakan di atas, dapat diperoleh beberapa ***insights*** sebagai berikut:

- Fitur numerik “Hari_Diasuransikan” untuk kedua dataset menyerupai distribusi *uniform* dan memiliki kesamaan karakteristik walaupun tidak sama secara persis. Dapat disimpulkan bahwa tidak ada kecenderungan tertentu untuk lama waktu pembelian asuransi bagi masyarakat yang terekam di dalam dataset terkait
- Fitur numerik “Umur” untuk kedua dataset memiliki karakteristik distribusi yang serupa walaupun distribusi frekuensi datanya tidak memiliki titik-titik yang sama persis
- Fitur numerik “Biaya” memiliki distribusi yang cukup serupa secara visual. Ditemukan pula bahwa terdapat banyak sekali frekuensi data yang nilainya mendekati 0. Setelah dilakukan penyelidikan, ditemukan bahwa nilai biaya sebesar 2630.0 berjumlah 44088 *records* pada dataset *train* dan berjumlah 9292 *records* pada dataset *test*

- Dapat diambil kesimpulan bahwa dengan asumsi dataset ini merupakan dataset historis yang lengkap, maka kebanyakan orang yang memiliki asuransi pada dataset *train* dan *test* adalah mereka yang berumur 21 hingga 27 tahun. Di luar cakupan umur tersebut, jumlahnya lebih sedikit

Tabel 1 Perbandingan jumlah pelanggan terhadap umurnya

Train		***Test***	
Umur		Umur	
24.0	23033	24.0	4072
23.0	21560	23.0	3800
25.0	18636	25.0	3303
22.0	18367	22.0	3236
21.0	13996	21.0	2449
26.0	9776	26.0	2242
27.0	7748	27.0	1779

- Dengan visualisasi yang berukuran sangat kecil dan tidak berwarna dengan *range xticks* yang sangat jauh, dipostulasikan bahwa dataset *train* dan *test* memiliki *outliers* yang jumlahnya tidak sedikit. Hal ini akan menjadi pertimbangan untuk *data preprocessing*

Untuk dapat menjustrifikasi lebih jauh terkait dengan perbedaan distribusi dari kedua dataset, maka dilakukan uji Kolmogorov-Smirnov untuk menguji *goodness of fit* terhadap dua sampel distribusi yang berbeda, dalam hal ini adalah kolom-kolom dataset *train* dan *test* yang bersifat numerik. Namun, perlu diperhatikan bahwa pada dataset terdapat banyak ***missing values***. Maka dari itu, uji statistik dilakukan dua kali, yakni dimana dataset masih memiliki *missing values*, dan dengan tidak memilikinya. Hasilnya adalah sebagai berikut:

```
Kolmogorov-Smirnov Tests:
-Hari_Diasuransikan:
KstestResult(statistic=0.02211854676267455, pvalue=4.837805991158851e-28)
-Umur:
KstestResult(statistic=0.05621752740303387, pvalue=6.593721204575883e-179)
-Biaya:
KstestResult(statistic=0.06492331339752155, pvalue=1.6099716101311423e-238)
```

Gambar 5 KS Test pada dataset murni

Dengan masih adanya *missing values*, dapat dilihat nilai *p-value* berada di bawah 0.05 (*confidence level* yang ditinjau). Maka dari itu, *null hypothesis* dapat ditolak dan dapat diambil kesimpulan bahwa perbandingan distribusi tiga kolom numerik *train* dan *test* adalah berbeda.

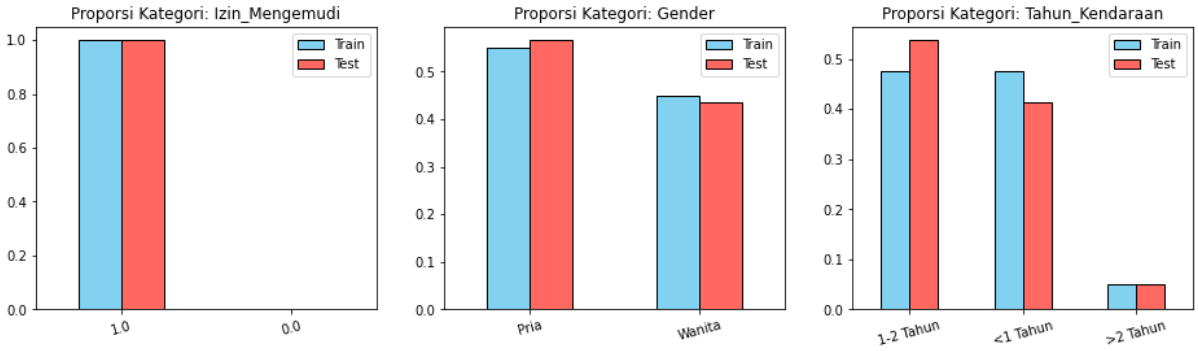
```
Kolmogorov-Smirnov Tests [NO MISSING VALUES]:
-Hari_Diasuransikan:
KstestResult(statistic=0.009365446174616765, pvalue=0.24096749566291065)
-Umur:
KstestResult(statistic=0.12019394125314753, pvalue=9.492902575944933e-152)
-Biaya:
KstestResult(statistic=0.03314823405569223, pvalue=6.587742107379246e-12)
```

Gambar 6 KS Test pada dataset bersih

Didapatkan bahwa untuk *missing values* yang sudah dihilangkan, uji statistik menunjukkan adanya peningkatan kesamaan dibandingkan dengan adanya *missing values*. Lebih menariknya lagi, ditemukan bahwa tanpa *missing values*, distribusi dari "Hari_Diasuransikan" untuk kedua dataset **sama secara statistik** dengan nilai di atas *confidence level* 0.05.

2.2 Proporsi Kategori Antara Train & Test

Setelah mengetahui bahwa terdapat 5 kolom kategorikal dari *train* dan *test* yang dapat digunakan sebagai prediktor dalam model, perlu dilakukan peninjauan apakah proporsi-proposri kategori pada setiap kolom kategorikal sama atau paling tidak serupa antara dataset *train* dan *test*. Hal ini penting untuk diketahui karena dalam *predictive analysis & modelling*, karakteristik dari proporsi kategori pada dataset dapat mempengaruhi hasil dari prediksi, apalagi ketika proporsi antara dataset *train* dan *test* berbeda. Maka dari itu, perlu dilakukan perbandingan untuk setiap kolom kategorikal yang ada.

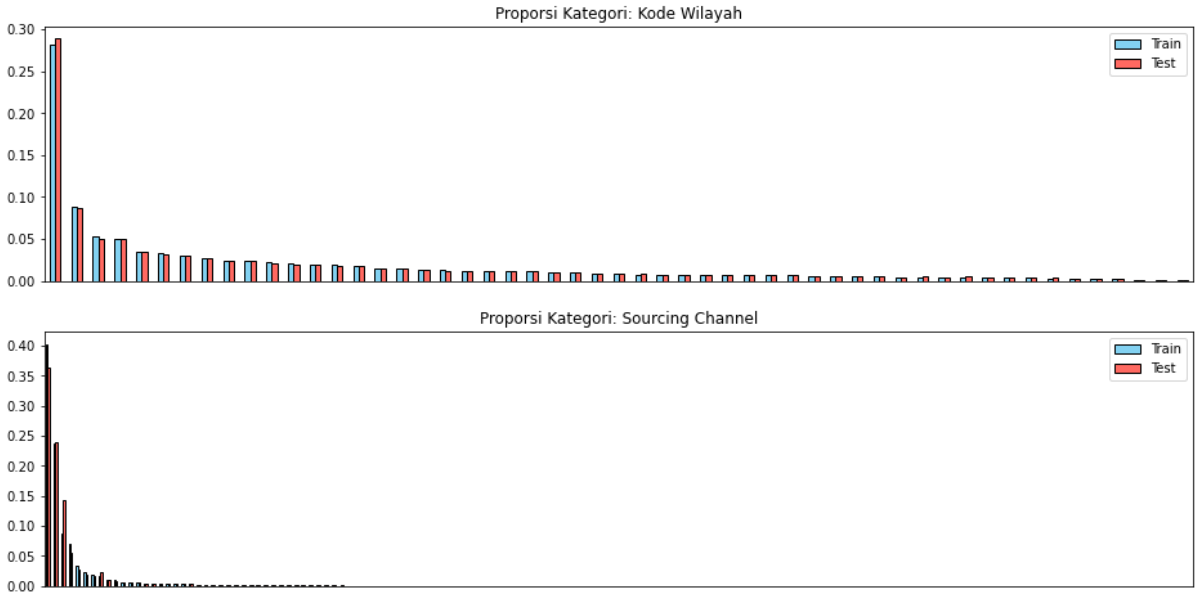


Gambar 7 Proporsi kategori kolom "Izin_Mengemudi", "Gender", "Tahun_kendaraan"

Perhatikan bahwa proporsi untuk kolom kategorikal Izin_Mengemudi, Gender, dan Tahun_Kendaraan tidak begitu berbeda diantara dataset *train* dan *test*. Beberapa informasi lainnya yang dapat diperoleh dari *plot* ini adalah bahwa:

- Kebanyakan yang terdaftar dalam asuransi ini adalah Pria dengan jumlah pria sebanyak 192814 & 39971 dan perempuan sebanyak 157572 & 30724 pada dataset *train* dan *test* secara berturut-turut.
- Lama waktu kepemilikan kendaraan kebanyakan berada di bawah 2 tahun, sedangkan yang di atas 2 tahun hanya berjumlah di bawah 10% untuk kedua dataset.
- Mayoritas dari orang yang terdaftar di dalam dataset ini sudah memiliki izin mengemudi, yakni 99.8815% dan 99.8449% berturut-turut pada dataset *train* dan *test*.

Informasi ini nantinya dapat digunakan sebagai pertimbangan untuk *preprocessing*, terutama terkait dengan mayoritas orang yang sudah memiliki izin mengemudi.



Gambar 8 Proporsi kategori kolom “Kode_Wilayah” dan “Sourcing_Channel”

Dengan melihat dua kolom kategorikal lainnya yakni kode wilayah serta *sourcing channel*, ditemukan bahwa terdapat banyak sekali nilai-nilai kategori yang berbeda-beda. Oleh karena itu, label pada sumbu x dihilangkan untuk dapat mengamati dengan lebih rapih. Ditemukan bahwa untuk tiap *sourcing channel* dan kode wilayah, proporsi pada dataset *train* dan *test* cukup serupa. Selain itu, ditemukan pula bahwa kode wilayah dan *sourcing channel* berjumlah cukup banyak yakni di atas 30% pada satu kategori saja.

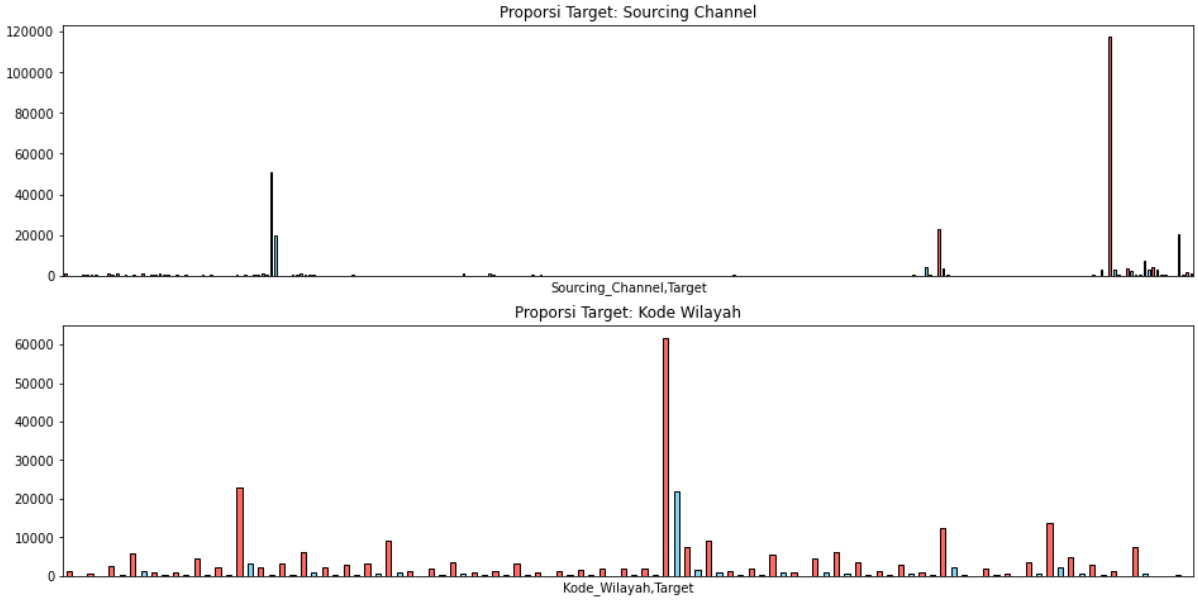
2.3 Proporsi Target Terhadap Kolom Kategorikal

Dengan menemukan bahwa proporsi dari kolom izin mengemudi banyak bernilai 1, yakni memiliki izin mengemudi, dicurigai bahwa kolom kategorikal ini tidak akan memiliki pengaruh signifikan terhadap keinginan seseorang untuk membeli atau melanjutkan asuransinya, yakni kolom target. Dengan dasar seperti itu, perlu dicari tahu apakah kolom-kolom kategorikal ini memiliki distribusi nilai target yang bermakna untuk setiap kategori ataukah tidak.

Gambar 9 Proporsi target kolom "Izin_Mengemudi", "Gender", "Tahun_kendaraan"

Untuk tiga kolom kategori pertama, ditemukan bahwa proporsi dari target 0 dan 1 pada kolom Gender dan Tahun_Kendaraan cukup normal walaupun jumlahnya yang kalah jauh dengan rasio perbedaan yang sangat tidak seimbang. Namun, sesuatu yang dapat diperhatikan adalah bahwa pada kolom izin mengemudi – sesuai dengan deskripsi poin sebelumnya – tidak terdapat banyak jumlah target pada orang-orang yang tidak memiliki izin mengemudi. Maka dari itu, pengetahuan terkait bahwa orang-

orang banyak yang sudah memiliki izin mengemudi, kepentingan dari kolom ini menjadi berkurang yang akan menjadi pertimbangan untuk *data preprocessing*.



Gambar 10 Proporsi target kolom "Kode_Wilayah" dan "Sourcing_Channel"

Penelaahan untuk kolom kategorikal dari *sourcing channel* dan kode wilayah menunjukkan bahwa lebih banyak target yang bernilai 0 untuk kedua kolom. Distribusi dari target untuk kedua kolom ini jauh lebih sulit terlihat karena kategori yang ada berjumlah cukup banyak. Namun sekilas, terlihat bahwa target berniali 0 lebih mendominasi dataset pada setiap kategori.

2.4 Persentase *Missing Values*

Nilai yang hilang atau yang sering disebut dengan *missing values* adalah masalah yang kerap kali dihadapi pada dataset riil. Ketika *missing values* ditemukan pada suatu dataset, hal ini dapat menjadi masalah yang mengganggu performa prediksi yang dilakukan oleh model. Oleh karena itu, menjadi penting untuk mengetahui berapa banyak bagian dari dataset yang dimiliki, memiliki *missing values*.

Gambar 11 Persentase *Missing Values*

Dari *bar* yang telah dipetakan, ditemukan bahwa *missing values* pada dataset *train* dan *test* cukup banyak, dengan semua kolom selain id dan target memiliki *missing values*. Selain kolom Gender, *missing values* pada kolom ada lebih dari 15% baik untuk *train* atau *test*. Kolom biaya memiliki persentase *missing values* tertinggi untuk kedua dataset. Untuk rincian proporsinya dapat dilihat pada tabel di bawah ini.

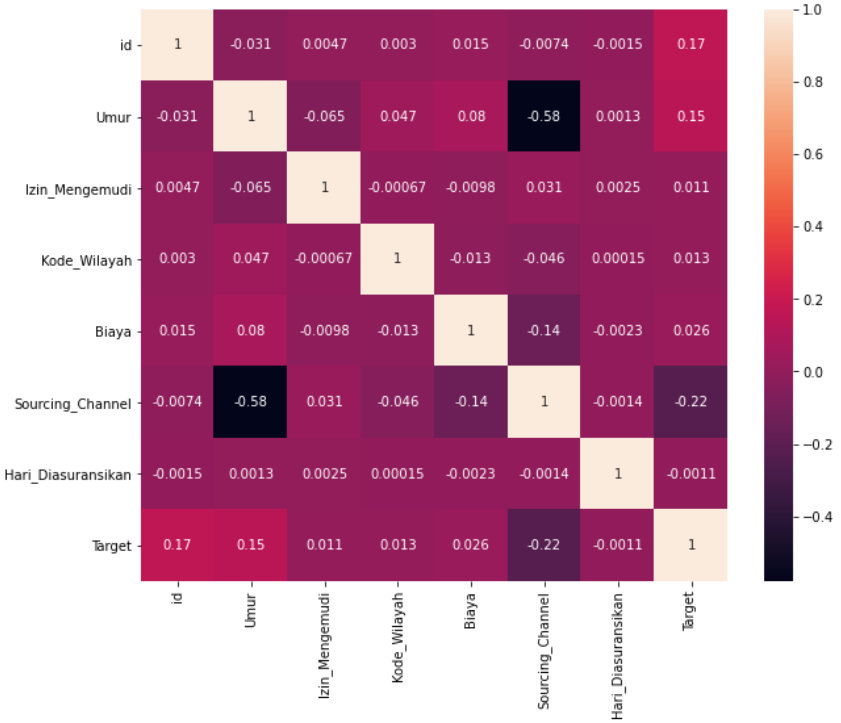
Tabel 2 Persentase missing values pada test dan train

Train Missing Values Percentage		***Test Missing Values Percentage***	
id	0.000000	id	0.000000
Gender	0.083129	Gender	0.096815
Umur	0.251883	Umur	0.199060
Izin_Mengemudi	0.200566	Izin_Mengemudi	0.200746
Kode_Wilayah	0.220000	Kode_Wilayah	0.249805
Tanggal_Asuransi	0.204326	Tanggal_Asuransi	0.143753
Tahun_Kendaraan	0.173857	Tahun_Kendaraan	0.173125
Biaya	0.331115	Biaya	0.266337
Sourcing_Channel	0.218878	Sourcing_Channel	0.169139
Hari_Diasuransikan	0.197999	Hari_Diasuransikan	0.219999
Target	0.000000		

Dengan persentase *missing value* yang cukup banyak, sebelum dilakukan *training* perlu dilakukan *data preprocessing* yang mencakup imputasi *missing values*.

2.5 *Correlation Matrix (Heatmap)*

Untuk mempertimbangkan konstruk dan struktur dari model lebih jauh, peninjauan terhadap nilai korelasi antara setiap variabel – kategorikal ataupun numerik – dapat dilakukan untuk mendapatkan informasi lebih terkait dengan hubungan yang ada. Maka dari itu, digunakan *heatmap* dari *library seaborn* untuk memvisualisasikan hal ini:



Gambar 12 Correlation matrix antar variabel

Didapatkan bahwa variabel-variabel tidak berkorelasi tinggi untuk satu sama lain, namun terdapat beberapa poin informasi yang penting untuk dipertimbangkan:

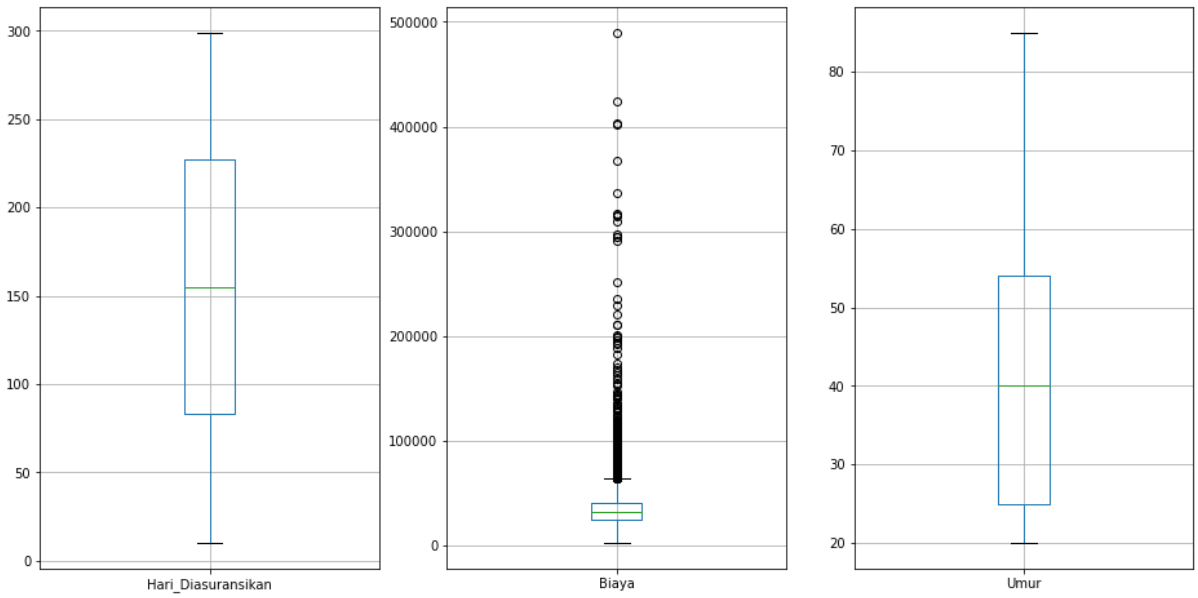
- Umur dan *Sourcing_Channel* memiliki korelasi yang cukup besar -0.58, namun perlu diperhatikan bahwa *sourcing channel* adalah variabel yang sifatnya sebenarnya kategorikal sehingga tidak terdapat kesimpulan hubungan yang jelas dalam hal ini.
- Korelasi-korelasi yang ada pada kolom-kolom yang ada tidak begitu kuat.

2.6 Deteksi Outlier

Outliers dapat didefinisikan sebagai suatu observasi yang menyimpang jauh dari observasi lainnya. Jika tidak ditangani, *outliers* dapat menghambat kemampuan model untuk menggeneralisasi terhadap set data lainnya. Pengeluaran *outliers* biasa diasosiasikan dengan data bertipe numerik sehingga pada kasus ini, akan diteliti sebaran data untuk tiga kolom saja, yaitu Hari_Diasuransikan, Biaya, dan Umur. Untuk mendeteksi *outlier,* digunakan visualisasi berupa *box plot* yang mendiskriminasi *data points* pada set data berdasarkan interval yang dihitung dari beberapa ukuran *quantile*. *Box plot* untuk ketiga kolom numerik tersebut, beserta rumus untuk memperoleh batas-batasnya dapat dilihat sebagai berikut.

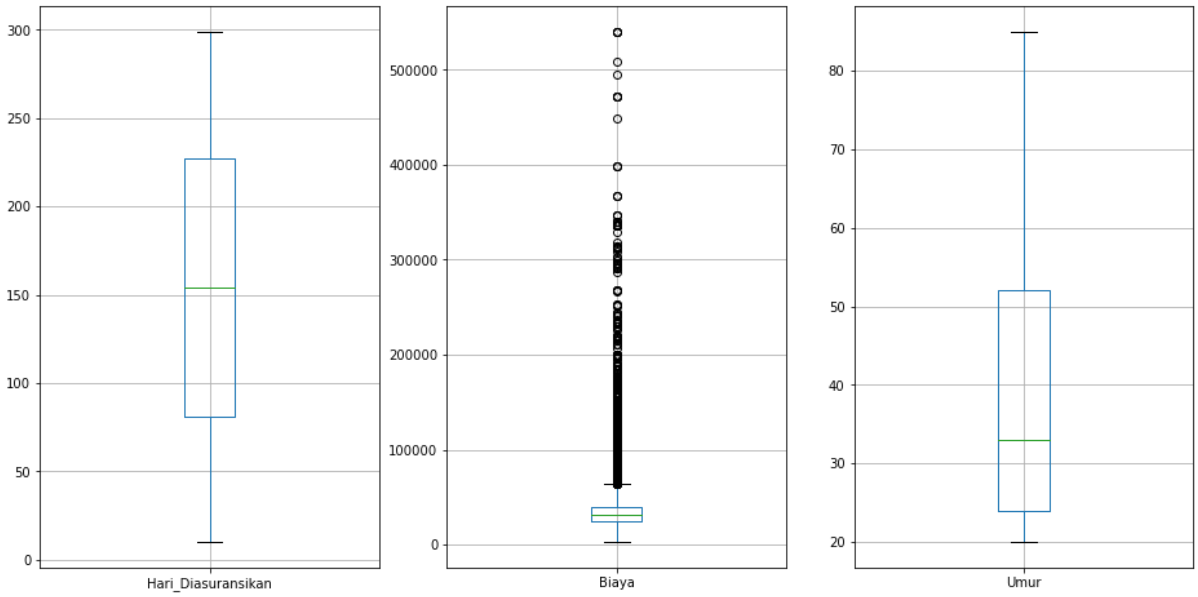
Tabel 3 Rumusan batasan

Outlier Threshold	**Keterangan**
$Batas\ minimum = Q1 + 1.5 \cdot IQR$ $Batas\ maksimum = Q3 + 1.5 \cdot IQR$	$Q1: 0.25\ quantile$ $Q3: 0.75\ quantile$ $IQR\ (interquartile\ range) = Q3 - Q1$



Gambar 13 Boxplot kolom numerik *test*

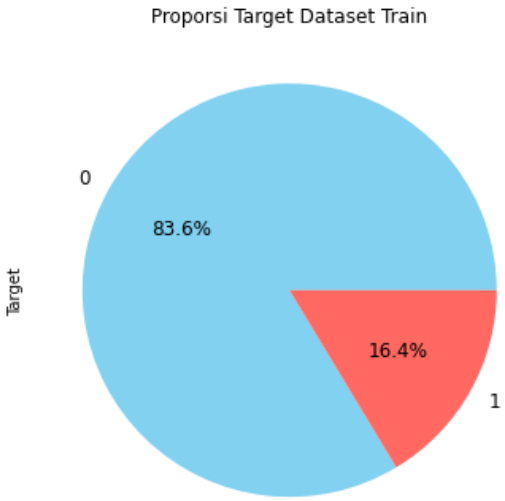
Perhatikan bahwa pada dataset *test*, terdapat banyak *outliers* yang terdeteksi pada kolom numerik Biaya. Hal ini menunjukkan bahwa perlu dilakukan peninjauan lebih lanjut terkait dengan dataset *train*.



Gambar 14 Boxplot kolom numerik *train*

2.7 Proporsi Target Dataset Train

Prediksi yang dihasilkan berdasarkan sebuah model *Machine Learning* akan terpengaruh ketika sebuah *dataset* memiliki variabel target yang tidak terbagi secara rata. Ada kemungkinan bahwa model akan cenderung "bersandar" pada prediksi terhadap target yang lebih banyak jumlahnya di dataset *train*. Maka dari itu, identifikasi terhadap proporsi ini perlu dilakukan. Dengan menggunakan *pie chart*, diperoleh visualisasi sebagai berikut.



Gambar 15 Proporsi kolom "Target" pada dataset *train*

Perhatikan bahwa target pada dataset *train* terdapat jumlah target bernilai 1 yang jauh lebih sedikit dari target bernilai 0 dengan rasio mendekati 1:4. Dapat disimpulkan bahwa *dataset* ini *extremely imbalanced* dan perlu dilakukan *processing* atau langkah intervensi tertentu seperti SMOTE yang akan dibahas lebih lanjut nantinya.

3.8 ID Duplikat Pada Train & Test

Dalam suatu dataset tertentu, tidak menutup kemungkinan bahwa terdapat suatu probabilitas terjadinya *data leakage*. *Data leakage* dalam kasus ini dapat didefinisikan sebagai saat suatu *record* yang memiliki identitas dan karakteristik yang sama berada pada kedua dataset, yakni dataset *train* untuk melatih model dan dataset *test* untuk menguji model. Keberadaan *leakage* ini akan mempengaruhi performansi dan perilaku model terhadap prediksinya, oleh karena itu identifikasi perlu dilakukan terlebih dahulu. Telah ditemukan bahwa terdapat paling tidak 4406 *records* yang berada pada kedua dataset. Terdapat postulasi bahwa id yang berada di kedua dataset sebenarnya adalah id yang sama dengan karakteristik kolom-kolom yang sama. Untuk itu, dilakukan sampling 5 baris id yang ada pada kedua dataset.

	id	Gender	Umur	Izin_Mengemudi	Kode_Wilayah	Tanggal_Asuransi	Tahun_Kendaraan	Biaya	Sourcing_Channel	Hari_Diasuransikan	Target
263815	12028	Pria	64.0	1.0	13.0	2019-10-01	1-2 Tahun	53081.0	124.0	135.0	0
28469	179464	Wanita	64.0	1.0	41.0	2019-01-17	NaN	NaN	26.0	40.0	0
53944	306525	NaN	62.0	1.0	5.0	2019-12-19	1-2 Tahun	NaN	NaN	NaN	0
302364	372649	Pria	65.0	1.0	28.0	NaT	>2 Tahun	2630.0	55.0	218.0	0
111725	448635	Pria	66.0	1.0	23.0	2018-07-08	1-2 Tahun	NaN	26.0	NaN	0

Gambar 16 ID Duplikat pada *Train*

	id	Gender	Umur	Izin_Mengemudi	Kode_Wilayah	Tanggal_Asuransi	Tahun_Kendaraan	Biaya	Sourcing_Channel	Hari_Diasuransikan
38	12028	Pria	64.0	1.0	13.0	10/1/2019	1-2 Tahun	53081.0	124.0	135.0
86	179464	Wanita	64.0	1.0	NaN	1/17/2019	1-2 Tahun	NaN	26.0	40.0
9	306525	Pria	62.0	NaN	5.0	12/19/2019	1-2 Tahun	NaN	NaN	NaN
76	372649	Pria	NaN	1.0	28.0	NaN	>2 Tahun	2630.0	55.0	218.0
18	448635	Pria	NaN	NaN	23.0	7/8/2018	1-2 Tahun	NaN	NaN	256.0

Gambar 17 ID Duplikat pada *Test*

Perhatikan bahwa pada kedua tabel di atas, seluruh kolom yang nilainya tidak *missing* memiliki nilai yang sama persis. Untuk *missing values* yang ada pada tiap *record* ternyata memiliki kekosongan yang berbeda untuk *train* dan *test*. Misalnya saja untuk id 306525 ditemukan bahwa Gender pada *Train* bernilai NaN, namun Gender-nya terlihat pada dataset *Test* yakni Pria. Hal ini menunjukkan adanya kemungkinan kuat bahwa id yang sama ini merupakan data / *records* yang sama yang dapat menjadi pertimbangan untuk *model-building* ataupun untuk implikasi manajerial bagi perusahaan asuransi J.

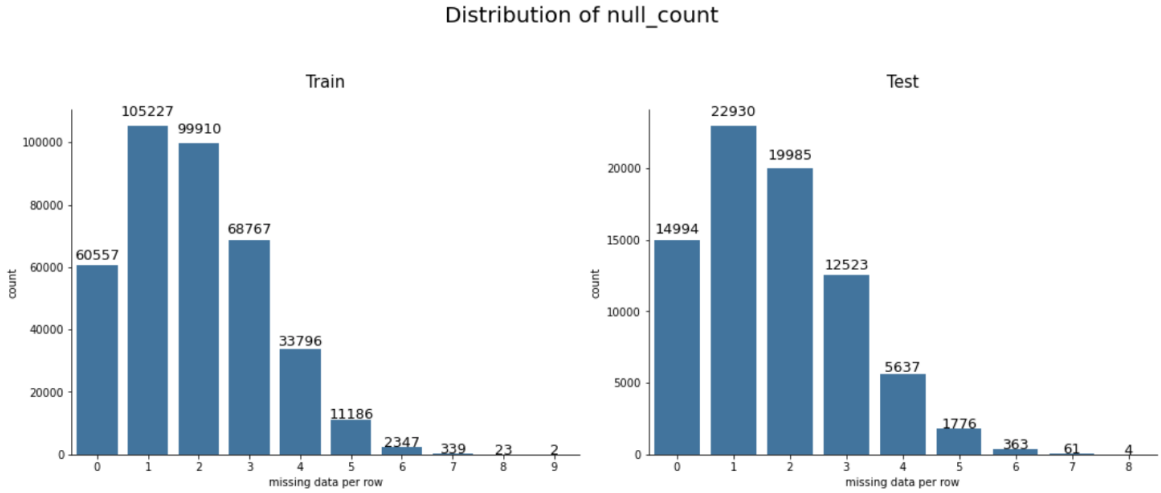
BAB III
METODE DATA MINING

3.1 Feature Engineering

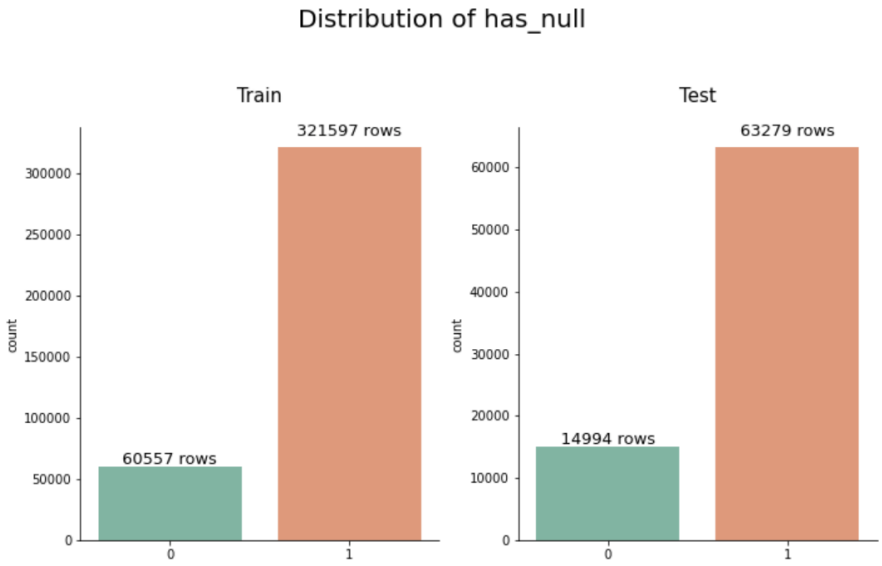
Di samping 10 kolom prediktor yang sudah tersedia, akan dibuat beberapa kolom baru yang merupakan hasil ekstraksi informasi dari kolom yang sudah ada. Dengan dibuatnya kolom baru tersebut, diharapkan model yang akan digunakan dapat belajar lebih banyak dari data yang ada.

3.1.1 Penambahan kolom null_count & has_null

Pertama, akan dibuat dua buah kolom baru bernama null_count dan has_null yang merefleksikan komposisi *missing data* pada setiap baris. Kolom null_count merepresentasikan jumlah nilai NaN per baris. Mengingat bahwa terdapat 10 kolom prediktor di awal, maka jumlah maksimal NaN per baris adalah 10. Dapat terlihat di bawah ini bahwa distribusi null_count untuk data *train* dan *test* cukup mirip. Selain itu, terlihat juga bahwa pada data *train* maupun *test* tidak terdapat satu baris pun yang memiliki nilai NaN keseluruhan. Jumlah NaN maksimal per baris untuk data *train* dan test berturut-turut adalah 9 kolom dan 8 kolom.



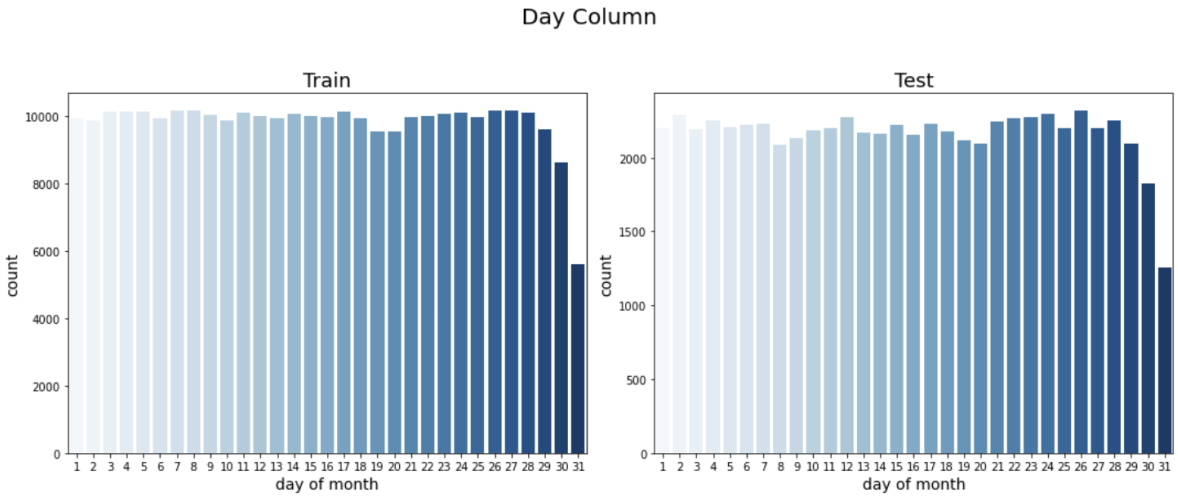
Kolom baru kedua yang akan ditambahkan bernama has_null dan hanya merefleksikan apakah sebuah baris memiliki paling sedikit satu nilai NaN atau tidak memiliki NaN sama sekali. Tentunya, kolom ini berjenis biner di mana kategori 1 menandakan terdapat paling sedikit satu NaN dan kategori 0 menandakan baris yang lengkap. Ini berarti bahwa baris yang memiliki null_count lebih dari nol akan menghasilkan kolom has_null bernilai 1 dan sebaliknya. Berikut merupakan visualisasi dari kolom has_null.

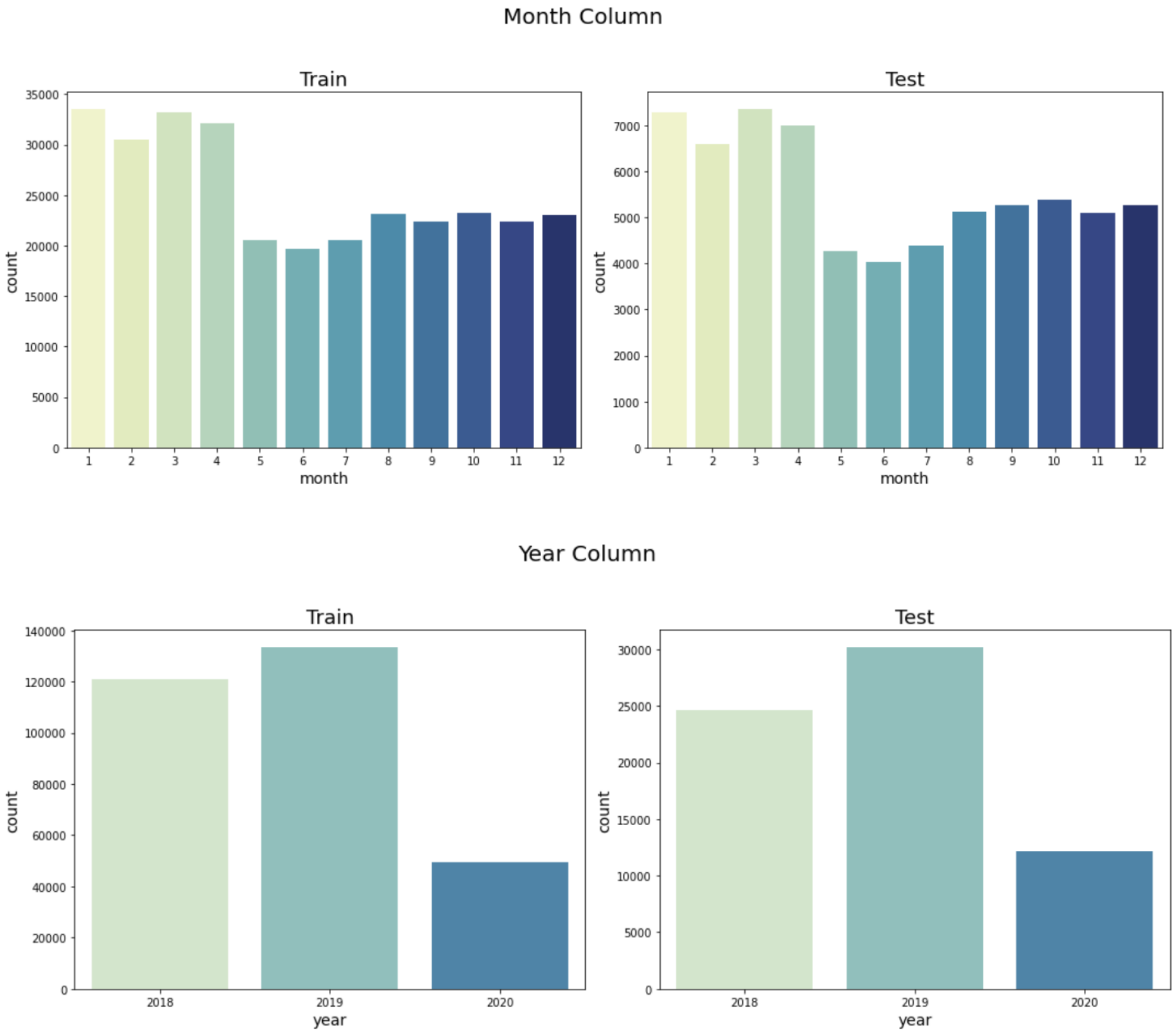


Pada *bar plot* di atas terlihat pula bahwa distribusi has_null untuk data *train* dan *test* sangat mirip. Karena proporsi kedua kategori tersebut identik, maka kolom tersebut dapat digeneralisasikan oleh model ketika melakukan prediksi.

3.1.2 Ekstraksi Tanggal Asuransi

Salah satu kolom yang berbasis waktu pada set data adalah Tanggal_Asuransi. Kolom tersebut didefinisikan sebagai tanggal pelanggan mulai melakukan layanan asuransi. Perlu diketahui juga bahwa Tanggal_Asuransi memiliki format MM/DD/YYY sehingga masing-masing komponen tanggal tersebut, yakni bulan, hari, dan tahun dapat diekstraksi menjadi kolom tersendiri. Setelah dibuat tiga kolom tersebut, kolom Tanggal_Asuransi pun dibuang. Berikut ditampilkan distribusi dari ketiga kolom baru tersebut.



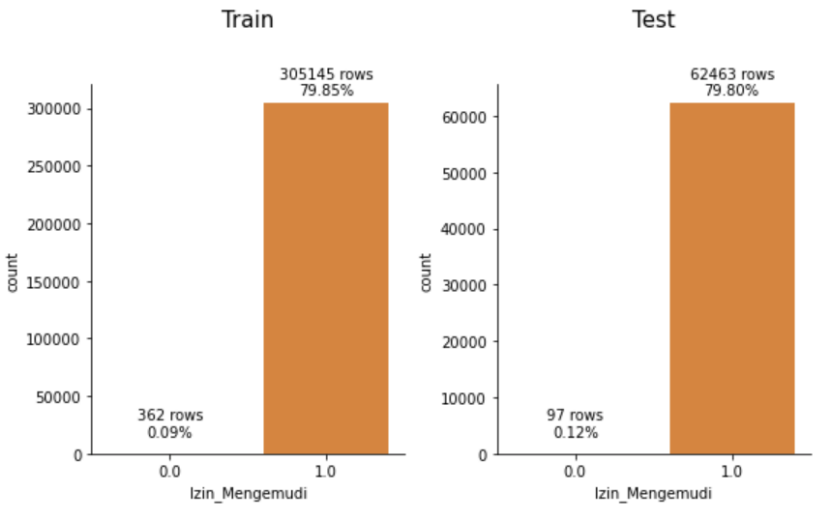


3.2 Data Preprocessing

Pemrosesan data dilakukan untuk mencegah terjadinya *Garbage In Garbage Out* (GIGO). Setelah memperoleh beberapa *insights* pada *exploratory data analysis*, terdapat beberapa metode pemrosesan yang diharapkan dapat menjamin kualitas data ketika melakukan *training model*. Metode pemrosesan yang digunakan adalah sebagai berikut.

3.2.1 Penghapusan kolom Izin_Mengemudi

Setelah meneliti proporsi kategori untuk kolom Izin_Mengemudi, ditemukan bahwa proporsi nilai 1.0 (memiliki izin mengemudi) jauh lebih tinggi dibandingkan dengan proporsi nilai 0.0 (tidak memiliki izin mengemudi) untuk jenis dataset. Dapat terlihat pada *bar plot* di bawah ini bahwa untuk kolom Izin_Mengemudi di data *train,* proporsi baris yang bernilai 1.0 mencapai 79.85% dan yang bernilai 0.0 hanya 0.09%. Di sisi lain, pada data *test* proporsi baris yang bernilai 1.0 adalah 79.80% dan yang bernilai 0.0 hanya 0.12%. Dapat diperhatikan juga bahwa penjumlahan dari proporsi tersebut tidak mencapai 100%, sebab sisa barisnya bernilai NaN atau *missing*.



Fenomena tersebut cukup menarik, sebab proporsi kategori 1.0 yang jauh melebihi kategori 0.0 berarti bahwa status kepemilikan Izin_Mengemudi tidak akan berpengaruh signifikan terhadap prediksi variabel target. Maka, kolom Izin_Mengemudi akan dihapus pada data *train* dan *test*.

3.2.1 Binning umur: penambahan kolom

Salah satu metode pemrosesan yang dapat digunakan untuk mencegah *overfitting* pada saat *training* adalah *binning*. Pada dasarnya, *binning* meng-*assign* data numerik ke kelompok tertentu berdasarkan interval yang telah ditetapkan. Interval yang digunakan beserta nilai yang di-*assign* dapat dilihat pada tabel di bawah ini.

Bin interval	*Assigned value*
$1-20$	0
$21-30$	1
$31-40$	2
$41-55$	3
>55	4
NaN	-1

Assignment nilai yang dilakukan pada kolom Umur pada dasarnya mengubah data yang semulanya bertipe numerik kontinu menjadi data bertipe kategori ordinal. Data umur yang telah dikelompokkan ke dalam *bin* menjadi ordinal sebab *assignment* nilai mengindikasikan sebuah *ranking.* Kolom baru ini bukan merupakan skala interval, sebab selisih antar *assigned value* belum tentu memiliki jarak yang sama. Transformasi ini cocok dilakukan sebab data umur dapat direpresentasikan menggunakan kategori berurutan, seperti $remaja < paruh\ baya < dewasa < orang\ tua$. Selain itu, patut diperhatikan juga bahwa keberadaan *missing data* atau NaN pada kolom Umur ditindaklanjuti dengan *assignment* nilai -1 (*flagging*).

3.2.2 Imputasi *missing values*

Sebagaimana telah dijelaskan pada subbab 2.4, terdapat cukup banyak *missing values* pada data *train* dan data *test*. Pada data *train* sendiri, terdapat 84,15% baris yang memiliki paling sedikit satu *missing value*. Tentunya, angka tersebut sangat besar sehingga akan sangat *costly* jika setiap baris yang memiliki nilai NaN dibuang begitu saja. Salah satu alternatif yang dapat dilakukan untuk menindaklanjuti masalah ini adalah dengan imputasi *missing value*. Tabel di bawah ini menunjukkan metode imputasi yang digunakan untuk beberapa kolom.

Nama kolom	Metode Imputasi
Umur	Median
Biaya	Median
Hari_Diasuransikan	Mean
Kode_Wilayah	Flagging (-1)
Sourcing_Channel	Flagging (-1)
day	Flagging (-1)
month	Flagging (-1)
year	Flagging (-1)

Perlu ditekankan juga bahwa ketika melakukan men-*split* data menjadi *train* dan *validation set*, data validasi hanya diimputasi menggunakan metrik (median atau mean) yang diperoleh dari data *train.* Hal ini dilakukan untuk mencegah *data leakage*, yaitu kejadian di mana karakteristik data *test* terungkap di dalam data validasi. Tentunya, sebelum melakukan prediksi dengan model, data *test* juga diimputasi seperti cara di atas.

Untuk kolom Kode_Wilayah, Sourcing_Channel, day, month, dan year merupakan kolom yang kami identifikasi sebagai data kategorikal. Maka dari itu, proses flagging juga dapat dikatakan membuat sebuah kategori baru pada kolom-kolom tersebut yang mengindikasikan adanya missing value. Kemudian kolom-kolom ini juga akan diubah menjadi data tipe *integer* atau bilangan bulat dari sebelumnya data bertipe *float* atau bilangan *real.* Hal ini dilakukan agar kelak model yang digunakan dapat membaca kolom-kolom ini sebagai data kategorikal.

3.2.1 Label Encoding

Selain data bertipe numerik, perlu diingat juga bahwa pada data *train* dan *test* terdapat dua kolom yang bertipe *string* atau *object,* yaitu Gender dan Tahun_Kendaraan. Agar dapat digunakan sebagai input model, kedua kolom tersebut perlu dikonversi menjadi data bertipe numerik. Untuk melakukan ini, akan digunakan metode *label encoding*, yaitu *assignment* sebuah integer yang unik untuk setiap kategori yang berbeda.

3.3 Modelling

Dari hasil *preprocessing* dan *feature engineering*, berikut merupakan *feature* yang digunakan sebagai input dari model-model yang akan dipakai. Kolom Id digunakan sebagai *predictor* karena dari hasil EDA diperoleh informasi bahwa terdapat Id yang sama pada test set, sehingga jika kolom Id dimasukkan sebagai salah satu *predictor* maka model dipercaya dapat memprediksi test set lebih baik.

Column	Type
Id	Numerikal
Gender	Kategorikal
Umur	Numerikal
Kelas_Umur	Kategorikal
Kode_Wilayah	Kategorikal
Tahun_Kendaraan	Kategorikal
Biaya	Numerikal
Sourcing_Channel	Kategorikal
Hari_Diasuransikan	Kategorikal
Null_count	Numerikal
Has_null	Kategorikal
Day	Kategorikal
Month	Kategorikal
Year	Kategorikal

Pada kasus kali ini, tim ElsiX menggunakan beberapa model untuk menyelesaikan permasalahan pada perusahaan J Insurance terkait prediksi apakah suatu pelanggan akan lanjut meneruskan layanan asuransinya atau tidak. Model yang digunakan semua merupakan model yang berbasis *decision tree*. Maka dari itu terdapat beberapa kelebihan, yaitu:

1. *Multicollinearity* atau *feature-feature* yang memiliki korelasi tinggi tidak berpengaruh terhadap model
2. Distribusi data input tidak perlu memiliki distribusi yang normal
3. Data input tidak perlu *scaling*

Model pertama yang digunakan adalah XGBoost. XGBoost merupakan model dengan basis *decision tree* yang menggunakan *ensemble method* yaitu *boosting. Boosting* merupakan sebuah *ensemble method*, yaitu teknik yang menggabungkan model-model lemah, kemudian mempelajari model lemah tersebut secara sekuensial dengan cara yang adaptive. Jadi setiap mempelajari model yang lemah, maka model utamanya akan mempelajari kesalahannya dan memperbaiki pada model lemah selanjutnya. Dengan teknik ini model-model lemah yang digabungkan dapat menjadi model yang *powerful.* XGBoost memiliki beberapa kelebihan seperti memiliki regulasi sehingga mencegah *overfitting,* memiliki waktu komputasi yang relatif lebih cepat, dan memiliki *tree pruning* sehingga meningkatkan *computational performance*.

Model kedua adalah CatBoost, yang merupakan singkatan dari Categorical Boosting dan merupakan model berbasis *decision tree* yang menggunakan *ensemble method boosting.* Sesuai

dengan namanya juga, CatBoost dikenal memiliki performansi yang baik jika input data banyak berupa data kategorikal. Terbukti juga bahwa di antara semua model yang digunakan, model CatBoost menghasilkan nilai performansi yang paling baik. Hasil perbandingan antar model dapat dilihat pada bagian setelah ini.

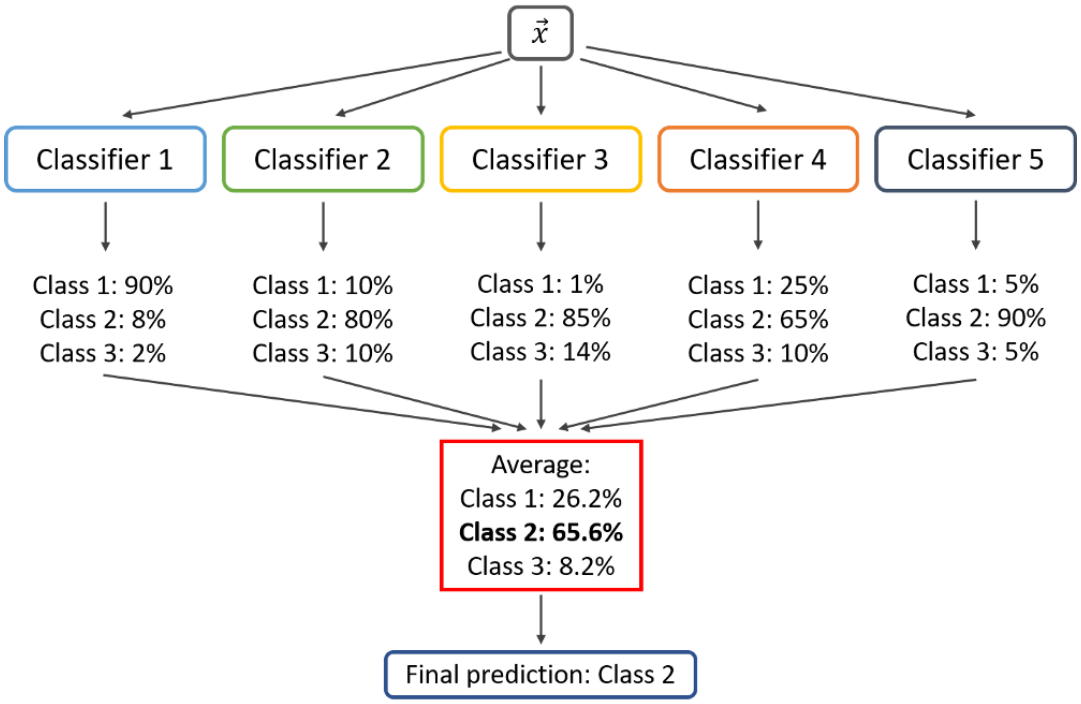
Model ketiga adalah LightGBM yang menggunakan *ensemble method boosting* juga. LightGBM menggunakan dasar histogram pada modelnya sehingga meningkatkan kecepatan *training* dan juga kebutuhan *memory* yang lebih kecil. LightGBM cocok digunakan untuk diaplikasikan terhadap dataset berukuran besar. LightGBM juga merupakan salah satu model *boosting* yang dikenal memiliki tingkat performansi baik.

Model keempat yaitu Random Forest yang juga masih merupakan model berdasarkan *decision tree.* Random forest menggunakan *ensemble method* bukan *boosting* lagi, melainkan *bagging (bootstrap aggregating).* Sederhananya jika *boosting* menggabungkan dan belajar dari model-model lemah secara sekuensial, maka *bagging* menggabungkan model lemah secara paralel dan kemudian menyatukannya untuk menjadi model yang lebih baik. Random Forest mampu untuk mengolah data berukuran secara besar sehingga cocok untuk kasus ini dan juga merupakan salah satu model dengan historis performansi yang bagus.

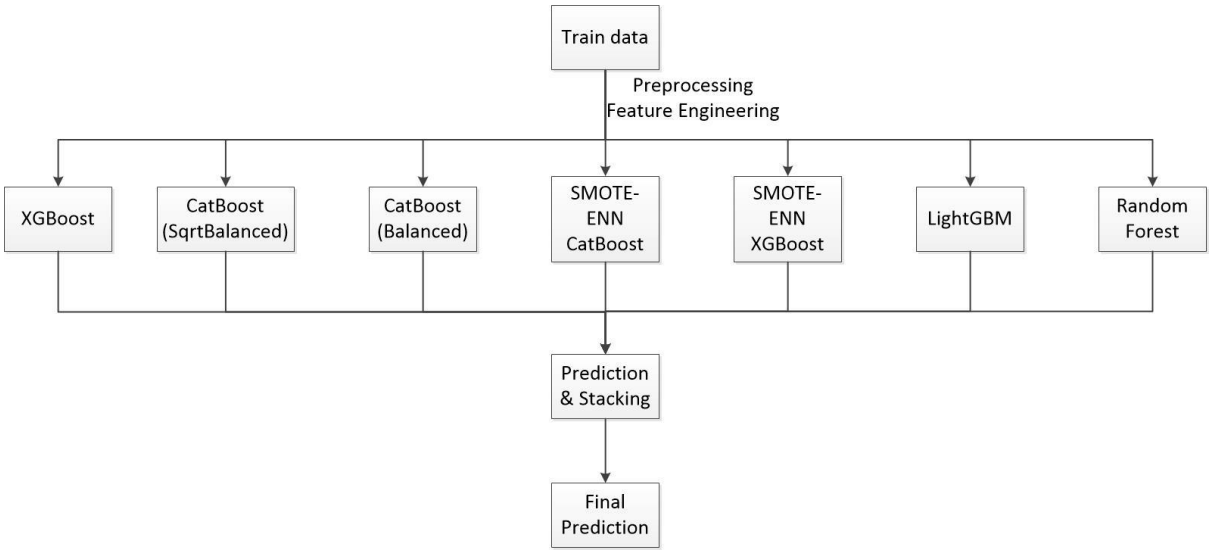
Kemudian dikarenakan ditemukan bahwa dataset yang ada merupakan *imbalanced dataset*, yaitu proporsi target atau *dependent variable* tidak seimbang, yaitu sekitar 85% kelas 0 dan 15% kelas 1, maka dilakukan teknik *oversampling*. *Oversampling* merupakan teknik mereplikasi data sehingga data memiliki kelas target yang seimbang. Dengan input data yang lebih baik maka diharapkan model dapat menghasilkan performansi yang semakin baik juga, sesuai dengan pepatah “Garbage in, garbage out”. Teknik *oversampling* yang digunakan berupa SMOTE Edited Nearest Neighbors atau SMOTE-ENN. Dengan SMOTE-ENN dilakukan *resampling* terhadap dataset sehingga dataset memiliki proporsi terhadap target lebih seimbang. Untuk metode SMOTE-ENN ini diaplikasikan terhadap model XGBoost dan CatBoost saja.

3.4 Ensemble Method (Voting)

Untuk membuat model yang menjadi lebih *robust* atau kokoh, maka dapat dilakukan *ensemble method* yang merupakan suatu teknik *machine learning* yang menggabungkan banyak model sehingga dapat diperoleh suatu model baru yang memiliki tingkat performansi lebih tinggi. Pada kali ini, teknik *ensemble* yang digunakan berupa *voting.* Sesuai dengan namanya, teknik ini pada dasarnya memilih hasil akhir prediksi sesuai dengan suara terbanyak dari model-model yang digabungkan. Teknik *voting* yang dilakukan berupa *soft voting*¸ yaitu melakukan *voting* terhadap hasil prediksi model namun dengan hasil prediksi yang berupa probabilitas. *Soft voting* melakukan perhitungan rata-rata dari probabilitas tiap kelas *dependent variable*-nya kemudian melakukan penentuan hasil prediksi berdasarkan rata-rata tersebut. Berikut merupakan ilustrasi dari *soft voting*.



Maka dari itu dilakukan teknik *soft voting* terhadap model-model yang sudah dijelaskan sebelumnya. Struktur dari teknik *ensemble voting* terhadap model-model yang dipakai sebagai berikut.



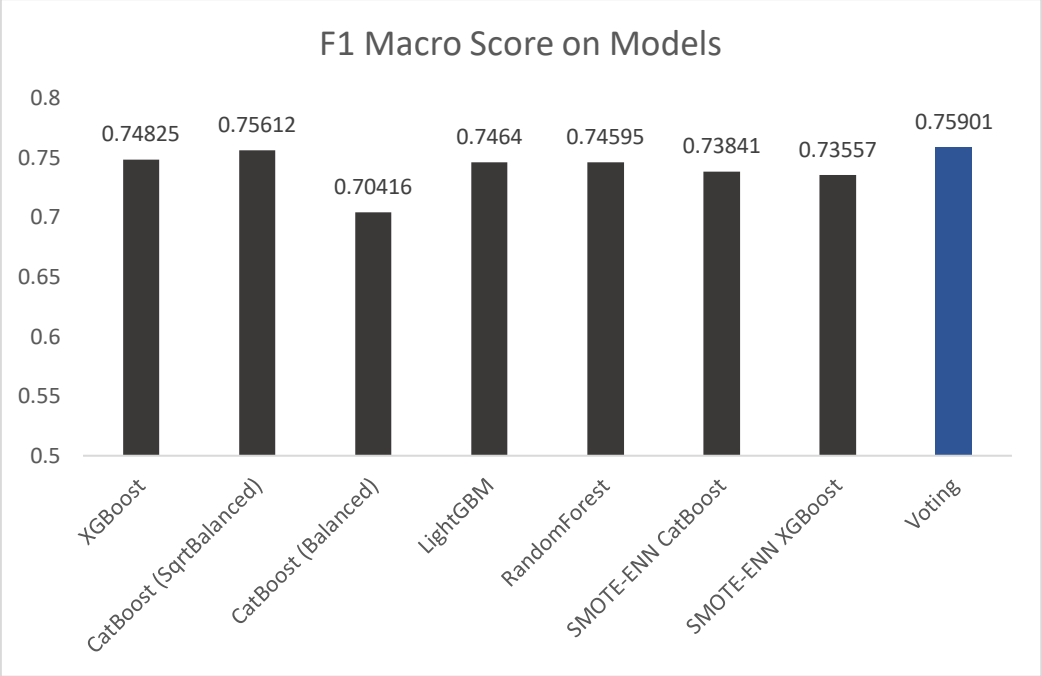
BAB IV
HASIL & PEMBAHASAN

4.1 Validation

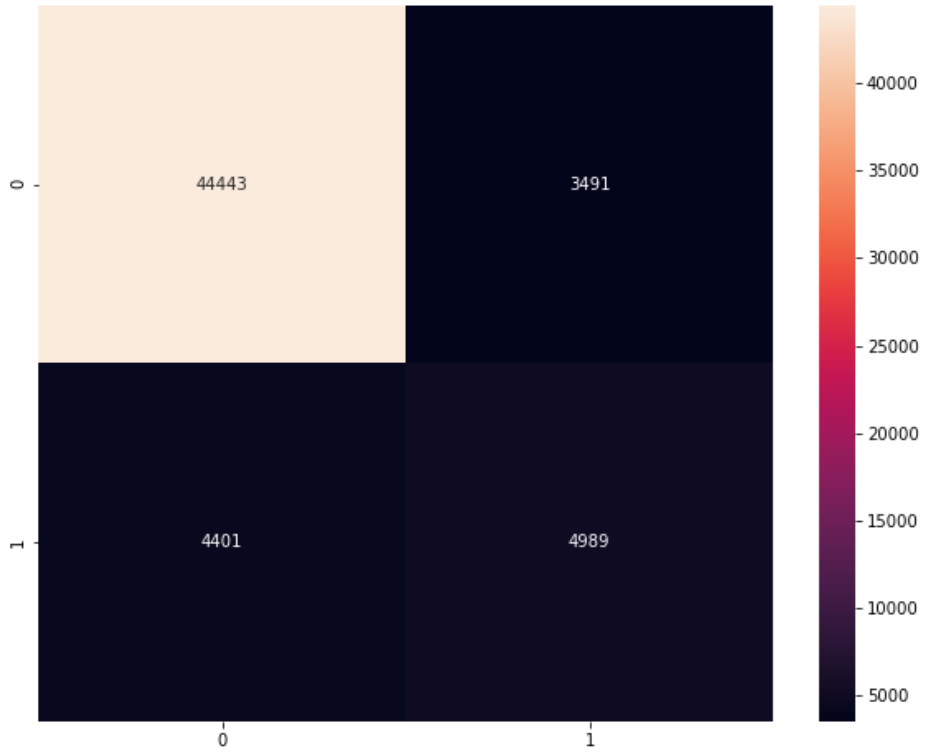
Metric untuk validasi dipilih skor F1 Macro. F1 Macro ini menilai *importance* dari tiap label secara sama sehingga bagus untuk digunakan pada *dataset* yang *imbalance*. Jika menggunakan *accuracy* pada dataset yang *imbalance*, maka hasil dari *metric accuracy* tidak *reliable* karena dengan mengisi target yang memiliki kelas lebih banyak akan menghasilkan skor *accuracy* yang tinggi. Maka dari itu dipilih F1 Macro yang dianggap lebih *reliabel* ketika dihadapi dengan *imbalanced dataset*. Ada pun rumus F1 Macro sebagai berikut. Dari rumus dibawah, dapat dikatakan bahwa F1 Macro merupakan rata-rata dari F1 Score semua kelasnya.

$$F1\ Macro = \frac{\sum_{i=1}^{n} F1\ Score}{n}$$

Setelah melakukan *training* dan *validation* pada tiap model, diperoleh bahwa memang benar teknik *voting* menghasilkan kenaikan pada skor F1 Macro dibandingkan dengan model-model lainnya. Berikut merupakan hasil validasi dari model yang digunakan beserta teknik *voting.*

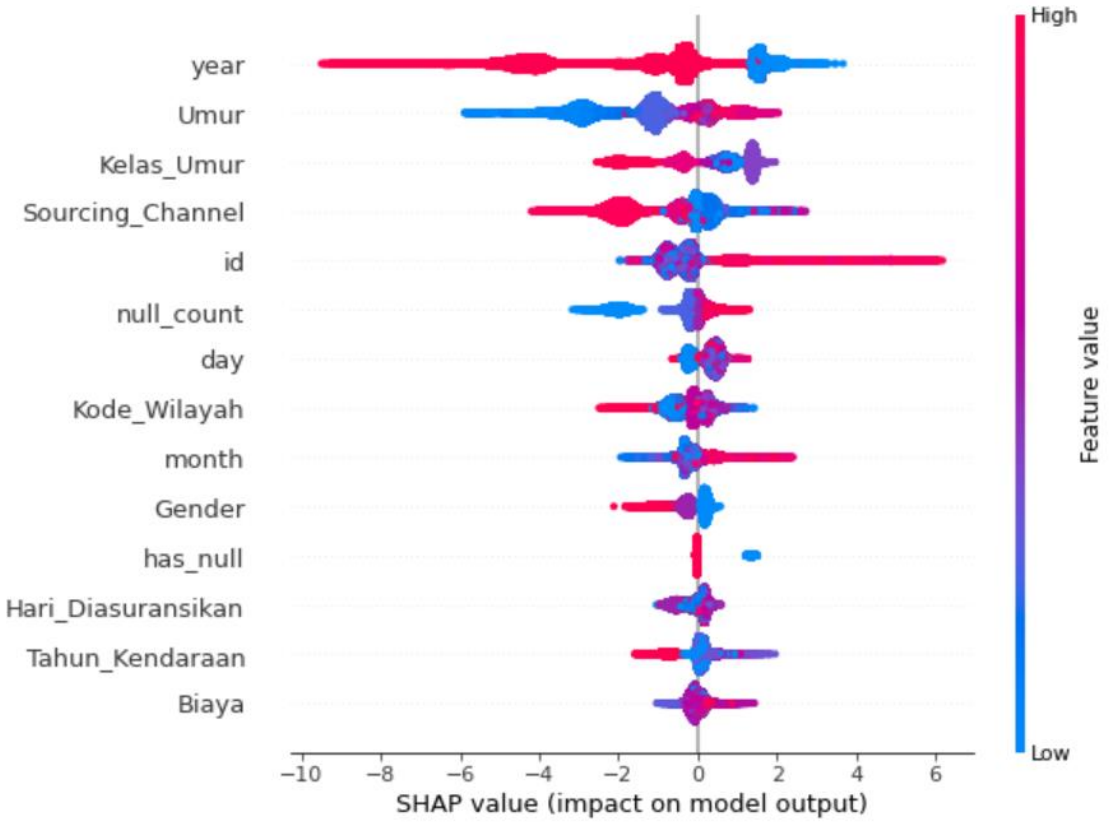


Dari hasil validasi di atas, dapat dilihat bahwa model-model memiliki skor F1 Macro dengan kisaran 0.70-0.75. Dari grafik di atas juga terbukti bahwa dengan melakukan teknik *ensemble voting* maka diperoleh hasil skor F1 Macro paling tinggi di antara model lainnya, yaitu sebesar 0.75901.



Setelah itu dilakukan visualisasi terhadap *confusion matrix* dari hasil validasi model *voting.* Dapat dilihat bahwa angka *false negative* dan *false positive* relatif sama, angka *true negative* juga sangat mendominasi dibandingkan dengan angka *true positive*. Hal ini dapat disebabkan karena dataset awal yang memang *imbalance.*

Setelah itu, dari salah satu model yang digunakan yaitu XGBoost dihitung Shap values untuk melihat *feature importance* dari kolom-kolom yang digunakan. Berikut merupakan hasil visualisasi dari shap values yang telah dihitung. Dari hasil shap values dapat dilihat bahwa ternyata kolom Umur dan year yang memiliki dampak paling signifikan terhadap klasifikasi apakah pelanggan akan meneruskan asuransi atau tidak. Kemudian terdapat kolom Kelas_Umur dan null_count yang berada pada posisi tengah dan atas, yang menandakan bahwa hipotesis pada *feature engineering* memiliki dampak cukup signifikan terhadap model.



4.2 Classification Report Best Model

Berikut merupakan *classification report* terhadap *validation set*, yang merupakan subset dari *train set* sebesar 15%, oleh model terbaik yaitu model *voting*. Diperoleh hasil F1 Macro tertinggi sebesar 0.75901.

Classification report of Ensemble Method Voting

	precision	recall	f1-score	support
0	0.90186	0.97661	0.93775	47934
1	0.79306	0.45751	0.58027	9390
accuracy			0.89158	57324
macro avg	0.84746	0.71706	0.75901	57324
weighted avg	0.88404	0.89158	0.87919	57324

BAB V
PENUTUP

5.1 Kesimpulan

Setelah dilakukan EDA, preprocessing, model building, serta validation, diperoleh beberapa kesimpulan yang mencakup poin-poin berikut:

1. Dataset yang diperoleh untuk prediksi masih memiliki banyak *missing values*, *imbalanced*, memiliki *outliers*, serta terdapat *data leakage.* Maka dari itu dilakukan berbagai *preprocessing* dan *feature engineering* yang dapat dilihat di bagian Feature Engineering dan Data Preprocessing.
2. Setelah melakukan *training* pada berbagai macam model, yaitu XGBoost, CatBoost, LightGBM, Random Forest, SMOTE-ENN CatBoost, SMOTE-ENN XGBoost, dan Voting, diperoleh model terbaik yang mampu menghasilkan nilai validasi tertinggi yaitu model Voting.
3. Model *Voting* mampu menghasilkan nilai F1 Macro di *validation set* sebesar 0.75901.

DAFTAR PUSTAKA

Geron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, & TensorFlow.* Sebastopol: O'REILLY.

What is Data Leakage? (2021). Retrieved from Force point: https://www.forcepoint.com/cyber-edu/data-leakage#:~:text=Data%20leakage%20is%20the%20unauthorized,is%20transferred%20electronically%20or%20physically.