Tugas Besar
Pembelajaran Mesin 2 : Classification

Kaenova Mahendra Auditama (1301190324) and Moh. Adi Ikfini M. (1301194160)

CII3C3-IF-43-02
Fakultas Informatika, S1 Informatika
Universitas Telkom

1 Pendahuluan

Tugas Besar Klasifikasi pada Mata Kuliah Pembelajaran Mesin (CII3C3-IF-43-02) merupakan tugas kedua dari dua projek tugas yang ada. Pada tugas ini, kami diminta untuk membuat suatu sistem atau model yang dapat memprediksi dan mengklasifikasi dari dataset yang disediakan. Data yang diberikan merupakan data pelanggan terhadap ketertarikan untuk memiliki kendaraan baru. Ada beberapa atribut dalam dataset tersebut, seperti *id, Jenis_Kelamin, Umur, SIM, Kode_Daerah, Sudah_Asuransi, Umur_Kendaraan, Kendaraan_Rusak, Premi, Kanal_Penjualan, Lama_Berlangganan, Tertarik*.

Dengan data-data tersebut kami diminta untuk memprediksi dan mengklasifikasi ketertarikan suatu pelanggan terhadap pembelian mobil dari data-data yang ada. Kami melakukan beberapa percobaan untuk melihat dan mendapatkan hasil yang terbaik. Pendekatan yang digunakan diantaranya Decision Tree, Naive Bayes, dan Artificial Neural Network.

2 Formulasi Masalah

Pada kasus ini kami diberikan suatu dataset terkait ketertarikan pelanggan dengan beberapa atribut-atribut yang sudah disebutkan pada bagian 1. Dari sana, kami diminta untuk memprediksi dan mengklasifikasi suatu data berdasarkan atributnya yang lain dengan menggunakan salah satu teknik dalam pembelajaran mesin *supervised learning*. Data yang diberikan sebesar 285.831 *records*. Dari sana kami bisa mengira akan ada beberapa data yang tidak lengkap ataupun salah, sehingga dibutuhkan pra-pemrosesan data sebelum kami lanjutkan untuk melakukan pemodelan dan klasifikasi.

Ada beberapa pendekatan dalam membuat model pembelajaran mesin yang bertipe *supervised learning* untuk melakukan klasifikasi dan prediksi data. Beberapa model diantaranya ialah *Decisiion Tree, Naive Bayes, Support Vector Machine*, dan masih banyak lagi [Hans, 2020]. Dari beberapa model-model tersebut, kami memilih Decision Tree, Naive Bayes, dan Artificial Neural Network sebagai model yang digunakan dalam melakukan prediksi klasifikasi data-data kami.

3 Eksplorasi Data

Dari 285.831 *records* dan atribut-atribut *id, Jenis_Kelamin, Umur, SIM, Kode_Daerah, Sudah_Asuransi, Umur_Kendaraan, Kendaraan_Rusak, Premi, Kanal_Penjualan, Lama_Berlangganan, Tertarik*, kami

melihat berbagai jenis tipe data. Hanya saja ada beberapa tipe data yang tidak menurut kami asing.

Dengan hal tersebut, kami melakukan eksplorasi yang lebih mendalam kepada beberapa atribut: *Kode_Daerah, Premi, Kanal_Penjualan*. Tetapi sebelum kami mengeksplorasi lebih jauh, kami melakukan pendefinisian jenis variable. Berdasarkan gambar 1 kita dapat melihat bahwa ada data berjenis kualitatif dan kuantitaitf[1].

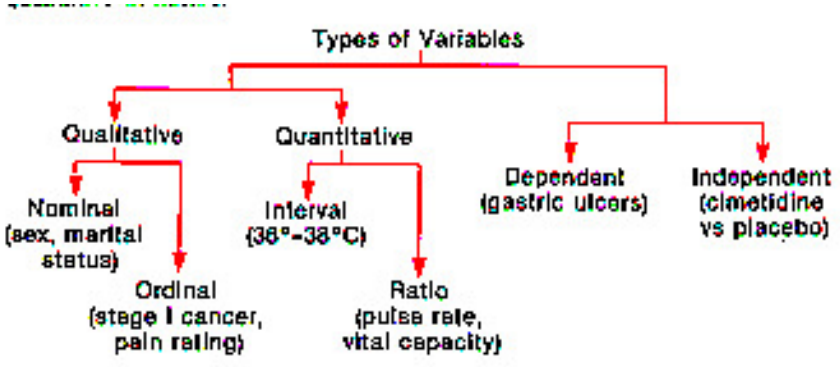


Figure 1: Jenis-jenis variable

Setelah melakukan pendefinisian variable, kami mulai untuk melakukan analisa general dengan atribut-atribut yang sudah pasti, setelah itu melakukan analisa terhadap atribut-atribut yang menurut kami aneh, dan yang terakhir kesimpulan yang bisa diambil dari analisa yang sudah dilakukan sehingga kami mengetahui apa yang harus dilakukan untuk pra-pemrosesan data.

3.1 Analisa Atribut General

Pertama-tama kami melihat data yang belum dilakukan pengolahan. Disini kami diberikan dua data: *train* dan *test*. Kami melihat berapa perbandingan dari target prediksi kami, yaitu perbandingan 0 atau 1 pada kedua data tersebut. Didapatkan pada data *train* dari total 285831 data, yang bernilai positif (1) hanya sekitar 35006 atau 12.25%. Untuk data testingnya pun serupa, dari 47639 data, terdapat 5861 data yang bernilai positif (1) atau sekitar 12.30%. Hal ini menunjukkan bahwa kita mendapatkan data yang *imbalance*.

Setelah dilakukan pemriksaan data, kami mencoba melihat kembali data-data tersebut dan ditemukan beberapa data yang tidak lengkap atau terdapat *NaN* pada salah satu kolomnya, sehingga kami mengatasi hal tersebut dengan cara menghilangkannya. Dengan menghilangkannya, kami melihat bahwa data-data kategorikal lainnya tidak akan menjadi rusak, selain itu kami melakukan ini karena data yang didapatkan walaupun melakukan penghilangan *NaN* masih terbilang banyak, yaitu sekitar 114763 data dan memiliki target atribut yang positif (1) sekitar 14208 atau 12.38%. Disini kami mendapatkan insight baru, walaupun ada beberapa data yang dihapus, tetapi ratio antara target data positif (1) dan negatif (0) tetap menyerupai yaitu disekitar 12%, sehingga bisa dikatakan penghilangan data yang memilik *NaN* tidak terlalu merusak data.

3.2 Analisa Atribut Mendalam

Disini, kami melakukan beberapa anlisa yang lebih mendalam terhadap atribut-atribut yang tidak atau belum kami mengerti.

[1]Gambar dapat dilihat melalui link berikut:https://www.uth.tmc.edu/uth_orgs/educ_dev/oser/L1_2.HTM

3.2.1 Kode_Daerah

Pertama, kami melihat dari nama tribut itu sendiri. Kode daerah ini walaupun pada datanya terlihat memiliki angka *float*, tetapi kami menganggap sebagai kategorikal kualitatif nominal. Untuk melihat tersebut, pertama saya melihat apakah angka-angka apakah memiliki angka tanpa koma semua dengan menggunakan melihat angka-angka yang unik.

```
In [77]: np.sort(df_train_analyze_daerah.unique())

Out[77]: array([ 0.,  1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10., 11., 12.,
                13., 14., 15., 16., 17., 18., 19., 20., 21., 22., 23., 24., 25.,
                26., 27., 28., 29., 30., 31., 32., 33., 34., 35., 36., 37., 38.,
                39., 40., 41., 42., 43., 44., 45., 46., 47., 48., 49., 50., 51.,
                52.])
```

Figure 2: Angka-angka unik pada atribut Kode_Daerah

Berdasarkan gambar 2 kita bisa melihat bahwa tidak ada nilai koma. Hal ini menguatkan bahwa atribut ini merupakan atribut kategorikal nominal.

Selanjutnya kami melakukan analisa bagaimana bentuk data-data yang ada pada data training dan data testing. Dapat terlihat pada gambar 3 distribusi yang ada pada data training dan data testing sangatlah mirip, sehingga kami bisa menyimpulkan atribut ini merupakan atribut kategorikal nominal yang dapat dilakukan *one hot encoding*.

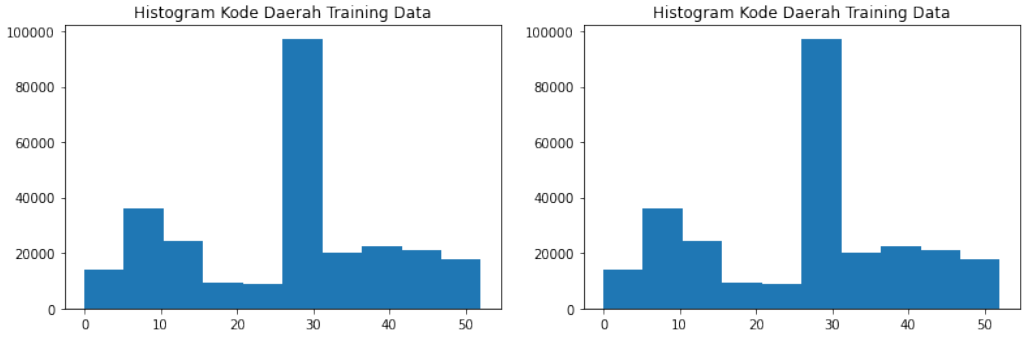


Figure 3: Histogram distribusi Kode_Daerah pada data Training dan Data Testing

3.2.2 Premi

Dalam menganalisa atribut ini, kami mencaritahu apa yang dimaksud Premi itu sendiri, menurut KBBI premi[2] sendiri memiliki beberapa arti, diantaranya:

- n hadiah (uang dan sebagainya) yang diberikan sebagai perangsang untuk meningkatkan prestasi kerja
- n hadiah (dalam undian, perlombaan, pembelian)
- n jumlah uang yang harus dibayarkan pada waktu tertentu kepada asuransi sosial: – asuransi

Berdasarkan KBBI kita bisa mengetahui bahwa atribut ini merupakan atribut kuantitatif. Maka selanjutnya, kami menganggap bahwa atribut ini merupakan atribut kuantitatif.

Selanjutnya kami melihat bagaimana bentuk data premi ini sendiri. Kami melihat menggunakan boxplot untuk melihat persebaran data itu sendiri. Terlihat pada gambar 4 bahwa ada

[2]KBBI terkait Premi dapat dibuka melalui: https://kbbi.kemdikbud.go.id/entri/premi

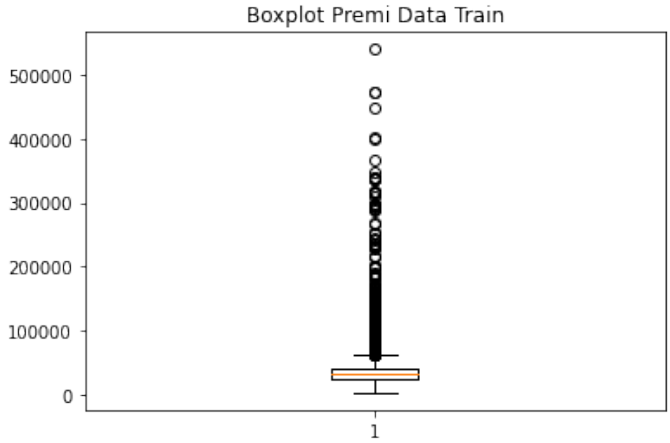


Figure 4: Boxplot Atribut Premi pada Data Train

banyak data *outlier*. Kami sudah mencoba menghilangkan data outler tersebut menggunakan metode *interquartile-range*, tetapi data yang dihilangkan sangatlah banyak, dari 171068 data menjadi 85534, dimana hampir setengah dari data yang ada. Kami memutuskan untuk tidak menggunakan ini sebagai fitur dalam pembuatan model ini.

3.2.3 Kanal_Penjualan

Untuk atribut ini sendiri, kami mengartikan secara harfiah dimana "kanal" merupakan "saluran." Pada umumnya saluran-saluran ini tidak ada pengaruhnya ketika angkanya tinggi ataupun rendah, kami melihatnya bahwa nilai-nilai pada atribut ini adalah hanya kode untuk menamai suatu kanal penjualan, sehingga kami melihatnya atribut ini merupakan berjenis variable kategorikal nominal. Selanjutnya untuk memastikan hal itu, kami juga melihat data apa saja yang ada pada atribut ini.

Terlihat pada gambar 5 bahwa tidak ada angka koma (float), selain itu kami melihat juga bahwa ada beberapa kode yang tidak ada dalam data pada atribut tersebut, terlihat pada list dengan angka [5, 41, 43, 72, 77, 85, 141, 142, 143, 149, 161, 162]. Hal ini sering dijumpai juga ketika kita mendapatkan atribut-atribut bertipe kategorikal nominal.

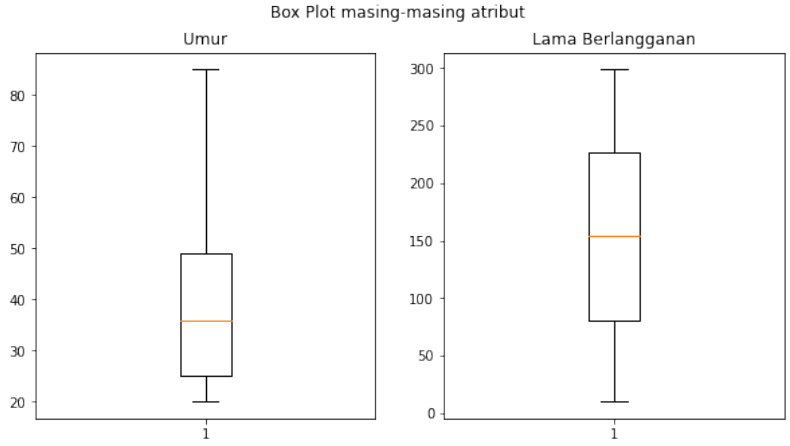
```
In [169]: np.sort(df_train_analyze_kanal.unique())

Out[169]: array([  1.,   2.,   3.,   4.,   6.,   7.,   8.,   9.,  10.,  11.,  12.,
                  13.,  14.,  15.,  16.,  17.,  18.,  19.,  20.,  21.,  22.,  23.,
                  24.,  25.,  26.,  27.,  28.,  29.,  30.,  31.,  32.,  33.,  34.,
                  35.,  36.,  37.,  38.,  39.,  40.,  42.,  44.,  45.,  46.,  47.,
                  48.,  49.,  50.,  51.,  52.,  53.,  54.,  55.,  56.,  57.,  58.,
                  59.,  60.,  61.,  62.,  63.,  64.,  65.,  66.,  67.,  68.,  69.,
                  70.,  71.,  73.,  74.,  75.,  76.,  78.,  79.,  80.,  81.,  82.,
                  83.,  84.,  86.,  87.,  88.,  89.,  90.,  91.,  92.,  93.,  94.,
                  95.,  96.,  97.,  98.,  99., 100., 101., 102., 103., 104., 105.,
                 106., 107., 108., 109., 110., 111., 112., 113., 114., 115., 116.,
                 117., 118., 119., 120., 121., 122., 123., 124., 125., 126., 127.,
                 128., 129., 130., 131., 132., 133., 134., 135., 136., 137., 138.,
                 139., 140., 144., 145., 146., 147., 148., 150., 151., 152., 153.,
                 154., 155., 156., 157., 158., 159., 160., 163.])

In [177]: # Let's check what number that didn't appear in the array
          kanal_unique = np.sort(df_train_analyze_kanal.unique())
          for i in range(1, int(kanal_unique[len(kanal_unique) - 1])):
              if i not in kanal_unique:
                  print(i, end=", ")

          5, 41, 43, 72, 77, 85, 141, 142, 143, 149, 161, 162,
```

Figure 5: Boxplot Atribut Premi pada Data Train

3.3 Analisa Atribut Akhir

Pada akhirnya, kami memiliki dua jenis atribut, pertama berjenis kualitatif dan kuantitatif. Atribut kuantitatif diantaranya *Umur* dan *Lama_Berlangganan* dan atribut kualitatif diantaranya: *Jenis_Kelamin, SIM, Kode_Daerah, Sudah_Asuransi, Umur_Kendaraan, Kendaraan_Rusak, Kanal_Penjualan*. Atribut-atribut ini kami lakukan analisa kembali untuk mengetahui bentuk data serta fitur terbaik untuk dimasukkan ke dalam model kami.

Pada data kuantitatif, untuk melihat baiknya data tersebut, pertama kami cek terlebih dahulu bagaimana bentuk datanya menggunakan boxplot. Terlihat pada gambar 6 bahwa kita tidak perlu melakuka pra-pemrosesan menghilangkan outlier pada atribut ini, dikarenakan pada boxplot nilai outlier tidak terlihat untuk kedua atribut.



(a)

▾ Correlation Numerical - Numerical

Menggunakan Pearson

```
[ ]  1 df_numerical = df_train_mapped[["Umur", "Lama_Berlangganan"]]
     2 df_numerical.corr()
```

	Umur	Lama_Berlangganan
Umur	1.000000	-0.001032
Lama_Berlangganan	-0.001032	1.000000

(b)

Figure 6: (a) Boxplot pada atribut dengan tipe data kuantitatif (b) Korelasi Atribut bertipe numerikal kuantitatif

Selanjutnya kami memeriksa nilai korelasi antar data kuantitatif ini. Dengan menggunakan *Pearson Correlation Coefficient* berdasarkan rumus 1 kita dapat menentukan berapa besar keterhubungan antar atribut ini [sta, 2021]. Terlihat bahwa kedua atribut ini tidak memiliki keterhubungan dengan korelasi diantara atribut tersebut mendekati 0, dimana jika diartikan pada *Pearson Correlation* tidak memiliki korelasi sama sekali

$$\rho = \frac{\text{cov}(X, Y)}{\sigma_x \sigma_y} \tag{1}$$

Setelah memeriksa atribut yang bersifat kuantitatif, selanjutnya kami memeriksa data yang bersifat kualitatif nominal. Disini kami mengambil atribut: *Jenis_Kelamin, SIM, Kode_Daerah,*

Sudah_Asuransi, Umur_Kendaraan, Kendaraan_Rusak, Kanal_Penjualan. Untuk menganalisa atribut-atribut tersebut, kami melakukan penggantian label untuk atribut: *Jenis_Kelamiin, Kendaraan_Rusah, Umur_Kendaraan* menjadi angka.

	Jenis_Kelamin	SIM	Kode_Daerah	Sudah_Asuransi	Umur_Kendaraan	Kendaraan_Rusak	Kanal_Penjualan
Jenis_Kelamin	1.00	0.0	0.01	0.01	0.03	0.01	0.04
SIM	0.00	1.0	0.00	0.00	0.00	0.00	0.00
Kode_Daerah	0.01	0.0	1.00	0.06	0.09	0.05	0.02
Sudah_Asuransi	0.01	0.0	0.06	1.00	0.15	0.68	0.20
Umur_Kendaraan	0.03	0.0	0.09	0.15	1.00	0.16	0.39
Kendaraan_Rusak	0.01	0.0	0.05	0.68	0.16	1.00	0.21
Kanal_Penjualan	0.04	0.0	0.02	0.20	0.39	0.21	1.00

Figure 7: Korelasi Atribut bertipe nominal kualitatif

Selanjutnya kami melihat korelasi dengan menggunakan metode *Crammerś V* pada tipe atribut kualitatif nominal, hal ini dapat dilakukan untuk melihat keterkaitan antar data-data nominal tersebut [Cramer, 2021]. Pada gambar 7 bisa terlihat bahwa nilai korelasi tertinggi pada atribut *Sudah_Asuransi* dan *Kendaraan_Rusak*. Kedua atribut ini kami jadikan sebagai hipotesa fitur terbaik (Best Feature) kami.

4 Pra-Pemrosesan

Berdasarkan hasil eksplorasi yang kami lakukan terhadap data yang diberikan, kami akhirnya melakukan pra-pemrosesan terhadap data yang digunakan. Kami juga memperhatikan jenis atribut yang akan dijadikan fitur. Pada bagian 3.3 kami menentukan atribut kategorikal (kualitatif) dan numerikal (kuantitatif). Adapun juga atribut yang kami tidak akan digunakan, salah satunya ialah *Premi*.

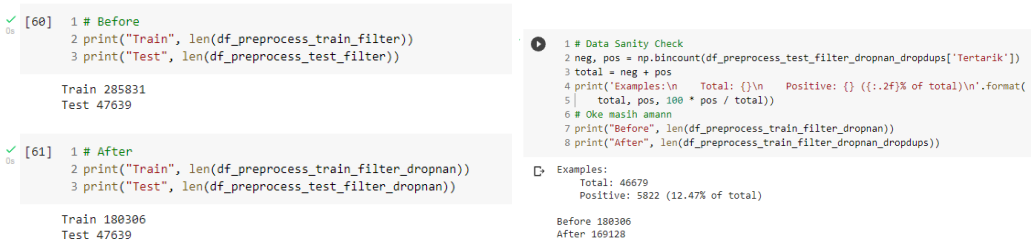
Pada akhir prapemrosesan membagi pemrosesan menjadi 2 bagian, yaitu pra-pemrosesan untuk atribut berjenis kategorikal dan berjenis numerikal. Dengan langkah ini, data yang sebelumnya belum bisa digunakan di dalam model, akan menjadi lebih baik digunakan oleh model. Selain itu, kami juga ingin menlakukan perbandingan terhadap pra-pemrosesan data, sehingga dari data yang diberikan akan menjadi dua data yang siap dimasukkan ke dalam model. Data pertama ialah data yang menggunakan semua fitur, sementara data kedua merupakan data yang menggunakan fitur terbaik, disini hipotesa fitur terbaik kami ialah *Kendaraan_Rusak* dan *Sudah_Asuransi* seperti yang dilihat pada gambar 7.

4.1 Pra-Pemrosesan Utama

Pada bagian ini, kami memasukkan semua data berjenis tipe apapun. Pada tahap ini kami melkaukan *filter* pada fitur yang tidak digunakan. Pada bagian 3.2.2 kami sudah sepakat untuk menghilangkan atribut *Premi*, dikarenakan data yang terdistribusi tidak baik, dan memiliki outlier yang jika dihilangkan akan merusak data serta menghilangkan data sangat banyak.

Setelah menghilangkan atribut tersebut, kami selanjutnya melakukan penghilangan *NaN*. Penghilangan data ini mengikuti aturan, "Jika pada salah satu atribut memiliki *NaN*, maka hilangkan record atau baris tersebut." Seperti yang ada pada gambar 8(a) bahwa dengan melakukan hal ini, tidak terlalu merusak data train, selain itu dengan melakukan ini, kita tidak akan merusak model.

Selanjutnya, kami melakukan drop duplicate. Dengan menghilangkan data yang memiliki duplikat, akan menghilangkan juga bias suatu model terhadap data tertentu. Sehingga dengan melakukan hal ini diharapkan model akan memiliki bias yang sangat kecil terhadap suatu data.



(a) (b)

Figure 8: (a) Jumlah data setelah dilakukan penghilangan *NaN* (b) Hasil jumlah data setelah dilakukan penghilangan data duplikat pada data train

Setelah kami melakukan drop duplicate, kami melakukan *data sanity check*. Dapat dilihat pada 8(b) bahwa hasil yang didapatkan memiliki target ratio yang serupa saat kita melakukan eksplorasi data pada bagian 3.1, dimana dengan melakukan pra-pemrosesan data tersebut, kami berhasil untuk tidak merusak data.

Pada akhir pra-pemrosesan atribut numerikal dan kategorikal, kami lakukan oversampling menggunakan metode SMOTE. Hal ini dilakukan dikarenakan karena data yang didapatkan tidak balance, sehingga jika dilakukan train akan condong pada label yang major. Dengan menggunakan SMOTE, kami tidak hanya menduplikasi data-data yang manoritas, tetapi kami membentuk data-data baru dengan pendekatan nearest neighbor[Chawla et al., 2002]. Tetapi perlu diperhatikan bahwa karana oversampling merupakan penambahan data terhadap label yang minortias, kami mencoba untuk tetap mempertahankan tingkat minoritasnya itu sehingga melakukan penambahan data minoritas menjadi 44% dari total data.

4.2 Pra-Pemrosesan Atribut Numerikal

Pada data yang kita miliki, terdapat atribut bertipe numerikal yaitu *Umur* dan *Lama_Berlangganan*. Dengan data yang akan kami gunakan untuk melakukan pemodelan fitur penuh, kami harus melakukan *feature scaling* pada kedua atribut ini. Menggunakan library dari *sckit-learn* dengan MinMaxScaler kami melakukan *feature scaling* pada data train dan data test. Hal ini dilakukan untuk mempercepat pemrosesan dalam perhitungan, serta menormalkan data-data yang ada pada atribut-atribut tersebut.

4.3 Pra-Pemrosesan Atribut Kategorikal

Dari atribut-atribut yang ada, banyak atribut yang berjenis kategorikal nominal. Terlihat bahwa atribut-atribut tersebut masih dalam bentuk *Ordinal Coding*. Kelemahan dari Ordinal Coding adalah kurangnya ekspresi kode tersebut didalam mesin, sehingga kami melakukan *One Hot Encoding* pada atribut-atribut tersebut. Dengan *One Hot Encoding*, kode-kode yang ada pada atribut kategorikal akan menjadi lebih "ekspresif" dan lebih independen. Selain itu, melakukan *One Hot Encoding* dapat meningkatkan akurasi yang lebih tinggi dibandingkan dengan *Ordinal Coding* [Potdar et al., 2017]. Tetapi pada atribut kategorikal yang data awalnya berbentuk dari sebuah text, kami ubah menjadi representasi angka, atribut diantaranya ialah Jenis_Kelamin, Kendaraan_Rusak, dan Umur_Kendaraan, dimana atribut Umur_Kendaraan dilakukan Ordinal Encoding.

Dalam pemrosesan atribut kategorikal ini, atribut *Kanal_Penjualan* harus diperhatikan, dalam eksplorasi data, kami menemukan bahwa data pada training tidak serupa dengan data testing. Sehingga dalam melakukan *One Hot Encoding*, kami harus membuat kolom sendiri berdasarkan data

yang tidak ada tersebut. Seperti yang tertera pada gambar 9 bahwa ada beberapa *Kanal_Penjualan* yang tidak terdapat dalam data train dan data testing atau sebaliknya.

```
[244]  1 a = np.sort(df_preprocess_train_scaling["Kanal_Penjualan"].unique()).tolist()
       2 b = np.sort(df_preprocess_test_scaling["Kanal_Penjualan"].unique()).tolist()
```

```
[245]  1 not_in_a = []
       2 not_in_b = []
       3 for kanal in set(a+b):
       4     if kanal not in a:
       5         not_in_a.append(kanal)
       6     if kanal not in b:
       7         not_in_b.append(kanal)
       8
       9 print(not_in_a)
      10 print(not_in_b)
      11 print(len(not_in_a))
      12 print(len(not_in_b))
```

```
[33, 104, 112]
[2, 6, 17, 28, 50, 57, 69, 71, 74, 75, 82, 84, 100, 101, 110, 117, 123, 134, 137]
3
19
```

Figure 9: Hasil data yang tidak lengkap pada atribut *Kanal_Penjualan* dari data train ke data test dan sebaliknya

5 Pembangunan Model (Pemodelan)

5.1 Metrics yang Digunakan

Sebelum kami melakukan seleksi model-model yang akan digunakan, kami menentukan metric-metric yang digunakan untuk mengkuantisasikan suatu model merupakan model yang baik ataupun tidak. Pada tugas yang diberikan, kami diminta untuk melakukan pengecekkan terhadap data testing menggunakan metric akurasi. Dimana akurasi merupakan rasio prediksi berapa banyak model dapat menentukan hasil yang tepat yang dapat dijelaskan pada rumus 2, dimana T dan F merupakan *True* dan *False*, serta P dan N merupakan *Positive* dan *Negative*.

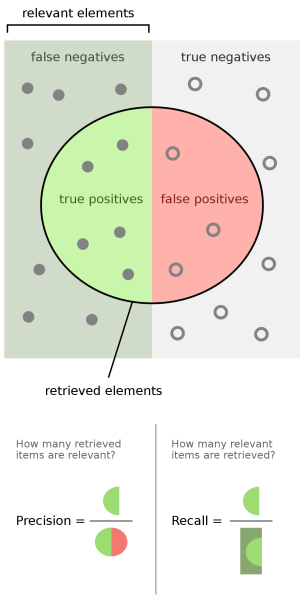


Figure 10: Presisi dan Recall[3]

$$\text{Akurasi} = \frac{\text{TP} + \text{FP}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{2}$$

Selain akurasi, kami juga mempertimbangkan imbalance data train ini. Karena data yang tidak balance ini akan menyebabkan akurasi yang bias. Kami menggunakan metric presisi dan recall. Recall dapat memberikan kita informasi seberapa benar suatu prediksi mesin pada domain yang sama. Hal ini sangat membantu ketika kita memiliki data yang imbalance dikarenakan kita bisa melihat seberapa benar prediksi terhadap domain yang sama. Sementara, presisi bisa memberikan kita informasi terhadap hasil dari data prediksi tersebut. Hal ini dapat digambarkan pada gambar 10. Selain itu juga, dari data yang disediakan, karena imbalance kita akan lebih baik jika bisa memprediksi yang aslinya memiliki nilai 1 (*Tertarik*) terprediksi benar bernilai 1. Hal ini dapat dilihat dari model bisnis yang menyerupai pada data yang ada, dimana kita harus bisa memprediksi benar bahwa orang yang tertarik benar-benar dapat terprediksi tertarik. Selain itu untuk melihat secara keseluruhan dari presisi dan recall, kami menggunakan F1 Score.

[3]Gambar dapat diunduh mellaui https://en.wikipedia.org/wiki/File:Precisionrecall.svg

5.2 Seleksi Model

Beberapa model telah kami pilih untuk melakukan prediksi dari data yang siapkan. Dimulai dari model mesin yang sederhana, hingga kompleks. Kami memilih Decision Tree, Naive Bayes[4], dan Artificial Neural Network sebagai pilihan model yang kami bandingkan. Model-model tersebut kami pilih berdasarkan pendekatan yang digunakan setiap model untuk menyelesaikan masalah untuk memprediksi ketertarikan pelanggan secara tepat dan diambil model yang dapat menyelesaikan masalah secara efektif. Untuk model Naive Bayes sendiri, kami membuatnya secara manual. Algoritma

Dengan decision tree dan naive bayes, keduanya menggunakan pendekatan statistikal dan probabilitas untuk meyelesaikan permasalahan tersebut. Kedua model ini memiliki pendekatan yang sama, tetapi memiliki cara pemecahan masalahnya tersendiri. Dimulai dengan decision tree dimana ia melihat dari struktur data yang ada dan mencari suatu nilai yang dapat berdampak untuk bisa memprediksi dengan tepat. Sementara, naive bayes melihat dengan pendekatan probabilitas terhadap data-data yang ada serta mengukur berat dari setiap nilai untuk dapat memberikan hasil yang tepat.

Menggunakan Artificial Neural Network merupakan salah satu *state of the art* dalam pemecahan masalah yang kompleks. Kami mencoba model ini berdasarkan ide tersebut untuk menyelesaikan permasalahan memprediksi terhadap data-data yang ada. Artificial Neural Network menggunakan konsep regresi kompleks untuk menyelesaikan suatu masalah. Dengan menambah *hidden layer* pada model tersebut, dapat membantu untuk menyelesaikan masalah yang lebih kompleks karena parameter yang lebih banyak sehingga dapat dilakukan pendapatan parameter yang sangat presisi dan tepat.

5.3 Hasil dan Diskusi

Model	Metrics					
	Akurasi	W. Precision	W. Recall	W. F1 Score	Precision (Tertarik)	Recall (Tertarik)
"Hyphothesis" Best Feature						
Decision Tree	0.64	0.99	0.64	0.70	0.25	0.98
Gaussian NB	0.64	0.99	0.64	0.70	0.25	0.98
ANN	0.64	0.99	0.64	0.70	0.25	0.98
All Feature						
Decision Tree	**0.82**	0.83	**0.82**	**0.82**	**0.29**	0.30
ANN	0.73	**0.89**	0.73	0.77	**0.29**	**0.88**

Table 1: Performa Data Testing pada setiap Model (NB = Naive Bayes, ANN = Artificial Neural Network, W. = Weighted)

Berdasarkan tabel 1 yang dibuat berdasarkan luaran model yang telah kami coba[5]. Kita bisa lihat bahwa nilai-nilai pada data yang sebelumnya dikatakan sebagai hipotesa fitur terbaik pada bagian 3.3 memiliki nilai recall dan precision yang serupa. Menurut kami, hal ini terjadi kerena nilai kemungkinan kombinasi yang sangat sedikit untuk model dapat pelajari, sehingga ketiga model tersebut dapat memberikan nilai-nilai yang serupa.

Pada hasil-hasil model tersebut kami menekankan metrics menggunakan metode *Weighted*. Hal ini diperlukan karena data testing yang ada juga tidak balance, sehingga jika kita menggunakan nilai asli dari setiap metrics akan tidak sama. Selain itu, kami juga melihat dari nilai F1 Score.

[4]Kode model Naive Bayes dapat dilihat melalui https://colab.research.google.com/drive/1ohoNu0x_BJRmOgpDVl69MOEucA3RXOFr?usp=sharing

[5]Hasil luaran model dapat dilihat malalui: https://github.com/kaenova/static_raw_file/blob/main/Malin2_Output_ModelSelection.txt

Pada tabel 1, kami menekankan Precision dan Recall dari label 1 (Tertarik) hal ini digunakan untuk menggambarkan seberapa baik model dapat memprediksi yang benar memprediksi tertarik.

Pada data Best Feature, kita bisa melihat bahwa nilai-nilai yang dihasilkan model sama. Ini menunjukkan sedikitnya kemungkinan yang ada jika kita gunakan fitur tersebut. Tetapi kita bisa melihat seberapa baik hasil yang diberikan dari model-model tersebut. Pada hasil ini, terlihat akurasi tidak terlalu baik pada ketiga model tersebut, tetapi nilai recall untuk melabeli tertarik bisa mencapai 0.98. Ini artinya bahwa model bisa memprediksi dan mengklasifikasi secara benar untuk label tertarik. Perlu diperhatikan pada nilai precision untuk nilai tertarik ini memiliki angka yang cukup rendah karena data yang ada tidaklah balance. Karena jumlah data pada label tertarik pada data testing sedikit dan label tidak tertarik banyak seperti yang disebutkan dalam 3.1. Kesalahan dalam memprediksi dari label yang tidak tertarik akan lebih banyak. Hal ini yang menyebabkan nilai precision menjadi kecil. Dengan hal tersebut, untuk kasus memprediksi dan mengklasifikasi orang yang tertarik secara tepat dengan menggunakan fitur-fitur tersebut sudah cukup. Ketiga model dapat memprediksi serta mengklasifikasi dengan tepat orang yang tertarik.

Pada data All Feature, kita bisa melihat berbagai variansi nilai yang dihasilkan oleh kedua model. Kita bisa melihat kekurangann dari decision tree yang tidak bisa memberikan hasil yang tepat untuk memprediksi benar orang yang tertarik. Hal ini terlihat dari nilai Recall yang dibandingkan dengan Artificial Neural Network. Pada kasus ini, Artificial Neural Network yang sebelumnya sering dikatakan sebagai *state of the art* dapat menyelesaikan masalah dengan dimensi yang lebih tinggi dibandingkan decision tree. Serta bisa menghasilkan nilai recall yang lebih tinggi.

6 Kesimpulan

Pada Tugas Besar Klasifikasi Mata Kuliah Pembalajaran Mesin (CII3C3-IF-43-02) merupakan tugas dimana kita diharuskan bisa mengklasifikasi orang yang tertarik dan yang tidak. Pada laporan ini dapat ditunjukkan bahwa ada beberapa masalah dalam melakukan prediksi terhadap ketertarikan orang dikarenakan data yang tidak begitu baik. Keterbatasan utama adalah pada data. Data yang tidak seimbang dengan perbandingan 1 kelas dengan kelas yang lain ialah 1/10. Dengan data yang tidak baik kami mencoba untuk membandingkan beberapa model: Decision Tree, Naive Bayes, dan Artificial Neural Network sebagai tolak ukur pemilihan fitur serta pemilihan model terbaik.

Dari hasil pemodelan yang sudah dijelaskan pada bagian 5.3, pada akhirnya kami tidak bisa mengukur hasil model kami dengan metrics akurasi karena data yang tidak balance. Sehingga kami menggunakan recall terhadap label tertarik sebagai tolak ukur. Hal ini dikarenakan akan lebih baik jika kita bisa melakukan prediksi yang benar terhadap orang yang tertarik. Hanya dengan menggunakan beberapa atribut (Kendaraan_Rusak dan Sudah_Asuransi) yang sudah kami pilih menjadi Best Feature, model Decision Tree, Gaussian Naive Bayes, ataupun Artificial Neural Network dapat memprediksi benar yang tertarik. Tetapi untuk keefisienan, model Naive Bayes sudah cukup karena pemrosesan waktu yang cepat dengan F1 Score 0.70 dan akurasi 0.64. Sehingga bisa kita temukan bahwa kita tidak perlu model yang yang rumit atau data yang berdimensi besar untuk menyelesaikan permasalahan klasifikasi ini.

Lampiran utama [Video Penjelasan, Google Colab, dan Hasil Klasifikasi] dapat diakses melalui
https://kaenova-link.pages.dev/school/malin_tubes2
atau
https://bit.ly/KaenovaMalinTubes2

References

[sta, 2021] (2021). Correlation coefficient: Simple definition, formula, easy steps.

[Chawla et al., 2002] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

[Cramer, 2021] Cramer, H. (2021). *Mathematical Methods of Statistics*.

[Hans, 2020] Hans, R. (2020). Jenis-jenis algoritma supervised machine learning.

[Potdar et al., 2017] Potdar, K., S., T., and D., C. (2017). A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers. *International Journal of Computer Applications*, 175(4):7–9.