Mata Kuliah - Penggalian Data

Nama Kelompok :

Anggota :

[202110370311029 - Andi Aswad]

[202110370311047 - Muhammad Daffa]

[202110370311334 - Umi Nursyafika]

Berikut ini merupakan update template laporan Mini Project kuliah Penggalian Data.

Nilai Total: 120 poin

Tahap 0 (poin: 25): Business Objective

Memberikan panduan kepada calon pengusaha atau pemilik usaha kost untuk menentukan harga kos yang sesuai berdasarkan fasilitas.

Tahap 1 (poin: 25): Original Data

- Urgensi topik/kasus yang dipilih.
    - Memudahkan dan memberikan referensi bagi pelaku usaha kos dalam menyediakan fasilitas serta memberi harga yang sesuai agar banyak peminat.
- Data yang digunakan.
    - Deskripsi singkat
        - Data yang kami gunakan adalah data primer dan sekunder, yang dimana data primer tersebut dikumpulkan dari aplikasi mamikos, dan data sekunder diambil dari kaggle.
    - Sebutkan dan jelaskan atribut pada data tersebut.
      Pada data kami terdapat 7 atribut yaitu:
        - Nama Kos : Untuk mengidentifikasi setiap kos
        - Jenis kos : Yang membedakan penghuni kos berdasarkan gender, misalnya untuk putri atau putra ataupun campur
        - Fasilitas Kamar: deskripsi dari fasilitas yang tersedia pada kamar kos
        - Fasilitas Umum: deskripsi fasilitas yang tersedia dan dapat digunakan bersama oleh seluruh penghuni kos
        - Harga: Tarif sewa kos

    - Rating: Penilaian yang diberikan oleh penghuni kos pada ratting dibagi menjadi ratting kebersihan, kenyamanan, keamanan, harga, fasilitas umum fasilitas kamar, total ratting
    - Jarak ke kampus: jarak dari kos ke kampus yang terdekat
  - Jelaskan data mining task yang akan digunakan (*classification, clustering, regression, association rule mining, anomaly detection,* dsb.)

    Dalam studi kasus ini, data mining task yang digunakan adalah classification. Pada kasus ini, kami menggunakan klasifikasi untuk menghubungkan beberapa variabel dan mengelompokkan variabel tertentu guna memprediksi variabel output, yaitu kategori harga kos (murah, sesuai, atau mahal). Dengan menggunakan data historis dan fitur-fitur yang relevan, seperti fasilitas dan harga kos, kami melatih model klasifikasi untuk memahami pola hubungan antara variabel-variabel ini. Hasilnya, model ini dapat memprediksi kategori harga dari kos-kosan baru berdasarkan atribut yang diberikan, membantu dalam menentukan apakah suatu kos termasuk dalam kategori harga murah, sesuai, atau mahal.
- Sumber data (sertakan link).
    - Primer : Aplikasi/web Mamikos (https://docs.google.com/spreadsheets/d/1qU6FbC3c_HIJ5JzRPmn-EGUNEtfYa0W_/edit?usp=drive_link&ouid=116204949216856815064&rtpof=true&sd=true)
    - sekunder : kaggle (https://www.kaggle.com/datasets/wirantomillennium/mamikostdataset)

Tahap 2 (poin: 10): Target Data (Optional)

- Target data yang digunakan mengahapus beberapa atribut yang ada pada original data. Atribut yang digunakan pada target data sebagai berikut:
  - Harga kos
  - Fasilitas kamar
  - Fasilitas Umum
  - Total rating

Tahap 3-4 (poin: 25): Data Pre-processing & Transformation

Berikut adalah beberapa teknik yang kami terapkan dalam studi kasus ini:

- Data Cleaning (pembersihan data):
  - Kami melakukan pembersihan data untuk menghapus baris yang mengandung nilai yang hilang atau kosong (NaN). Hal ini dilakukan agar data yang digunakan dalam analisis tidak terpengaruh oleh ketidaksesuaian atau ketidakkonsistenan data.
  - Selain itu, kami juga melakukan perubahan format pada data harga untuk memastikan konsistensi dalam representasi numeriknya. Misalnya, kami bisa mengonversi harga menjadi format numerik standar atau melakukan normalisasi.

    - Dan juga menghapus kolom yang tidak relevan atau tidak termasuk dalam target data, sehingga hanya menyisakan fitur-fitur yang paling penting dan memiliki dampak signifikan terhadap prediksi harga kos-kosan.

- Feature Engineering (rekayasa fitur):
    - Kami melakukan rekayasa fitur dengan menggabungkan beberapa kolom yang memiliki keterkaitan atau kesamaan dalam konteksnya. Misalnya, kami menggabungkan kolom fasilitas umum dan fasilitas kamar menjadi satu fitur tunggal yang disebut "Fasilitas". Hal ini memungkinkan kami untuk lebih memahami pengaruh dari berbagai jenis fasilitas terhadap harga kos-kosan.
    - Kami juga menambahkan fitur tambahan, seperti fitur outcome untuk pengelompokan harga kos-kosan menjadi kategori harga (murah, sesuai, atau mahal), yang akan menjadi target prediksi dalam model klasifikasi kami.

- Feature Selection (pemilihan fitur):
    - Setelah melakukan rekayasa fitur, kami memilih fitur-fitur yang paling relevan dan memiliki dampak yang signifikan terhadap prediksi harga kos-kosan. Proses ini membantu mengurangi dimensi data dan meningkatkan efisiensi pemodelan.
    - Kami memilih 4 fitur yang kami anggap paling penting dalam studi kasus ini, seperti harga, fasilitas kamar, fasilitas umum, dan total ranting. Fitur-fitur lain yang kurang relevan atau memiliki korelasi yang rendah dengan target data dihapus.

- Data Integration (integrasi data):
    - Kami menggabungkan dua dataset dari sumber berbeda untuk meningkatkan keragaman dan kualitas data yang digunakan dalam analisis. Proses integrasi ini memungkinkan kami untuk memperoleh wawasan yang lebih komprehensif dan akurat tentang pola-pola yang ada dalam data kos-kosan.

Dengan menerapkan teknik-teknik ini secara sistematis, kami dapat mengoptimalkan kualitas data, mengurangi dimensi data, dan meningkatkan relevansi fitur-fitur yang digunakan dalam model klasifikasi kami, sehingga menghasilkan prediksi harga kos-kosan yang lebih akurat.

Berikut hasil data kami setelah dilakukan pre-processing dan transformasi :

combined_data_final_terbaru.csv

	A	B	C	D
1	Harga	Rating	Fasilitas	Outcome
2	802000	4.5	0	2
3	913000	5	0	2
4	813750	5	0	2
5	850000	4.9	0	2
6	700000	4.8	0	2
7	2600000	5	3	2
8	600000	5	0	1
9	1875000	4.9	2	1
10	1350000	5	2	1
11	1275000	5	1	2
12	2525000	5	2	2
13	1525000	4.9	2	2
14	2375000	4.6	3	2
15	2425000	4.8	3	2
16	2175000	4.6	3	2
17	800000	5	2	0
18	650000	5	0	1
19	913000	5	0	2
20	813750	5	0	2
21	850000	4.9	0	2
22	700000	4.8	0	2
23	2600000	5	2	2
24	600000	5	0	1
25	700000	1.3	2	0
26	1000000	4.3	0	2
27	850000	4.9	0	2
28	1400000	4.9	2	1
29	1650000	5	2	2
30	750000	5	0	2
31	750000	4	1	1

Tahap 5 (poin: 25): Data Mining

- Algoritma data mining yang digunakan (sesuai data mining task).

  Penelitian ini menggunakan 5 model algoritma Machine Learning, dan pada setiap algoritma terdapat beberpa percobaan dalam penerapan modelnya yaitu pada data original, data resampling, dan penerapan feature selection. Berikut penjelasan lebih lengkap pada tiap model algoritma:

- Random Forest:

  Random Forest terdiri dari banyak pohon keputusan yang dihasilkan secara acak, yang kemudian digabungkan untuk membuat prediksi. Ini membantu dalam menghindari overfitting dan menghasilkan model yang lebih umum. Algoritma ini juga mampu menangani kumpulan data besar dengan fitur yang bervariasi dengan baik, sehingga cocok untuk dataset kos-kosan yang kompleks. Karena sifat ensemble-nya, Random Forest dapat mengidentifikasi pola yang rumit dalam data dan memberikan prediksi yang akurat. Sehingga, model ini cukup baik dalam memprediksi kategori harga kos-kosan.

  Dari beberapa penerapan yang digunakan pada model random forest, yang memiliki hasil nilai akurasi tertinggi adalah pada penerapan Model di data Resampling yaitu dengan hasil akurasi 0.96

  Rumus Matematis yang digunakan Pada Model random Forest:

  a. Pembentukan Sub-sampel (Bootstrap Sampling)
     Untuk setiap pohon dalam hutan, kita membentuk subset bootstrap dari dataset asli D yang memiliki n contoh data. Misalkan dataset asli D terdiri dari fitur X dan target y:

     $$\boldsymbol{D'} = \{(\boldsymbol{X'_1}, \boldsymbol{y'_1}), (\boldsymbol{X'_2}, \boldsymbol{y'_2},)$$

     Dimana (X'_i, y'_i) dipilih secara acak dengan pengembalian dari D
  b. Pemilihan Fitur Acak (Random Feature Selection
     Pada setiap node dalam keputusan, kita memilih subset acak dari fitur. Misalkan kita memiliki p fitur total, kita memilih k fitur acak:

     $$\boldsymbol{F_i} = \{\boldsymbol{f_1 f_2}, \dots, \boldsymbol{f_k}\}$$

     Nilai k biasanya $\sqrt{p}$ untuk Klasifikasi
  c. Kriteria Pemisahan (Split Criterion)
     Salah satu kriteria pemisahan yang umum dan default pada random forest adalah Gini impurity

     $$\boldsymbol{I_{gini}} = \mathbf{1} - \sum_{\boldsymbol{i=1}}^{\boldsymbol{C}} \boldsymbol{p_i^2}$$

     Keterangan :

     p_i adalah proporsi contoh dari kelas i di node tersebut.

     C adalah jumlah total kelas

     Untuk setiap pemisahan yang mungkin, impurity total setelah pemisahan dihitung sebagai berikut:

     $$\boldsymbol{Gini_{split}} = \frac{\boldsymbol{n_{left}}}{\boldsymbol{n}} \boldsymbol{I_{gini}}(\boldsymbol{D_{left}}) + \frac{\boldsymbol{n_{right}}}{\boldsymbol{n}} \boldsymbol{I_{gini}}(\boldsymbol{D_{right}})$$

Keterangan :

n_{left} dan n_{right} adalah jumlah contoh di bagian kiri dan kanan setelah pemisahan.

D_{left} dan D_{right} adalah subset data setelah pemishan

d. Pembangunan pohon keputusan

Proses pemilihan fitur dan titik pemisahan berlanjut secara rekursif untuk setiap node hingga salah satu kondisi berikut terpenuhi:

- Pohon mencapai kedalaman maksimum.
- Node tidak dapat dibagi lebih lanjut (misalnya, semua contoh dalam node memiliki label yang sama).

e. Penggabungan Prediksi (Ensemble Voting)

Misalkan kita memiliki m pohon, dan setiap pohon h_i memberikan prediksi kelas $h_i(x)$ untuk input x. Prediksi akhir adalah kelas yang paling sering diprediksi:

$$\hat{y} = \boldsymbol{mode}\ (\{\boldsymbol{h_1}(\boldsymbol{x}), \boldsymbol{h_2}(\boldsymbol{x}), \dots, \boldsymbol{h_m}(\boldsymbol{x})\})$$

Perhitungan matrix berdasarkan confusion matrix:

Kelas 0

TP : 161
FP : 2 (kelas 1) + 0 (kelas 2) = 2
FN : 4 (kelas 1) + 1 (kelas 2) = 5

$$Precission\ 0 = \frac{TP0}{TP0 + FP0} = \frac{161}{161 + 2} = \frac{161}{163} = 0{,}9877$$

$$Recall\ 0 = \frac{TP0}{TP0 + FN0} = \frac{161}{161 + 5} = \frac{161}{166} = 0{,}9699$$

$$F1\ Score\ 0 = 2 \times \frac{Precission\ 0 \times Recall\ 0}{Precission\ 0 + Recall\ 0} = \frac{0{,}9877 \times 0.9699}{0{,}9877 + 0{,}9699} = 0{,}9787$$

Kelas 1

TP1 : 176
FP1 : 4 (kelas 0) + 5 (kelas 2) = 9
FN1 : 2 (kelas 0) + 6 (kelas 2) = 8

$$Precission\ 1 = \frac{TP1}{TP1 + FP1} = \frac{176}{176 + 9} = \frac{175}{185} = 0{,}9514$$

$$Recall\ 1 = \frac{TP1}{TP1 + FN1} = \frac{176}{176 + 8} = \frac{176}{184} = 0{,}9565$$

$$F1\ Score\ 1\ = 2 \times \frac{Precission\ 1\ \times Recall\ 1}{Precission\ 1 + Recall\ 1} = \frac{0{,}9514 \times 0.9565}{0{,}9514 + 0{,}9565}$$
$$= 0{,}9539$$

Kelas 2

TP2 : 152
FP2 : 1 (kelas 0) + 6 (kelas 1) = 7
FN2 : 5 (kelas 1)

$$Precission\ 2 = \frac{TP2}{TP2 + FP2} = \frac{152}{152 + 7} = \frac{152}{159} = 0{,}9566$$

$$Recall\ 2 = \frac{TP2}{TP2 + FN2} = \frac{152}{152 + 5} = \frac{152}{157} = 0{,}9682$$

$$F1\ Score\ 2\ = 2 \times \frac{Precission\ 2\ \times Recall\ 2}{Precission\ 2 + Recall\ 2} = \frac{0{,}9566 \times 0.9682}{0{,}9566 + 0{,}9682}$$
$$= 0{,}9624$$

$$\boldsymbol{Akurasi} = \frac{Jumlah\ Prediksi\ Benar}{Jumlah\ Total\ Data} = \frac{TP0+TP1+TP2}{507} = \frac{161+176+152}{507} = \frac{489}{507} = 0{,}9654$$

- Decision Tree:

  Decision Tree menghasilkan model berbentuk pohon keputusan, yang mudah diinterpretasikan oleh manusia. Ini membantu dalam memahami faktor-faktor yang mempengaruhi keputusan klasifikasi. Algoritma ini mampu menangani data dengan baik bahkan tanpa persyaratan pengolahan yang rumit. Hal ini membuatnya cocok untuk dataset yang tidak terlalu besar atau rumit. Decision Tree cenderung cepat dalam pembentukan model dan prediksi, sehingga cocok untuk digunakan dan menunjukkan bahwa model ini sangat baik dalam memprediksi kategori harga kos-kosan.

  Dari beberapa penerapan yang digunakan pada model Decission Tree, yang memiliki hasil nilai akurasi tertinggi adalah pada penerapan Model di data Resampling yaitu dengan hasil akurasi 0.96

Rumus Matematis yang digunakan Pada Model Decission Tree:

a. Entropy digunakan untuk mengukur impurity dalam sebuah dataset. Rumus yang digunakan:

$$\boldsymbol{H(D) = -\sum_{i=1}^{c} p_i \, log_2 \, (P_i)}$$

Keterangan:
$H(D)$ = Entropy dari dataset D
c = Jumlah kelas
p_i = Proporsi sampel yang termasuk kelas i dalam dataset D

b. Gini impurity juga digunakan untuk mengukur impurity dalam dataset. Rumusnya adalah:

$$\boldsymbol{G(D) = 1 - \sum_{i=1}^{c} p_i^2}$$

Keterangan:
$G(D)$ = Gini impurity dari dataset D
c = Jumlah kelas
p_i = Proporsi sampel yang termasuk kelas i dalam dataset D

c. Information Gain (IG) digunakan untuk menentukan fitur terbaik untuk membagi dataset. Rumusnya adalah:

$$\boldsymbol{IG(D, A) = H(D) - \sum_{v \in values(A)} \frac{|D_v|}{|D|} H(D_v)}$$

Keterangan:
$IG(D, A)$: Information Gain dari pemisahan dataset D berdasarkan fitur A
$H(D)$: Entropy dari dataset D
v : Nilai unik dari fitur A
D_v : Subset dari D dimana fitur A memiliki nilai υ
$|D_v|$: Jumlah sampel dalam subset $D_υ$
$|D|$: Jumlah sampel dalam dataset D

d. Split Criterion (Kriteria pemisah)

Untuk setiap fitur, kita membagi dataset ke dalam dua subset berdasarkan nilai tertentu dari fitur tersebut dan menghitung impurity total setelah pemisahan. Contohnya, jika kita menggunakan entropy:

$$\boldsymbol{H_{split}\ (D, A) = \sum_{v \in values(A)} \frac{|D_v|}{|D|} H(D_v)}$$

Keterangan:

- H_{split} (D, A) : Entropy total setelah dataset D dibagi berdasarkan fitur A
- $\frac{|D_v|}{|D|}$: sampel yang termasuk subset $D_υ$ terhadap total sampel dalam dataset D
- $H(D_v)$: Entropy dari subset $D_υ$

Perhitungan matrix berdasarkan confusion matrix:

Kelas 0

TP : 161
FP : 2 (kelas 1) + 0 (kelas 2) = 2
FN : 4 (kelas 1) + 1 (kelas 2) = 5

$$Precission\ 0 = \frac{TP0}{TP0 + FP0} = \frac{161}{161 + 2} = \frac{161}{163} = 0{,}9877$$

$$Recall\ 0 = \frac{TP0}{TP0 + FN0} = \frac{161}{161 + 5} = \frac{161}{166} = 0{,}9699$$

$$F1\ Score\ 0 = 2 \times \frac{Precission\ 0 \times Recall\ 0}{Precission\ 0 + Recall\ 0} = \frac{0{,}9877 \times 0.9699}{0{,}9877 + 0{,}9699} = 0{,}9787$$

Kelas 1

TP : 176
FP : 4 (kelas 0) + 6 (kelas 2) = 10
FN : 2 (kelas 0) + 6 (kelas 2) = 8

$$Precission\ 1 = \frac{TP1}{TP1 + FP1} = \frac{176}{176 + 10} = \frac{176}{186} = 0{,}9462$$

$$Recall\ 1 = \frac{TP1}{TP1 + FN1} = \frac{176}{176 + 8} = \frac{176}{184} = 0{,}9565$$

$$F1\ Score\ 1 = 2 \times \frac{Precission\ 1 \times Recall\ 1}{Precission\ 1 + Recall\ 1} = \frac{0{,}9462 \times 0.9565}{0{,}9462 + 0{,}9565} = 0{,}9513$$

Kelas 2

TP : 151
FP : 1 (kelas 0) + 6 (kelas 1) = 7
FN : 6 (kelas 1)

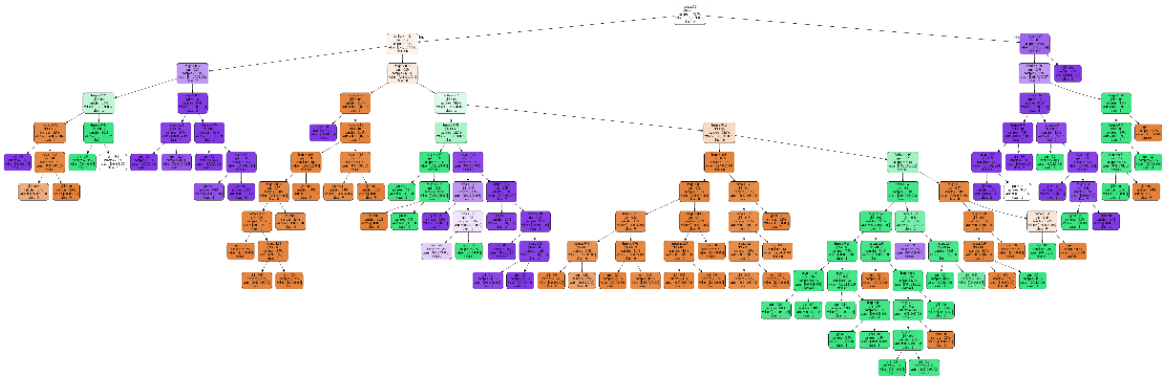
$$Precission\ 2 = \frac{TP2}{TP2 + FP2} = \frac{151}{151 + 7} = \frac{151}{158} = 0{,}9557$$

$$Recall\ 2 = \frac{TP2}{TP2 + FN2} = \frac{151}{151 + 6} = \frac{151}{157} = 0{,}9618$$

$$F1\ Score\ 2 = 2 \times \frac{Precission\ 2 \times Recall\ 2}{Precission\ 2 + Recall\ 2} = \frac{0{,}9557 \times 0.9618}{0{,}9557 + 0{,}9618} = 0{,}9588$$

$$\boldsymbol{Akurasi} = \frac{Jumlah\ Prediksi\ Benar}{Jumlah\ Total\ Data} = \frac{TP0+TP1+TP2}{507} = \frac{161+176+151}{507} = \frac{488}{507} = 0{,}9625$$

Berikut visualisasi Tree yang di hasilkan :



- Naïve Bayes:

  Naïve Bayes adalah algoritma yang sederhana namun efektif, yang bergantung pada asumsi independensi antar fitur. Meskipun asumsi ini mungkin tidak selalu terpenuhi, Naïve Bayes sering memberikan hasil yang cukup baik. Namun, dalam kasus ini Naïve Bayes relatif rendah dibandingkan dengan algoritma lainnya. Hal ini mungkin disebabkan oleh fakta bahwa asumsi independensi Naïve Bayes tidak sepenuhnya terpenuhi dalam data kos-kosan yang kompleks ini.

  Dari semua model yang digunakan, moadel naïve bayes memiliki rata rata akurasi yang paling rendah yaitu 0.65 akurasi tertinggi dari semua penerapan yang di lakukan

  Rumus Matematis yang digunakan Pada Model Naïve Bayes:

  a. Naïve bayes untuk klasifikasi

     Dalam konteks klasifikasi, kita ingin menemukan kelas $\boldsymbol{C}$ yang memaksimalkan probabilitas posterior $\boldsymbol{P(C|X)}$. Dengan menggunakan asumsi independensi antara fitur-fitur, kita dapat menulis ulang likelihood $\boldsymbol{P(C|X)}$ sebagai produk dari probabilitas kondisi individu dari setiap fitur:

     $$\boldsymbol{P(C|X) \propto P(C)\ \cdot\ \prod_{i=1}^{n} P\ (x_i|\mathrm{C})}$$

     Keterangan:

     $\propto$: Proporsional terhadap (karena $P(X)$ adalah konstan untuk semua kelas, kita bisa mengabaikannya dalam perhitungan komparatif)

     n : jumlah fitur

     x_i : Fitur ke-iii dalam vektor fitur X

b. Menghitung probabilitas prior

Probabilitas prior $\boldsymbol{P(C)}$ dihitung dari proporsi kelas dalam dataset pelatihan:

$$\boldsymbol{P(C) = \frac{jumlah\ sampel\ dalam\ kelas\ C}{total\ jumlah\ sampel}}$$

c. Menghitung Likelihood

Probabilitas kondisi $P(x_i|\mathrm{C})$ bergantung pada jenis fitur:

- Untuk fitur kategorikal

$$\boldsymbol{P(x_i|C) = \frac{jumlah\ sampel\ dalam\ kelas\ C\ dengan\ x_i}{jumlah\ sampel\ dalam\ kelas\ C}}$$

d. Untuk fitur kontinu (menggunakan distribusi gaussian):

$$\boldsymbol{P(x_i|C) = \frac{1}{\sqrt{2\pi\sigma^2 c}}\ exp\left(-\frac{(x_{i-\mu c})^2}{dx2\sigma^2 c}\right)}$$

Keterangan:

μc : Mean dari fitur x_i dalam kelas C.

σc : Standard deviation dari fitur x_i dalam kelas C.

e. Klasifikasi

Untuk mengklasifikasikan sampel baru XXX, kita hitung $\boldsymbol{P(C|X)}$ untuk setiap kelas C dan pilih kelas dengan probabilitas tertinggi:

$$\boldsymbol{\widehat{C} = \arg max_c\ P(C) \cdot \prod_{i=1}^{n} P(x_i \mid C)}$$

Keterangan:

$\hat{C}$: Kelas prediksi dengan probabilitas tertinggi.

Perhitungan matrix berdasarkan confusion matrix:

- Logistic Regression:

Logistic Regression memodelkan hubungan antara fitur-fitur input dan probabilitas terjadinya suatu kejadian dengan menggunakan fungsi logistik. Ini membantu dalam memahami dampak relatif dari setiap fitur terhadap prediksi kelas target. Algoritma ini relatif sederhana dan mudah diinterpretasikan, sehingga cocok untuk kasus di mana interpretasi model adalah faktor penting. Logistic Regression juga stabil dalam kinerjanya dan biasanya memberikan hasil yang konsisten, terutama jika asumsi yang mendasarinya terpenuhi. Sehingga menunjukkan bahwa model ini cukup baik dalam memprediksi kategori harga kos-kosan.

Dari beberapa penerapan yang digunakan pada model random forest, yang memeiliki hasil nilai akurasi tertinggi adalah pada penerapan Model di data Resampling yaitu dengan hasil akurasi 0.92

Rumus Matematis yang digunakan Pada Model Logistic Regression:

a. Fungsi Linear (Linear Function)

Logistic Regression dimulai dengan menghitung skor linear dari input fitur.

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$$

keterangan:

- z adalah skor linear atau logit.
- β_0 adalah bias atau intercept.
- $\beta_1, \beta_2 \ldots \beta_n$ adalah koefisien regresi.
- $x_1, x_2 \ldots x_n$ adalah fitur input.

b. Fungsi Sigmoid (Sigmoid Function)

Skor linear z kemudian dilewatkan melalui fungsi sigmoid untuk memetakan nilai $\sigma\,(z)$ ke rentang probabilitas (0 sampai 1).

$$\hat{y} = \sigma\,(z) = \frac{1}{1 + e^{-z}}$$

keterangan:

- $\hat{y}$ adalah probabilitas prediksi dari kelas positif (misalnya, kelas 1).
- $\sigma\,(z) =$ adalah fungsi sigmoid.

c. Prediksi Klasifikasi (Classification Prediction)

Untuk menentukan kelas akhir berdasarkan probabilitas prediksi, digunakan ambang batas (threshold) default 0.5.

$$Prediksi\ Kelas = \begin{cases} 1\ jika\ \hat{y}\ \geq 0.5 \\ 0\ jika\ \hat{y} \leq 0.5 \end{cases}$$

d. Fungsi Loss (Loss Function)

Logistic Regression menggunakan log-loss (juga dikenal sebagai binary cross-entropy loss) untuk mengukur performa model.

$$L(\hat{y}, y) = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$$

keterangan:

- $L(\hat{y}, y)$ adalah nilai loss.
- y adalah label sebenarnya (0 atau 1).
- $\hat{y}$ adalah probabilitas prediksi dari kelas positif.

e. Training Model (Model Training)

Koefisien $\beta_1, \beta_2 \ldots \beta_n$ dioptimalkan untuk meminimalkan fungsi loss di seluruh data pelatihan. Ini biasanya dilakukan menggunakan algoritma optimasi seperti Gradient Descent.

Proses Optimasi

Untuk meminimalkan fungsi loss dan menemukan koefisien optimal, algoritma optimasi seperti Gradient Descent digunakan. Gradient Descent memperbarui koefisien secara iteratif:

$$\beta_j = \beta_j - \propto \frac{\partial L}{\partial \beta_j}$$

keterangan:

- $\propto$ adalah learning rate.
- $\frac{\partial L}{\partial \beta_j}$ adalah turunan parsial dari fungsi loss terhadap koefisien β_j

Perhitungan matrix berdasarkan confusion matrix:
Kelas 0

TP : 153

FP : 14 (kelas 1) + 1 (kelas 2) = 15

FN : 12 (kelas 1) + 1 (kelas 2) = 13

$$Precission\ 0 = \frac{TP0}{TP0 + FP0} = \frac{153}{153 + 15} = \frac{153}{168} = 0{,}9107$$

$$Recall\ 0 = \frac{TP0}{TP0 + FN0} = \frac{153}{153 + 13} = \frac{153}{166} = 0{,}9217$$

$$F1\ Score\ 0 = 2 \times \frac{Precission\ 0 \times Recall\ 0}{Precission\ 0 + Recall\ 0} = \frac{0{,}9107 \times 0.9217}{0{,}9107 + 0{,}9217} = 0{,}9162$$

Kelas 1

TP : 163
FP : 12 (kelas 0) + 6 (kelas 2) = 18
FN : 14 (kelas 0) + 7 (kelas 2) = 21

$$Precission\ 1 = \frac{TP1}{TP1 + FP1} = \frac{163}{163 + 18} = \frac{163}{181} = 0{,}9006$$

$$Recall\ 1 = \frac{TP1}{TP1 + FN1} = \frac{163}{163 + 21} = \frac{163}{184} = 0{,}8869$$

$$F1\ Score\ 1 = 2 \times \frac{Precission\ 1 \times Recall\ 1}{Precission\ 1 + Recall\ 1} = \frac{0{,}9006 \times 0.8869}{0{,}9006 + 0{,}8869} = 0{,}8937$$

Kelas 2

TP : 150
FP : 1 (kelas 0) + 7 (kelas 1) = 8
FN : 6 (kelas 1)

$$Precission\ 2 = \frac{TP2}{TP2 + FP2} = \frac{150}{150 + 6} = \frac{150}{158} = 0{,}9494$$

$$Recall\ 2 = \frac{TP2}{TP2 + FN2} = \frac{150}{150 + 6} = \frac{150}{156} = 0{,}9615$$

$$F1\ Score\ 2 = 2 \times \frac{Precission\ 2 \times Recall\ 2}{Precission\ 2 + Recall\ 2} = \frac{0{,}9557 \times 0.9618}{0{,}9557 + 0{,}9618} = 0{,}9588$$

$$\boldsymbol{Akurasi} = \frac{Jumlah\ Prediksi\ Benar}{Jumlah\ Total\ Data} = \frac{TP0+TP1+TP2}{507} = \frac{153+163+150}{507} = \frac{466}{507} = 0{,}9193$$

- Xgboost:

  Xgboost menggabungkan keuntungan dari beberapa model lemah untuk membuat model yang lebih kuat. Ini membantu dalam meningkatkan akurasi prediksi dan mengurangi overfitting. Algoritma ini sangat efisien dalam hal waktu dan sumber daya komputasi, sehingga cocok untuk digunakan dalam skenario di mana kita perlu menangani dataset besar atau kompleks. Xgboost memiliki banyak parameter yang dapat disesuaikan, sehingga memungkinkan penyesuaian yang lebih lanjut untuk meningkatkan kinerja model. Model ini juga menunjukkan bahwa sangat baik dalam memprediksi kategori harga kos-kosan.

Dari beberapa penerapan yang digunakan pada model XGBoost, keseluruhan hasil nilai akurasinya itu hampir sama yaitu 0,95

Rumus Matematis yang digunakan Pada Model XGBoost:

a. Fungsi Objektif
XGBoost mengoptimalkan fungsi objektif yang terdiri dari dua bagian: fungsi kerugian (loss function) dan regularisasi (penalty). Untuk klasifikasi biner dengan XGBoost, fungsi objektif yang umum digunakan adalah:

$$\mathcal{L}(\boldsymbol{\phi}) = \sum_{i=1}^{n} \left[y_i \cdot \log\left(1 + e^{-\phi(x_i)}\right) + (1 - y_i) \cdot \log\left(1 + e^{-\phi(x_i)}\right) \right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2$$

Keterangan:
- y_i adalah label kelas (0 atau 1) dari instance data ke-i,
- $\phi(x_i)$ adalah prediksi dari model XGBoost untuk instance data ke-i,
- T adalah jumlah pohon (trees) yang dibangun,
- γ adalah parameter yang mengontrol kompleksitas model (regularisasi),
- λ adalah parameter regularisasi L2,
- ω_j adalah bobot (weight) dari node ke-j dalam pohon.

b. Prediksi Probabilitas
XGBoost menggunakan fungsi logistik (sigmoid function) untuk menghitung probabilitas kelas positif (biasanya kelas 1):

$$P(y = 1|x) = \frac{1}{1 + e^{-\phi(x)}}$$

c. Gradient dan Hessian
Untuk mengoptimalkan fungsi objektif, XGBoost menggunakan turunan pertama (gradient) dan kedua (hessian) dari fungsi objektif terhadap prediksi $\phi(X_i)$:

- Gradient

$$g_i = \frac{\partial \mathcal{L}(\boldsymbol{\phi})}{\partial \boldsymbol{\phi}(x_i)} = -\frac{y_i - P(y = 1|x_i)}{P(y = 1|x_i) \cdot (1 - P(y = 1|x_i))}$$

- Hessian:

$$h_i = \frac{\partial^2 \mathcal{L}(\boldsymbol{\phi})}{\partial \boldsymbol{\phi}(x_i)^2} = P(y = 1|x_i) \cdot (1 - P(y = 1|x_i))$$

d. Proses Booting
XGBoost melakukan boosting dengan menambahkan pohon baru pada setiap iterasi untuk mengurangi kesalahan residual (selisih antara prediksi aktual dan prediksi model saat ini) menggunakan gradient descent:

$$\boldsymbol{\phi}^{(t)}(\boldsymbol{x}) = \boldsymbol{\phi}^{(t-1)}(\boldsymbol{x}) + \boldsymbol{\eta} \cdot \boldsymbol{h_t}(\boldsymbol{x})$$

Di mana η adalah learning rate yang mengontrol seberapa besar kontribusi setiap pohon terhadap model akhir

Kesimpulan:

Rumus-rumus di atas membentuk dasar matematis dari bagaimana XGBoost bekerja dalam konteks klasifikasi. Algoritma ini mengoptimalkan fungsi objektif yang mencakup fungsi kerugian, regularisasi, dan menggunakan pendekatan boosting untuk meningkatkan prediksi secara bertahap.

Perhitungan matrix berdasarkan confusion matrix:

Kelas 0

TP : 158
FP : 2 (kelas 1) + 1 (kelas 2) = 3
FN : 7 (kelas 1) + 1 (kelas 2) = 8

$$Precission\ 0 = \frac{TP0}{TP0 + FP0} = \frac{158}{158 + 3} = \frac{158}{161} = 0{,}9814$$

$$Recall\ 0 = \frac{TP0}{TP0 + FN0} = \frac{158}{158 + 8} = \frac{158}{166} = 0{,}9518$$

$$F1\ Score\ 0 = 2 \times \frac{Precission\ 0 \times Recall\ 0}{Precission\ 0 + Recall\ 0} = \frac{0{,}9814 \times 0.9518}{0{,}9814 + 0{,}9518} = 0{,}9663$$

Kelas 1

TP : 176
FP : 7 (kelas 0) + 6 (kelas 2) = 13
FN : 2 (kelas 0) + 6 (kelas 2) = 8

$$Precission\ 1 = \frac{TP1}{TP1 + FP1} = \frac{176}{176 + 13} = \frac{176}{189} = 0{,}9312$$

$$Recall\ 1 = \frac{TP1}{TP1 + FN1} = \frac{176}{176 + 8} = \frac{176}{184} = 0{,}9565$$

$$F1\ Score\ 1 = 2 \times \frac{Precission\ 1 \times Recall\ 1}{Precission\ 1 + Recall\ 1} = \frac{0{,}9312 \times 0.9565}{0{,}9312 + 0{,}9565} = 0{,}9437$$

Kelas 2

TP : 150
FP : 1 (kelas 0) + 6 (kelas 1) = 7
FN : 6 (kelas 1)

$$Precission\ 2 = \frac{TP2}{TP2 + FP2} = \frac{150}{150 + 7} = \frac{150}{157} = 0{,}9554$$

$$Recall\ 2 = \frac{TP2}{TP2 + FN2} = \frac{150}{150 + 6} = \frac{150}{156} = 0{,}9615$$

$$F1\ Score\ 2 = 2 \times \frac{Precission\ 2 \times Recall\ 2}{Precission\ 2 + Recall\ 2} = \frac{0{,}9554 \times 0.9615}{0{,}9554 + 0{,}9615} = 0{,}9584$$

$$\boldsymbol{Akurasi} = \frac{Jumlah\ Prediksi\ Benar}{Jumlah\ Total\ Data} = \frac{TP0+TP1+TP2}{507} = \frac{158+176+150}{507} = \frac{484}{507} = 0{,}9546$$

- Skenario eksperiment sederhana.

  Eksperimen ini bertujuan untuk mengklasifikasikan harga kos-kosan berdasarkan beberapa fitur yang relevan, dengan tujuan akhir untuk menentukan apakah harga tersebut tergolong murah, sesuai, atau mahal. Untuk mencapai tujuan ini, kami menggunakan berbagai teknik analisis data dan algoritma pembelajaran mesin.

  Dalam eksperimen ini, kami mengkaji data kos-kosan yang diperoleh dari aplikasi Mamikos dan Kaggle. Tahap pertama melibatkan pemahaman data, di mana kami menganalisis dataset yang berisi 8 fitur dan memutuskan untuk menggunakan hanya 3 fitur untuk eksperimen ini. Selanjutnya, kami melakukan pembersihan data dengan menghapus beberapa fitur berdasarkan hasil analisis awal, menghilangkan baris dengan nilai NaN, serta mentransformasi data agar lebih mudah diolah oleh mesin. Kami memilih metode klasifikasi untuk menentukan harga kos apakah tergolong murah, sesuai, atau mahal.

  Data kemudian dibagi menjadi data pelatihan dan data uji. Untuk penerapan model, kami menggunakan lima algoritma yaitu Random Forest, Decision Tree, dan Naïve Bayes, Logistic Regression , dan Xgboost. Hasil evaluasi menunjukkan bahwa Decision Tree, Xgboost dan Random Forest mencapai akurasi 95%, Logistic Regression mencapai akurasi 87 dan Naïve Bayes hanya mencapai akurasi 65%. Berikut hasil dari table perbandingan tiap tiap model dan gambar confusion matrix untuk masing-masing model dapat dilihat dibawah:

- **Tabel Perbandingan**

Data Original					
Algoritma	Class	Accuracy	Precission	Recall	f1-score
Naïve Bayes	Harga Murah	0.65	0.00	0.00	0.00
	Harga Sesuai		0.65	0.95	0.77
	Harga Mahal		0.61	0.21	0.31
Decission Tree	Harga Murah	0.95	0.90	0.97	0.94
	Harga Sesuai		0.96	0.97	0.96
	Harga Mahal		0.98	0.90	0.94
Logistic Regression	Harga Murah	0.87	0.86	0.66	0,75
	Harga Sesuai		0.85	0.98	0.91
	Harga Mahal		1.00	0.71	0.83
Random Forest	Harga Murah	0.95	0.90	0.97	0.94
	Harga Sesuai		0.96	0.97	0.96
	Harga Mahal		0.98	0.90	0.94
Xgboost	Harga Murah	0.95	0.90	0.97	0.94
	Harga Sesuai		0.96	0.97	0.96
	Harga Mahal		0.98	0.90	0.94
Data setelah Resampling					
Naïve Bayes	Harga Murah	0.55	0.55	0.86	0.67
	Harga Sesuai		0.53	0.38	0.44
	Harga Mahal		0.60	0.43	0.50
Decission Tree	Harga Murah	0.96	0.99	0.97	0.98

	Harga Sesuai		0.95	0.96	0.95
	Harga Mahal		0.96	0.96	0.96
Logistic Regression	Harga Murah	0.92	0.91	0.92	0.92
	Harga Sesuai		0.90	0.89	0.89
	Harga Mahal		0.95	0.96	0.95
Random Forest	Harga Murah	0.96	0.99	0.97	0.98
	Harga Sesuai		0.95	0.96	0.95
	Harga Mahal		0.96	0.97	0.96
Xgboost	Harga Murah	0.95	0.98	0.95	0.97
	Harga Sesuai		0.93	0.96	0.94
	Harga Mahal		0.96	0.96	0.96
Penerapan Chi2 pada data Original					
Naïve Bayes	Harga Murah	0.68	0.00	0.00	0.00
	Harga Sesuai		0.66	1.00	0.80
	Harga Mahal		1.00	0.22	0.36
Decission Tree	Harga Murah	0.95	0.90	0.97	0.94
	Harga Sesuai		0.96	0.97	0.96
	Harga Mahal		0.98	0.90	0.94
Logistic Regression	Harga Murah	0.62	0.00	0.00	0.00
	Harga Sesuai		0.63	1.00	0.77
	Harga Mahal		0.00	0.00	0.00
Random Forest	Harga Murah	0.95	0.90	0.97	0.94
	Harga Sesuai		0.96	0.97	0.96

	Harga Mahal		0.98	0.90	0.94
Xgboost	Harga Murah	0.95	0.90	0.97	0.94
	Harga Sesuai		0.96	0.97	0.96
	Harga Mahal		0.98	0.90	0.94
Penerapan Chi2 pada data resampling					
Naïve Bayes	Harga Murah	0.55	0.48	0.83	0.61
	Harga Sesuai		0.58	0.34	0.43
	Harga Mahal		0.70	0.50	0.58
Decission Tree	Harga Murah	0.96	0.99	0.97	0.98
	Harga Sesuai		0.95	0.96	0.95
	Harga Mahal		0.96	0.96	0.96
Logistic Regression	Harga Murah	0.31	0.00	0.00	0.00
	Harga Sesuai		0.00	0.00	0.00
	Harga Mahal		0.31	1.00	0.47
Random Forest	Harga Murah	0.96	0.99	0.97	0.98
	Harga Sesuai		0.95	0.96	0.95
	Harga Mahal		0.96	0.97	0.96
Xgboost	Harga Murah	0.95	0.98	0.95	0.97
	Harga Sesuai		0.93	0.96	0.94
	Harga Mahal		0.96	0.96	0.96
Penerapan SFS Forward pada data Original					
Naïve Bayes	Harga Murah	0.68	0.00	0.00	0.00
	Harga Sesuai		0.66	1.00	0.80

	Harga Mahal		1.00	0.22	0.36
Decission Tree	Harga Murah	0.95	0.90	0.97	0.94
	Harga Sesuai		0.96	0.97	0.96
	Harga Mahal		0.98	0.90	0.94
Logistic Regression	Harga Murah	0.87	0.86	0.66	0.75
	Harga Sesuai		0.85	0.98	0.91
	Harga Mahal		1.00	0.71	0.83
Random Forest	Harga Murah	0.95	0.90	0.97	0.94
	Harga Sesuai		0.96	0.97	0.96
	Harga Mahal		0.98	0.90	0.94
Xgboost	Harga Murah	0.95	0.90	0.97	0.94
	Harga Sesuai		0.96	0.97	0.96
	Harga Mahal		0.98	0.90	0.94
Penerapan SFS forward pada data resampling					
Naïve Bayes	Harga Murah	0.55	0.55	0.86	0.67
	Harga Sesuai		0.53	0.38	0.44
	Harga Mahal		0.60	0.43	0.50
Decission Tree	Harga Murah	0.96	0.99	0.97	0.98
	Harga Sesuai		0.95	0.96	0.95
	Harga Mahal		0.96	0.96	0.96
Logistic Regression	Harga Murah	0.92	0.91	0.92	0.92
	Harga Sesuai		0.90	0.89	0.89
	Harga Mahal		0.95	0.96	0.95

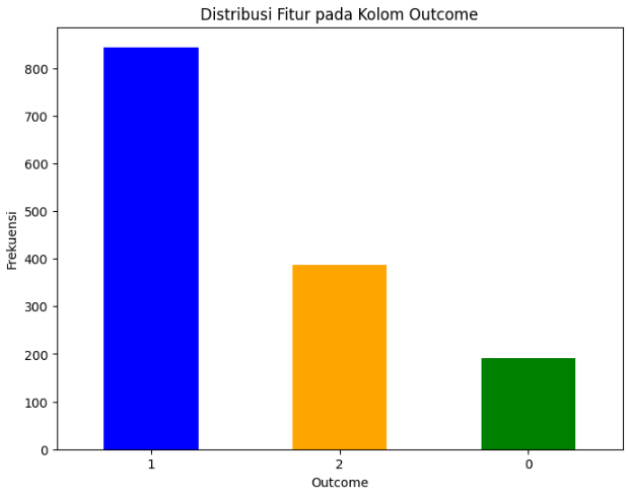
Random Forest	Harga Murah	0.96	0.99	0.97	0.98
	Harga Sesuai		0.95	0.96	0.95
	Harga Mahal		0.96	0.97	0.96
Xgboost	Harga Murah	0.95	0.98	0.95	0.97
	Harga Sesuai		0.93	0.96	0.94
	Harga Mahal		0.96	0.96	0.96
Penerapan SFS Backward pada data Original					
Naïve Bayes	Harga Murah	0.68	0.00	0.00	0.00
	Harga Sesuai		0.66	1.00	0.80
	Harga Mahal		1.00	0.22	0.36
Decission Tree	Harga Murah	0.95	0.90	0.97	0.94
	Harga Sesuai		0.96	0.97	0.96
	Harga Mahal		0.98	0.90	0.94
Logistic Regression	Harga Murah	0.87	0.86	0.66	0.75
	Harga Sesuai		0.85	0.98	0.91
	Harga Mahal		1.00	0.71	0.83
Random Forest	Harga Murah	0.95	0.90	0.97	0.94
	Harga Sesuai		0.96	0.97	0.96
	Harga Mahal		0.98	0.90	0.94
Xgboost	Harga Murah	0.95	0.90	0.97	0.94
	Harga Sesuai		0.96	0.97	0.96
	Harga Mahal		0.98	0.90	0.94
Penerapan SFS Backward pada data resampling					

Naïve Bayes	Harga Murah	0.55	0.48	0.83	0.61
	Harga Sesuai		0.58	0.34	0.43
	Harga Mahal		0.70	0.50	0.58
Decission Tree	Harga Murah	0.96	0.99	0.97	0.98
	Harga Sesuai		0.95	0.96	0.95
	Harga Mahal		0.96	0.96	0.96
Logistic Regression	Harga Murah	0.92	0.91	0.92	0.92
	Harga Sesuai		0.90	0.89	0.89
	Harga Mahal		0.95	0.96	0.95
Random Forest	Harga Murah	0.96	0.99	0.97	0.98
	Harga Sesuai		0.95	0.96	0.95
	Harga Mahal		0.96	0.97	0.96
Xgboost	Harga Murah	0.95	0.90	0.97	0.94
	Harga Sesuai		0.96	0.97	0.96
	Harga Mahal		0.98	0.90	0.94

- **Distribusi Fitur**

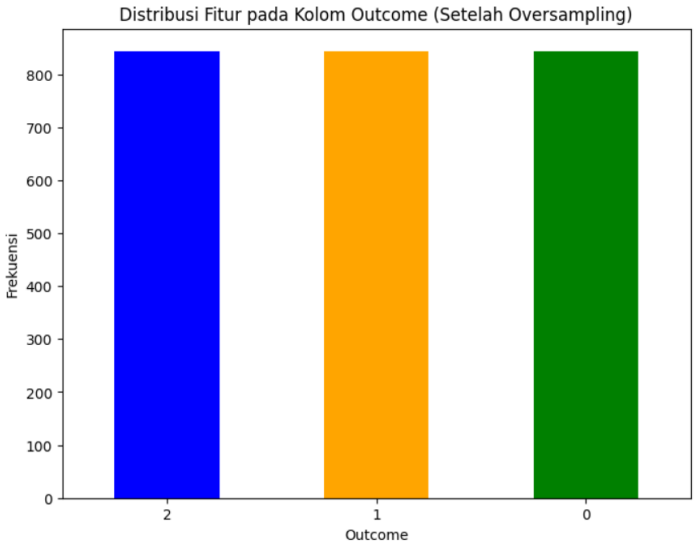
  Distribusi dibawah mencakup Algoritma Random Forest, Decission Tree, Xgboost, Logistic Regression, dan Naïve Bayes.

  Distribusi Fitur kolom outcome



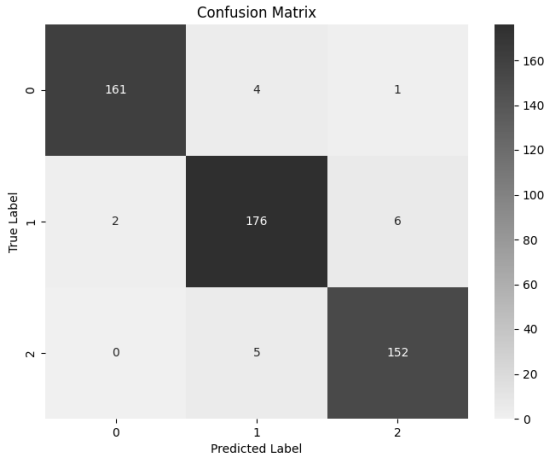
  Grafik distribusi class menunjukan ketidakseimbangan jumlah data pada setiap kelas yang dimana pada data pada class 1 terdapat 800 data, class 2 terdapat 400 data, dan pada class 0 terdapat 200 data. Oleh karena itu perlu dilakukan balancing data agar data dari tiap kelas seimbang dan dapat memaksimalkan kinerja model.

  Distribusi Fitur kolom outcome (Setelah oversampling)



  Grafik distribusi class setelah dilakukan oversampling menunjukan bahwa sudah tidak ada perbedaan yang signifikan atau bisa dikatakan data telah seimbang setiap classnya

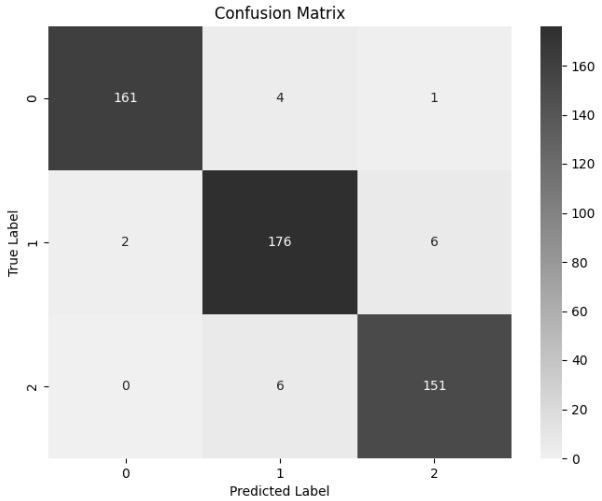
- **Confusion Matrix**
  - *Random Forest*



Gambar 1. Confusion Matrix *Random Forest*

Confusion matrix Random Forest menunjukkan bahwa model memiliki kinerja yang sangat baik dengan akurasi tinggi dalam mengklasifikasikan instance dari setiap kelas.

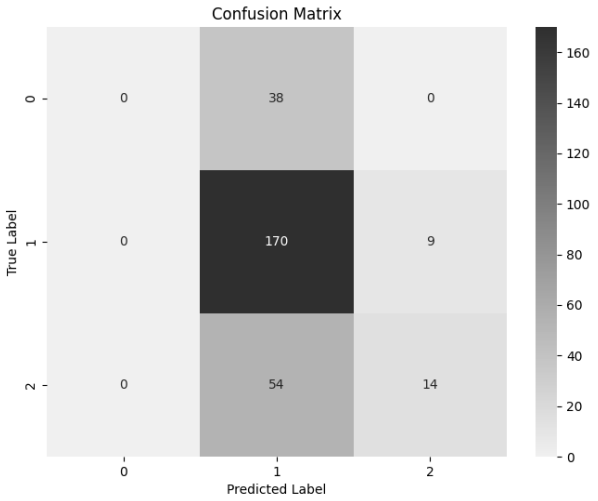
  - *Decision Tree*



Gambar 2. Confusion Matrix *Decision Tree*

Confusion matrix Decision Tree menunjukkan bahwa model memiliki kinerja yang sangat baik dengan akurasi tinggi dalam mengklasifikasikan instance dari setiap kelas.

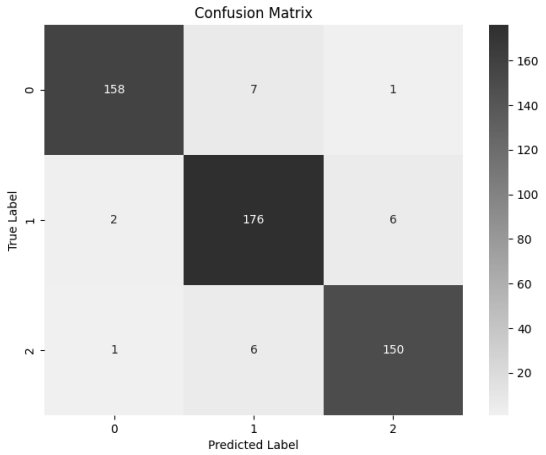
- *Naïve Bayes*



Gambar 3. Confusion Matrix *Naïve Bayes*

Confusion matrix Naïve Bayes menunjukkan bahwa model memiliki kinerja yang kurang baik dengan akurasi rendah dalam mengklasifikasikan instance dari setiap kelas.

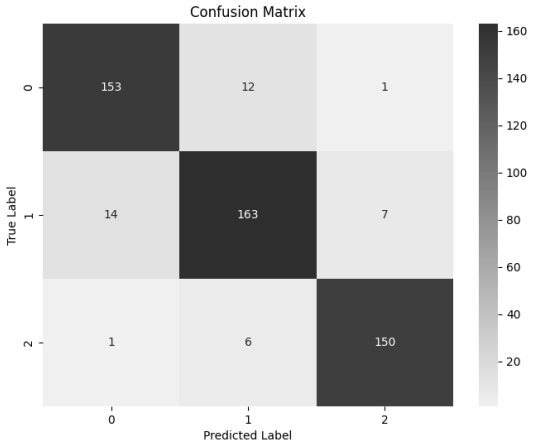
- *XGboost*



Gambar 4. Confusion Matrix *XGboost*

Confusion matrix Xgboost menunjukkan bahwa model memiliki kinerja yang sangat baik dengan akurasi tinggi dalam mengklasifikasikan instance dari setiap kelas.

- Logistic Regression



Gambar 5. Confusion Matrix Logistic Regression

Confusion matrix Logistic Regression menunjukkan bahwa model memiliki kinerja yang sangat baik dengan akurasi tinggi dalam mengklasifikasikan instance dari setiap kelas.

Tahap 6 (poin: 20): Knowledge Interpretation

- Pola-pola *useful* yang telah ditemukan

Dalam pembelajaran data mining, pengetahuan yang diinterpretasikan dari data sering kali berupa pola-pola yang berguna untuk memahami hubungan antar atribut dan memprediksi nilai-nilai baru. Pada studi kasus ini, kami menemukan dua jenis pola utama yaitu pola asosiasi dan pola prediktif.

Pola asosiasi adalah hubungan atau keterkaitan antara beberapa atribut dalam data yang sering muncul bersamaan. Pada kasus ini, kami menemukan bahwa terdapat hubungan yang kuat antara atribut harga dan fasilitas yang ditawarkan. Secara khusus, semakin tinggi harga kos, semakin banyak dan semakin bagus fasilitas yang disediakan. Sebaliknya, kos dengan harga lebih rendah cenderung memiliki fasilitas yang lebih sedikit atau kualitas yang lebih rendah. Pola ini menunjukkan bahwa harga kos bisa menjadi indikator dari jenis dan kualitas fasilitas yang dapat diharapkan.

Dan untuk pola prediktif, di sisi lain, berfokus pada kemampuan untuk memprediksi nilai-nilai baru berdasarkan model yang dibangun dari data yang ada. Dalam studi kasus ini, kami menggunakan pola klasifikasi untuk membangun model yang dapat mengklasifikasikan kos-kosan ke dalam kategori harga murah, sesuai, atau mahal. Model ini dilatih dengan menggunakan data historis dan fitur-fitur yang relevan untuk

memprediksi kategori harga suatu kos. Dengan demikian, pola prediktif ini membantu dalam membuat prediksi yang lebih akurat mengenai kategori harga kos-kosan berdasarkan

- Knowladge Interpretation
  - Naïve Bayes

    Akurasi pada Naïve Bayes menunjukan bahwa algoritma dapat memprediksi data dengan ketepatan 55%. 45% tidak dapat memprediksi dengan tepat. Dalam 45% ini dapat di analisis bahwa Pada Kelas 0, 10 data kelas 0 salah diprediksi sebagai kelas 1, 13 data kelas 0 salah diprediksi sebagai kelas 2. Pada Kelas 1, 70 data kelas 1 salah diprediksi sebagai kelas 2, 82 data kelas 1 salah diprediksi sebagai kelas 0. Pada kelas 2, 32 data kelas 2 salah diprediksi sebagai kelas 1, 37 data kelas 2 salah diprediksi sebagai kelas 0.

  - Decission Tree

    Akurasi pada Decission Tree menunjukan bahwa algoritma dapat memprediksi data dengan ketepatan 96%. 4% tidak dapat memprediksi dengan tepat. Dalam 4% ini dapat dianalisis bahwa Pada Kelas 0, 4 data kelas 0 salah diprediksi sebagai kelas 1, 1 data kelas 0 salah diprediksi sebagai kelas 2.Pada Kelas 1, 176 data kelas 1 salah diprediksi sebagai kelas 2, 2 data kelas 1 salah diprediksi sebagai kelas 0. Pada kelas 2, 6 data kelas 2 salah diprediksi sebagai kelas 1, 0 data kelas 2 salah diprediksi sebagai kelas 0.

  - Logistic Regression

    Akurasi pada logistic Regression menunjukan bahwa algoritma dapat memprediksi data dengan ketepatan 92%. 8% tidak dapat memprediksi dengan tepat. Dalam 8% ini dapat dianalisis bahwa Pada Kelas 0, 12 data kelas 0 salah diprediksi sebagai kelas 1, 1 data kelas 1 salah diprediksi sebagai kelas 2. Pada Kelas 1, 163 data kelas 1 salah diprediksi sebagai kelas 2, 14 data kelas 1 salah diprediksi sebagai kelas 0. Pada kelas 2, 7 data kelas 2 salah diprediksi sebagai kelas 1, 1 data kelas 2 salah diprediksi sebagai kelas 0.

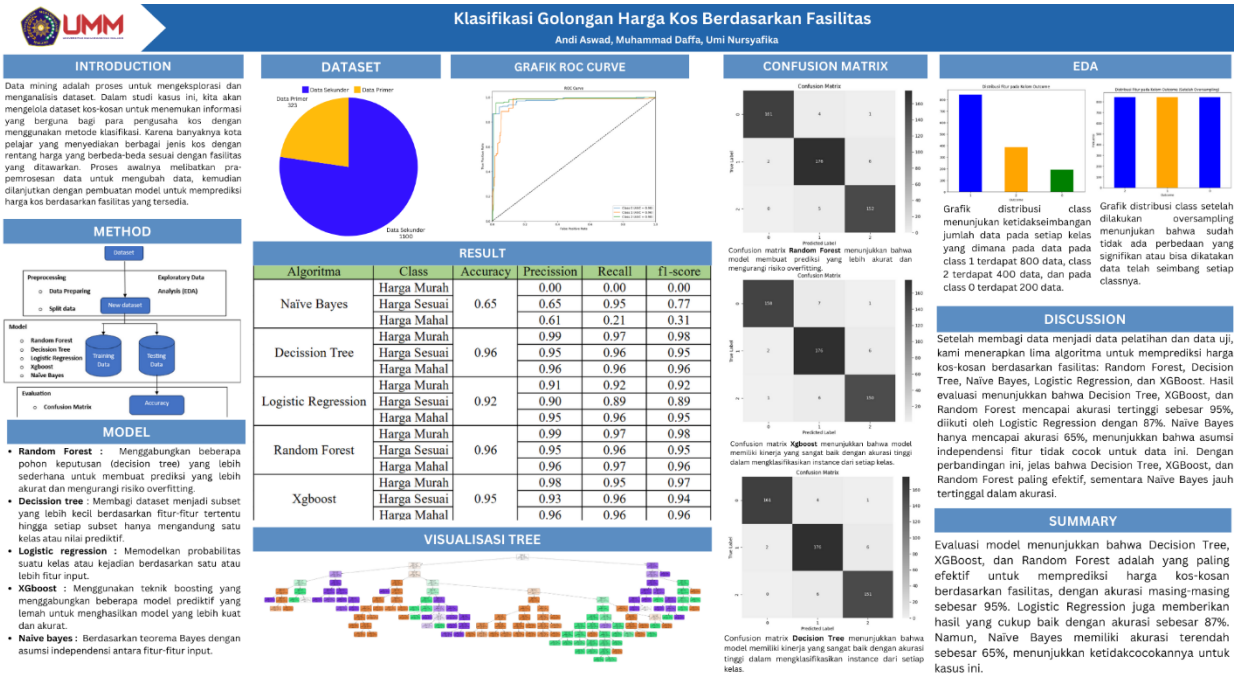
  - Random Forest

    Akurasi pada logistic Regression menunjukan bahwa algoritma dapat memprediksi data dengan ketepatan 96%. 4% tidak dapat memprediksi dengan tepat. Dalam 4% ini dapat dianalisis bahwa Pada Kelas 0, 4 data kelas 0 salah diprediksi sebagai kelas 1, 1 data kelas 1 salah diprediksi sebagai kelas 2.Pada Kelas 1, 176 data kelas 1 salah diprediksi sebagai kelas 2, 2 data kelas 1 salah diprediksi sebagai kelas 0. Pada kelas 2, 6 data kelas 2 salah diprediksi sebagai kelas 1, 0 data kelas 2 salah diprediksi sebagai kelas 0.

- XGBoost
  Akurasi pada logistic Regression menunjukan bahwa algoritma dapat memprediksi data dengan ketepatan 95%. 5% tidak dapat memprediksi dengan tepat. Dalam 5% ini dapat dianalisis bahwa Pada Kelas 0, 2 data kelas 0 salah diprediksi sebagai kelas 1, 1 data kelas 1 salah diprediksi sebagai kelas 2. Pada Kelas 1, 176 data kelas 1 salah diprediksi sebagai kelas 2, 2 data kelas 1 salah diprediksi sebagai kelas 0. Pada kelas 2, 6 data kelas 2 salah diprediksi sebagai kelas 1, 1 data kelas 2 salah diprediksi sebagai kelas 0.

Tahap 7 (poin: 15): Reporting

- **Simple academic Poster pada slide berikutnya**



Link Editable Poster :
https://www.canva.com/design/DAGHvoHgQsk/LsMRkuAeSHNI7l1np10nkQ/edit

- Jupiter Notebook (Python)
  Link Collab :
  https://colab.research.google.com/drive/1ViTa4p4Qq0TV1rYL6nk5sKd42izbs0yN?usp=sharing