

Kelompok 5

KLASIFIKASI PENYAKIT STROKE MENGGUNAKAN METODE NAIVE BAYES

UTS SISTEM CERDAS



NAIVE BAYES CLASSIFICATION

S I S T E M C E R D A S



Muhammad Reva Alief
Fathoni (082111733006)



Mutiara Dewi Ayu
Antika (082111733003)



Krishna Alvian R.
(082111733024)



Putri Azzahra A.
(082111733035)



Yuma Zahran Ewaldo
(082111733069)



Nur Alifiadewi
(082111733060)



Rona Kamilia
(082111733050)



Ahmad Rizki Nur
Permana (082111733001)



Sherly Angelica
(082111733040)



Edwardo Alpian
(082111733004)



PENDAHULUAN

01

LATAR BELAKANG

Pada tahun 2019, angka kasus dan penyitas stroke di Indonesia mencapai 293,33 dan 1.097,22 per 100.000 orang dengan Kalimantan Timur sebagai provinsi dengan angka kasus dan penyitas stroke tertinggi dan 90,5% dari keseluruhan faktor penyebab stroke merupakan faktor yang dapat dikontrol, dan sebanyak 74,2% berkaitan dengan faktor gaya hidup (Widyasari, 2022; Setyopranoto, 2019).

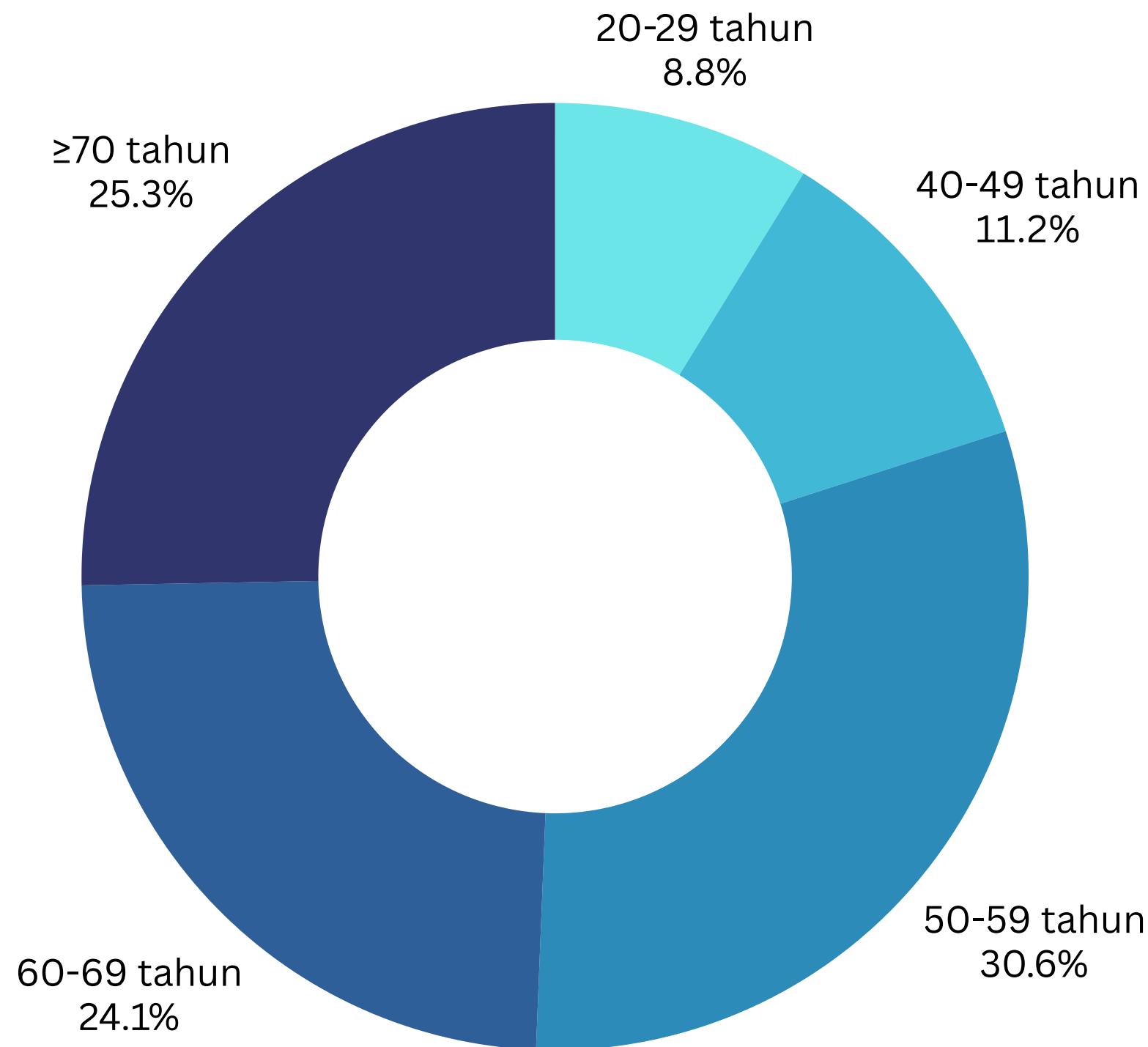


Diagram prevalensi penyitas stroke berdasarkan usia per 100.000 orang (Setyopranoto, 2019)

LATAR BELAKANG

Terlepas dari meningkatnya harapan hidup bagi penderita stroke, dalam pengobatannya, sangat penting untuk dapat mendiagnosis fase awal stroke dan mengobati pasien sesuai dengan diagnosisnya.

Penelitian yang dilakukan oleh Dritsas (2022) menggunakan Naïve Bayes Classification untuk mengklasifikasi kategori stroke dan non-stroke pada pasien dengan dan tanpa stroke.

Dengan mempertimbangkan setiap fitur dari riwayat medis pasien dan mengelompokkannya ke dalam kategori yang telah ditentukan Naïve Bayes dapat memprediksi kemungkinan terjadinya stroke yang dapat mendukung pengambilan keputusan klinis



RUMUSAN MASALAH



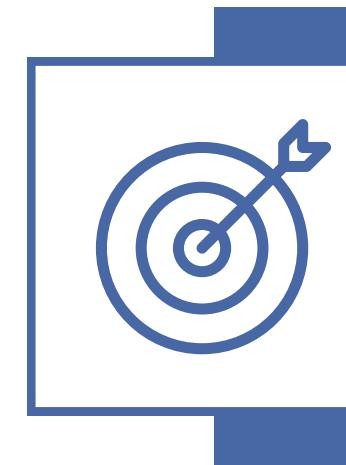
Bagaimana pengaruh proporsi pembagian data training dan data testing terhadap performa model Naïve Bayes dalam mengklasifikasi pasien penyakit stroke?



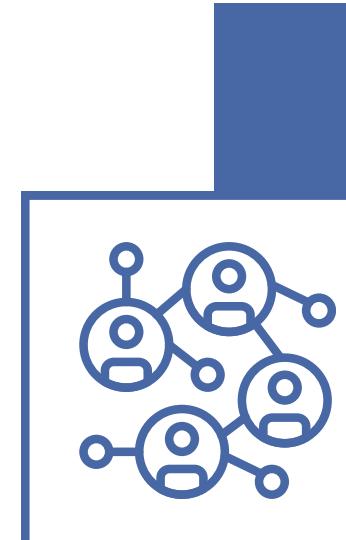
Faktor-faktor apa saja yang dapat mempengaruhi akurasi klasifikasi penyakit stroke selain proporsi pembagian data training dan testing?

04

TUJUAN PENELITIAN

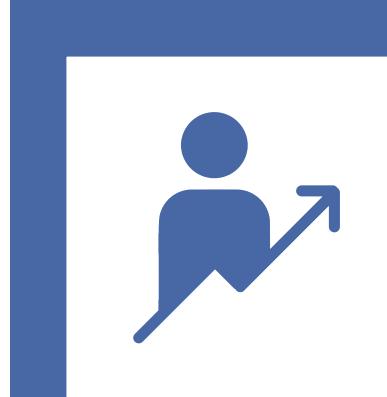


Evaluasi efektivitas model Naïve Bayes untuk klasifikasi pasien stroke dengan variasi proporsi data training dan testing. **Identifikasi faktor-faktor** yang memengaruhi performa model untuk pengembangan sistem klasifikasi yang lebih baik.



menevaluasi pengaruh proporsi pembagian data training/testing dan **faktor-faktor lain** seperti fitur klinis, ukuran sampel, dan kualitas data terhadap akurasi model Naïve Bayes dalam mengklasifikasi pasien penyakit stroke. Tujuan tambahan adalah **menemukan proporsi optimal** pembagian data yang dapat meningkatkan performa model serta **menganalisis kontribusi relatif** dari faktor-faktor tersebut untuk pengembangan model klasifikasi yang lebih akurat.

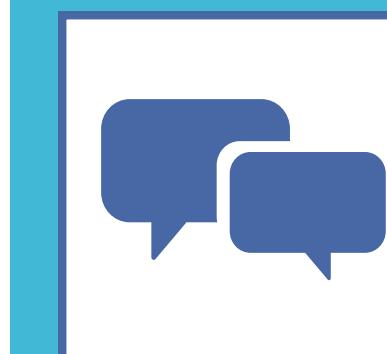
MANFAAT PENELITIAN



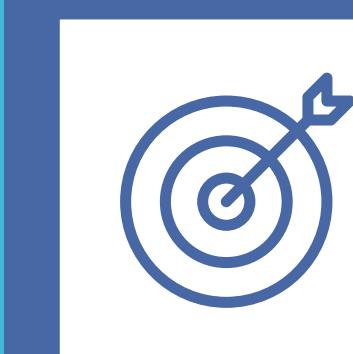
Optimisasi Klasifikasi



Pemahaman Lebih Lanjut



Pengembangan Model



Implikasi Klinis



TINJAUAN PUSTAKA

07

STROKE

Stroke adalah kondisi medis yang terjadi ketika **suplai darah ke bagian otak terganggu**, baik karena **penyumbatan pembuluh darah** (stroke iskemik) atau **pecahnya pembuluh darah** (stroke hemoragik)

Faktor risiko yang **tidak dapat dimodifikasi**: usia, jenis kelamin, dan riwayat keluarga

Faktor risiko yang **dapat dimodifikasi**: BMI, hipertensi, merokok, dislipidemia, diabetes melitus, obesitas, alkohol dan atrial fibrillation

NBC

Naïve Bayes adalah suatu pengklasifikasian probabilistik sederhana yang menghitung peluang dari menambahkan frekuensi dan kombinasi nilai dari dataset.

Persamaan teorema Bayes dituliskan sebagai berikut:

$$P(Y|X) = \frac{P(X \cap Y)}{P(X)} = \frac{P(X|Y) \cdot P(Y)}{P(X)}$$

dimana:

$P(Y|X)$: Peluang terjadinya Y berdasarkan kondisi X (posteriori prob)

$P(Y)$: Peluang terjadinya Y (prior prob)

$P(X|Y)$: Peluang terjadinya X berdasarkan kondisi pada hipotesis Y

$P(X)$: Peluang terjadinya X

Klasifikasi Naïve Bayes dapat dituliskan sebagai berikut:

$$P(Y|X_1, \dots, X_n) = P(Y) \cdot P(X_1|Y) \cdot P(X_2|Y) \cdots$$

$$= P(Y) \prod_{i=1}^n P(X_i|Y)$$

Rumus Gaussian Naïve Bayes:

$$P(X_i = x_i | Y_i = y_i) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(x_i - u_{ij})^2}{2\sigma_{ij}^2}}$$

DATA MINING

- Proses dimana **data berukuran besar dikumpulkan** dengan **ekstraksi** dan **identifikasi** pola-pola penting atau **mencari data yang berada di basis data** (Wulandari, et al., 2020).
- Proses menggunakan **teknik statistika, matematika, kecerdasan kecerdasan buatan** dan **machine learning** untuk **mengidentifikasi** dan **memperoleh informasi** yang berguna dan pengetahuan yang berhubungan dengan berbagai **basis data besar** (Turban, et al., 2005).

PREPROCESSING DATA

- Tujuan: untuk **mempersiapkan data** untuk digunakan dalam proses selanjutnya.
- Pertama, **pembersihan data** dilakukan dengan pengecekan data hilang dan data duplikat (Fikri, et al., 2020)
- Tahapan selanjutnya adalah **pengubahan bentuk data** agar dapat mempermudah dalam proses pengklasifikasian (Adi, 2018).
- Terakhir, **data yang digunakan dibagi** menjadi data training dan data testing.

CLASSIFICATION

- Cara untuk **menemukan properti yang sama** pada himpunan objek di dalam sebuah data dan **mengklasifikasikannya ke dalam kelas-kelas yang berbeda** menurut model klasifikasi yang ditetapkan.
- Tujuan: **menemukan model dari training set** yang membedakan kelas yang sesuai, model tersebut selanjutnya digunakan **untuk mengklasifikasikan atribut yang kelasnya belum diketahui**.

CONFUSION MATRIX

Confusion matrix merupakan suatu metode yang digunakan untuk melakukan perhitungan akurasi pada konsep data mining (Rosandy, 2016).

10

- **True positive** adalah jumlah dari record positif yang diklasifikasikan sebagai positif
- **False positive** adalah jumlah dari record negatif yang diklasifikasikan sebagai positif
- **False negative** adalah jumlah dari record positif yang diklasifikasikan sebagai negatif
- **True negative** adalah jumlah dari record negatif yang diklasifikasikan sebagai negatif

Kelas Asli	Kelas Prediksi	
	Positif	Negatif
Positif	True Positive (TP)	False Negative (FN)
Negatif	False Positive (FP)	True Negative (TN)

AKURASI

Tingkat kedekatan antara nilai prediksi dengan nilai sebenarnya

$$Akurasi = \frac{TP + TN}{TP + FP + TN + FN} \times 100\%$$

PRESISI

Menghitung nilai proporsi kelas positif yang berhasil diprediksi benar dari keseluruhan hasil kelas positif

$$Presisi = \frac{TP}{TP + FP} \times 100\%$$

F1-SCORE / F1-MEASURE

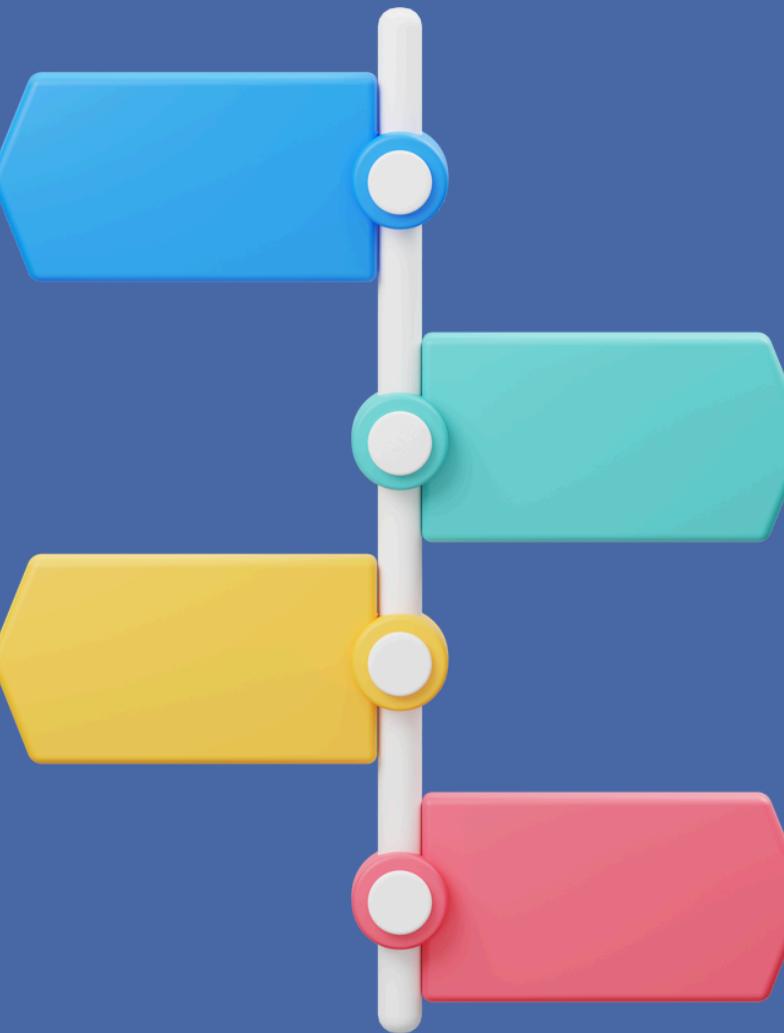
Rata-rata harmonik dari presisi dan recall

$$F1 - score = 2 \times \frac{\text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}}$$

RECALL

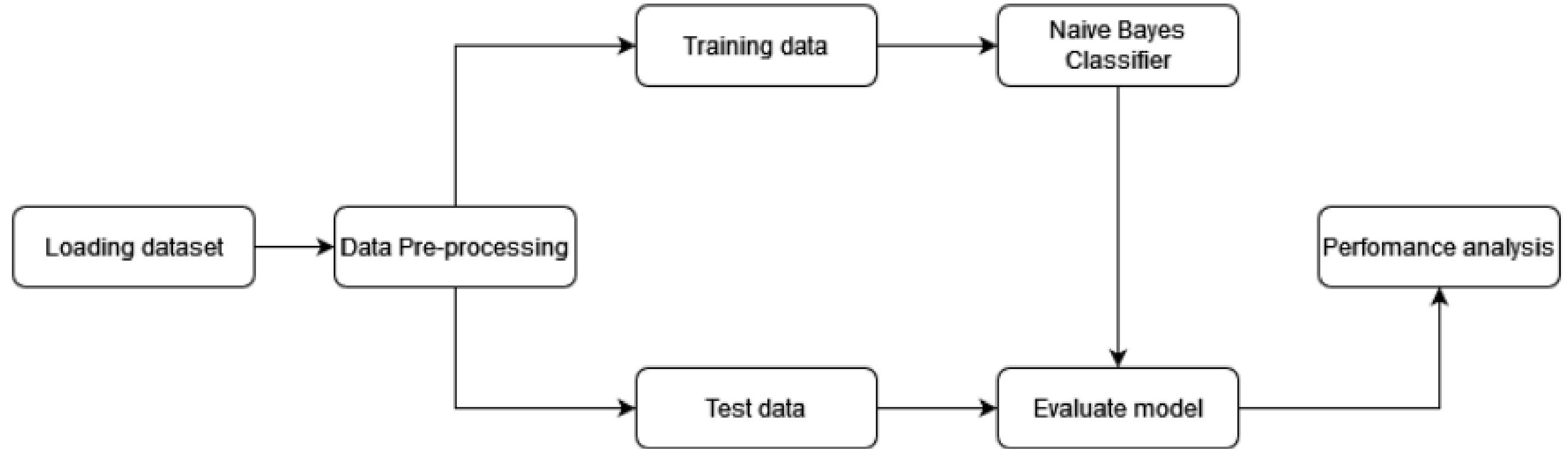
Menghitung persentase kelas data positif yang berhasil diprediksi benar dari keseluruhan data kelas positif

$$Recall = \frac{TP}{TP + FN} \times 100\%$$



METODE PENELITIAN

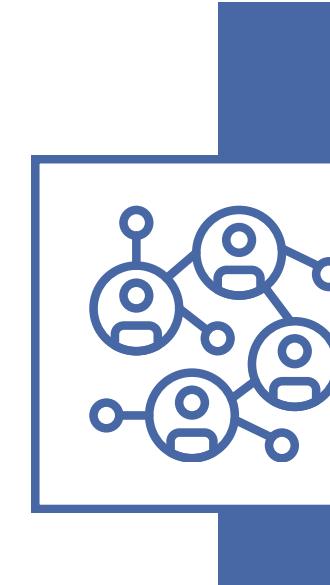
WORKFLOW



PERSIAPAN DATA



Dataset yang digunakan berjudul "Stroke Prediction Data".
[Stroke Prediction Dataset \(kaggle.com\)](#)



Dataset terdiri dari beberapa atribut seperti usia, jenis kelamin, riwayat hipertensi, riwayat penyakit jantung, rata-rata level glukosa, indeks massa tubuh (BMI), status perokok, dan status apakah pasien pernah mengalami stroke atau tidak.

DATA PREPROCESSING

Data Cleaning

Mengatasi Missing Values.

Analisis Data

Visualisasi Target dan Atribut. Distribusi Korelasi Antara Kelas Atribut.

Pemilihan Atribut

Menentukan Atribut yang Relevan: Usia, Riwayat Hipertensi, Riwayat Penyakit Jantung, Glukosa Rata-rata, dan Indeks Massa Tubuh (BMI).

GAUSSIAN NAIVE BAYES CLASSIFIER

- Menggunakan algoritma **Gaussian Naive Bayes Classifier** untuk melakukan **prediksi** penyakit stroke **berdasarkan atribut-atribut** yang telah dipilih sebelumnya.
- Metode Naive Bayes umumnya digunakan untuk data yang digolongkan secara binary, true atau false dan juga kategorik

- Metode Gaussian Naive Bayes merupakan subset dari metode Naive Bayes yang mendukung atribut dan model kontinu sesuai dengan distribusi normal Gaussian yang berbasis pada nilai mean dan standar deviasi.

NAIVE BAYES CLASSIFIER

Data handling

Data Separation and Summarization

Predicting Using Gaussian PDF

Output Analysis with Performance Metrics

- Impor file CSV dari file yang telah di-*preprocess*
- Memisahkan *training set* dan *testing set*
- Memisahkan record data berdasarkan label *class*
- Menyederhanakan atau men-summarize data menjadi mean, stdev dari tiap *attribute*
- Menghitung *probability* kategori dataset dengan *Gaussian*
- Membandingkan *probability* untuk mendapat prediksi *class*
- Membentuk confusion matrix
- Kalkulasi *accuracy*, *precision*, *recall*, *specificity*, *negative prediction value*, dan *f1 score*

EVALUASI MODEL

Accuracy

$$(TP + TN) / (TP + TN + FP + FN)$$

Precision

$$TP / (TP + FP)$$

Recall

$$TP / (TP + FN)$$

Negative predictive value

$$TN / (TN + FN)$$

Specificity

$$TN / (TN + FP)$$

F1 score

$$2 * (\text{Presisi} * \text{Recall}) / (\text{Presisi} + \text{Recall})$$



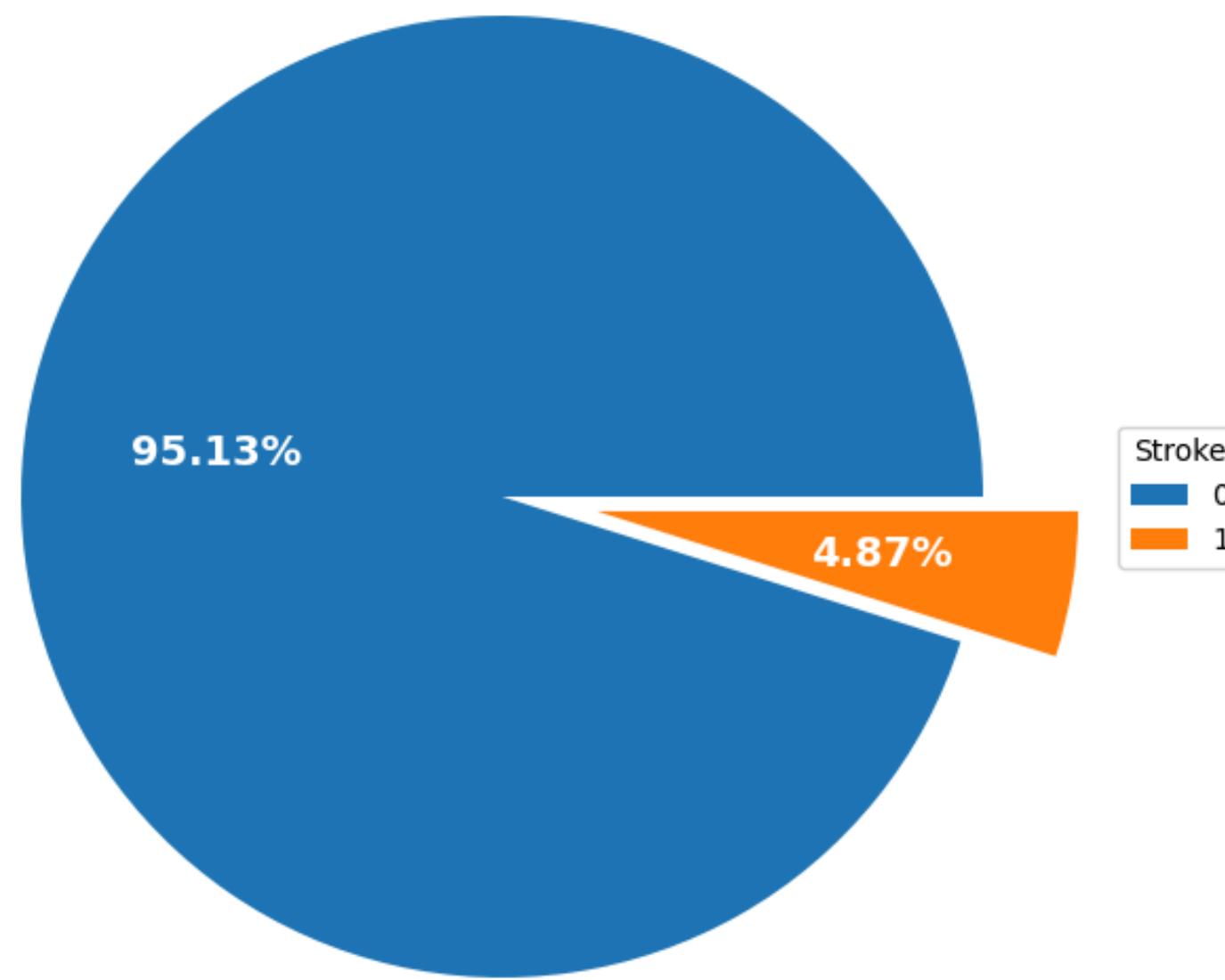


HASIL DAN PEMBAHASAN

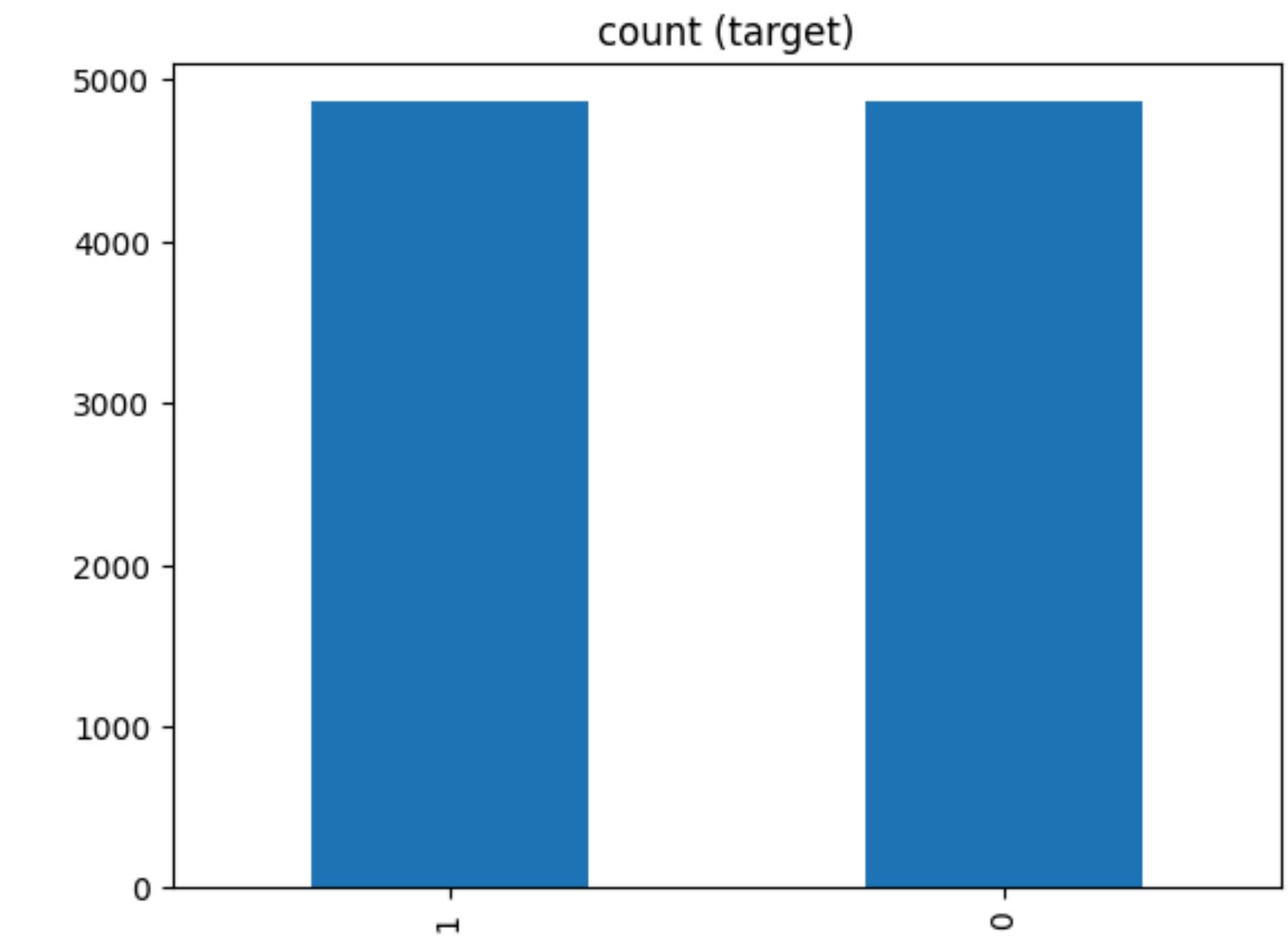
EDA

EXPLORATORY DATA ANALYSIS

18



class distribution

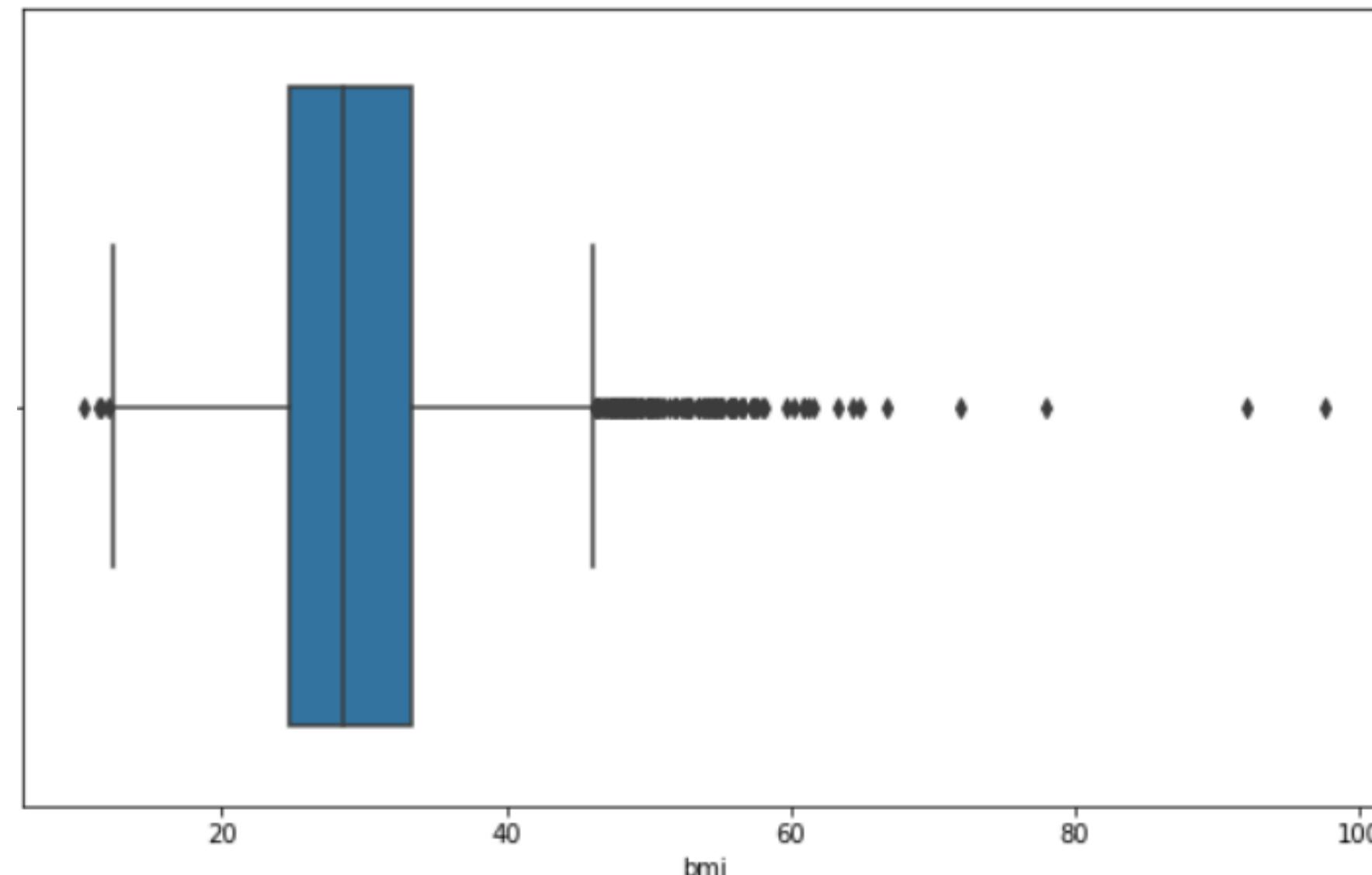


oversampling

EDA

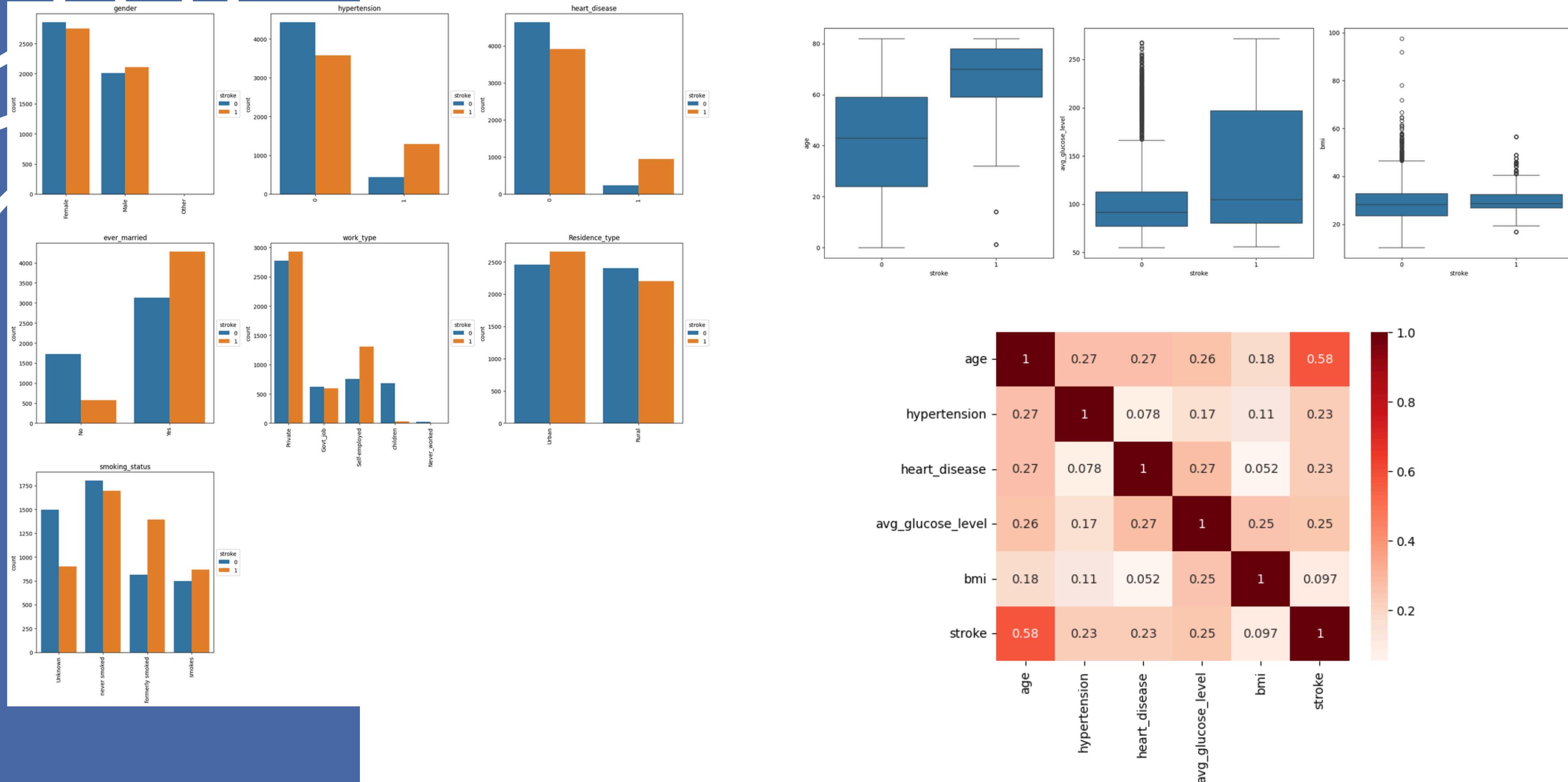
EXPLORATORY DATA ANALYSIS

19



DROP UNKNOWN VALUE IN BMI ATTRIBUTES

EDA (EXPLORATORY DATA ANALYSIS)



EDA (EXPLORATORY DATA ANALYSIS)

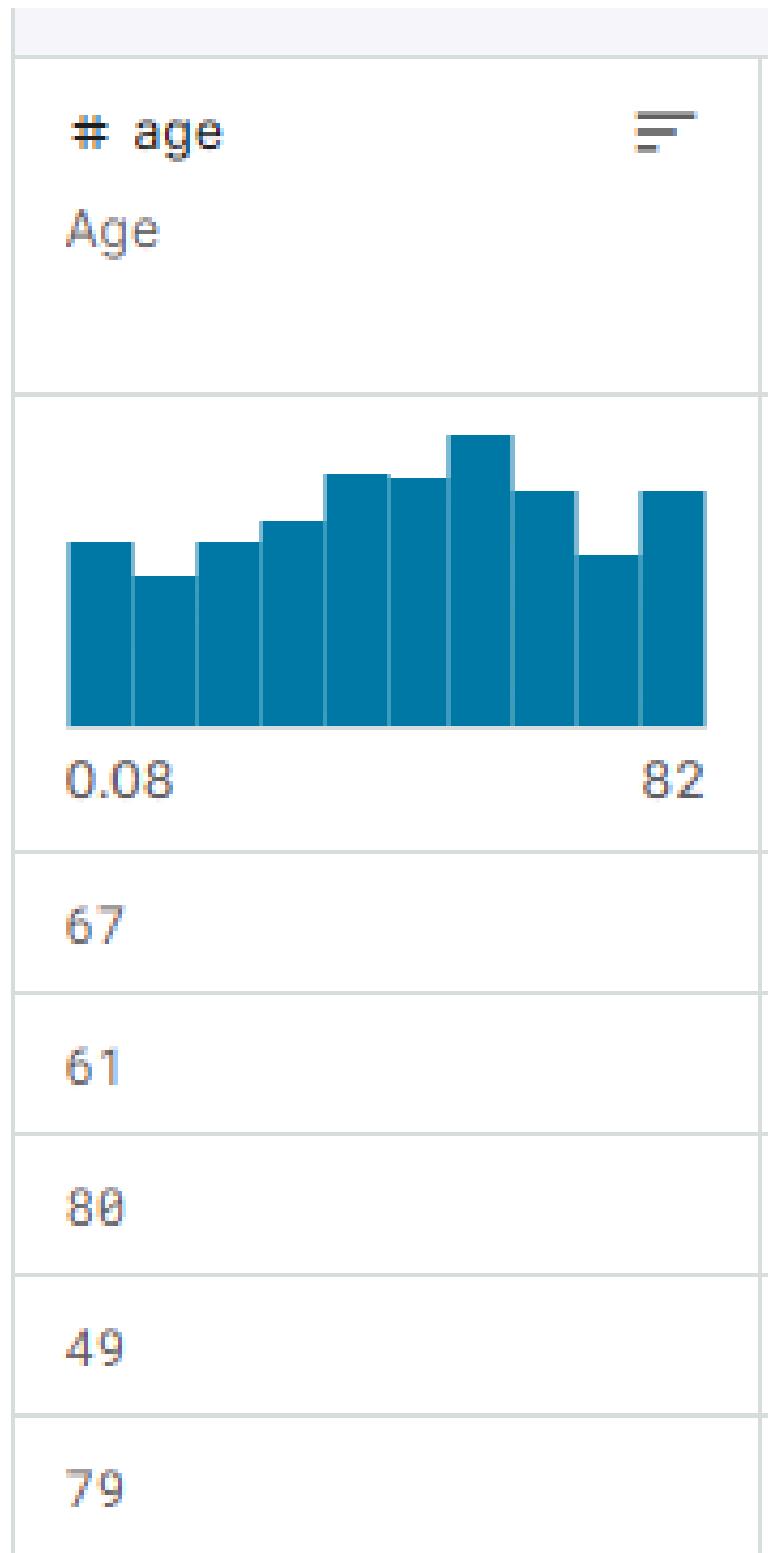
	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke
103	81.0	0	1	78.70	19.4	1
149	70.0	0	1	239.07	26.1	1
128	82.0	0	0	200.59	29.0	1
108	79.0	0	0	93.05	24.2	1
157	57.0	0	0	221.89	37.3	1
94	45.0	0	0	64.14	29.4	1
193	68.0	1	1	271.74	31.1	1
153	68.0	0	0	77.82	27.5	1
151	68.0	0	1	223.83	31.9	1
91	81.0	0	0	72.81	26.3	1

ATTRIBUTES OF INTEREST

PEMILIHAN ATTRIBUT DAN ALASANNYA

AGE

- Faktor umur sangat berpengaruh terhadap kualitas fisik manusia, termasuk saraf.
- Stroke merupakan penyakit di bagian saraf yang menyerang pembuluh darah di bagian otak.
- Semakin meningkatnya usia, faktor risiko terkena penyakit stroke semakin tinggi.



PEMILIHAN ATRIBUT DAN ALASANNYA

BMI

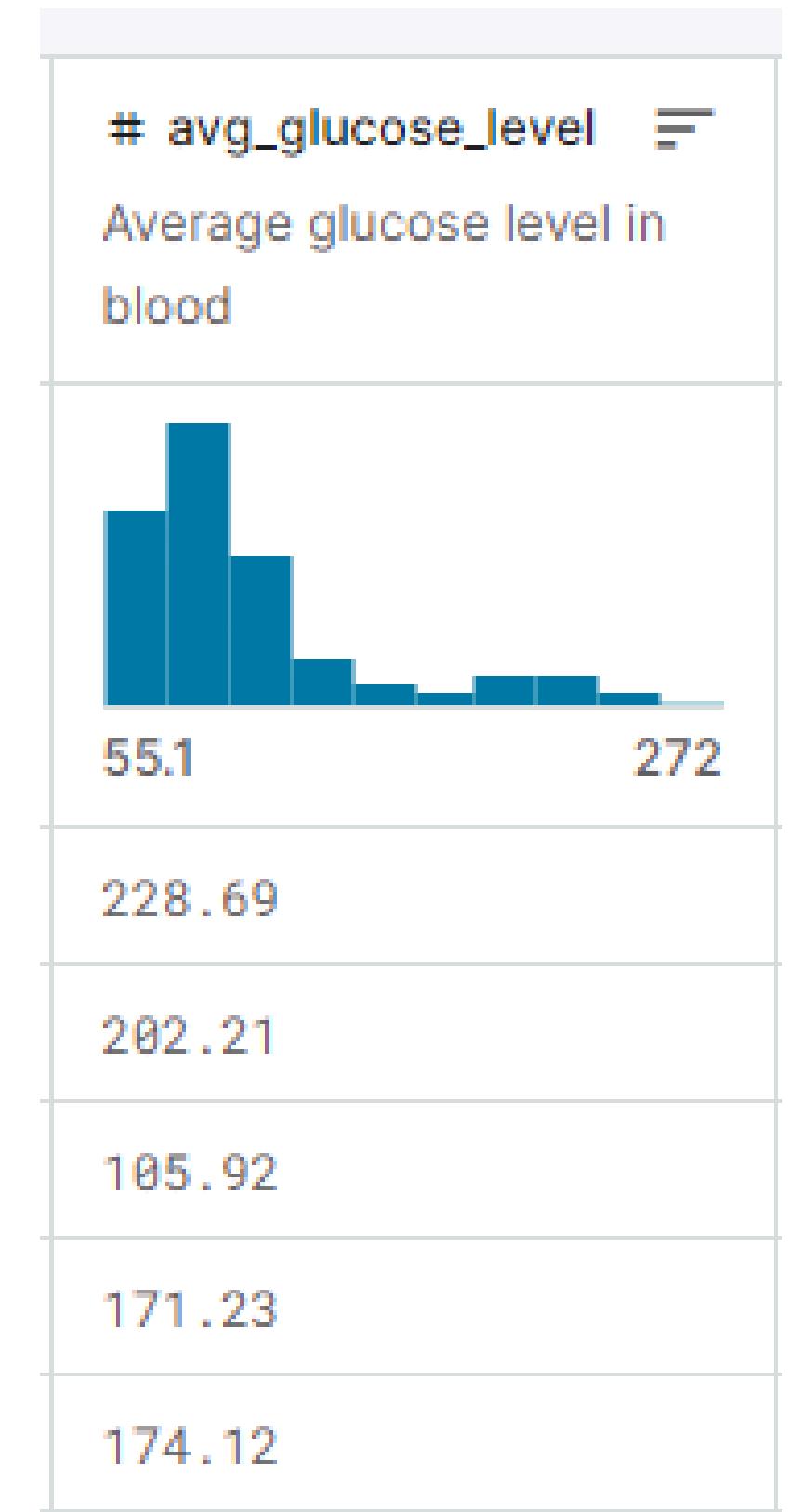
- BMI merupakan faktor risiko penting karena obesitas yang diukur melalui indeks massa tubuh (BMI) dapat menyebabkan peradangan sistemik akibat lebih banyak jaringan lemak yang diproduksi
- Kondisi obesitas juga dikaitkan dengan peningkatan risiko terkena stroke iskemik maupun hemoragik karena berpotensi meningkatkan resistensi pembuluh darah dan memicu aterosklerosis.

A bmi	F
Body Mass Index	
N/A	4%
28.7	1%
Other (4868)	95%
36.6	
N/A	
32.5	
34.4	
24	

PEMILIHAN ATRIBUT DAN ALASANNYA

Average Glukose (kadar gula darah)

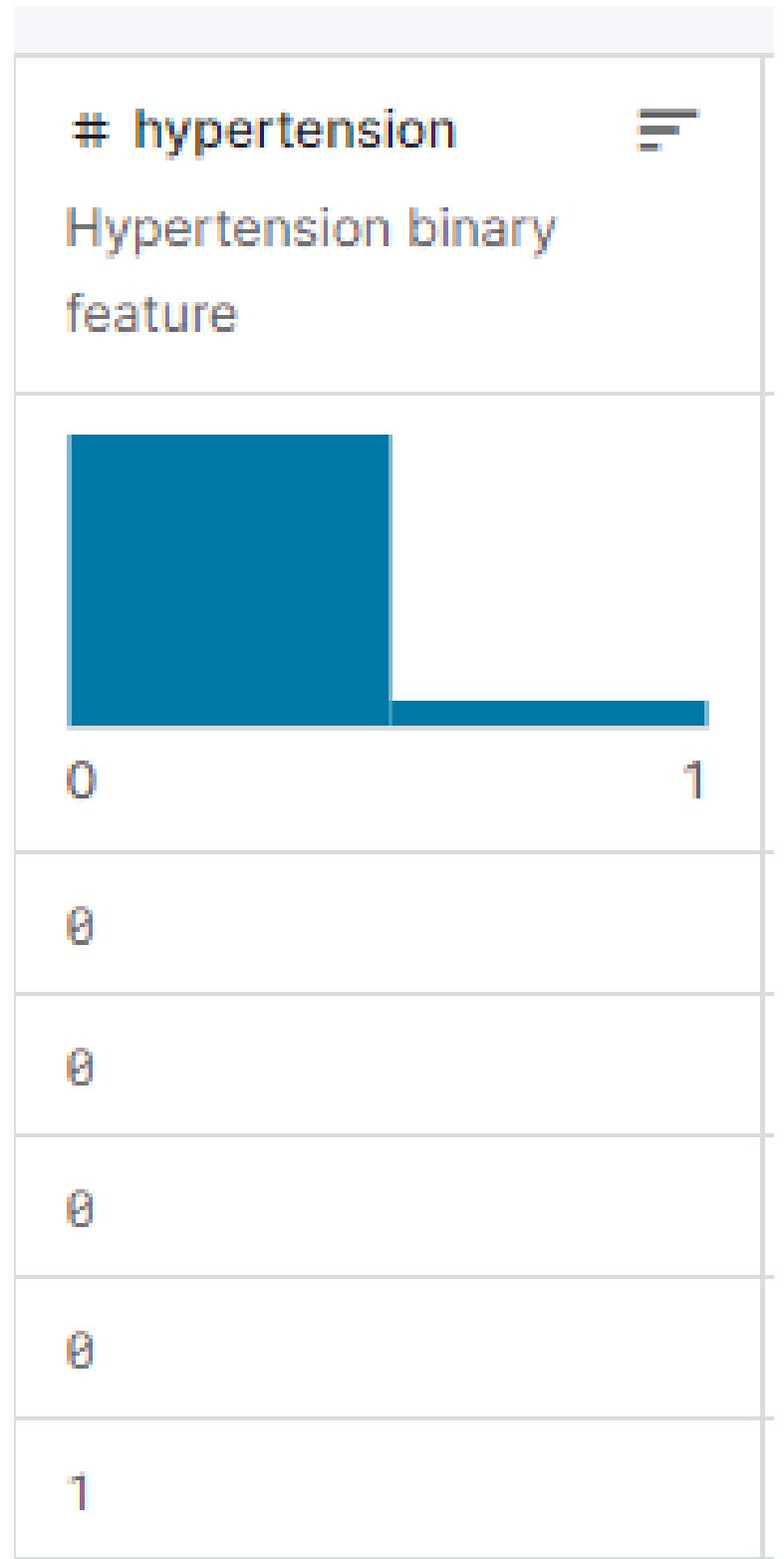
- Kadar glukosa darah yang tinggi dapat menyebabkan terbentuknya atheroma dan penumpukan lemak dalam pembuluh darah.
- Riset menunjukkan semakin tinggi kadar glukosa darah, risiko terkena stroke juga semakin meningkat.



PEMILIHAN ATTRIBUT DAN ALASANNYA

Hipertensi

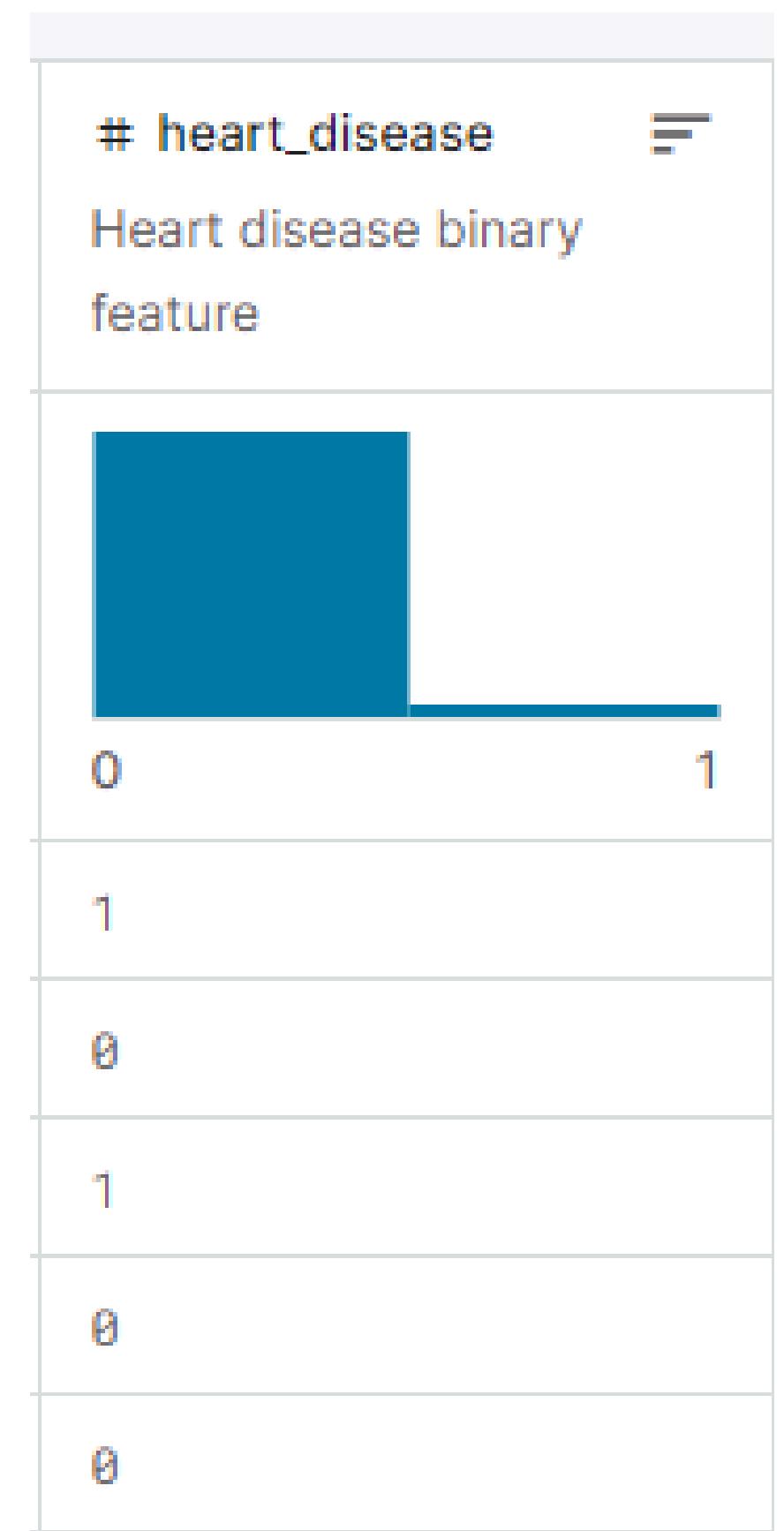
- Hipertensi berpotensi menyebabkan peningkatan tekanan darah perifer sehingga menimbulkan gangguan hemodinamik dan terjadinya penebalan pembuluh darah serta hipertrofi miokardium.
- Tekanan darah tinggi berisiko menyebabkan pecahnya pembuluh darah. Jika terjadi di otak, hal ini dapat menimbulkan stroke akibat perdarahan.



PEMILIHAN ATTRIBUT DAN ALASANNYA

Heart Disease

- Penyakit jantung rentan memicu stroke karena adanya kemungkinan terganggunya aliran darah akibat ketidaksinkronan kerja jantung.



DATA HANDLING

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9722 entries, 103 to 5109
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype  
---  --  
 0   age              9722 non-null    float64 
 1   hypertension     9722 non-null    int64   
 2   heart_disease   9722 non-null    int64   
 3   avg_glucose_level 9722 non-null    float64 
 4   bmi              9722 non-null    float64 
 5   stroke           9722 non-null    int64   
dtypes: float64(3), int64(3)
memory usage: 531.7 KB
```

Pada *preprocessed_Stroke_Data.csv* terdapat 9722 buah record

▼ Data Handling

```
[1] # Load CSV from the directory
def loadCsv(filename):
    lines = csv.reader(open(filename))
    dataset = list(lines)
    dataset.pop(0)
    for i in range(len(dataset)):
        dataset[i] = [float(x) for x in dataset[i]]
    return dataset

# Separate dataset into training set and testing set
def splitDataset(dataset, splitRatio):
    trainSize = int(len(dataset) * splitRatio)
    trainSet = []
    copy = list(dataset)
    while len(trainSet) < trainSize:
        index = random.randrange(len(copy))
        trainSet.append(copy.pop(index))
    return [trainSet, copy]

# Load/import the preprocessed dataset
filename = 'preprocessed_stroke_data.csv'
splitRatio = 0.2 #Calculating the percentage ratio of test data and training data.
dataset = loadCsv(filename)

# Dividing the dataset into training dataset and testing dataset
trainingSet, testSet = splitDataset(dataset, splitRatio)
print('Split {0} rows into train = {1} and test = {2} rows'.format(len(dataset), len(trainingSet), len(testSet)))

Split 9722 rows into train = 1944 and test = 7778 rows
```

Dipecah dengan split ratio 0.2 secara acak menjadi 1944 data training dan 7778 data testing

DATA SEPARATION AND SUMMARIZATION

▼ Data Separation by Class

```
[ ] def separateByClass(dataset):
separated = {}
for i in range(len(dataset)):
    vector = dataset[i]
    if vector[-1] not in separated:
        separated[vector[-1]] = []
    separated[vector[-1]].append(vector)
return separated
```

```
[ ] def summarize(dataset):
summaries = [(np.mean(attribute), np.std(attribute, ddof=1)) for attribute in zip(*dataset)]
del summaries[-1]
return summaries

[ ] def summarizeByClass(dataset):
separated = separateByClass(dataset)
summaries = {}
for classValue, instances in separated.items():
    summaries[classValue] = summarize(instances)
return summaries
```

PREDICTING USING GAUSSIAN PDF

▼ Prediction using Gaussian PDF

```
[ ] # Formula of gaussian probability density function
def calculateProbability(x, mean, stdev):
    exponent = math.exp(-(math.pow(x-mean,2)/(2*math.pow(stdev,2))))
    return (1/(math.sqrt(2*math.pi)*stdev))*exponent

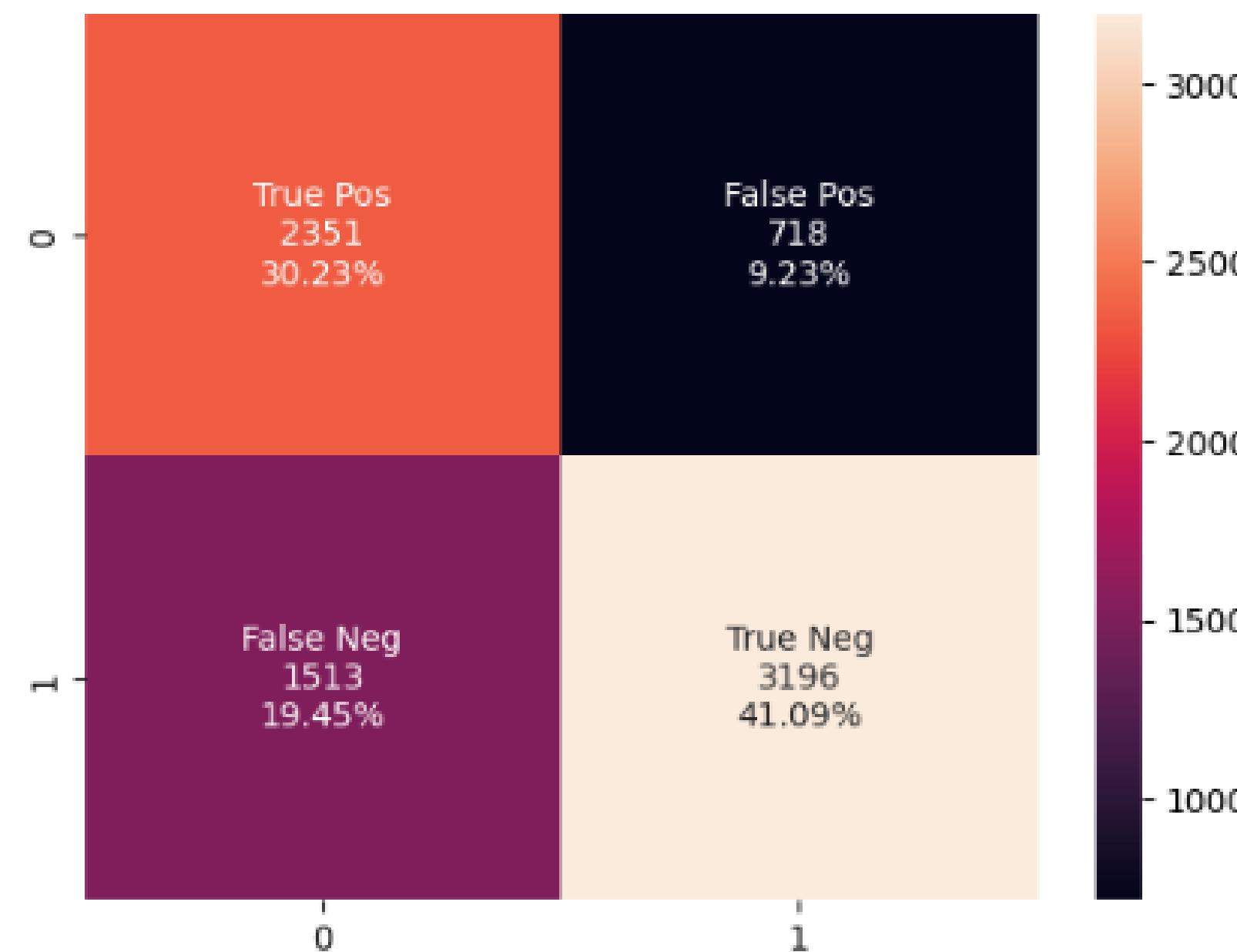
[ ] # Calculating the class categorization probability of input vector
def calculateClassProbabilites(summaries, inputVector):
    probabilities = {}
    for classValue, classSummaries in summaries.items():
        probabilities[classValue] = 1
        for i in range(len(classSummaries)):
            mean, stdev = classSummaries[i]
            x = inputVector[i]
            probabilities[classValue] *= calculateProbability(x, mean, stdev)
    return probabilities
```

```
[ ] # Comparing the class categorization probability (highest) to pick the best prediction
def predict(summaries, inputVector):
    probabilities = calculateclassProbabilites(summaries, inputVector)
    bestLabel, bestProb = None, -1
    for classValue, probability in probabilities.items():
        if bestLabel is None or probability > bestProb:
            bestProb = probability
            bestLabel = classValue
    return bestLabel

[ ] # Organizing predictions into a list
def getPredictions(summaries, testSet):
    predictions = []
    for i in range(len(testSet)):
        result = predict(summaries, testSet[i])
        predictions.append(result)
    return predictions
```

OUTPUT ANALYSIS WITH PERFORMANCE METRICS

30



- Accuracy: 71.3%
- Precision: 76.6%
- Recall: 60.8%
- Specificity: 81.6%
- F1 Score: 67.8%



KESIMPULAN

SUMMARY REPORT

32

KESIMPULAN

- Proporsi data training dan data testing memengaruhi performa model Naive Bayes. Proporsi yang tidak tepat dapat mengakibatkan overfitting atau underfitting.
- Preprocessing data dengan data cleaning dan pemilihan atribut yang relevan seperti umur, hipertensi, riwayat penyakit jantung, rata-rata glukosa, dan BMI mempengaruhi akurasi klasifikasi.

SARAN

- Melakukan penelitian lanjutan seperti riwayat penyakit, faktor risiko, gejala penyakit stroke, faktor risiko, mekanisme patofisiologi, dan strategi pencegahan penyakit stroke disarankan untuk mendukung pengembangan model klasifikasi yang lebih akurat dan efektif
- Menguji model pada sampel populasi yang lebih besar dan beragam untuk memvalidasi generalisasi model.

THANK YOU

Let's Discuss!